

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA**  
**COMPUTAÇÃO**

**Rafael Luiz Cancian**

**AVALIAÇÃO DE DESEMPENHO DE ALGORITMOS DE**  
**ESCALONAMENTO DE TEMPO REAL PARA O**  
**AMBIENTE DO MULTICOMPUTADOR CRUX.**

Dissertação apresentada ao Curso de Pós-Graduação em Ciência da Computação – Universidade Federal de Santa Catarina, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação.

Professor Orientador: Luis Fernando Friedrich, Dr

Florianópolis, agosto de 2001.

# AVALIAÇÃO DE DESEMPENHO DE ALGORITMOS DE ESCALONAMENTO DE TEMPO REAL PARA O AMBIENTE DO MULTICOMPUTADOR CRUX.

Rafael Luiz Cancian

Esta dissertação foi julgada adequada para obtenção do título de **Mestre em Ciência da Computação** Área de Concentração **Computação Paralela** e aprovada em sua forma final pelo Curso de Pós-Graduação em Ciência da Computação

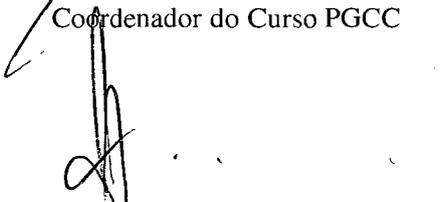


Prof. Luis Fernando Friedrich, Dr  
Orientador, INE - UFSC



Fernando A. O. Gauthier, Dr  
Coordenador do Curso PGCC

Banca Examinadora



Prof. Paulo José de Freitas Filho, Dr,  
INE - UFSC



Prof. Rômulo Silva de Oliveira, Dr  
DAS - UFSC



Prof. Thadeu Botteri Corso, Dr  
INE - UFSC

"Não acredite em qualquer coisa simplesmente porque você escutou.

Não acredite em qualquer coisa simplesmente porque foi dito e fofocado por muitos. Não acredite em qualquer coisa simplesmente porque foi encontrado escrito em seus livros religiosos. Não acredite em qualquer coisa meramente na autoridade de seus professores e anciãos. Não acredite em tradições porque elas foram passadas abaixo por gerações. Mas após observação e análise, quando você descobre que qualquer coisa concorda com a razão e é condutivo ao bem e benefício de um e todos, então aceite e viva para isso."

Siddartha Gautama (o Buda)

## Dedicatória

Aos meus pais Sérgio Gabriel e Kacilda Yara e ao meu irmão Felipe Augusto.

## Agradecimentos

À Universidade Federal de Santa Catarina.

Aos professores e funcionários do Curso de Pós-Graduação em Ciência da Computação.

Ao meu orientador, L. F. Friedrich, e demais membros da banca pela ajuda fornecida.

Aos meus sempre amigos Humberto A. Carvalho, Jorge A. Cruz e Patrícia M. Siqueira.

## SUMÁRIO

LISTA DE FIGURAS .....	vi
LISTA DE TABELAS .....	ix
LISTA DE ABREVIATURAS E SIGLAS.....	xii
RESUMO.....	xiv
ABSTRACT.....	xv
1. INTRODUÇÃO E JUSTIFICATIVA .....	01
2. SISTEMAS DISTRIBUÍDOS .....	05
2.1. Introdução .....	05
2.2. Características.....	06
2.3. Classificação .....	07
2.3.1. <u>Quanto ao Modo de Interação</u> .....	07
2.3.2. <u>Quanto ao Modo de Processamento</u> .....	09
2.4. <u>Redes de Interconexão</u> .....	10
2.4.1. <u>Topologias Estáticas</u> .....	12
2.4.2. <u>Topologias Dinâmicas</u> .....	13
2.5. <u>Resumo</u> .....	14
3. SISTEMAS DE TEMPO REAL.....	16
3.1. <u>Introdução</u> .....	16

<b>3.2. Características</b> .....	17
<b>3.3. Classificação</b> .....	19
<b>3.4. Sistema Operacional de Tempo Real</b> .....	19
<b>3.4.1. Introdução</b> .....	19
<b>3.4.2. Processos</b> .....	21
<b>3.4.3. Comunicação</b> .....	21
<b>3.4.4. Sincronização</b> .....	22
<b>3.5. Resumo</b> .....	23
<b>4. ESCALONAMENTO DE TEMPO REAL</b> .....	25
<b>4.1. Introdução</b> .....	25
<b>4.2. Classificação dos Algoritmos de Escalonamento</b> .....	26
<b>4.3. Caracterização das Tarefas de Tempo Real</b> .....	26
<b>4.4. Precedência de Tarefas e Alocação de Recursos</b> .....	28
<b>4.5. Algoritmos de Escalonamento de Tempo Real</b> .....	30
<b>4.5.1. Taxa Monotônica (RM – Rate Monotonic)</b> .....	30
<b>4.5.2. Menor Deadline Primeiro (EDF – Earliest Deadline First)</b> .....	33
<b>4.5.3. Deadline Monotônica (MD – Monotonic Deadline)</b> .....	35
<b>4.5.4. Escalonamento em Background (BS – Background Scheduling)</b> .....	36
<b>4.5.5. Servidor Pooling (PS – Pooling Server)</b> .....	37
<b>4.5.6. Servidor Deferrable (DS – Deferrable Server)</b> .....	39
<b>4.5.7. Servidor Esporádico (SS – Sporadic Server)</b> .....	40
<b>4.5.8. Menor Deadline Primeiro Robusto (RED – Robust Earliest Deadline First)</b> .....	43

4.5.9. Servidor Esporádico Dinâmico (DSS – <i>Dinamic Sporadic Server</i> ) .....	47
4.5.10. Servidor de Largura de Banda Constante (CBS – <i>Constant Bandwidth Server</i> ) ..	49
4.5.11. Reserva de Banda Adaptativa (ABR – <i>Adaptative Bandwidth Reservation</i> ).....	52
<b>5. SIMULAÇÃO DIGITAL .....</b>	<b>54</b>
5.1. Introdução .....	54
5.2. Vantagens e Desvantagens .....	54
5.3. Classificação .....	56
5.4. Análise Estatística.....	57
5.5. Ferramentas de Simulação .....	62
5.5.1. <u>GHOST</u> .....	62
5.5.2. <u>Arena</u> .....	64
<b>6. AVALIAÇÃO DE DESEMPENHO.....</b>	<b>67</b>
6.1. Introdução .....	67
6.2. Métricas de Desempenho.....	70
6.3. Carga de Trabalho .....	74
6.3.1. <u>Benchmarks</u> .....	76
6.4. Projeto Experimental .....	78
6.4.1. <u>Projeto Fatorial 2<sup>k</sup></u> .....	82
6.4.2. <u>Projeto Fatorial 2<sup>k</sup>r</u> .....	83
<b>7. O AMBIENTE MULTICOMPUTADOR CRUX.....</b>	<b>85</b>

<b>7.1. Arquitetura CRUX</b> .....	85
<b>7.2. O sistema operacional <math>\alpha</math>-CRUX</b> .....	87
<b>8. METODOLOGIA DE PESQUISA</b> .....	89
<b>8.1. Introdução</b> .....	89
<b>8.2. Projeto de Avaliação de Desempenho</b> .....	89
<b>8.2.1. <u>Definição dos Objetivos do Sistema</u></b> .....	89
<b>8.2.2. <u>Relação dos Serviços e Saídas</u></b> .....	90
<b>8.2.3. <u>Seleção das Métricas</u></b> .....	91
<b>8.2.4. <u>Relação dos Parâmetros</u></b> .....	91
<b>8.2.5. <u>Seleção dos Fatores</u></b> .....	92
<b>8.2.6. <u>Técnica de Avaliação</u></b> .....	92
<b>8.2.7. <u>Seleção da Carga de Trabalho</u></b> .....	93
<b>8.2.8. <u>Projeto de Experimentos</u></b> .....	94
<b>8.2.9. <u>Análise e Interpretação dos Resultados</u></b> .....	95
<b>8.2.10. <u>Apresentação dos Resultados</u></b> .....	96
<b>9. DESENVOLVIMENTO DO MODELO DE SIMULAÇÃO</b> .....	97
<b>9.1. Introdução</b> .....	97
<b>9.2. Descrição do Comportamento do Modelo</b> .....	99
<b>9.2.1. <u>Comportamento Geral</u></b> .....	99
<b>9.2.2. <u>Comportamento de Atividades <i>Hard Real-Time</i></u></b> .....	102
<b>9.2.3. <u>Comportamento de Atividades <i>Soft Real-Time VBR</i></u></b> .....	105

9.2.4. <u>Comportamento de Atividades <i>Soft Real-Time CBR</i></u> .....	112
9.2.5. <u>Comportamento dos Elementos do Sistema</u> .....	117
9.3. <b>Parâmetros do Modelo</b> .....	119
9.4. <b>Descrição da Interface VBA</b> .....	120
9.5. <b>Verificação e Validação do Modelo</b> .....	122
10. <b>ANÁLISE DOS RESULTADOS</b> .....	125
10.1. <b>Introdução</b> .....	125
10.1. <b>Primeiro Conjunto de Experimentos</b> .....	125
10.1.1. <u>Análise do Tempo de Resposta <i>soft VBR</i></u> .....	127
10.1.2. <u>Análise da Variação de Execução <i>soft CBR</i></u> .....	128
10.1.3. <u>Análise do Tempo de Conexão</u> .....	129
9.1. <b>Segundo Conjunto de Experimentos</b> .....	131
10.2.1. <u>Análise do Tempo de Resposta <i>soft VBR</i></u> .....	133
10.2.2. <u>Análise da Variação de Execução <i>soft CBR</i></u> .....	141
10.2.3. <u>Análise do Tempo de Conexão</u> .....	149
11. <b>CONCLUSÕES</b> .....	156
12. <b>CONSIDERAÇÕES FINAIS</b> .....	159
13. <b>REFERÊNCIA BIBLIOGRÁFICA</b> .....	161

## LISTA DE FIGURAS

Figura 2.1 – Representação de uma máquina SISD .....	07
Figura 2.2 – Representação de uma máquina SIMD.....	08
Figura 2.3 – Representação de uma máquina MIMD .....	08
Figura 2.4 – Representação de um multiprocessador conectado por barramento .....	09
Figura 2.5 – Representação de um multicomputador conectado por barramento .....	09
Figura 2.6 – Topologias de redes de interconexão estáticas .....	12
Figura 2.7 – Elemento comutador de estágio único .....	13
Figura 2.8 – Rede dinâmica com comutadores multi-estágio .....	13
Figura 2.9 – Rede dinâmica tipo <i>crossbar</i> .....	14
Figura 4.1 – Escalonamento de tarefas com RM .....	31
Figura 4.2 – Diagrama de blocos do algoritmo RED.....	45
Figura 5.1 – Tela da ferramenta de simulação GHOST.....	63
Figura 5.2 – Estrutura do simulador GHOST .....	63
Figura 5.3 – Exemplo de definição do algoritmo RM no GHOST .....	64
Figura 5.4 – Tela da ferramenta de simulação Arena.....	65
Figura 6.1 – Etapas de um projeto de avaliação de desempenho .....	70
Figura 7.1 – Arquitetura do multicomputador CRUX .....	86

Figura 8.1 – Distribuição <i>Poisson</i> de acesso aos arquivos da classe 0 .....	94
Figura 8.2 – Distribuição <i>Poisson</i> de acesso aos arquivos da classe 1 .....	94
Figura 9.1 – Elementos do modelo do multicomputador CRUX.....	98
Figura 9.2 – Ocupação do NT por uma atividade <i>hard real-time</i> .....	102
Figura 9.3 – Ocupação de NTs servidores e cliente por uma atividade <i>soft VBR</i> .....	109
Figura 9.4 – Seqüência de eventos de uma atividade <i>soft real-time VBR</i> .....	111
Figura 9.5 – Ocupação de NTs servidor e cliente por uma atividade <i>soft CBR</i> .....	114
Figura 9.6 – Ocupação de NTs servidor e cliente por uma atividade <i>soft CBR</i> .....	114
Figura 9.7 – Variação da taxa de execução de uma atividade <i>soft CBR</i> .....	114
Figura 9.8 – Seqüência de eventos de uma atividade <i>soft real-time CBR</i> .....	116
Figura 9.9 – Tela da interface VBA criada para o modelo .....	121
Figura 9.10 – Tela de animação para verificação do modelo.....	123
Figura 10.1 - Gráfico das diferenças entre as médias de tempo de resposta <i>VBR</i> .....	131
Figura 10.2 – Gráficos de resíduo e <i>quantile-quantile</i> do Tempo de resposta <i>VBR</i> .....	131
Figura 10.3 - Gráfico das diferenças entre as médias de variação de execução <i>CBR</i> .....	134
Figura 10.4 – Gráficos de resíduo e <i>quantile-quantile</i> da variação de execução <i>CBR</i> .....	135
Figura 10.5 - Gráfico das diferenças entre as médias de tempo de conexão.....	138
Figura 10.6 – Gráficos de resíduo e <i>quantile-quantile</i> do tempo de conexão.....	139
Figura 10.7 – Valores atuais x Valores preditos do Tempo de Resposta <i>VBR</i> .....	147
Figura 10.8 – Gráficos de resíduo e <i>quantile-quantile</i> do Tempo de resposta <i>VBR</i> .....	148

Figura 10.9 – Gráficos de interação do tempo de resposta <i>VBR</i> .....	149
Figura 10.10 –Valores atuais x Valores preditos para a variação de execução <i>CBR</i> .....	155
Figura 10.11 – Gráficos de resíduo e <i>quantile-quantile</i> para variação de execução <i>CBR</i>	156
Figura 10.12 – Gráficos de interação da variação de execução <i>CBR</i> .....	157
Figura 10.13 – Valores atuais x Valores preditos para o tempo de conexão .....	162
Figura 10.14 – Gráficos de resíduo e <i>quantile-quantile</i> para o tempo de conexão.....	163
Figura 10.15 – Gráficos de interação do o tempo de conexão .....	164

## LISTA DE TABELAS

Quadro 4.1 – Conjunto de tarefas do exemplo .....	29
Quadro 4.2 – Algoritmo de escalonamento RM .....	32
Quadro 4.3 – Algoritmo garantia de escalonabilidade no RM.....	32
Quadro 4.4 – Algoritmo de escalonamento EDF .....	34
Quadro 4.5 – Algoritmo garantia de escalonabilidade no EDF .....	34
Quadro 4.6 – Algoritmo de escalonamento DM.....	35
Quadro 4.7 – Algoritmo de escalonamento BS.....	36
Quadro 4.8 – Algoritmo de escalonamento PS .....	38
Quadro 4.9 – Algoritmo de escalonamento DS .....	38
Quadro 4.10 – Algoritmo de escalonamento SS .....	42
Quadro 4.11 – Algoritmo de escalonamento RED.....	45
Quadro 4.12 – Algoritmo garantia de escalonabilidade no RED.....	46
Quadro 4.13 – Algoritmo de escalonamento DSS .....	48
Quadro 4.14 – Algoritmo de escalonamento CBS.....	51
Quadro 4.15 – Algoritmo de escalonamento ABR .....	53
Tabela 6.1 – Tamanho de arquivos em cada classe de acessos.....	78
Tabela 6.2 – Equações para o projeto experimental com um fator .....	81
Tabela 6.3 – Equações para o projeto experimental $2^k$ .....	82
Tabela 6.4 – Equações para o projeto experimental $2^{k_r}$ .....	83

Quadro 7.1 - Estrutura em camada do sistema operacional $\alpha$ -CRUX .....	88
Quadro 8.1 – Breve descrição dos serviços oferecidos pelo sistema.....	90
Quadro 9.1 – Tempos envolvidos no processamento de atividade <i>soft real-time VBR</i> ....	110
Quadro 9.2 – Tempos envolvidos no processamento de atividade <i>soft real-time CBR</i> ...	115
Quadro 9.3 – Parâmetros do modelo .....	119
Tabela 10.1 - Parâmetros fixos no primeiro conjunto de experimentos .....	126
Tabela 10.2 - Resultados das simulações do projeto experimental de um fator .....	127
Tabela 10.3 – Resultados da ANOVA sobre o tempo de resposta VBR .....	128
Tabela 10.4 – Média do tempo de resposta VBR para os algoritmos de escalonamento..	129
Tabela 10.5 – Resultados das diferenças entre as médias de tempo de resposta VBR.....	129
Tabela 10.6 – Resultados da ANOVA sobre a taxa de variação CBR.....	132
Tabela 10.7 – Média da taxa de variação CBR para os algoritmos de escalonamento.....	132
Tabela 10.8 – Resultados das diferenças entre as médias de variação de execução CBR	133
Tabela 10.9 – Resultados da ANOVA sobre o tempo de conexão .....	136
Tabela 10.10 – Tempo médio de conexão para os algoritmos de escalonamento .....	136
Tabela 10.11 – Resultados das diferenças entre as médias de tempo de conexão .....	137
Tabela 10.12 – Fatores e suas características do projeto experimental fatorial $2^5$ .....	140
Tabela 10.13 – Resultados das simulações projeto experimental fatorial completo $2^5$ ..	141
Tabela 10.14 – Contribuição dos fatores sobre o tempo de resposta <i>soft VBR</i> .....	142
Tabela 10.15 – Resultados da ANOVA sobre o tempo de resposta VBR .....	144

Tabela 10.16 – Estatísticas da ANOVA sobre o tempo de resposta <i>VBR</i> .....	145
Tabela 10.17 – Contribuição dos fatores sobre a variação de execução <i>CBR</i> .....	151
Tabela 10.18 – Resultados da ANOVA sobre a variação de execução <i>CBR</i> .....	153
Tabela 10.19 – Estatísticas da ANOVA sobre a variação de execução <i>CBR</i> .....	153
Tabela 10.20 – Contribuição dos fatores e suas interações sobre o tempo de conexão ....	159
Tabela 10.21 – Resultados da ANOVA sobre o tempo de conexão .....	160
Tabela 10.22 – Estatísticas da ANOVA sobre o tempo de conexão .....	161

## LISTA DE ABREVIATURAS E SIGLAS

<b>ABR</b>	<i>Adaptative Bandwidth Reservation</i>
<b>BS</b>	<i>Background Server</i>
<b>CB</b>	<i>Crossbar</i>
<b>CPI</b>	<i>Cicles Per Instruction</i>
<b>CPU</b>	<i>Central Processor Unit</i>
<b>DAO</b>	<i>Data Access Objects</i>
<b>DS</b>	<i>Deferreble Server</i>
<b>DSS</b>	<i>Dinamic Sporadic Server</i>
<b>E/S</b>	(dispositivos de) <i>Entrada/Saída</i>
<b>FCFS</b>	<i>First Come First Served</i>
<b>FIFO</b>	<i>First In First Out</i>
<b>GHOST</b>	<i>General Hard real-time Oriented Simulator Tool</i>
<b>IC</b>	<i>Instruction Counter</i>
<b>IEEE</b>	<i>Institute of Eletrical and Eletronics Enguneers</i>
<b>ISSO</b>	<i>International Srandardization Organization</i>
<b>LAN</b>	<i>Local Area Network</i>
<b>NC</b>	<i>Nó de Controle</i>
<b>NT</b>	<i>Nó de Trabalho</i>
<b>MFLOPS</b>	<i>Million of Floating-Point Operations Per Second</i>
<b>MISD</b>	<i>Multiple-Instruction Simple-Data stream</i>
<b>MIMD</b>	<i>Multiple-Instruction Multiple-Data stream</i>

<b>MIPS</b>	<i>Million of Instructions Per Second</i>
<b>ODBC</b>	<i>Open Database Conectivity</i>
<b>OSG</b>	<i>Open Systems Group</i>
<b>PE</b>	Priority Exchange
<b>PCP</b>	<i>Priority Ceiling Protocol</i>
<b>PIP</b>	<i>Priority Inheritance Protocol</i>
<b>POSIX</b>	<i>Portable Operating System Interface</i>
<b>ROM</b>	<i>Read Only Memory</i>
<b>SISD</b>	<i>Simple-Instruction Simple-Data stream</i>
<b>SIMD</b>	<i>Simple-Instruction Multiple-Data stream</i>
<b>SMF</b>	<i>Shortest Message First</i>
<b>SOTR</b>	Sistema Operacional de Tempo Real
<b>SPEC</b>	<i>System Performance Evaluation Cooperative</i>
<b>SS</b>	<i>Sporadic Server</i>
<b>TR</b>	Tempo Real
<b>UP</b>	Unidade de Processamento
<b>UC</b>	Unidade de Controle
<b>VBA</b>	<i>Visual Basic for Applications</i>

## RESUMO

Sistemas de tempo real são aqueles onde a execução correta dos programas não depende apenas dos resultados lógicos da computação, mas também do tempo no qual os resultados são produzidos, ou seja, se ocorre dentro do tempo previsto. Os sistemas de tempo real devem ser projetados de forma a atender as restrições temporais e demais restrições impostas, como confiabilidade e tolerância a falhas. Aplicações de tempo real estão cada vez mais presentes no dia-a-dia e em muitas áreas, tais como: multimídia, robótica, controle de processos, atividades militares, biomédica, etc.

Este trabalho está inserido no contexto do projeto CRUX, em desenvolvimento no Curso de Pós-Graduação em Ciência da Computação (CPGCC) da Universidade Federal de Santa Catarina, que visa a concepção de um ambiente completo para programação paralela. Esse projeto inclui a construção de um multicomputador que servirá como base para o projeto e a implementação de um sistema operacional e uma linguagem de programação paralela.

Este trabalho faz uma avaliação de desempenho de diversos algoritmos de escalonamento de tempo real encontrados na bibliografia e algumas variações novas, representando tanto o processamento quanto a comunicação do multicomputador CRUX. O objetivo foi o de quantificar o efeito do algoritmo de escalonamento e outros fatores sobre algumas métricas de desempenho selecionadas, de modo a verificar a aplicabilidade do CRUX em sistemas de tempo real e também propor um escalonador de tempo real adequado ao ambiente.

## ABSTRACT

Real time systems are those where the correct execution of the programs doesn't depend only on the logical results of the computation, but also on the time at which the results are produced, in other words, if it happens within the predicted time. The real time systems should be projected in a way to ensure the time restrictions and also other imposed restrictions, as reliability and fault tolerance. Real time applications are more and more present in the day by day and in a lot of areas, such as: multimedia, robotics, processes' control, military activities, biomedical, etc.

This work is inserted in the context of the CRUX project, in development in the Masters degree Course of in Computing Science (CPGCC) of the Federal University of Santa Catarina, that seeks the conception of a complete environment for parallel programming. This work includes the construction of a multicomputer that will serve as base for the project and implementation of an operating system and a parallel programming language.

This work makes an performance evaluation of several real time scheduler algorithms found in the bibliography and some new variations, representing the processing and the communication of the CRUX multicomputer. The goal was of quantifying the effect of the scheduler algorithm and other factors on some selected performance metrics, in way to verify the usability of CRUX in real time systems and also to propose an appropriate real time scheduler to that environment.

# 1. Introdução e Justificativa

Os sistemas paralelos são uma solução eficiente à demanda por alta capacidade de processamento, difícil de ser suprida por sistemas seqüenciais, em especial microcomputadores monoprocessados. Os sistemas paralelos normalmente compõem-se de vários elementos (nós) processadores e podem ser, basicamente, multiprocessadores ou multicomputadores.

Uma das aplicações que demanda alto poder de processamento e confiança, alvo das arquiteturas paralelas, são as aplicações de tempo real, que incluem muitas áreas, como controle de processos industriais, militares, aéreos, robótica, multimídia e vários outros.

Uma aplicação de tempo real exige certas garantias temporais, de desempenho, de confiabilidade e de tolerância a falhas, que podem ser supridas apenas por arquiteturas paralelas controladas por um sistema que seja projetado para dar suporte às necessidades básicas de tais aplicações.

Num sistema operacional de tempo real, um dos principais aspectos diz respeito ao escalonamento de processos nos nós processadores, responsável por escolher quando e qual processo deve executar, com o objetivo de melhor cumprir as restrições impostas por aplicações de tempo real.

Um exemplo de arquitetura paralela multicomputador é o ambiente CRUX, projeto desenvolvido no Curso de Pós-Graduação em Ciência da Computação (CPGCC), do Centro Tecnológico (CTC) da Universidade Federal de Santa Catarina (UFSC).

O projeto CRUX visa a construção de um sistema paralelo multicomputador, composto de nós de trabalho e um nó de controle do tipo microcomputador pessoal, e possui uma rede de interconexão dinâmica do tipo *crossbar* que permite a troca de mensagens entre os nós de trabalho e uma rede tipo barramento no qual os nós de trabalho solicitam as conexões do *crossbar* ao nó de controle.

O ambiente do CRUX pode servir como base para a execução de programas paralelos, projetos que envolvam aplicações paralelas, a construção de um sistema operacional e uma linguagem de programação paralelos, e outros projetos importantes que demandem alto desempenho.

Por estar atualmente em estágio de projeto, é importante que se possa estimar o comportamento e o desempenho de tal sistema quando concluído, para permitir que se conheça, de antemão, a capacidade e a aplicabilidade do sistema.

Este trabalho está incluído no contexto do projeto CRUX, e são basicamente dois os seus objetivos:

O primeiro é modelar o comportamento do ambiente CRUX para, através de simulação digital, obter dados que permitam avaliar o desempenho desse ambiente e o comportamento de vários algoritmos de escalonamento de tempo real e seus efeitos sobre as métricas de desempenho escolhidas.

O segundo é que tal modelo seja reusável, possuindo uma interface com o usuário que permita alterações nas características do sistema sem a necessidade de alterar a estrutura do modelo, o que pode ser muito útil para futuros trabalhos sobre simulação, avaliação de desempenho ou planejamento da capacidade do ambiente CRUX, mesmo sob um enfoque de análise diferente.

Este trabalho também aborda alguns pontos apresentados por outros autores como possíveis trabalhos futuros, em suas dissertações de mestrado no CPGCC da UFSC, também inseridos no contexto do projeto do CRUX.

MONTEZ (1995) sugere a adoção de *threads* para programação de eventos paralelos, o que facilitaria o desenvolvimento desse tipo de algoritmos (paralelos). Apesar de que a implementação de *threads* não ser parte do escopo deste trabalho, a simulação do ambiente assume que a unidade básica de alocação de processos é a *thread*.

SILVA (1996) também sugere que “o modelo de simulação desenvolvido pode ser melhorado implementando uma modelagem mais genérica das aplicações existentes. Devem ainda, serem feitos experimentos das várias combinações possíveis da configuração da máquina. Por exemplo, experimentos com base no aumento da

quantidade de comunicações em cada NT, na variação do tempo de processamento e na utilização de mensagens de tamanhos variados”. Essas sugestões também foram incluídas do âmbito do presente trabalho.

CORRÊA (1998) aponta como pontos-alvo para futuro trabalhos “uma nova análise de avaliação de desempenho quanto ao esquema de escalonamento”, e a “implementação dos serviços de suporte para aplicações de tempo real”.

Na mesma linha de raciocínio, CUNHA (1998) também sugere a “criação de um modelo genérico [...] no *software* de simulação que aceitasse características específicas através de uma interface orientada a menu, possibilitando a simulação do funcionamento [...] com dezenas de diferentes características”. Tais características desejadas são agora contempladas neste trabalho.

Aos leitores interessados em informações mais específicas, é importante compreender que o presente trabalho está organizado da seguinte maneira:

A primeira parte, de revisão bibliográfica e contextualização do trabalho. Inicialmente, no Capítulo 1, é feita uma Introdução e Justificativa à realização deste trabalho. No Capítulo 2, são apresentadas características de Sistemas Distribuídos, o que permite classificar o ambiente CRUX. Em seguida, no Capítulo 3, sobre Sistemas de Tempo Real, pode-se compreender que tipo de restrições se deseja submeter ao ambiente CRUX. O Capítulo 4 continua acerca de sistemas de tempo real, e apresenta mais especificamente Algoritmos de Escalonamento de Tempo Real, onde é possível conhecer os principais escalonadores encontrados na bibliografia recente e que serão utilizados neste trabalho. O Capítulo 5 conceitua, caracteriza e classifica Simulação Digital de Sistemas, ferramenta utilizada como meio de realizar a Avaliação de Desempenho (Capítulo 6) do ambiente CRUX no que se refere aos escalonadores de tempo real. O Capítulo 7 apresenta com mais detalhes o próprio Ambiente CRUX.

A segunda parte, de desenvolvimento e conclusões alcançadas. No Capítulo 8 é descrita Metodologia de Pesquisa para a avaliação de desempenho. O Capítulo 9 trata especificamente do Desenvolvimento do Modelo de Simulação, onde é descrito em detalhes o modelo (ARENA) e o programa (VBA) desenvolvidos para simular o ambiente CRUX. Os resultados obtidos nas simulações são apresentados e analisados no

décimo capítulo, Análise dos Resultados. Por fim, o Capítulo 11 discute os resultados e apresenta as Conclusões alcançadas, e o Capítulo 12 trata de algumas Considerações Finais a respeito de trabalho desenvolvido e de possíveis aperfeiçoamentos. A Bibliografia utilizada e referenciada no texto é apresentada no Capítulo 13.

## 2. Sistemas Distribuídos

### 2.1. Introdução

Dois grandes avanços conseguidos na década de 80 foram o desenvolvimento de microprocessadores com poder de processamento cada vez maior e o das redes locais (LAN) de alta velocidade. Segundo TANENBAUM (1995), esses avanços permitiram a criação de novos sistemas, compostos por vários processadores e conectados através de redes de alta velocidade e que são denominados Sistemas Distribuídos.

A vantagem mais evidente do uso de sistemas distribuídos é o compartilhamento de recursos computacionais, como memória, processadores, unidades de discos, impressoras, *plotters* e outros. Quanto maior o custo de um recurso, mais vantajoso se torna o seu compartilhamento. Uma outra vantagem do uso de sistemas distribuídos advém do fato de muitas aplicações serem distribuídas por natureza, como acesso à saldos e extratos numa rede bancária ou o controle de um grupo de robôs numa planta industrial.

TANENBAUM (1995) cita ainda como outras vantagens que os sistemas distribuídos têm sobre os sistemas centralizados ou processadores independentes (*standalone*) a confiabilidade de um sistema distribuído, pois a queda de uma máquina não vai derrubar o sistema inteiro. Além disso, o “crescimento incremental”, que é a capacidade de aumentar o poder de processamento de um sistema também é outra vantagem importante dos sistemas distribuídos, juntamente com sua flexibilidade e a possibilidade de comunicação entre processos e usuários geograficamente distribuídos.

Apesar das vantagens, a complexidade dos sistemas distribuídos cresce consideravelmente, e deve ser enfocada. É necessário conhecer e entender diversos novos problemas, principalmente quanto à topologia e ao uso correto e eficiente da rede de interconexão, quanto à comunicação consistente entre processos distribuídos, quanto à sincronização e escalonamento de processos e processadores distribuídos e quanto à organização e segurança do sistema de arquivos distribuído (SILBERSCHATZ & GALVIN 2000).

## 2.2. Características

Conforme COULOURIS et al (1995), são seis as características principais de Sistemas Distribuídos:

- Compartilhamento de Recursos: A possibilidade de compartilhar recursos, como memória, impressoras,
- Expansibilidade: Um sistema pode ser extensível ou não em relação a hardware - por exemplo, na adição de periféricos, memória ou novos terminais - ou em relação a software - por exemplo, na adição de novas facilidade ao sistema operacional ou protocolos de comunicação. A expansibilidade de software passa pela documentação e especificação de uma interface pública dos principais softwares para os desenvolvedores;
- Concorrência: Quando mais de um processo tenta executar num único processador, temos concorrência. Num sistema distribuído, onde há mais de um processador, os processos podem executar realmente em paralelo. Acessos a processadores e recursos distribuídos requerem cuidados, e a sincronização deve ser bem planejada para garantir que os benefícios da concorrência não sejam perdidos (COULOURIS et al 1995);
- Escalabilidade: A escala de um sistema refere-se a quantidade de elementos que é composto e sua facilidade em ampliar essa quantidade. A principal característica de um sistema escalável é que as aplicações não precisam ser modificadas quando a escala do sistema aumentar. Mais ainda, o aumento da escala não deve gerar conflitos de hardware ou diminuição na performance do sistema;
- Tolerância a Falhas: Falhas em sistemas computacionais, oriundas de hardware ou software, podem levar programas a se tornarem indisponíveis ou a produzirem resultados incorretos. “Um sistema tolerante a falhas deve ser capaz de detectar falhas e de reconfigurar o sistema” (SILBERSCHATZ & GALVIN 2000). Um sistema tolerante a falhas é alcançado através de duas abordagens principais: redundância de hardware e softwares especiais para cobrir falhas;
- Transparência: Transparência significa fazer com que usuários utilizem o sistema como se estivessem num único computador, sem a preocupação sobre a distribuição

de cada componente. Existem modos de transparência em particular que são desejáveis em um sistema distribuído, são elas: Transparência de acesso, Transparência de localidade, Transparência de replicação, Transparência à falhas, Transparência de migração e Transparência de escala.

### 2.3. Classificação

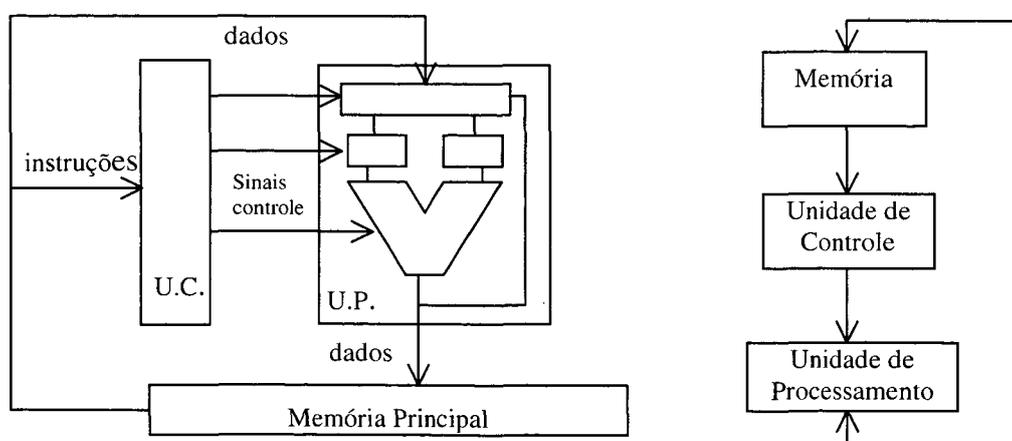
Existem várias maneiras de se classificar um sistema distribuído. As mais comuns dizem respeito ao modo como os processadores interagem entre si e quanto ao modo de processamento.

#### 2.3.1. Quanto ao Modo de Processamento

Uma das classificações quanto aos sistemas computadorizados por desenvolvida por M. J. Flynn, que os classifica de acordo com seu modo de processamento em quatro categorias:

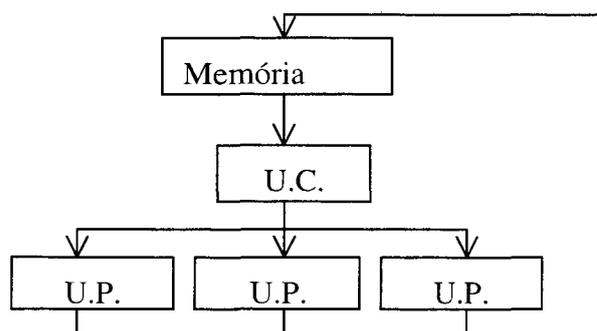
- **SISD** (*Single-Instruction Single-Data stream*): Um computador SISD é uma tradicional máquina de Von Neuman, que executa apenas uma instrução por vez, seqüencialmente (figura 2.1);

Figura 2.1 – Representações de uma máquina SISD



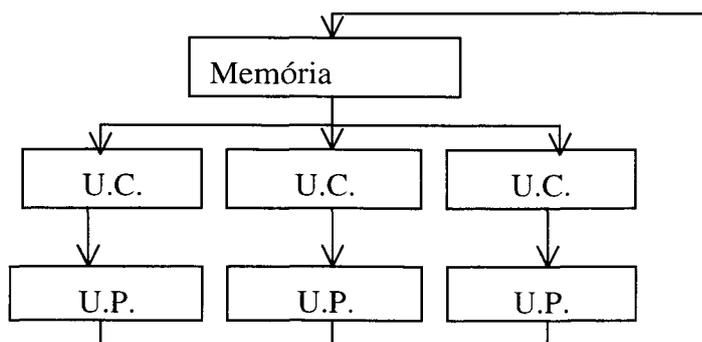
- **MISD** (*Multiple-Instruction Single-Data stream*): Um computador MISD é capaz de executar múltiplas instruções e possui apenas uma unidade de processamento. Apesar de ser uma configuração possível não há exemplos reais desses computadores;
- **SIMD** (*Single-Instruction Multiple-Data stream*): Um computador SIMD possui normalmente uma única unidade de controle (U.C.) que busca e decodifica instruções, mas múltiplas unidades de processamento (U.P.). Exemplos são os processadores vetoriais (figura 2.2);

Figura 2.2 – Representação de uma máquina SIMD



- **MIMD** (*Multiple-Instruction Multiple-Data stream*): Um sistema MIMD possui múltiplas unidades de controle e múltiplas unidades de processamento e são verdadeiramente paralelos. Exemplos são os multiprocessadores, multicomputadores e demais arquiteturas paralelas (figura 2.3);

Figura 2.3 – Representação de uma máquina MIMD

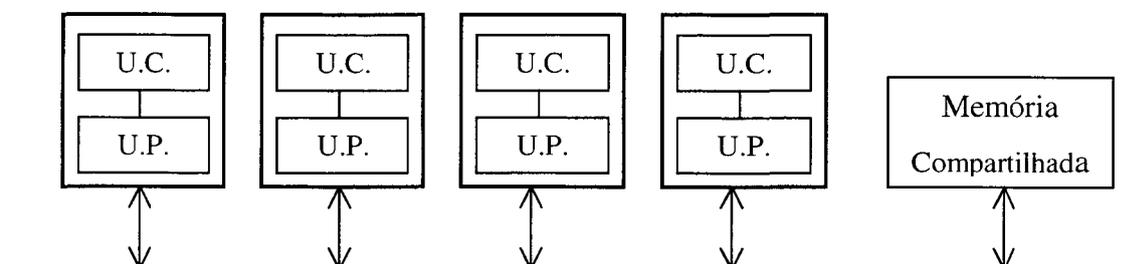


### 2.3.2. Quanto ao Modo de Interação

Em relação ao modo de interação, os sistemas paralelos podem ser classificados de acordo com o grau de acoplamento e a natureza da intercomunicação entre os processadores. Ressalta-se que um multiprocessador com memória compartilhada não é considerado um sistema distribuído.

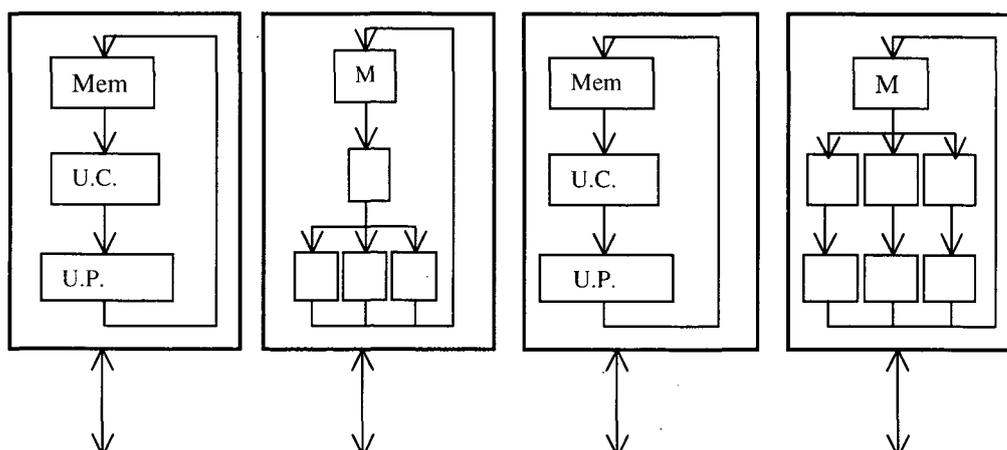
- **Multiprocessadores:** Possuem memória compartilhada e todos os processadores “enxergam” um único espaço de endereçamento virtual (figura 2.4);

Figura 2.4 – Representação de um multiprocessador conectado por barramento



- **Multicomputadores:** Cada processador possui sua própria memória e espaço de endereçamento. Não há memória compartilhada (figura 2.5);

Figura 2.5 – Representação de um multicomputador conectado por barramento



TANENBAUM (1995) cita outra maneira de classificar os sistemas distribuídos ainda quanto ao modo de interação os apresenta também em dois grupos:

- Fortemente Acoplados: “Em um sistema fortemente acoplado, o retardo ocasionado pelo envio de uma mensagem de uma máquina para outra é baixo, e a velocidade de transmissão é alta.” (TANENBAUM 1995). Outras características são: Memória compartilhada e sistema operacional em comum, além da distribuição dinâmica da carga. Os multiprocessadores tendem a ser fortemente acoplados e essa normalmente é a classificação dos sistemas paralelos;
- Fracamente Acoplados: Em um sistema fracamente acoplado, ocorre o oposto: “o retardo experimentado pela transmissão de mensagens é alto, e a velocidade de transmissão é baixa”. Algumas características apresentadas pelos sistemas fracamente acoplados são: Elementos autônomos, processadores dedicados a uma tarefa, comunicação no nível de dados e balanceamento estático da carga. Os multicomputadores tendem a ser fracamente acoplados e essa normalmente é a classificação dos sistemas distribuídos.

#### **2.4. Redes de Interconexão**

Para a implantação da Computação Distribuída é fundamental uma arquitetura de rede robusta e eficiente, que seja confiável para a transmissão das informações e suporte o sistema e as aplicações instaladas. FENG (1981) identifica basicamente quatro decisões de projeto acerca das redes de interconexão:

- Modo de Operação: Pode ser síncrona ou assíncrona. Um sistema distribuído deve prover suporte aos dois modos de operação;
- Estratégia de Controle: O controle da rede de interconexão pode ser feito através de um único elemento, numa estratégia centralizada, ou através de alguma estratégia distribuída;

- Metodologia de Chaveamento: As principais metodologias são o chaveamento por pacote e o chaveamento por circuito. Em geral o chaveamento por pacote é mais eficiente para mensagens curtas e o chaveamento por circuito para mensagens muito grandes;
- Topologia da Rede: A topologia da rede de interconexão é um fator determinante no projeto de um sistema distribuído. As topologias regulares podem ser estáticas ou dinâmicas;

Além destas decisões, deve-se lembrar que o desempenho de uma rede de interconexão tem aumentado muito nos últimos anos. PATTERSON (1998) lembra que "a banda passante de pico por *link* melhorou, em uma década, de 0,5 MB/segundo no Intel iPSC (1986) para 800 MB/segundo no SGI origin (1996)". Conforme CORRÊA (1998), o desempenho depende basicamente de quatro fatores:

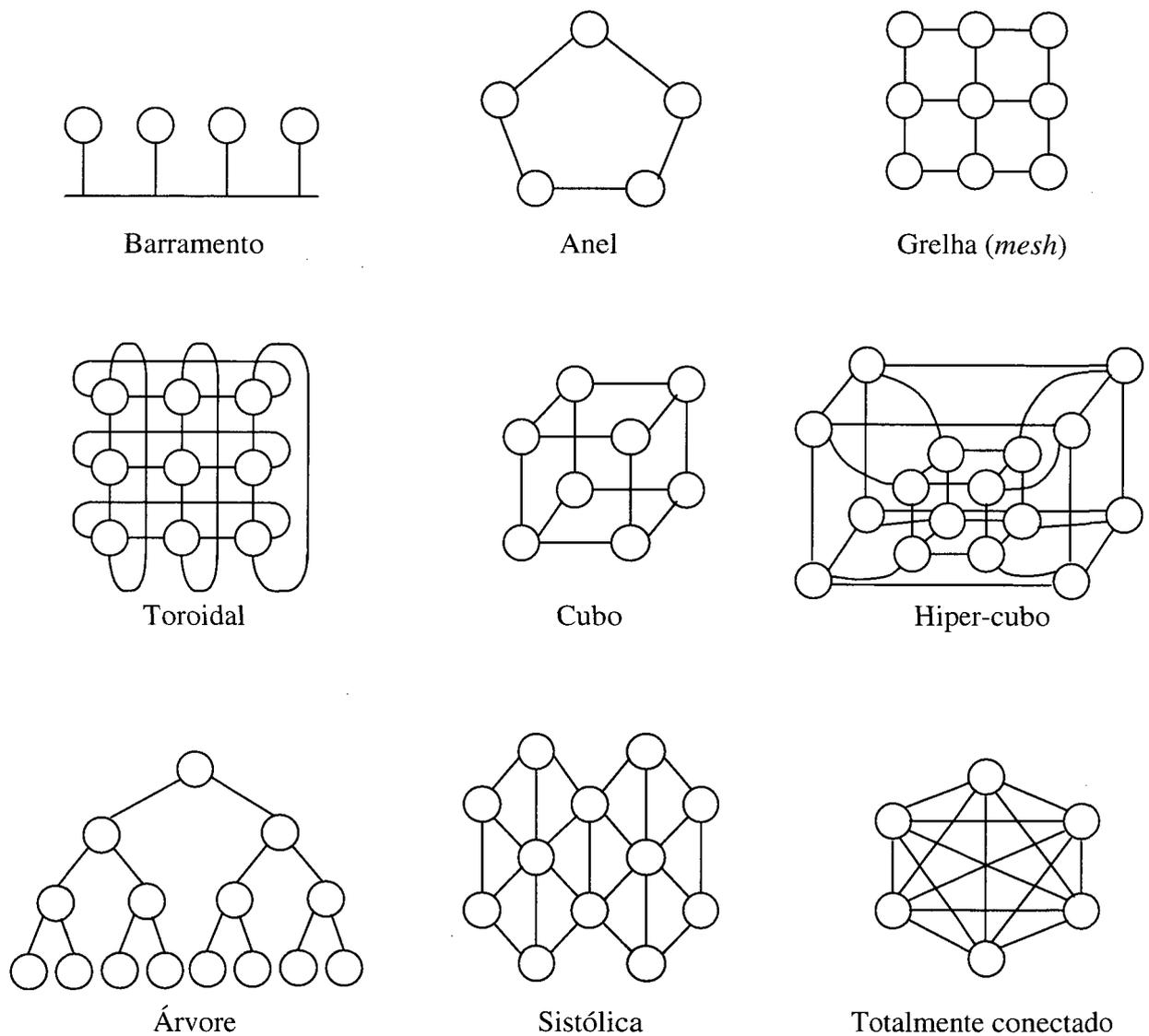
- Funcionalidade: Refere-se a como a rede suporta o roteamento de dados e a manipulação de interrupções, entre outros.;
- Largura de Banda: Determina o maior atraso na transmissão de uma unidade de mensagem;
- Complexidade de Hardware: Refere-se ao equipamento que é necessário para implementar a rede física;
- Escalabilidade: Capacidade da rede de aumentar sua performance com a inclusão de novos recursos na máquina;

As topologias de redes de interconexão podem ser representadas por um grafo, onde os vértices são os nós processadores, e as arestas são as vias de comunicação entre dois nós. "A topologia da rede de interconexão é muito importante na construção de uma máquina que pode vir a ter de 100 a 10.000 nós" (PATTERSON 1998). As principais topologias utilizadas são apresentadas a seguir:

### 2.4.1. Topologias Estáticas

Topologias estáticas são aquelas em que as conexões entre os elementos são estabelecidas em tempo de projeto e permanecem fixas. Entre as topologias estáticas, as mais comuns são apresentadas na figura 2.6.

Figura 2.6 – Topologias de redes de interconexão estáticas

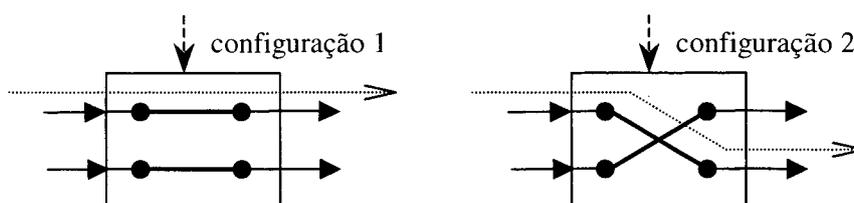


## 2.4.2. Topologias Dinâmicas

As topologias dinâmicas permitem modificar os nodos conectados dinamicamente, alterando a estrutura da rede, através de elementos comutadores (*switches*). FENG (1981) divide as topologias dinâmicas em três categorias:

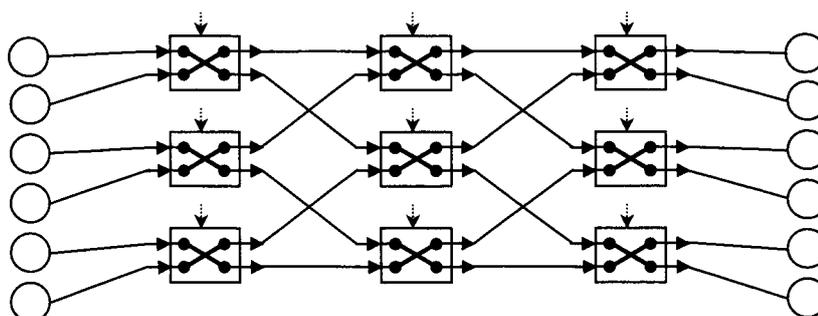
- Comutador de único-estágio: Elemento básico de comutação entre duas entradas e duas saídas. Um bit controla a configuração do comutador (figura 2.8). Não é usado isoladamente, mas em conjuntos;

Figura 2.7 – Elemento comutador de estágio único



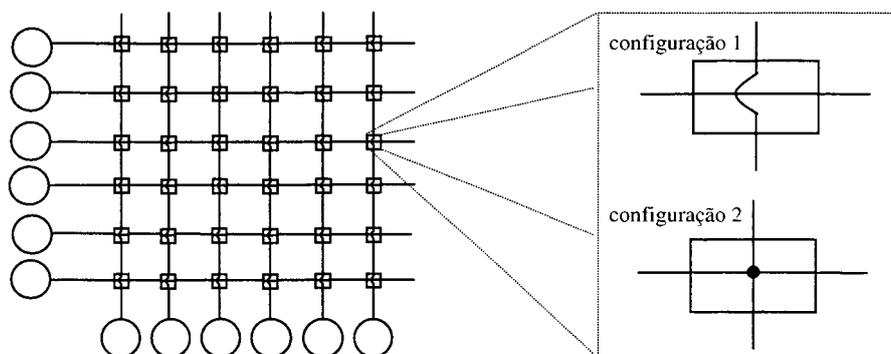
- Comutadores multi-estágio: É uma rede formada por um conjunto de comutadores agrupados em  $n$  estágios interconectados por meio de ligações fixas. Os comutadores são configurados dinamicamente e permitem o estabelecimento direto entre qualquer nó. Além do tamanho dos comutadores e dos padrões das conexões interestágios, este tipo de rede pode ter controle centralizado ou distribuído (figura 2.9);

Figura 2.8 – Rede dinâmica com comutadores multi-estágio



- **Crossbar:** Uma rede *crossbar* é formada por  $n^2$  comutadores que permitem conectar diretamente, sem roteamento ou contenção,  $n$  nós processadores. O *crossbar* é a rede de interconexão com as maiores largura de banda, conectividade e capacidade de roteamento, porém com um custo de implementação mais elevado (figura 2.10).

Figura 2.9 – Rede dinâmica tipo *crossbar*



Sobre os comutadores, ATHAS (1988) alerta que “A menos que os projetistas de multiprocessadores descubram métodos de implementar comutadores de rede com menor latência ou desenvolvam processadores que sejam menos sensíveis à latência de acesso à memória, o aumento de multiprocessadores para faixa de milhares de nodos será impraticável”.

## 2.5. Resumo

Os sistemas distribuídos possuem muitas vantagens em relação aos sistemas centralizados, como o compartilhamento de recursos e processamento paralelo, sendo que a principal desvantagem está no gerenciamento e na complexidade de tais sistemas. As principais características dos sistemas distribuídos são: Compartilhamento de recursos, Expansibilidade, Concorrência, Escalabilidade, Tolerância a falhas e Transparência.

Flynn classifica os sistemas computacionais conforme a quantidade de unidades de controle e unidades de processamento, em quatro classes: SISD, SIMD, MISD, MIMD, sendo que os sistemas MIMD são realmente paralelos. Outra classificação

separa os sistemas paralelos em Multiprocessadores (fortemente acoplado) e Multicomputadores (fracamente acoplado). Por possuir memória compartilhada entre os processadores, um multiprocessador não é considerado um sistema distribuído, apenas paralelo. Os multicomputadores são, efetivamente, sistemas paralelos distribuídos.

Num sistema distribuído, o modo como os processadores são conectados, ou seja, a rede de interconexão, é muito importante. O projeto da rede de interconexão exige principalmente quatro decisões quanto a: O modo de operação (síncrono ou assíncrono), A estratégia de controle (centralizada ou distribuída), A metodologia de chaveamento (pacote ou circuito) e a Topologia da Rede (estática ou dinâmica).

A topologia da rede de interconexão pode ser estática, se as conexões entre os processadores forem definidas em tempo de projeto, ou dinâmica, se as conexões entre os processadores forem estabelecidas dinamicamente durante o funcionamento do sistema. Entre as topologias dinâmicas está o *crossbar* (barramento cruzado), com grande desempenho, e que usa  $n^2$  comutadores para comunicar  $n$  processadores diretamente, sem roteamento ou contenção de mensagens.

Deste modo, torna-se compreensível a informação de que o CRUX, objeto deste trabalho, é um sistema paralelo e distribuído, tipo multicomputador (com processadores Intel80x86), com uma rede de interconexão dinâmica tipo *crossbar* e uma rede de serviços (controle) tipo barramento. Esse sistema, com grande capacidade de processamento, poderia ser utilizado, entre muitas aplicações científicas, como plataforma para sistemas de tempo real, que são discutidos no Capítulo 3.

## 3. Sistemas de Tempo Real (STR)

### 3.1. Introdução

STANKOVIC (1992) define um Sistema de Tempo Real (STR) como sendo um sistema no qual a ‘corretude’ do sistema depende não apenas dos resultados lógicos da computação, mas também do tempo no qual os resultados são produzidos.

BUTTAZZO (1997) ainda acrescenta que “sistemas em tempo real devem reagir sob restrições de tempo precisas a eventos do ambiente. Uma reação que ocorre muito tarde pode ser inútil ou mesmo perigosa. (...) A palavra real [de tempo real] indica que a reação do sistema a eventos externos deve ocorrer *durante* sua evolução. Como consequência, o tempo no sistema (tempo interno) deve ser medido usando a mesma escala de tempo usada para medir o tempo no ambiente controlado (tempo externo)”.

STANKOVIC (1992a) faz, ainda, um comentário interessante: “Sistema de tempo real é diferente de computação muito rápida”, indicando que existem outras restrições aplicadas aos STR além da grande necessidade de desempenho.

São alguns exemplos de áreas onde STR são necessários:

Controle de Tráfego Aéreo. “Além da necessidade de segurança, há a necessidade de uma base científica para tratamento [e representação] correto do tempo, funcionalidade e dependências temporais” (STANKOVIC 1992).

Sistemas de Controle de Processos, que incluem centenas de aplicações industriais, controle de equipamentos ou robótica. O controle de um processo químico ou um reator nuclear pode exigir tempos de reação muito pequenos (e rígidos), sendo que uma operação incorreta do sistema pode trazer sérios prejuízos.

Simuladores e Ambientes Virtuais, que exigem alta capacidade de processamento e interatividade com grandes restrições de tempo.

Sistemas Militares. Entre várias aplicações há o controle de armamentos como a trajetória dinâmica de um míssil ou jatos automatizados.

Biomédica, área em grande crescimento e onde a vida de um ser humano pode depender do correto funcionamento do sistema e de respostas rápidas e confiáveis.

### **3.2. Características**

Para que sistema possa atender a todas as restrições impostas, é necessário que eles possuam algumas características especiais. De acordo com STANKOVIC (1992a) a próxima geração deve ser de sistemas dinâmicos e flexíveis.

As restrições sobre o tempo de computação de tarefas num STR provêm do impacto físico que isso pode causar no ambiente. Esse tempo limite é chamado *deadline*. “Uma tarefa tem uma *deadline* no qual ele deve terminar ou começar [sua execução], ou pode haver restrições [*deadlines*] tanto para o tempo de início de execução quanto para seu término.” (STANKOVIC 1992).

Um STR interage de forma íntima e fortemente acoplada ao seu ambiente. Uma tarefa ativada por um estímulo do ambiente deve ser completada até seu prazo final (*deadline*).

BUTTAZZO (1997) diz que “a nível de processo [tarefa], a principal diferença entre uma tarefa de tempo real e uma tarefa comum é que a tarefa de tempo real é caracterizada por uma *deadline*, que é o tempo máximo no qual ela deve terminar sua execução”.

De modo geral, uma tarefa de tempo real difere de uma tarefa normal por apresentar três tipos de restrições:

- Temporais: A atividade deve completar sua execução antes de sua *dealine* para evitar danos ao sistema. Os efeitos desses danos podem ser desde a falta de sincronismo numa conferência em tempo real, até o erro de um alvo militar estratégico numa guerra;
- Precedência: Em certas aplicações algumas tarefas podem executar em paralelo (concorrente), e há aquelas que devem aguardar o término da execução de outra tarefa para poder executar. O ST deve considerar as ordens de precedência entre tarefas para efeitos de escalonamento e sincronização;

- Alocação de Recursos: Quando duas ou mais tarefas acessam dados de recursos exclusivos, deve-se garantir a exclusão mútua, bloqueando outras tarefas quando uma delas estiver acessando esses dados (seção crítica). Um ST deve garantir que tarefas prioritárias fiquem bloqueadas e percam sua *deadline* aguardando por tarefas menos prioritárias, causando a inversão de prioridade.

Algumas propriedades básicas que devem estar presentes num STR são as seguintes:

- Timeliness. “Resultados devem ser corretos não apenas nos seus valores mas também no domínio do tempo. Como consequência, o sistema operacional deve prover mecanismos específicos no *kernel* para gerenciar e controlar tarefas com restrições temporais explícitas e com níveis de prioridade diferentes” (BUTTAZZO 1997).
- Projeto para o Pior Caso. Ainda segundo BUTTAZZO (1997), “sistemas de tempo real não devem colapsar quando estão sujeitos a condições de sobrecarga”. Assim, eles devem ser projetados para funcionarem bem no pior momento, garantindo deste modo, o bom funcionamento em todas as outras situações.
- Preditabilidade. Para aumentar a performance e atender às restrições impostas, o sistema deve ser apto a prever as consequências de qualquer ação de escalonamento, e deve tirar proveito de todo o conhecimento disponível sobre o ambiente de modo a prever seu comportamento e tratá-lo de maneira mais adequada.
- Tolerância a Falhas. “Simples falhas de hardware ou software não devem causar a queda do sistema. Assim, componentes críticos de um sistema de tempo real devem ser projetados para serem tolerantes a falhas” (BUTTAZZO 1997).

Além disso, num STR, o mecanismo de tratamento de interrupções deve ser integrado ao mecanismo de escalonamento de processos, para que as rotinas tratadoras de interrupção possam ser escalonadas juntamente com qualquer outro processo, a fim de manter as garantias necessárias quanto ao tempo de execução dos processos e evitar indeterminismo.

### 3.3. Classificação

Os STR podem ser classificados facilmente em duas categorias básicas, de acordo com as consequências que podem ocorrer com a perda de uma *deadline*.

- “Uma tarefa de tempo real é dita *hard* [*hard real-time*] se a perda de sua *deadline* puder causar consequências catastróficas no ambiente sob controle”, (BUTTAZZO 1997);
- “Uma tarefa de tempo real é dita *soft* [*soft real-time*] se o cumprimento de sua *deadline* é desejável por motivos de performance, mas a perda da *deadline* não causa sérios prejuízos ao ambiente”, (BUTTAZZO 1997).

Em relação à ativação, as tarefas podem ser classificadas em *periódicas* ou *aperiódicas*. Basicamente, as tarefas periódicas são ativadas sempre a cada período de ‘T’ unidades de tempo. Já o momento da ativação de tarefas aperiódicas é imprevisível.

### 3.4. Sistema Operacional de Tempo Real (SOTR)

#### 3.4.1. Introdução

Um sistema operacional em um sistema de tempo real (SOTR) é um componente muito crítico, pois ele é o responsável pelo gerenciamento dos recursos da máquina, e deve prover mecanismos que permitam que as restrições impostas aos sistemas de tempo real sejam cumpridas.

O *kernel*, ou núcleo, é a parte mais interna do Sistema Operacional (SO), que tem conexão direta com o hardware, e normalmente prove alguns serviços básicos:

- Gerenciamento de Processos: Inclui funções como criação e término de processos, escalonamento, despacho, chaveamento de contexto e outras. Cada um desses serviços deve satisfazer as restrições impostas por um STR, como um rápido chaveamento de contexto, que normalmente é um dos maiores *overheads* do SO, a

representação das características de TR nas tabelas de processos e um escalonamento adequado ao sistema;

- Gerenciamento de Memória: Inclui funções de alocação e desalocação de trechos de memória, compartilhamento e cópia de trechos de memória, entre outras. O gerenciamento de memória costuma ser simples, utilizando normalmente um dos mecanismos básicos (partições fixas, variáveis, *buddy*, *etc*) mais adequado ao problema em particular. O mecanismo de memória virtual (paginação, segmentação) não é utilizado devido ao indeterminismo introduzido pelas faltas de página/segmentos e acessos ao disco para *swapping*. Na realidade, a própria cache introduz certo grau de indeterminismo;
- Tratamento de Interrupções: Permite o reconhecimento e tratamento de interrupções de hardware geradas por dispositivos de E/S. Num STR, as interrupções devem ser tratadas como tarefas e têm que ser escalonadas como tal, de tal forma que o efeito de muitas chamadas de interrupções e reentrância possam ser controlados, sem causar indeterminismo no tempo de execução das demais tarefas ou a possível perda de suas *deadlines*;
- Sincronização e Comunicação de Processos: Inclui mecanismos de bloqueio e desbloqueio de processos e troca de mensagens, essenciais a qualquer sistema multitarefa, principalmente distribuído, e também protocolos de acesso a recursos, que devem ter uma atenção especial em STR..

Esses serviços básicos, realizados por qualquer bom *kernel* de um sistema operacional, possuem restrições e características adicionais para que suportem aplicações em tempo real, como sincronização de relógios, tolerância a falhas e previsibilidade.

A seguir estão apresentados os principais aspectos de tempo real em um sistema operacional.

### 3.4.2. Processos

O gerenciamento de processos num SOTR deve ser eficiente. Por exigir alta performance, o *overhead* relacionado com gerenciamento de processos e troca de contexto deve ser minimizado.

Processos de tempo real possuem uma *deadline* associada, e outros atributos que permitem representar as restrições de tempo real ao qual estão submetidos e que são consideradas relevantes pelo sistema. De modo geral, esses atributos são utilizados pelo escalonador de processos, que é um dos principais responsáveis pelo cumprimento das restrições de tempo real impostas.

### 3.4.3. Comunicação

Num sistema multitarefa, onde vários processos podem executar, um processo pode necessitar se comunicar com outro para acessar recursos compartilhados ou trocar informações. O Sistema Operacional deve prover mecanismos de comunicação entre processos de forma uniforme e independente de sua localização, pois os processos são possivelmente remotos, no caso de um sistema distribuído.

A comunicação normalmente se dá por uma dentre duas maneiras:

- Área compartilhada de memória: Uma área de memória é compartilhada pelos processos comunicantes. Essa estratégia exige cuidados especiais para evitar uma condição de corrida, quando o resultado de um processamento determinístico poderia ser imprevisível. É necessário garantir a exclusão mútua, espera limitada e a não-interferência. Um mecanismo muito utilizado para sincronização através de área compartilhada é o semáforo, uma variável natural (inteira não-negativa) que pode ser acessada apenas através das operações UP (não-bloqueante) e DOWN (bloqueante).
- Troca de mensagens: A comunicação e a sincronização entre os processos são alcançadas pelo envio de uma mensagem de um processo a outro, e pela recepção vinda de um processo. A troca de mensagens é feita através de duas primitivas: *send* (bloqueante ou não-bloqueante) e *receive* (bloqueante). A comunicação síncrona (bloqueante) não necessita armazenar mensagens, enquanto que a assíncrona (não-

bloqueante) exige um *buffer* ou caixa-postal para armazenar a mensagem até que ela seja recebida pelo destinatário.

Num STR, o sistema de comunicação deve ser ainda mais confiável e eficiente que em sistemas comuns.

#### 3.4.4. Sincronização

Dois aspectos de sincronização devem ser considerados num STR: A sincronização de relógios, num nível mais baixo, e a sincronização de processos, num nível mais elevado.

A sincronização de relógios é um aspecto muito importante, principalmente quando há nós distribuídos e múltiplos elementos de relógio (*clocks*). Quando um sistema possui mais de um relógio, independentes entre si, diferenças mínimas no seu cristal (responsável pela oscilação que gera o sinal do relógio) e sua construção, fazem com que, os relógios marquem o tempo com alguma diferença entre si. Isso significa que, se todos os relógios do sistema forem inicializados exatamente no mesmo instante, após algum tempo eles irão divergir quanto ao valor do tempo decorrido.

As conseqüências dessas diferenças num STR podem ser muito graves. Como um STR deve interagir muito intimamente com o ambiente, e responder rapidamente aos eventos ocorridos, uma indeterminação ou erro no valor que denota o momento da ocorrência de um evento pode fazer com que o sistema responda erroneamente.

A necessidade de sincronização dos relógios, para que reflitam o mesmo tempo, ou valores muito próximos deste, leva à existência de algoritmos de sincronização de relógios.

Outro aspecto de temporização refere-se ao período do sinal (*tick*) do temporizador (*timer*), que gera uma interrupção periódica (e deve ser tratada como uma tarefa periódica) e é responsável por executar, regularmente, o SO para verificar o sistema e fazer o escalonamento de tarefas. Ainda conforme BUTTAZZO (1997), intervalos de *tick* muito pequenos atendem melhor tarefas com alta taxa de ativação, porém causam um grande *overhead*, um dos mais significantes gerados pelo *kernel*. Intervalos de *tick* muito grandes podem não atender adequadamente à tarefas críticas. O

período do sinal (*tick*) do temporizador deve então ser ajustado ao problema em particular, sendo que valores típicos estão entre 1 e 50 ms.

A sincronização de processos é necessária quando há comunicação entre os processos e é preciso que os processos aguardem pela ocorrência de eventos para poderem continuar sua execução. Essa sincronização pode ser conseguida através de áreas compartilhadas de memória ou por troca de mensagens, como apresentado anteriormente na comunicação entre processos.

Uma dos motivos da existência de sincronismo entre os processos advém da necessidade de evitar as condições de corrida causadas por sessões críticas (dados compartilhados). Uma das restrições impostas para evitar as condições de corrida é a exclusão mútua, ou seja, apenas um processo pode executar a sessão crítica num dado instante, e o restante dos processos que também deseja executar a sessão crítica deve esperar. Num STR, o SO deve garantir que um processo mais prioritário não espere por outro menos prioritário que está executando a sessão crítica, o que provoca o fenômeno da inversão de prioridades.

### **3.5. Resumo**

Sistemas de tempo real são aqueles que devem reagir sob restrições de tempo precisas a eventos do ambiente, e onde a conclusão tardia de uma tarefa pode ter conseqüências graves. Suas principais características são: Restrições temporais, Projeto para o pior caso, Preditabilidade e Tolerância a Falhas. Em relação às tarefas, num STR, elas possuem restrições Temporais (*deadline*), de Precedência e de Alocação de Recursos.

Os STR podem ser *hard* se a perda de sua *deadline* puder causar conseqüências catastróficas no ambiente sob controle, ou *soft* se a perda da *deadline* produz apenas perda de desempenho e não problemas mais sérios.

Um SOTR deve possuir inúmeras características para prover suporte às restrições de TR. São elas: Ser Multitarefa; Ter Escalonamento baseado em prioridade; Ser preemptivo; Ter a habilidade de responder rapidamente a interrupções externas; Possuir um *Microkernel* com funcionalidade mínima; Realizar a troca de contexto muito

rapidamente; Prover alarmes especiais e *timeouts*; Prover arquivos seqüenciais rápidos para armazenar dados; Abandonar o uso memória virtual ou *swap* de disco; Fazer a sincronização de relógios; etc.

Entre os principais componentes de um SOTR está o escalonador de processos, responsável por escolher qual tarefa deve executar em cada instante, de modo a garantir as restrições de TR. Esse assunto é tratado em maior profundidade no Capítulo 4.

## 4. Escalonamento de Tempo Real

### 4.1. Introdução

O escalonador (*scheduler*) é o componente do SO responsável pelo agendamento de processos, ou seja, determinar a ordem de execução dos processos prontos a executar. Esses processos permanecem em uma fila até serem escolhidos pelo escalonador, que os despacha ao processador (aloca o processador ao processo) que pode, então, executar suas instruções.

Para proteger o sistema e impedir que um processo mais importante espere na fila por outro menos importante que está executando, o Sistema Operacional deve ter a capacidade de interromper a execução de um processo (através de uma interrupção do *timer*), retirá-lo do processador (colocando-o novamente na fila) e escolher outro processo da fila para executar. À capacidade do escalonador de interromper a execução de processos em execução, damos o nome de preempção. Em sistemas de TR é essencial que o escalonamento seja preemptivo, o que melhora o desempenho do sistema e pode diminuir consideravelmente a taxa de perdas de *deadline*.

BUTTAZZO (1997) fornece uma definição mais formal:

“Suponha três conjuntos: um conjunto de  $n$  tarefas  $J = \{J_1, J_2, \dots, J_n\}$ , um conjunto de  $m$  processadores  $P = \{P_1, P_2, \dots, P_m\}$  e um conjunto de  $s$  tipos de recursos  $R = \{R_1, R_2, \dots, R_s\}$ . Além disso, relações de precedência entre tarefas podem ser especificadas através de um grafo acíclico dirigido, e restrições temporais podem ser associadas a cada tarefa. Neste contexto, escalonamento significa associar processadores de  $P$  e recursos de  $R$  a tarefas de  $J$  de modo a completar todas as tarefas sob as restrições impostas”.

Muitos algoritmos de escalonamento podem ser idealizados, sendo que os escalonadores para sistemas de TR devem ser elaborados considerando todas as restrições impostas a tais sistemas, objetivando maximizar o fator de utilização do

processador, minimizar a perda de *deadlines*, o tempo de resposta e a latência dos processos.

#### 4.2. Classificação dos Algoritmos de Escalonamento

Os algoritmos de escalonamento podem ser classificados sob vários aspectos. TANENBAUM (1995) :

- Heurístico ou Determinístico: Um algoritmo determinístico é apropriado quando se conhece, de antemão, o comportamento do sistema (preditabilidade). Quando o comportamento é variável e a carga de trabalho imprevisível, um algoritmo heurístico é mais adequado.
- Centralizado ou Distribuído: Um algoritmo de escalonamento é distribuído quando é composto por processos distribuídos que se comunicam.
- Ótimo ou Subótimo: Um algoritmo é dito ótimo em relação a certo indicador de desempenho se tal algoritmo maximiza ou minimiza tal indicador.
- Global ou Local: Um algoritmo de escalonamento é Local se considera apenas os processos e o processador do computador no qual está executando. Um algoritmo Global gerencia os processos entre processadores, visando balancear a carga do sistema.

#### 4.3. Caracterização das Tarefas de Tempo Real

Toda a tarefa de tempo real pode ser representada por um conjunto de atributos que a representam, conforme as necessidades de um STR específico. As características representadas podem diferir conforme o tipo da tarefa (*hard* ou *soft*) ou conforme as restrições impostas pelo algoritmo de escalonamento ao qual são submetidas.

No contexto do presente trabalho, foram representadas as características impostas pelos principais algoritmos de escalonamento de TR desenvolvidos, como mostrado a seguir.

O sistema deve atender um conjunto  $\tau = \{\tau_1, \tau_2, \dots, \tau_b, \dots, \tau_n\}$  de  $n$  tarefas. Cada tarefa  $\tau_i$  é representada como  $\tau_i = (CR_i, a_i, C_i, D_i, T_i, v_i)$ , onde:

- $CR_i$  refere-se ao Tipo da tarefa. Pode ser *soft* ou *hard*;
- $a_i$  é o Tempo de Ativação (ou chegada), instante de tempo em que a tarefa entra no sistema e se torna pronta para a execução. Em tarefas periódicas, esse valor representa a chegada da primeira instância (job) da tarefa, e nesse caso também é chamado de fase ( $\phi$ ) da tarefa;
- $C_i$  é o Tempo Máximo de Execução da tarefa;
- $D_i$  é a Deadline relativa da tarefa;  $D_i \geq C_i$  (1)
- $T_i$  é o Período de Ativação da tarefa. Se for uma tarefa aperiódica, tem-se  $T_i = \infty$ ;  
 $T_i \geq C_i$  (2)
- $vi$  é um valor que representa a importância relativa da tarefa em relação às outras do conjunto.

No caso de uma tarefa periódica, composta pela ativação de uma instância (denominada job) a cada período  $T_i$ , designa-se a  $j$ -ésima instância ou *job* da tarefa  $i$  por  $\tau_{i,j}$ .

Alguns indicadores também auxiliam na caracterização das tarefas:

- $b_i$  é o instante de tempo no qual a tarefa inicia sua execução;  $b_i \geq a_i$  (3)
- $R_i$  representa o Tempo de Resposta da tarefa;  $R_i = b_i - a_i$  (4)
- $F_i$  é o intervalo de tempo (relativo à chegada  $a_i$ ) no qual a tarefa termina completamente sua execução;
- $f_i$  representa o Tempo Absoluto de Término, que é o instante de tempo no qual a tarefa termina completamente sua execução;  $f_i = a_i + F_i$  (5)
- $d_i$  representa a Deadline absoluta da tarefa.  $d_i = a_i + D_i$  (6)
- $L_i$  é a Latência, o atraso de uma tarefa com respeito à *deadline*. É negativo se a tarefa atendeu a *deadline* e positivo e a perdeu.  $L_i = f_i - d_i$  (7)

- $X_i$  representa a Laxidade, que é o tempo em que uma tarefa poderia esperar sem perder sua *deadline* (*slack time*). 
$$X_i = d_i - C_i - a_i \quad (8)$$

Apenas como exemplo, uma tarefa aperiódica *soft* poderia ser descrita como  $\tau_1 = (\text{soft}, 0, 5, 10, \infty, 0)$ , ou seja, uma tarefa *soft real-time*, com tempo de ativação 0, que irá executar por até 5 u.t. e tem *deadline* de 10 u.t. (contados a partir da ativação), é aperiódica (período infinito) e tem importância relativa 0. Uma outra tarefa periódica *hard* poderia ser descrita como  $\tau_2 = (\text{hard}, 2, 15, 20, 20, 0)$ .

#### 4.4. Precedência de Tarefas e Alocação de Recursos

Em muitas aplicações de sistemas multitarefa, por exemplo, é necessário que as tarefas (*threads*) se sincronizem, e assim estabeleçam algumas relações de precedência entre si, ou seja, duas ou mais tarefas não podem executar de modo arbitrário, mas numa seqüência tal que respeite as precedências impostas pela sincronização.

Essas relações de dependência podem ser representadas através de um grafo acíclico dirigido, onde os nodos são tarefas e as arestas são as relações de precedência (BUTTAZZO 1997). O escalonador de tempo real deve levar em consideração a precedência de tarefas de tal modo garantir o cumprimento das *deadlines* e demais restrições dos STR.

Como exemplo do efeito da precedência de tarefas, considere o seguinte conjunto de tarefas:  $\tau = \{\tau_1(\text{hard}, 0, 5, 6, \infty, 1), \tau_2(\text{hard}, 4, 3, 4, \infty, 2), \tau_3(\text{hard}, 6, 4, 5, \infty, 0)\}$  também representado no quadro 4.1 (lembrando que um valor numérico menor no campo 'Importância' significa maior prioridade) e também o seguinte grafo de precedência:



onde " $\rightarrow$ " significa "é predecessor imediato de".

Nessas condições, um escalonador que não considere a precedência de tarefas, poderia preemptar  $\tau_2$  para executar  $\tau_3$  (mais importante), embora  $\tau_3$  tenha que esperar  $\tau_2$  terminar sua execução. Poderia também deixar de atender  $\tau_2$  para continuar atendendo  $\tau_1$  (mais importante que  $\tau_2$ ), o que causaria a perda da *deadline* de  $\tau_2$  e conseqüentemente de  $\tau_3$ , que é mais importante que  $\tau_1$ .

Quadro 4.1 – Conjunto de tarefas do exemplo

Tarefa	Tipo	Chegada	Computação	Deadline	Período	Importância
$\tau_1$	<i>hard</i>	0	5	6	$\infty$	1
$\tau_2$	<i>hard</i>	4	2	3	$\infty$	2
$\tau_3$	<i>hard</i>	6	4	5	$\infty$	0

Já um escalonador que considere as relações de precedência, deveria sacrificar (preemptar) a tarefa  $\tau_1$  (prioridade média) e atender a tarefa  $\tau_2$  (prioridade baixa) para que sua tarefa sucessora  $\tau_3$  (prioridade alta) possa ser atendida com sucesso.

Além da precedência, muitas tarefas também necessitam de recursos de vários tipos para poderem executar. “Do ponto de vista da tarefa, um recurso é qualquer estrutura de software que possa ser usada pela tarefa para avançar sua execução” (BUTTAZZO 1997). Um recurso pode ser compartilhado se várias tarefas podem utilizá-lo simultaneamente, ou exclusivo, se puder ser usado por uma única tarefa. As tarefas que aguardam um recurso exclusivo ser liberado são ditas bloqueadas. Essa situação pode levar ao fenômeno da Inversão de Prioridade, onde uma tarefa altamente prioritária fica bloqueada aguardando que outra tarefa menos prioritária libere um recurso necessário. Além disso, deve-se evitar *deadlocks* e o também limitar o tempo em que uma tarefa permanece bloqueada, o que produz indeterminismo no tempo de execução.

Com a finalidade de resolver esses problemas, vários algoritmos de alocação de recursos foram propostos, como o Protocolo de Herança de Prioridade (*Priority Inheritance Protocol – PIP*) e o Protocolo de Limite de Prioridade (*Priority Ceiling Protocol – PCP*), ambos propostos por SHA, RAJKUMAR e LEHOCZKY (1990).

## 4.5. Algoritmos de Escalonamento de Tempo Real

### 4.5.1. Taxa Monotônica (RM –Rate Monotonic)

Esse algoritmo faz o escalonamento de tarefas periódicas, e simplesmente associa à prioridade de cada tarefa a sua taxa de ativação. Ou seja, quanto menor o intervalo entre ativações da tarefa, maior será sua prioridade. O algoritmo escalona a tarefa com maior prioridade. Em 1973, Liu e Laylan mostraram que esse algoritmo é ótimo em relação aos algoritmos de prioridade fixa.

A escalonabilidade de um conjunto de  $n$  tarefas através do algoritmo RM pode ser verificada através das seguintes condições:

$$U = \sum_{i=1}^n \frac{C_i}{T_i} \quad (9)$$

$$U_{tub} = n(2^{1/n} - 1) \quad (10)$$

$$U \leq U_{tub} \quad (11)$$

A equação (9) descreve o “Fator de Utilização da CPU”, que é dado pelo somatório da relação entre o tempo de computação e o período de ativação de cada tarefa. Esse é um indicador importante, pois obviamente se  $U > 1$ , o conjunto de tarefas não é escalonável.

A equação (11) fornece uma condição suficiente mas não necessária à escalonabilidade do conjunto da tarefas pelo algoritmo RM, ou seja, “se um conjunto de tarefas tem um fator de utilização maior que  $U_{tub}$  e é menor que um (unidade), nada se pode afirmar sobre a escalonabilidade do conjunto” (BUTTAZZO 1997).

Como um exemplo de escalonamento através de RM e da aplicação da fórmula acima, considere o seguinte conjunto de tarefas:  $\tau_1=(hard, 0, 1, 3, 3, 0)$  e  $\tau_2=(hard, 3, 2, 4, 4, 0)$ . Assim, tem-se que:

$$n=2, C_1=1, T_1=3, C_2=2, T_2=4, \quad e \quad U = \sum_{i=1}^2 \frac{C_i}{T_i} = \frac{1}{3} + \frac{2}{4} = 0,8333 \quad e$$

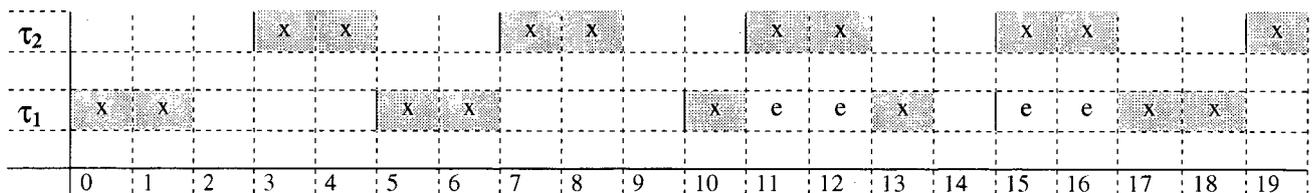
$$U_{ub} = n(2^{1/n} - 1) = 2(2^{1/2} - 1) = 0,8284$$

Sendo a equação  $U \leq U_{ub}$ , condição suficiente mas não necessária, se a condição for satisfeita, podemos garantir a escalonabilidade do conjunto de tarefas. Porém, se a condição não for satisfeita, ou seja,  $U_{ub} < U \leq 1$ , nada podemos afirmar, e se  $U > 1$ , podemos garantir que as tarefas não são escalonáveis.

Como neste exemplo temos  $1 > U > U_{ub}$ , nada podemos afirmar sobre sua escalonabilidade. Porém, conforme a figura 4.1, que apresenta a simulação do escalonamento dessas tarefas com o algoritmo RM, pode-se perceber que esse conjunto de tarefas é escalonável.

Os quadros 4.2 e 4.3 apresentam, respectivamente o algoritmo de escalonamento RM e o algoritmo de garantia de escalonabilidade para o RM.

Figura 4.1 – Escalonamento de tarefas com RM



Legenda: 

x
---

 : Tarefa executando

e
---

 : Tarefa esperando para executar

Quadro 4.2 – Algoritmo de escalonamento RM

## Algoritmo Taxa Monotonica – RM

```

void scheduler (void) {
    // A fila 'ReadyQueue' mantém os elementos (tarefas  $\tau$ ) por ordem crescente de Período(T)
    //  $\tau_{exec}$  representa a tarefa executando no momento
    // NQ( ) retorna a quantidade de elementos na fila (Number in Queue)
    if NQ(ReadyQueue) > 0) {
        if ( $\tau_{exec}.T > \text{Primeiro\_Elemento}(\text{ReadyQueue}).T$ ) {
            // preempta a tarefa atual
             $\tau_{exec}.\text{estado} = \text{READY}$ ;
            Insere_Elemento_Fila(ReadyQueue,  $\tau_{exec}$ );
            // despacha a nova tarefa
             $\tau_{exec} = \text{Retira\_Elemento\_Fila}(\text{ReadyQueue}, 1)$ ;
            Dispatch ( );
        }
    }
}

```

Quadro 4.3 – Algoritmo garantia de escalonabilidade no RM

## Algoritmo de Escalonabilidade com Taxa Monotonica – RM

```

Boolean conjunto_escalavel (tarefa t) {
    float Utub, U = 0;
    int i;

    Utub =  $n * (2^{1/n} - 1)$ ;
    // Calcula a utilização do conjunto de tarefas existente
    for (i=1; i <= num_tarefas; i++) U +=  $\tau_i.C / \tau_i.T$ ;
    // Inclui a utilização da nova tarefa
    U +=  $t.C / t.T$ ;

    return U <= Utub
}

```

#### 4.5.2. Menor Deadline Primeiro (EDF – *Earliest Deadline First*)

Esse é talvez o algoritmo de escalonamento de TR mais difundido. O EDF simplesmente escalone as tarefas com o *deadline* mais próximo, ou seja, aquele que deve terminar sua execução primeiro. Quanto mais próxima de seu *deadline*, maior a prioridade de uma tarefa, portanto o EDF é um algoritmo com prioridades dinâmicas, e não faz qualquer distinção ou menção quanto à periodicidade das tarefas, podendo ser usado para escalonar tarefas periódicas e aperiódicas. O quadro 4.4 apresenta o algoritmo EDF.

M. L. Dertouzos provou, em 1974, que o EDF é ótimo, no sentido de que se há uma seqüência de escalonamento possível para um conjunto de tarefas, então o EDF vai encontrá-la. (BUTTAZZO 1997). Apesar de muito conhecido e utilizado, o algoritmo EDF produz resultados ruins quando o sistema está sobrecarregado, devendo ser usado apenas quando o sistema está submetido a uma carga de trabalho (*workload*) abaixo do limite da utilização do processador. A equação (12) fornece a condição de escalonabilidade de um conjunto de tarefas sob o EDF, e o quadro 4.5 apresenta o algoritmo de garantia de escalonabilidade extraído dessa condição.

$$U = \sum_{i=1}^n \frac{C_i}{T_i} \leq 1 \quad (12)$$

Quadro 4.4 – Algoritmo de escalonamento EDF

## Algoritmo Menor Deadline Primeiro – EDF

```

void scheduler (void) {
    // A fila 'ReadyQueue' mantém os elementos (tarefas) por ordem crescente
    // de deadline absoluto (d), que é obtido somando o instante de criação (A) da tarefa à
    // sua deadline relativa (D). ( $d = A + D$ )
    //  $\tau_{exec}$  representa a tarefa executando no momento
    if NQ(ReadyQueue) > 0) {
        if ( $\tau_{exec}.A + \tau_{exec}.D > \text{Primeiro\_Elemento}(\text{ReadyQueue}).A + \text{Primeiro\_Elemento}(\text{ReadyQueue}).D$ ) {
            // preempta a tarefa atual
            tarefa_executando.estado = READY;
            Insere_Elemento_Fila(ReadyQueue,  $\tau_{exec}$ );
            // despacha a nova tarefa
             $\tau_{exec} = \text{Retira\_Elemento\_Fila}(\text{ReadyQueue}, 1)$ ;
            Dispatch ( );
        }
    }
}

```

Quadro 4.5 – Algoritmo garantia de escalonabilidade no EDF

## Algoritmo de Escalonabilidade com Menor Deadline Primeiro - EDF

```

Boolean EDF_Garantia (tarefa t) {
    float U = 0;
    int i;

    // Calcula a utilização do conjunto de tarefas existente
    for (i=1; i <= num_tarefas; i++) U +=  $\tau_i.C / \tau_i.T$ ;
    // Inclui a utilização da nova tarefa
    U +=  $t.C / t.T$ ;

    return U <= 1
}

```

### 4.5.3. Deadline Monotônica (DM – *Deadline Monotonic*)

No algoritmo de escalonamento DM, as tarefas representadas podem ter o período diferente de sua *deadline*, uma imposição feita a outros algoritmos de prioridade fixa. Nesse algoritmo, uma tarefa é representada por uma fase ( $\phi_i$ ), o tempo de computação no pior caso ( $C_i$ ), uma *deadline* relativa ( $D_i$ ) e um período ( $T_i$ ), sendo que

$$\begin{cases} C_i \leq D_i \leq T_i \\ r_{i,j} = \phi_i + (j-1)T_i \\ d_{i,k} = r_{i,k} + D_i \end{cases} \quad (13)$$

Trata-se de uma extensão do RM, sendo que a regra de escalonamento é a mesma (quadro 4.6).

Quadro 4.6 – Algoritmo de escalonamento DM

Algoritmo Deadline Monotônica – DM

```
void scheduler (void) {
    // A fila 'ReadyQueue' mantém os elementos (tarefas) por ordem crescente de Período (T)
    if NQ(ReadyQueue) > 0 {
        if (τexec.T > Primeiro_Elemento(ReadyQueue).T) {
            // preempta a tarefa atual
            τexec.estado = READY;
            Insere_Elemento_Fila(ReadyQueue, τexec);
            // despacha a nova tarefa
            τexec = Retira_Elemento_Fila(ReadyQueue, 1);
            Dispatch ();
        }
    }
}
```

#### 4.5.4. Escalonamento em Background (BS – *Background Scheduling*)

O algoritmo BS atende tarefas aperiódicas *soft real-time* juntamente com tarefas periódicas *hard real-time*. Os dois tipos de tarefas ficam em filas separadas. O escalonador atende primeiro as tarefas periódicas usando o algoritmo RM. Se não houver nenhuma tarefa na fila de periódicas, então o escalonador vai atender as tarefas aperiódicas da outra fila, através do algoritmo FCFS. O algoritmo BS costuma não levar a fatores de utilização muito altos, e sua principal vantagem é a simplicidade. (BUTTAZZO 1997). O BS é apresentado no quadro 4.7.

Quadro 4.7 – Algoritmo de escalonamento BS

```

Algoritmo Background Scheduling – BS
void scheduler (void) {
    // Há duas filas separadas: uma para tarefas periódicas 'PeriodicReadyQueue'
    // (mais prioritária) e outra para tarefas aperiódicas 'AperiodicReadyQueue'
    // (menos prioritárias). As tarefas periódicas são atendidas por RM
    // As tarefas aperiódicas são atendidas por outro critério qualquer. Neste caso, FCFS.
    if NQ(PeriodicReadyQueue) > 0) { // atende primeiro as tarefas periódicas
        if (τexec.T > Primeiro_Elemento(PeriodicReadyQueue).T)
            || (τexec.tipo == APERIODIC ) { // preempta a tarefa atual
                τexec.estado = READY;
                if (τexec.tipo == APERIODIC )
                    Insere_Elemento_Fila(AperiodicReadyQueue, τexec);
                Else
                    Insere_Elemento_Fila(PeriodicReadyQueue, τexec);
                // despacha a nova tarefa
                τexec = Retira_Elemento_Fila(PeriodicReadyQueue, 1);
                Dispatch ( );
            }
        } else { // Se não periódicas, atende as aperiódicas.
            if NQ(AperiodicReadyQueue) > 0) {
                if (τexec == IDLE) {
                    // escalona outra aperiódica apenas se a atual já acabou (FCFS)
                    // despacha a nova tarefa
                    τexec = Retira_Elemento_Fila(PeriodicReadyQueue, 1);
                    Dispatch ( );
                }
            }
        }
    }
}

```

#### 4.5.5. Servidor Pooling (PS – *Pooling Server*)

O algoritmo PS cria um servidor (PS) para servir tarefas aperiódicas. A cada intervalo de  $T_s$  u.t. o servidor se torna ativo e atende as tarefas aperiódicas pendentes, dentro do limite de sua capacidade  $C_s$ ,  $0 \leq C_s \leq C_{sMAX}$ . Se não há tarefas pendentes, ele se suspende até o próximo período ( $T_s$ ). A cada *tick* do temporizador em que o servidor está atendendo uma tarefa aperiódica, a capacidade  $C_s$  é decrementada.

Quando a capacidade for esgotada, o servidor fica inativo e a tarefa é preemptada. Se a tarefa acabar antes de consumir toda a capacidade do servidor, a capacidade é zerada e a execução é passada às tarefas periódicas, e cada início de período  $T_s$ , a capacidade é restabelecida para seu valor máximo  $C_{sMAX}$ . (BUTTAZZO 1997). Os parâmetros para esse algoritmo são o período  $T_s$  e a capacidade  $C_{sMAX}$  do servidor. O quadro 4.8 ilustra o PS.

Quadro 4.8 – Algoritmo de escalonamento PS

## Algoritmo Servidor Pooling – PS

```

void scheduler (void) {
    // Há duas filas separadas:
    // uma para tarefas periódicas 'PeriodicReadyQueue' (atendida por RM).
    // e outra para tarefas aperiódicas 'AperiodicReadyQueue' (não é atendida diretamente)
    // Uma tarefa periódica,  $\tau_{ps}$ , é criada para atender as tarefas aperiódicas
    if ( (TNOW mod Ts) == 0 ) {
        // início de um novo período do servidor
        if NQ(AperiodicReadyQueue) > 0) {
            // Há tarefas aperiódicas para serem atendidas
             $\tau_{ps}$  = Retira_Elemento_Fila (AperiodicReadyQueue, 1);
            // período de ativação do servidor
             $\tau_{ps}.T = Ts$ ;
            // reestabelece a capacidade;
            Cs = CsMAX;
            Inseire_Elemento_Fila (PeriodicReadyQueue,  $\tau_{ps}$ );
        }
        Else
            // não há tarefas aperiódicas a atender. Deixa o servidor inativo
            Cs = 0;
    }
}
if ( $\tau_{exec} == \tau_{ps}$ ) {
    // está executando o servidor. Consume uma unidade da capacidade
    Cs--;
}
if NQ(PeriodicReadyQueue) > 0) {
    if ( $\tau_{exec}.T > Primeiro\_Elemento(PeriodicReadyQueue).T$ )
    || ( ( $\tau_{exec} == \tau_{ps}$ ) && (Cs == 0) ) {
        // preempta a tarefa atual
         $\tau_{exec}.estado = READY$ ;
        Inseire_Elemento_Fila(PeriodicReadyQueue,  $\tau_{exec}$ );
        // despacha a nova tarefa
         $\tau_{exec} = Retira\_Elemento\_Fila(PeriodicReadyQueue, 1)$ ;
        Dispatch ();
    }
}
}

```

#### 4.5.6. Servidor Deferrable (DS – *Deferrable Server*)

O algoritmo DS cria um servidor (DS) para servir tarefas aperiódicas. A cada intervalo de  $T_s$  u.t. o servidor se torna ativo e atende as tarefas aperiódicas pendentes, dentro do limite de sua capacidade  $C_s$ ,  $0 \leq C_s \leq C_{sMAX}$ . Diferentemente do PS, se não há tarefas pendentes, ele mantém sua capacidade para atender possíveis tarefas aperiódicas que venham a surgir antes do próximo período ( $T_s$ ). A cada *tick* do temporizador em que o servidor está atendendo uma tarefa aperiódica, a capacidade  $C_s$  é decrementada.

Quando a capacidade for esgotada, o servidor fica inativo e a tarefa é preemptada. Se a tarefa acabar antes de consumir toda a capacidade do servidor, a capacidade restante é mantida para o próximo período, e cada início de período  $T_s$ , a capacidade é restabelecida para seu valor máximo  $C_{sMAX}$ . “O algoritmo DS produz respostas às tarefas aperiódicas muito melhores, uma vez que ele preserva a capacidade [do servidor] até ser necessária” (BUTTAZZO 1997). Os parâmetros para esse algoritmo são o período  $T_s$  e a capacidade  $C_{sMAX}$  do servidor. Quadro 4.9.

Quadro 4.9 – Algoritmo de escalonamento DS

```

Algoritmo Servidor Deferreble – DS
void scheduler (void) {
    // Há duas filas separadas:
    // uma para tarefas periódicas 'PeriodicReadyQueue' (atendida por RM).
    // e outra para tarefas aperiódicas 'AperiodicReadyQueue' (não é atendida diretamente)
    // Uma tarefa periódica,  $\tau_{ds}$ , é criada para atender as tarefas aperiódicas
    static int Cs = CSMAX;

    if ( ( $\tau_{ds}$  == IDLE) && (NQ(AperiodicReadyQueue) > 0) ) {
        // cria uma tarefa servidor
         $\tau_{ds}$  = Retira_Elemento_Fila(AperiodicReadyQueue, 1);
         $\tau_{ds}.T$  =  $T_s$ ;
        Insere_Elemento_Fila(PeriodicReadyQueue,  $\tau_{ds}$ );
    }
    if ( $\tau_{exec}$  ==  $\tau_{ds}$ ) {
        // está executando o servidor. Consome uma unidade da capacidade
        Cs--;
    }
    if ( (TNOW mod  $T_s$ ) == 0 ) {
        // início de um novo período do servidor
        // reestabelece a capacidade do servidor;
        Cs = CSMAX;
    }
    if NQ(PeriodicReadyQueue) > 0) {
        if ( $\tau_{exec}.T$  > Primeiro_Elemento(PeriodicReadyQueue).T)
        || ( ( $\tau_{exec}$  =  $\tau_{ds}$ ) && (Cs == 0) ) {
            // preempta a tarefa atual
             $\tau_{exec}.estado$  = READY;
            Insere_Elemento_Fila(PeriodicReadyQueue,  $\tau_{exec}$ );
            // despacha a nova tarefa
             $\tau_{exec}$  = Retira_Elemento_Fila(PeriodicReadyQueue, 1);
            Dispatch ( );
        }
    }
}

```

#### 4.5.7. Servidor Esporádico (SS – *Sporadic Server*)

O SS também cria um servidor para atender as tarefas aperiódicas, dentro de seu limite  $C_s$ . No SS, assim como no DS, a capacidade é mantida durante o período  $T_s$  para atender possíveis tarefas aperiódicas. Porém, ele difere em quando e na forma em que a

capacidade é restabelecida. No SS, o restabelecimento da capacidade segue as seguintes condições:

- ***Pexe***: É a prioridade da tarefa atualmente executando;
- ***Ps***: É a prioridade da tarefa do servidor SS;
- ***Ativo***: SS é dito ativo quando  $Pexe \geq Ps$  ;
- ***Ocioso***: SS é dito ocioso quando  $Pexe < Ps$  ;
- ***RT***: É o ‘tempo de reestabelecimento’, ou seja, o tempo no qual a capacidade do servidor SS será reestabelecida;
- ***RA***: É o ‘total de reestabelecimento’, ou seja, o total que será adicionado à capacidade  $C_s$  no instante  $RT$ ;
- ***ta***: Instante de tempo em que o SS se torna ativo e  $C_s > 0$ ;
- ***ti***: Instante de tempo em que o SS se torna ocioso ou sua capacidade é totalmente consumida ( $C_s = 0$ );
- *se  $TNOW = ta$  então  $RT = ta + Ts$ ;*
- *se  $TNOW = ti$  então  $RA = \text{capacidade consumida no intervalo } [ta, ti]$ .*

“O algoritmo SS permite melhorar o tempo de resposta de tarefas aperiódicas sem piorar o limite de utilização das tarefas periódicas” (Buttazzo 1997). Os parâmetros para esse algoritmo são a prioridade  $Ps$ , o período  $Ts$  e a capacidade  $C_{sMAX}$  do servidor. O quadro 4.10 apresenta o algoritmo SS.

Um conjunto de tarefas periódicas *hard* com fator de utilização  $U_p$  é escalonável pelo SS com utilização  $U_s$ , se:

$$U_p \leq n \left[ \left( \frac{2}{U_s + 1} \right)^{1/n} - 1 \right] \quad (14)$$

E, dado um fator de utilização  $U_p$ , o maior SS que garante a escalonabilidade das tarefas periódicas é dado por:

$$U_{ss}^* = 2 \left( \frac{U_p}{n} + 1 \right)^{-n} - 1 \quad (15)$$

Quadro 4.10 – Algoritmo de escalonamento SS

## Algoritmo Servidor Esporádico – SS

```

void scheduler (void) {
    // Há duas filas separadas: uma para tarefas periódicas 'PeriodicReadyQueue' (atendida
    // por RM) e outra para tarefas aperiódicas 'AperiodicReadyQueue' (não é atendida
    // diretamente). Uma tarefa periódica,  $\tau_{ss}$ , é criada para atender as tarefas aperiódicas.
    // A capacidade é reestabelecida de RA unidades no instante RT.
    bool SSativo;
    static int consumido = RT = RA = 0;
    static int Cs = CSMAX;

    SSativo = ( $\tau_{exec}$ .prioridade  $\geq$  Ps);
    if ( ( $\tau_{ss}$  == IDLE) && (NQ(AperiodicReadyQueue) > 0) ) { // cria uma tarefa servidor
         $\tau_{ss}$  = Retira_Elemento_Fila(AperiodicReadyQueue, 1);
         $\tau_{ss}$ .T = Ts;
         $\tau_{ss}$ .prioridade = Ps;
        Insere_Elemento_Fila(PeriodicReadyQueue,  $\tau_{ss}$ );
    } if ( $\tau_{exec}$  ==  $\tau_{ss}$ ) { // está executando o servidor. Consome uma unidade da capacidade
        Cs--;
        consumido++;
    } if (TNOW == RT) { // tempo de reestabelecimento. Reestabelece a capacidade.
        Cs = Cs + RA;
    } if NQ(PeriodicReadyQueue) > 0) {
        if ( $\tau_{exec}$ .prioridade > Primeiro_Elemento(PeriodicReadyQueue).prioridade) || ( ( $\tau_{exec}$ 
        ==  $\tau_{ss}$ ) && (Cs == 0) ) { // verifica se houve mudanças no estado do servidor SS
            if (!SSativo && ( $\tau_{exec}$ .prioridade  $\geq$  Ps) && (Cs > 0) ) {
                // tornou-se ativo E Cs > 0, ou seja, TNOW = ta
                consumido = 0;
                RT = TNOW + Ts;
            } else if (SSativo && !( $\tau_{exec}$ .prioridade  $\geq$  Ps) || (Cs == 0) ) {
                // tornou-se inativo OU Cs = 0, ou seja, TNOW = ti
                RA = consumido; }
            // preempta a tarefa atual
             $\tau_{exec}$ .estado = READY;
            Insere_Elemento_Fila(PeriodicReadyQueue,  $\tau_{exec}$ );
            // despacha a nova tarefa
             $\tau_{exec}$  = Retira_Elemento_Fila(PeriodicReadyQueue, 1);
            Dispatch ();
        } } }

```

#### 4.5.8. Menor Deadline Primeiro Robusto (RED – *Robust Earliest Deadline*)

O algoritmo EDF produz bons resultados quando o sistema está submetido a uma carga normal, mas degrada o desempenho quando o sistema for submetido a uma sobrecarga de trabalho. O algoritmo RED foi desenvolvido por Buttazzo & Stankovic para tratar de tarefas aperiódicas em ambientes sobrecarregados. “Ele opera em condições normais e de sobrecarga com excelente performance, e é apto não só a predizer perdas de *deadline*, mas também o ‘tamanho’ da sobrecarga, sua duração e o impacto geral sobre o sistema” (BUTTAZZO 1997).

No RED, cada tarefa é caracterizada por  $\tau_i (C_i, D_i, T_i, M_i, V_i)$ , onde  $C_i$  é o máximo tempo de computação necessário,  $D_i$  é a *deadline* relativo,  $T_i$  é o período de ativação,  $M_i$  é uma tolerância de *deadline*, e  $V_i$  é um valor de importância. A tolerância de *deadline* é o total de tempo o qual é permitido a uma tarefa atrasar. As tarefas são escalonadas usando seu *deadline* absoluto (chegada +  $D_i$ ), mas são aceitas conforme sua tolerância de *deadline*, que forma uma espécie de segundo *deadline*.

BUTTAZZO & STANKOVIC (1995), apresentam ainda as seguintes informações sobre a garantia de escalonabilidade e as estratégias utilizadas no RED:

Dado um conjunto  $J = \{J_1, J_2, \dots, J_n\}$  de tarefas aperiódicas ordenada por *deadline*, o tempo residual (laxidade)  $X_i$  de cada tarefa  $J_i$  num tempo  $t$  pode ser computado com a seguinte fórmula recursiva:

$$X_1 = d_1 - t - c_1 \quad (16)$$

$$X_i = X_{i-1} + (d_i - d_{i-1}) - c_i \quad (17)$$

Uma tarefa  $J_i$  é garantida que irá atender sua *deadline* se e apenas se  $X_i \geq 0$ . O conjunto de tarefas  $J$  é dito escalonável se  $X_i \geq 0$  para toda tarefa  $J_i$  pertencente a  $J$ . O RED divide as tarefas em *Hard* e *Critical*, sendo que todas as tarefas *hard* são garantidas em condições de carga normal e as tarefas *critical* são garantidas em situações de carga normal e também com sobrecarga.

O algoritmo calcula a carga do sistema na chegada e no término de uma tarefa. A tolerância de *deadline* funciona como espécie de compensação à avaliação pessimista

do tempo de computação, caso a tarefa acabe antes do seu tempo máximo. Para a determinação da carga, BUTTAZZO & STANKOVIC (1995) propõem um algoritmo eficiente, que permite criar um diagnóstico da carga, prever sua magnitude e o intervalo de tempo no qual pode ocorrer.  $p_i(t_a)$  indica a carga do processador no intervalo  $[t_a, d_i)$ , onde  $t_a$  é o tempo de chegada da última tarefa aperiódica no sistema. A computação da carga pode ser escrita como:

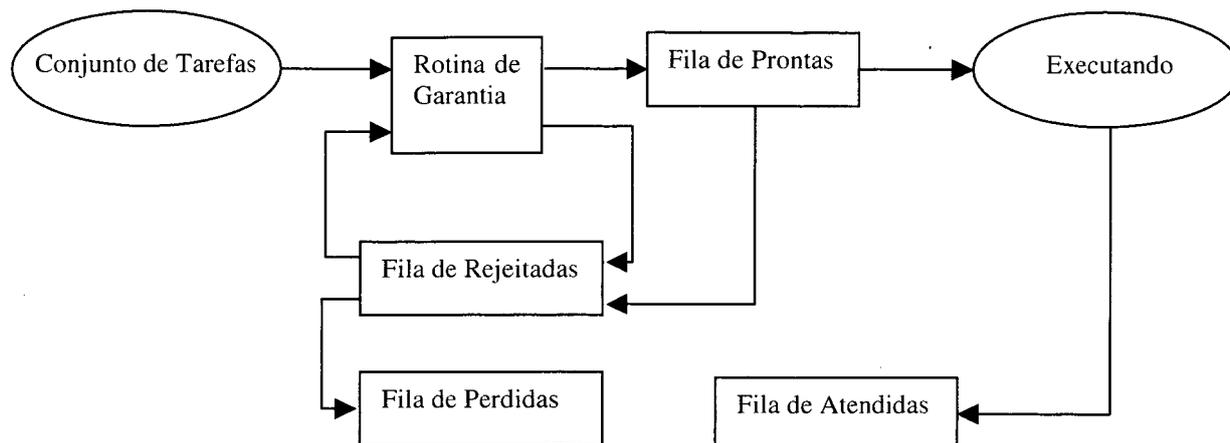
$$p_i(t_a) = \frac{\sum_{d_k < d_i} c_k}{d_i - t_a} \quad (18)$$

$$p_i = 1 - \frac{X_i}{d_i - t_a}; p_{i+1} = \frac{p_i(d_i - t_a) + c_{i+1}}{d_{i+1} - t_a} \quad (19)$$

Em condições de sobrecarga, tarefas *hard* podem ser rejeitadas para permitir atender as tarefas *critical*. As tarefas rejeitadas não são descartadas de imediato, mas ficam armazenadas numa fila separada (fila de rejeitadas), pois talvez possam ser reescaloadas depois, caso a carga diminua ou algumas tarefas terminem antes do esperado. As tarefas na fila de rejeitadas são ordenadas pelo valor  $v_i$ , de ordem decrescente. Assim, se alguma tarefa puder ser reescaloadada, o será a que tiver maior importância. Sempre que uma tarefa da fila de rejeitadas perde a *deadline*, ela é retirada definitivamente do sistema.

SPURI et al (1995) lembram que as diferenças do RED com o EDF se manifestam apenas em condições de sobrecarga de trabalho. A figura 4.2 ilustra o diagrama de blocos do RED. As filas de perdidas e atendidas servem apenas para ter uma trilha do comportamento do algoritmo. O quadro 4.11 ilustra o algoritmo RED, enquanto o quadro 4.12 apresenta o algoritmo para garantia de escalonabilidade das tarefas.

Figura 4.2 – Diagrama de blocos do algoritmo RED



Fonte: (ABENI 1999)

Quadro 4.11 – Algoritmo de escalonamento RED

## Algoritmo Menor Deadline Primeiro Robusto – RED

```

void scheduler (void) {
  if NQ(ReadyQueue) > 0) {
    if ( $\tau_{exec}.A + \tau_{exec}.D > \text{Primeiro\_Elemento}(\text{ReadyQueue}).A + \text{Primeiro\_Elemento}(\text{ReadyQueue}).D$ ) {
      // preempta a tarefa atual
       $\tau_{exec}.\text{estado} = \text{READY};$ 
      Insere_Elemento_Fila(ReadyQueue,  $\tau_{exec}$ );
      // despacha a nova tarefa
       $\tau_{exec} = \text{Retira\_Elemento\_Fila}(\text{ReadyQueue}, 1);$ 
      Dispatch ();
    }
  }
}

```

Quadro 4.12 – Algoritmo garantia de escalonabilidade no RED

```

boolean RED_Garantia (tarefa t) {
    int E, X0, X1;
    int i, k;
    int d0, d1 ;

    // Inseire a nova tarefa no conjunto de tarefas e retorna a posição da nova tarefa
    k = Inseire_Tarefa_Conjunto (t);
    E = 0;
    X0 = 0;
    d0 = TNOW;
    // Calcula a laxidade das tarefas posteriores à k
    for (i=k; i <= num_tarefas; i++) {
        d1 =  $\tau_i.A + \tau_i.D$ ; // deadline absoluta = chegada + deadline relativa
        X1 = X0 + (d1 - d0) -  $\tau_i.C$ ;
        if ( $X1 + \tau_i.M < -E$ )
            E =  $-(X1 + \tau_i.M)$ ;
        X0 = X1; // atualiza variáveis para próxima iteração
        d0 = d1;
    }
    if (E != 0) {
        // se E ≠ 0 é porque não pode garantir. Vai tentar descartar algumas tarefas
        tarefa rejeitadas[ ]; // define um conjunto (array) de tarefas
        rejeitadas = RED_Estrategia_Rejeicao( $\tau$ );
        if (Num_Elementos(rejeitadas) > 0) {
            // Algumas tarefas podem ser rejeitadas para garantir o atendimento das outras
            for (i == 0; i < Num_Elementos(rejeitadas); i++)
                Exclui_Tarefa_Conjunto (rejeitadas(i) );
            E = 0; // informa que está garantido
        }
    }
    // Resultado: se E=0 então 'garantido', senão 'não garantido'
    return E == 0;
}

```

#### 4.5.9. Servidor Esporádico Dinâmico (DSS – *Dinamic Sporadic Server*)

O algoritmo DSS é uma extensão do SS, e trabalha com prioridades dinâmicas através do EDF. Esse algoritmo cria um servidor DSS com período  $T_s$  para atender as tarefas aperiódicas, e difere do SS tradicional na atribuição da *deadline* do servidor e pelo modo como repõe a capacidade do DSS. O quadro 4.13 mostra o algoritmo do DSS que segue as regras:

- Quando o servidor é criado, sua capacidade  $C_s$  é inicializada com o valor máximo;
- O próximo instante de restabelecimento da capacidade  $RT$  e a *deadline* atual do servidor  $ds$ , são setados assim que  $C_s > 0$  e haja uma tarefa aperiódica pendente. Se  $ta$  é tal instante, então  $RT = ds = ta + T_s$  (20)
- O total da capacidade a ser restabelecida  $RA$  no instante  $RT$  é computado quando a última tarefa aperiódica é completada ou quando  $C_s = 0$ . Se  $ti$  é tal instante, então ao valor de  $RA$  é atribuído o total de capacidade consumida no intervalo  $[ta, ti]$ .

Quadro 4.13 – Algoritmo de escalonamento DSS

## Algoritmo Servidor Esporádico Dinâmico – DSS

```

Void scheduler (void) {
    // Há duas filas separadas: uma para tarefas periódicas 'PeriodicReadyQueue' (atendida
    // por RM) e outra para tarefas aperiódicas 'AperiodicReadyQueue' (não é atendida
    // diretamente) Uma tarefa periódica,  $\tau_{dss}$ , é criada para atender as tarefas aperiódicas
    // A capacidade é reestabelecida de RA unidades no instante RT.
    bool DSSativo;
    static int consumido = RT = RA = 0;
    static bool atendendo = false;
    static int Cs = CSMAX;

    if ( (Cs > 0) && ( $\tau_{dss}$  == IDLE) && (NQ(AperiodicReadyQueue) > 0) {
        // cria uma tarefa servidor
        ta = tnow;
         $\tau_{dss}$  = Retira_Elemento_Fila(AperiodicReadyQueue, 1);
         $\tau_{dss}.T = Ts$ ;  $\tau_{dss}.D = Ts$ ; // deadline relativa. A deadline absoluta seria  $d = ta + Ts$ 
         $\tau_{dss}.prioridade = ta + Ts$ ;
        Insere_Elemento_Fila(PeriodicReadyQueue,  $\tau_{dss}$ );
    } else if (( $\tau_{dss}$  == IDLE) && atendendo) {
        ti = tnow; // estava atendendo mas agora está ocioso. Recém atendeu a última aperiódica: ti
        RA = consumido;
        consumido = 0;
    }
    atendendo =  $\tau_{exec}$  ==  $\tau_{dss}$ ;
    if (atendendo) {
        Cs--; // está executando o servidor. Consome uma unidade da capacidade
        consumido++;
        if (Cs == 0) { // toda a capacidade foi consumida: ti
            ti = tnow;
            RA = consumido;
            consumido = 0;
        }
    }
    if (TNOW == RT) // tempo de reestabelecimento. Reestabelece a capacidade do servidor.
        Cs = Cs + RA;
    if (NQ(PeriodicReadyQueue) > 0) {
        if ( $\tau_{exec}.D > Primeiro_Elemento(PeriodicReadyQueue).D$ )
            // deadline = prioridade, portanto preempta a tarefa atual
             $\tau_{exec}.estado = READY$ ;
        Insere_Elemento_Fila(PeriodicReadyQueue,  $\tau_{exec}$ );
         $\tau_{exec}$  = Retira_Elemento_Fila(PeriodicReadyQueue, 1);
        Dispatch (); // despacha a nova tarefa
    }
}

```

#### 4.5.10. Servidor de Largura de Banda Constante (CBS – *Constant Bandwidth Server*)

O algoritmo CBS foi criado por BUTTAZZO e ABENI (1998) para atender aplicações multimídia em STR. “Atividades de mídia contínua, como áudio e vídeo *streams*, precisam de suporte de TR devido a sua sensibilidade ao atraso e *jitter*”.

O CBS considera três tipos de tarefas: *hard*, *soft* e tarefas não tempo-real. Para uma tarefa *soft*,  $C_i$  representa o tempo médio de execução de cada job, enquanto  $T_i$  representa o período de ativação *desejado* entre jobs sucessivos. Note a diferença em relação às tarefas *hard*, que não têm nenhum tratamento especial nesse algoritmo.

“Atividades de mídia contínua são tipicamente caracterizadas por tempos de execução altamente variáveis, fazendo com que o tempo máximo de execução seja muito maior que o tempo médio de execução [e por isso não é uma boa medida de representação]. (...) uma estratégia de reserva de largura de banda, que associa cada tarefa *soft* uma largura máxima de banda, calculada usando o tempo médio de execução e o período desejado de ativação, de modo a aumentar a utilização da CPU. Se uma tarefa precisa mais que sua largura de banda reservada, ela pode ficar mais lenta, mas não vai atrapalhar a escalonabilidade das tarefas *hard*”. (ABENI 1998)

Nesse algoritmo, cada tarefa *soft* é atendida por um Servidor de Largura de Banda Constante (CBS), e foi inspirado no algoritmo DSS, que também isola a influência das tarefas *soft*, garantindo a escalonabilidade das tarefas *hard*. O CBS é definido como segue:

- Um CBS é caracterizado por uma capacidade  $c_s$  e um par ordenado  $(Q_s, T_s)$ , onde  $Q_s$  é a capacidade máxima e  $T_s$  é o período do servidor. A taxa  $U_s = Q_s / T_s$  é denotada largura de banda do servidor. A cada instante, uma *deadline* fixada  $d_{s,k}$  é associada com o servidor. No início,  $d_{s,0} = 0$ ;
- A cada job servido  $J_{i,j}$  é associada uma *deadline* dinâmica  $d_{i,j}$  igual à *deadline* atual do servidor  $d_{s,k}$ ;
- Sempre que um job servido executa, a capacidade  $c_s$  é diminuída da mesma quantidade;

- Quando  $c_s = 0$ , a capacidade do servidor é recarregada para seu valor máximo  $Q_s$  e uma nova *deadline* do servidor é gerada como  $d_{s,k+1} = d_{s,k} + T_s$ . Note que não há intervalo de tempo no qual a capacidade é igual a zero.
- Um CBS é dito *ativo* no tempo  $t$  se há jobs pendentes. Isto é, existe um job servido  $J_{i,j}$  tal que  $a_{i,j} \leq t < f_{i,j}$ . Um CBS é dito *inativo* num tempo  $t$  se ele não está *ativo*.
- Quando um job  $J_{i,j}$  chega e o servidor está ativo, a requisição é enfileirada numa fila de jobs pendentes de acordo com algum algoritmo não preemptivo (por exemplo, FIFO);
- Quando um job  $J_{i,j}$  chega e o servidor está inativo, se  $c_s \geq (d_{s,k} - a_{i,j}) \cdot U_s$ , o servidor gera uma nova *deadline*  $d_{s,k+1} = a_{i,j} + T_s$  e  $c_s$  é recarregado para seu valor máximo  $Q_s$ . Senão o job é servido com a última *deadline* do servidor  $d_{s,k}$  usando a capacidade atual;
- Quando um job termina, o próximo job pendente, se existir, é servido usando a capacidade e a *deadline* atuais. Se não há jobs pendentes, o servidor se torna inativo;
- A qualquer instante, a um job é associada a última *deadline* gerada pelo servidor.

Conforme ABENI (1998), dado um conjunto de  $n$  tarefas periódicas *hard* com utilização do processador  $U_p$  e um CBS com utilização do processador  $U_s$ , todo o conjunto de tarefas é escalonável por EDF se e apenas se

$$U_p + U_s \leq 1 \quad (21)$$

Os parâmetros do CBS são o período  $T_s$  e a capacidade  $C_s$  do servidor. O quadro 4.14 mostra o algoritmo CBS.

Quadro 4.14 – Algoritmo de escalonamento CBS

```

Algoritmo Servidor de Largura de Banda Constante – CBS
Void scheduler (void) {
    // Uma tarefa periódica,  $\tau_{cbs}$ , é criada para atender as tarefas aperiódicas
    // A capacidade é restabelecida de RA unidades no instante RT.
    static int Cs = Qs;

    if (( $\tau_{cbs} == IDLE$ ) && (NQ(AperiodicReadyQueue) > 0) {
        // cria uma tarefa servidor
        if (Cs >= ( $\tau_{cbs}.A + \tau_{cbs}.D - Primeiro\_Elemento(ReadyQueue).A$ ) * (Cs / Ts)) {
            // Gera nova deadline e recarrega a capacidade
             $\tau_{cbs} = Retira\_Elemento\_Fila(AperiodicReadyQueue, 1)$ ;
             $\tau_{cbs}.D = Ts$ ; // deadline relativa. A deadline absoluta seria  $d = A + Ts$ 
            Cs = Qs;
        } else
            // Atende com a deadline e a capacidade atuais
             $\tau_{cbs} = Retira\_Elemento\_Fila(AperiodicReadyQueue, 1)$ ;
         $\tau_{cbs}.T = Ts$ ;
        InseRe_Elemento_Fila(PeriodicReadyQueue,  $\tau_{cbs}$ );
    }
    if ( $\tau_{exec} == \tau_{cbs}$ ) {
        // está executando o servidor. Consume uma unidade da capacidade
        Cs--;
        if (Cs == 0) {
            // toda a capacidade foi consumida: Restabelece capacidade e gera nova deadline
            Cs = Qs;
             $\tau_{cbs}.D = \tau_{cbs}.D + Ts$ 
        }
    }
    if NQ(PeriodicReadyQueue) > 0) {
        if ( $\tau_{exec}.A + \tau_{exec}.D > Primeiro\_Elemento(PeriodicReadyQueue).A +$ 
        Primeiro_Elemento(PeriodicReadyQueue).D)
            // deadline = prioridade
            // preempta a tarefa atual
             $\tau_{exec}.estado = READY$ ;
            InseRe_Elemento_Fila(PeriodicReadyQueue,  $\tau_{exec}$ );
            // despacha a nova tarefa
             $\tau_{exec} = Retira\_Elemento\_Fila(PeriodicReadyQueue, 1)$ ;
            Dispatch ();
        }
    }
}

```

#### 4.5.11. Reserva de Largura de Banda Adaptativa (ABR – *Adaptive Bandwidth Reservation*)

Esse algoritmo classifica um tipo de tarefa como ‘tarefas multimídia’, que são *streams* de *jobs* periodicamente ativados e que devem executar com taxa constante, sendo que o período entre *jobs* da tarefa (ou a taxa de ativação) é especificado, mas o tempo de execução não precisa ser conhecido.

O algoritmo ABR é uma extensão do CBS proposto por ABENI e BUTTAZZO (1999) para controlar a fração da largura de banda do processador alocado a tarefas cujo tempo de execução não é conhecido *a priori*. “Uma medida da qualidade do serviço é usada como *feedback* para controlar os recursos atribuídos (através da estratégia de reserva de largura de banda). A qualidade do serviço das tarefas pode ser ajustada mudando os períodos ou o tempo reservado da CPU” (ABENI 1999).

O algoritmo usado para escalonamento dinâmico (em baixo-nível) é o EDF, sendo que o período do servidor é mantido fixo, com valor igual ao período  $T_i$  especificado e a capacidade do servidor  $Q_s$  é ajustada dinamicamente de modo a atender os requisitos de TR. Defini-se o erro de controle como a diferença entre a *deadline* do servidor  $d_s$  e a *deadline* da tarefa.

$$e = d_s - (a_{i,j} + T_i) \quad (22)$$

O objetivo do controle é manter  $e = 0$  através de uma função de controle do tipo  $Q_s = f(e)$  que ajuste capacidade máxima do servidor  $Q_s$ . A determinação dessa função de controle é essencial para o desempenho do escalonador, sendo que muitas funções, inclusive PID e algoritmos inteligentes poderiam ser usados. ABENI e BUTTAZZO (1999) concluem em seu trabalho que “estamos planejando investigar esquemas de controle *feedback* e algoritmos de ajustamento de largura de banda mais sofisticados. Em particular, estamos interessados numa função de controle que possa ser usada para dinamicamente ajustar a largura de banda de modo a permitir uma garantia probabilística de *deadline*”. O algoritmo de escalonamento ABR é idêntico ao CBS, apenas podendo variar o valor de  $Q_s$  de acordo com a função de controle apresentada no quadro 4.15, que deve ser avaliada antes de cada chamada ao escalonador.

Quadro 4.15 – Algoritmo de adaptação do ABR

```

Algoritmo Reserva de Largura de Banda Adaptativo – ABR
Void Funcao_Feedback (void) {
    float e, Bs, sumB = 0;
    int j;

    e =  $\tau_{cbs}.A + \tau_{cbs}.D - (\tau_i.A + \tau_i.D)$ ;
    if (e!= 0) {
        // A saída é diferente da referência. Calcula novo Qs.
        dQ = (e / Ts) * Q
        Qs = Qs + dQ;
        // Escala a saída caso tenha aumentado muito a capacidade Qs
        Bs = Qs / Ts;
        if (Bs > BsMAX) {
            for (j=1; j <= num_tarefas; j++)
                SumB += ( $\tau_j.C / \tau_j.T$ ) *  $\tau_j.V$ ;
            Bs = Bs *  $\tau_i.V * (BsMAX / SumB)$ 
            Qs = Bs * Ts;
        }
    }
}

```

## 5. Simulação Digital de Sistemas

### 5.1. Introdução

A simulação digital de sistemas, ou apenas simulação, consiste na utilização de determinadas técnicas matemáticas e estatísticas, empregadas em computadores digitais, e que permitem imitar o comportamento de, praticamente, qualquer tipo de operação ou processo do mundo real.

Alguns dos principais conceitos envolvidos na simulação de sistemas são apresentados a seguir:

- Variável de Estado: “As variáveis cujos valores definem o estado do estado são chamadas de *variáveis de estado*” (JAIN 1991);
- Evento: Uma mudança no estado do sistema é denominada *evento*. Por exemplo, uma tarefa pode passar do estado ‘Pronto’ para ‘Executando’ devido a ocorrência do evento ‘escalonamento’;
- Entidade: Uma entidade representa um objeto que necessita ser claramente definido. As entidades normalmente “percorrem” o sistema e são processadas durante o caminho. São exemplos de entidades num sistema: clientes numa loja, peças sendo fabricadas, mensagens sendo enviadas numa rede, processos executando na CPU, etc.
- Recurso: Um recurso é um tipo de entidade estática que provê serviços às demais entidades. Um recurso num sistema pode ser uma máquina, um balconista, uma CPU, etc.

### 5.2. Vantagens e Desvantagens

Pode ser simplesmente impossível ou inviável de algum modo observar determinados sistemas do mundo real, desde sistemas muito rápidos ou microscópicos até sistemas de ordem astronômica.

“O sistema real pode ser tão complexo que se torne impossível descrevê-lo em termos de um conjunto de equações matemáticas e para o qual seja possível obtermos soluções analíticas, que enfim seriam usadas para fins preditivos” (NAYLOR 1966).

A simulação de sistemas, em especial a simulação digital em computador, traz inúmeras vantagens na representação, compreensão e predição do comportamento de um sistema real:

- A simulação permite estudar e experimentar complexas interações internas de um dado sistema;
- “Através da simulação podem-se estudar os efeitos de certas variações do meio ambiente [sistema], ..., fazendo alterações no seu modelo e observando os efeitos dessas alterações no comportamento desse sistema”, (NAYLOR 1966);
- A simulação de sistemas complexos permite descobrir quais são as variáveis mais importantes no sistema, e como elas interagem;
- Pode-se identificar pontos de estrangulamento (gargalos) no sistema e verificar a formação de filas;
- A simulação pode ser usada em experiências com novas situações sobre as quais se tem pouca ou nenhuma informação;
- Pode-se dividir um sistema complexo em sub-sistemas mais simples, podendo-se chegar a um nível de detalhamento bastante alto;
- Uma vez criado o modelo do sistema, ele pode ser utilizado inúmeras vezes, podendo ter suas variáveis alteradas ou não;
- “A simulação permite o estudo de sistemas dinâmicos quer em tempo real, tempo reduzido ou dilatado” (NAYLOR 1966);
- A simulação permite testar hipóteses sobre o comportamento do sistema.

Entre as desvantagens do uso de simulação, pode-se citar:

- É necessário identificar e coletar os dados de entrada para alimentar o modelo do sistema. Os dados de entrada devem ser bem escolhidos, ou os resultados podem não ter valor;
- É necessário descrever e especificar o modelo que representará o sistema. O modelo deve ser implementado em algum aplicativo de simulação digital;
- É necessário verificar o modelo (inexistência de erros) e validá-lo (representa o sistema desejado);
- É necessário interpretar os resultados, normalmente fornecidos na forma de números ou tabelas, de modo a poder tomar decisões e verificar hipóteses;

### **5.3. Classificação**

Os modelos das simulações podem ser classificados da seguinte forma:

- Quanto à Aplicação:
  - Modelos de imagens: As características que representam o sistema são representadas em forma de imagens, como um simulador de vôo, por exemplo;
  - Modelos simbólicos: As características do sistema são representadas de forma matemática ou simbólica;
- Quanto à Previsibilidade:
  - Modelos determinísticos: Onde os resultados podem ser pré-determinados e não envolvem variáveis aleatórias, ou seja, várias replicações do mesmo experimento, sob as mesmas condições, produzem exatamente os mesmos resultados;
  - Modelos estocásticos: Modelos onde replicações de um mesmo experimento, sob as mesmas condições, podem produzir resultados diferentes, pois há elementos de aleatoriedade e indeterminismo no modelo;
- Quanto ao Tempo:

- Modelos estáticos: São aqueles onde o comportamento do sistema é fixo ao longo do tempo, ou seja, o tempo é irrelevante ao modelo. Por exemplo, o modelo de uma equação matemática;
- Modelos dinâmicos: São aqueles em que o comportamento do sistema evolui com o tempo. Constituem a maioria dos processos conhecidos;
- Quanto à Troca de Estados:
  - Modelos discretos: Apresentam trocas de estado apenas em pontos discretos no tempo;
  - Modelos contínuos: São aqueles em que a troca de estados ocorre continuamente;
- Quanto às Entradas:
  - Modelos abertos: Possuem entradas externas e independentes do modelo;
  - Modelos fechados: Não possuem entradas externas;
- Quanto à Estabilidade:
  - Modelos estáveis: Se o comportamento dinâmico do sistema tende a um estado determinado, o modelo é dito estável;
  - Modelos instáveis: São aqueles em que o comportamento está mudando continuamente;

#### **5.4. Análise Estatística**

“É impossível fazer uma avaliação da qualidade do ajuste de um modelo sem recorrer a alguns conceitos básicos de estatística” (BARROS et al 1995).

Em qualquer estudo científico, antes mesmo da coleta dos dados é imprescindível fazer uma análise estatística que defina que tipo de dados devem ser coletados, em qual quantidade e de que modo, para permitir que deles se extraiam as informações que poderão levar ao cumprimento dos objetivos desse estudo. REIS (1998) alerta que “sem a utilização de métodos estatísticos os resultados de uma pesquisa não têm validade científica”.

O descumprimento de alguma suposição ou imposição estatística, como talvez a independência das amostras e distribuição normal dos erros, pode invalidar os resultados da análise estatística e levar o estudo a lugar nenhum. “Uma ignorância total dos diversos métodos de planejamento e análise de experimentos tende a produzir, para tristeza dos envolvidos, conclusões de natureza duvidosa. Pior ainda: pode levar à realização de experimentos incapazes de fornecer qualquer conclusão, duvidosa ou não...” (BARROS et al 1995).

Os conceitos abaixo, definidos conforme LEVINE et al (2000), MENDENHALL (1985), BARBETTA (1994), BUSSAB (1995) e REIS (1998), estão presentes na maioria das análises estatísticas realizadas e são de fundamental importância para a interpretação dos resultados estatísticos produzidos.

Conceitos básicos e de síntese estatística:

- Estatística: O objetivo da estatística é permitir inferências (previsões, decisões) acerca de uma população, tomando como base as informações contidas na amostra;
- Experimento aleatório: Um experimento que, sempre que for reproduzido nas mesmas condições, pode fornecer respostas diferentes das obtidas anteriormente;
- Probabilidade: É a chance de certo evento ocorrer. Varia de 0 se o evento não pode ocorrer até 1, se o evento certamente ocorrerá;
- Espaço amostral: É o conjunto de todas as possíveis respostas ocasionadas por um evento aleatório. “Para cada experimento aleatório haverá um espaço amostral único associado a ele” (REIS 1998);
- Evento: “Evento é qualquer subconjunto do espaço amostral. Um evento pode conter um ou mais resultados. Se pelo menos um dos resultados ocorrer, o evento ocorre” (REIS 1998);
- Variável aleatória: “Variável aleatória é uma função numérica definida sobre um espaço amostral” (MENDENHALL 1985). “Ela associa um número real ao resultado de espaço amostral associado a um experimento aleatório” (REIS 1998). Pode ser quantitativa ou qualitativa, nominal, ordinal ou intervalar, independente ou dependente;

- População: População é o conjunto que representa todas as medidas de interesse para o coletor de amostras;
- Amostra: Amostra é um subconjunto de medidas extraídas da população de interesse. A amostra deve ser representativa da população, suficiente e aleatória;
- Medidas de síntese: As medidas de síntese permitem representar todo o conjunto de dados em poucos indicadores;
- Medidas de posição: As medidas de posição procuram caracterizar a tendência central do conjunto, um valor numérico que represente o conjunto. As mais conhecidas são a média ( $\mu$ ), a mediana e a moda;
- Medidas de dispersão: “O objetivo das medidas de dispersão é medir o quão próximos uns dos outros estão os valores de um grupo” (REIS 1998). As mais comuns são o intervalo, a variância ( $s^2$ ,  $\sigma^2$ ), o desvio-padrão ( $s$ ,  $\sigma$ ), coeficiente de variação e as separatrizes;
- Agrupamento (Clusterização): “O objetivo básico da *clusterização* é particionar os componentes fazendo com que os membros dentro do grupo sejam tão similares quanto possível e diferentes entre os grupos” (FREITAS 2000);
- Correlação: A correlação é um número entre [-1,1] que resume o relacionamento entre as variáveis, indicando a força e a direção do relacionamento. Pode-se dizer que a correlação é a covariância normalizada;
- Regressão: A regressão é uma equação matemática que descreve a “natureza” (linear, parabólica, logarítmica, etc) do relacionamento entre as duas variáveis, permitindo inclusive que sejam feitas previsões dos valores de uma delas em função dos valores da outra. A função de regressão ‘explica’ grande parte da variação da variável independente com a dependente. Uma parcela da variação permanece sem ser explicada, e é atribuída ao acaso ou a outras variáveis;

Conceitos relacionados às distribuições de probabilidade:

- Função Densidade de Probabilidade: É uma função matemática que expressa a probabilidade da ocorrência de cada elemento pertencente ao espaço amostral. “A distribuição de probabilidade de uma variável aleatória contínua pode ser

representada por uma função não negativa, com área formada entre o eixo das abcissas e a curva dessa função igual a 1. Os eventos podem ser representados por intervalos nos eixos das abcissas, enquanto as correspondentes probabilidades por áreas sob a curva” (BARBETTA 1994);

- Distribuição Uniforme: É uma função densidade de probabilidade onde todos os elementos do espaço amostral possuem a mesma probabilidade de ocorrência;
- Distribuição Binomial: É uma função densidade de probabilidade discreta que indica a probabilidade de obtermos sucesso  $x$  vezes em um total de  $n$  repetições de um experimento aleatório, sabendo que a probabilidade de obter sucesso em um único experimento é  $p$ .
- Distribuição Normal: É uma função densidade de probabilidade contínua que representa muitos fenômenos reais. Está simetricamente distribuída em torno de sua média  $\mu$ , sendo o valor da média é o de maior probabilidade, que decai com a distância da média. Em torno de 63% da probabilidade está no intervalo  $\mu - \sigma$  e  $\mu + \sigma$ .
- Teorema Central do Limite: Se a variável de interesse não segue uma distribuição normal na população (ou não se sabe qual é sua distribuição), a distribuição amostral das médias de amostras aleatórias retiradas desta população será normal se o tamanho dessas amostras for suficientemente grande;
- Distribuição Exponencial: É uma distribuição que representa o tempo entre ocorrências de certo fenômeno.

Conceitos relacionados à inferência estatística:

- Inferência Estatística: “A Inferência Estatística consiste em fazer afirmações probabilísticas sobre as características do modelo probabilístico [o experimento aleatório], que se supõe representar uma população, a partir dos dados de uma amostra aleatória desta mesma população” (REIS 1998);
- Estimação de Parâmetros: A estimação de parâmetros consiste em, baseado numa amostra, inferir características (parâmetros) de toda a população, como sua média

ou proporção. A estimação por intervalo consiste em colocar um Intervalo de Confiança em torno da estimativa pontual do parâmetro;

- Confiança e Significância: Quando fazemos alguma afirmação sobre um experimento aleatório, estamos fazendo uma afirmação probabilística, e como tal, envolve certo grau de incerteza. Deve-se, então, informar qual é a confiança dessa afirmação, ou seja, qual a probabilidade mínima da afirmação ser verdadeira. Valores típicos são 95% e 99%. Significância é definida como 100% - confiança (valores típicos 5% e 1%);
- Intervalo de Confiança: O intervalo de confiança representa a incerteza a respeito de inferências feitas sobre a população. Desta maneira, não se pode estimar um valor pontual para um parâmetro populacional. É necessário estimar um intervalo de valores, centrado na estimativa pontual obtida e também a probabilidade (grau de confiança) do real valor do parâmetro estar dentro deste intervalo, com base nos dados coletados.
- Testes de Hipóteses: Um teste de hipóteses compreende um método científico para a tomada de decisão acerca de fenômenos estocásticos. Num teste de hipóteses há a hipótese nula ( $H_0$ ), aceita *a priori*, e uma hipótese alternativa ( $H_1$ ), contrária à hipótese nula, e que costuma representar aquilo que se deseja provar ou verificar. O resultado de um teste de hipóteses informa se os dados coletados possuem ou não indicações estatísticas suficientemente fortes para que se rejeite a hipótese nula, em favor da hipótese alternativa
- Teste F: É um teste usado para determinar, probabilisticamente, se duas médias amostrais são significativamente diferentes ou não. A hipótese nula, aceita *a priori*, é de que as médias são iguais ( $H_0$ ), e a segunda hipótese, aceita apenas se houver indicações estatísticas suficientes contrárias à hipótese nula, é a que as médias são diferentes ( $H_1$ ). A área da distribuição de probabilidade F abaixo do ponto  $p$  amostral obtido no teste fornece a certeza associada à resposta.
- Análise da Variância: A análise da variância (ANOVA – *ANalysis Of VAriance*) é importante pois permite verificar estatisticamente se as médias dos valores amostrados têm diferenças significativas, a um certo nível de confiança. Aceitar a hipótese de que há diferenças significativas significa dizer que uma mudança nesse

fator irá alterar o valor medido (variável de resposta). Para decidir sobre a igualdade estatística ou não das médias amostrais, utiliza-se o teste F.

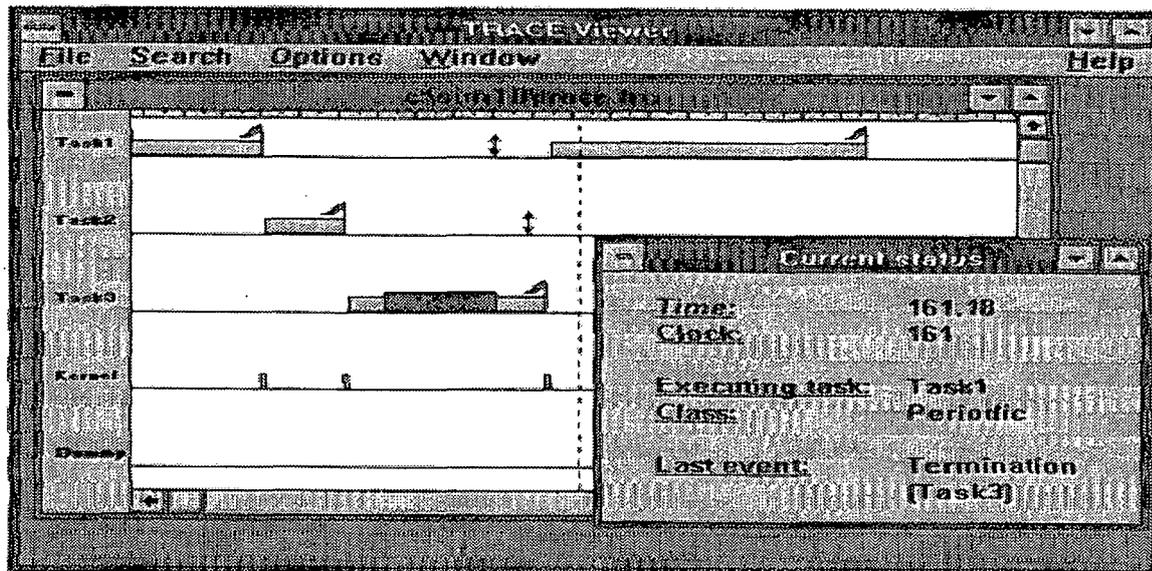
## **5.5. Ferramentas de Simulação**

### **5.5.1. GHOST**

A sigla GHOST significa “*General Hard real-time Oriented Simulator Tool*”. “GHOST é um simulador de escalonamento que foi desenvolvido como uma ferramenta de suporte ao estudo de desempenho de diferentes estratégias de escalonamento em um ambiente de tempo-real. É parte de um *kit* completo de ferramentas (ANCILIOTTI et al 1996) e desenvolvido juntamente com o projeto MORRIS (NATALE et al 1996) para suportar o projeto de uma complexa aplicação de controle robótico.” (SENSINI et al 1997).

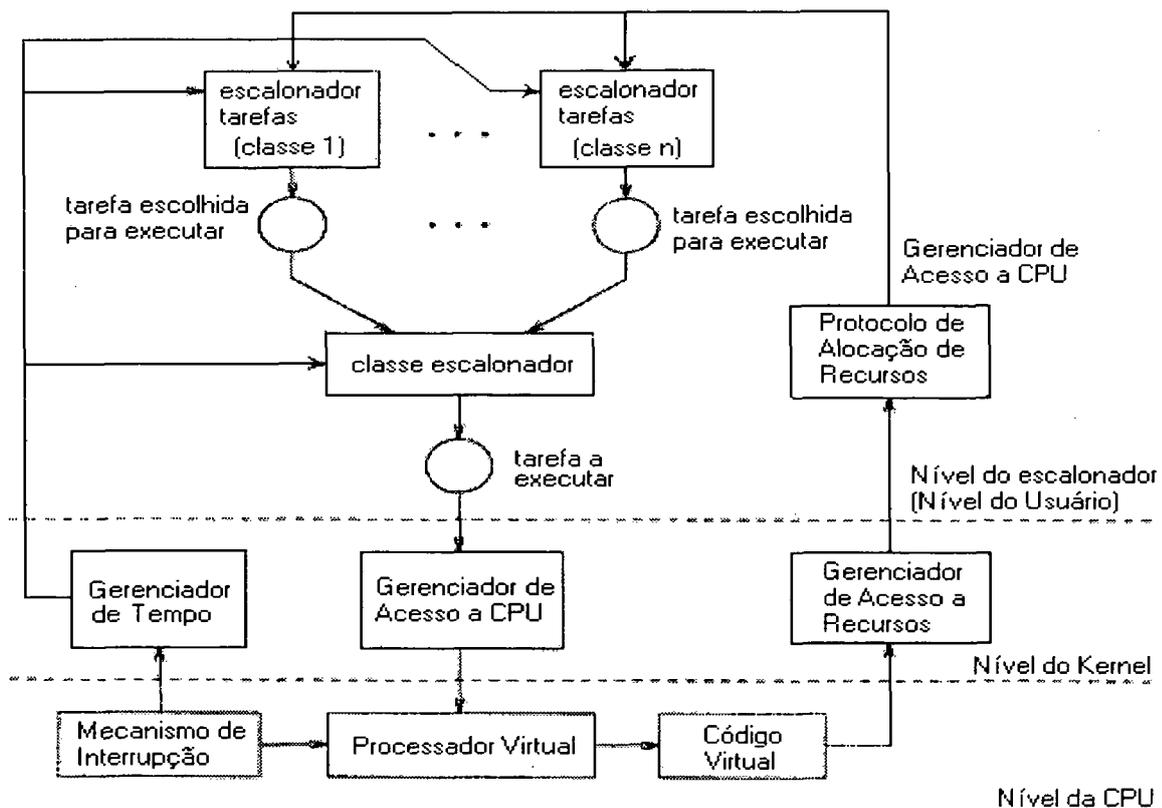
A principal diferença entre o GHOST e outros simuladores, é a sua capacidade de controlar tarefas com diferentes características (periódicas, aperiódicas, *hard*, *soft*, etc), permite definir novos algoritmos de escalonamento e protocolos de alocação de recursos, e suporta uma linguagem simples para descrever funções e distribuições estatísticas (SENSINI et al 1997). As figuras 5.1, 5.2 e 5.3 apresentam, respectivamente, uma tela do programa, a estrutura do (bloco) simulador e um exemplo da especificação de um algoritmo de escalonamento no GHOST.

Figura 5.1 – Tela da ferramenta de simulação GHOST



Fonte: (SENSINI et al 1997)

Figura 5.2 – Estrutura do simulador do GHOST



Fonte: (SENSINI et al 1997)

Figura 5.3 – Exemplo de definição do algoritmo RM no GHOST

```

#include "ghost.h"

void SchedInit ( ) {
    /* Cria uma classe periódica tendo "Psched" como escalonador */

    ClassCreate (PERIODIC, Psched);
    SetQueDis (SysQueue (PERIODIC), RM_PRIO);
}

void Psched (EVENT_T ev, TASK t) {

    switch (ev) {
        case ARRIVAL:
            TaskInsert (t, SysQueue (PERIODIC) );
            t = FirstTask (SysQueue (PERIODIC) );
            TaskSchedule (t, PERIODIC);
            break;
        case END:
            TaskExtract (t, SysQueue (PERIODIC) );
            t = FirstTask (SysQueue (PERIODIC) );
            TaskSchedule (t, PERIODIC);
            break;
        case DEADLINE:
            SimException ("Perda de Deadline");
    }
}

```

Fonte: Adaptação de (SENSINI et al 1997)

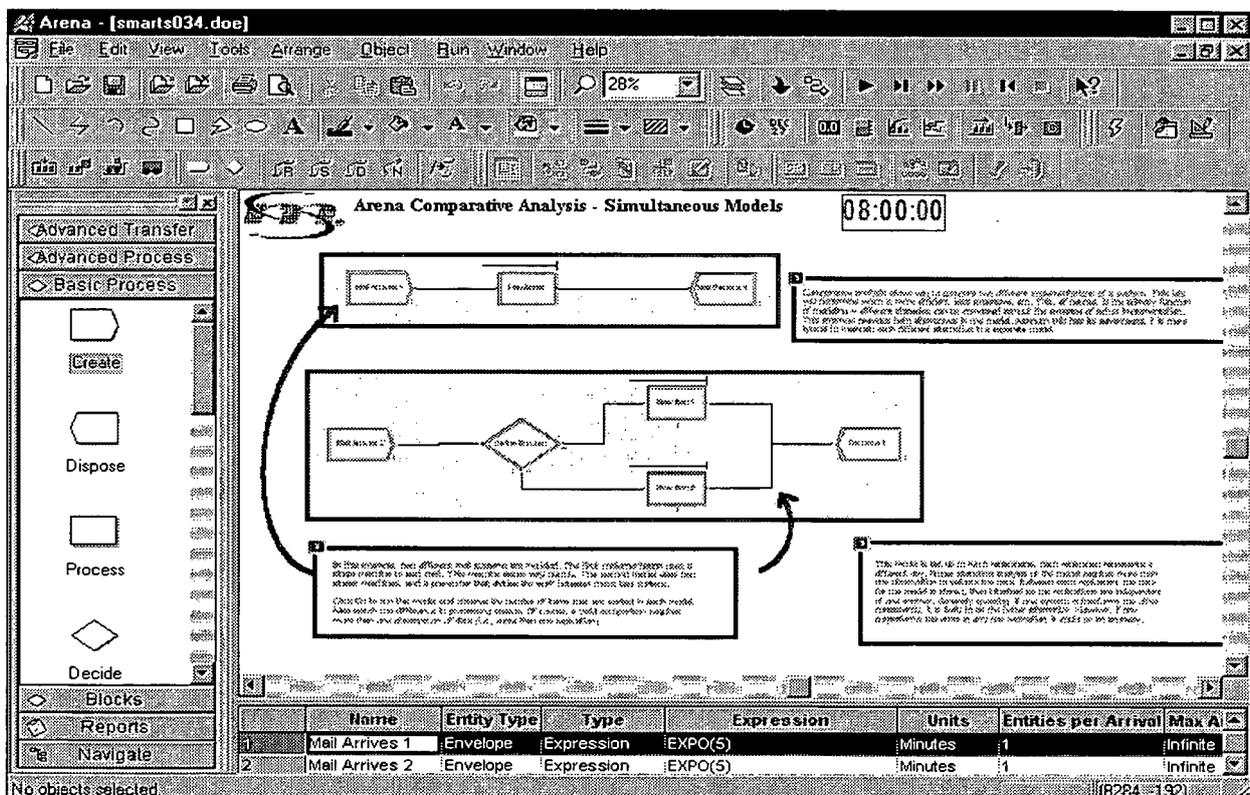
### 5.5.2. Arena

O Arena, da Systems Modeling, é uma ferramenta flexível e poderosa que permite aos analistas criar modelos de simulação animados que representam virtualmente qualquer sistema. Lançado inicialmente em 1993, a ferramenta Arena emprega um design orientado a objeto para desenvolvimento de modelos. Analistas de simulação objetos para definir componentes de sistemas como máquinas, operadores, e dispositivos de manipulação materiais. A ferramenta Arena é baseada na linguagem de simulação SIMAN. Depois de criar um modelo de simulação graficamente, o Arena gera automaticamente o código-fonte SIMAN usado para executar as simulações (TAKUS e PROFOZICH 1997). A figura 5.4 apresenta a tela principal do Arena.

Os *templates* do Arena são uma coleção de mais de 60 módulos, distribuídos junto com a ferramenta. Esses *templates* foram projetados para prover uma coleção de propósito geral para modelar características de todos os tipos de aplicações.

Três painéis compõem os *templates* do Arena: o painel Comum (*common*), contendo módulos que representam simulação fundamental de processos, como chegadas, serviço, e partidas; o painel de Apoio (*support*), contendo módulos suplementares para ações específicas e lógica de decisão; e o painel de Transferência (*transfer*), cujos módulos são usados para modelar a transferência ou fluxo de entidades pelo sistema. Os módulos destes painéis podem ser combinados no mesmo modelo. A animação é incluída automaticamente com muitos dos módulos do Arena para permitir o desenvolvimento rápido de um modelo de simulação.

Figura 5.4 – Tela da ferramenta Arena 4.0



A ferramenta Arena foi projetada para fazer a criação de um modelo de simulação um processo completamente gráfico. Todos os comportamentos de sistemas

são representados usando os módulos gráficos descritos acima (*templates*). Para lógicas como *IF-THEN-ELSE* e *WHILE-UNTIL*, o usuário cria um fluxograma do sistema colocando os módulos gráficos apropriados no modelo Arena e conecta estes módulos diretamente (TAKUS e PROFOZICH 1997).

A tecnologia por trás da ferramenta Arena é a linguagem de simulação SIMAN. Os módulos contidos nos templates Arena foram criados usando os blocos de modelagem do SIMAN como seus componentes. Módulos do SIMAN proporcionam flexibilidade para o usuário e aumentaram controle de lógica do sistema. Os usuários que foram acostumados a escrever para SIMAN codificam diretamente em um editor de texto pode usado de dentro de Arena. Neste caso, Arena provê uma opção para reconhecer este código, que é transformado em módulos gráficos automaticamente (TAKUS e PROFOZICH 1997).

A ferramenta Arena é produto compatível com Microsoft Windows 95 e Windows NT. O produto inteiro usa as Classes de Fundação da Microsoft e foi desenvolvido em Visual C++. Isto permite os usuários usem a arquitetura aberta da Microsoft para integrar dados externos e aplicações com modelos do Arena.

O suporte da ferramenta Arena a ActiveX (OLE) permite para o usuário embutir outras tecnologias, como planilhas eletrônicas, arquivos de Microsoft Word, *clipart*, e apresentações do PowerPoint dentro de modelos de simulação. Os suportes a DAO (*Data Access Objects*) e ODBC (*Open Database Connectivity*) permitem ao usuário integrar todos os sistemas de banco de dados que são compatíveis com o padrão ODBC da Microsoft. (TAKUS e PROFOZICH 1997).

Arena contém ferramentas adicionais que são valiosas para administrar projetos de simulação. O Analisador de Entrada (*input analyser*) é útil para determinar uma distribuição apropriada para um conjunto de dados. Esta distribuição pode ser então incorporada diretamente no modelo.

O Analisador de Saída (*output analyser*) é usado exibir e analisar dados modelo depois da execução da simulação. Opções de exibição gráficas incluem, correlogramas, histogramas, *charts* e mais. Ele também provê análises estatísticas, como intervalos de confiança, análise de variância, e comparações de múltiplos sistemas.

## 6. Avaliação de Desempenho

### 6.1. Introdução

A avaliação de desempenho de sistemas computacionais é complexa, e pode ser muito importante quando se pretende comparar duas máquinas diferentes (talvez para decidir por uma compra importante) ou para comprovar que certa máquina atende os requisitos necessários para certa aplicação.

A avaliação de desempenho de uma máquina pode ser feita sob vários aspectos e níveis diferentes. Para entender o significado e o impacto de algumas métricas (indicadores) de desempenho, é importante conhecer alguns aspectos do *software* e do *hardware* relacionados à medida de performance do sistema.

“A falta de planejamento muitas vezes é a causa do insucesso e uma investigação, e no entanto raros são os pesquisadores que pensam em estatística *antes* de realizar seus experimentos. Em geral eles costumam achar que uma análise de seus dados, não importa como tenham sido obtidos, eventualmente levará sua pesquisa a bom termo. Havendo dificuldades, pensam eles, basta recorrer a um entendido nessas coisas que o problema estará resolvido. Ledo engano”. (BARROS et al 1995)

Cada projeto de avaliação de desempenho possui características ímpares, dependendo do sistema de interesse. Porém, algumas etapas são as mesmas para a grande maioria dos projetos. JAIN (1991) apresenta 10 passos básicos que podem ser seguidos para realizar o projeto de avaliação de desempenho de qualquer sistema:

- Declare os objetivos e defina o sistema: O primeiro passo em qualquer avaliação de desempenho é estabelecer os objetivos e delinear o sistema sob estudo. A escolha dos objetivos e os limites do sistema afetam todas as demais etapas do processo, e por isso devem ser bem definidos;
- Liste serviços e saídas: Cada sistema provê uma série de serviços. Escolha os serviços que são importantes no sistema de acordo com a definição feita. Quando algum desses serviços é requisitado, o sistema pode responder com saídas, algumas

importantes, outras não. Uma lista de serviços e possíveis saídas é útil para posteriormente selecionar métricas e cargas de trabalho corretas;

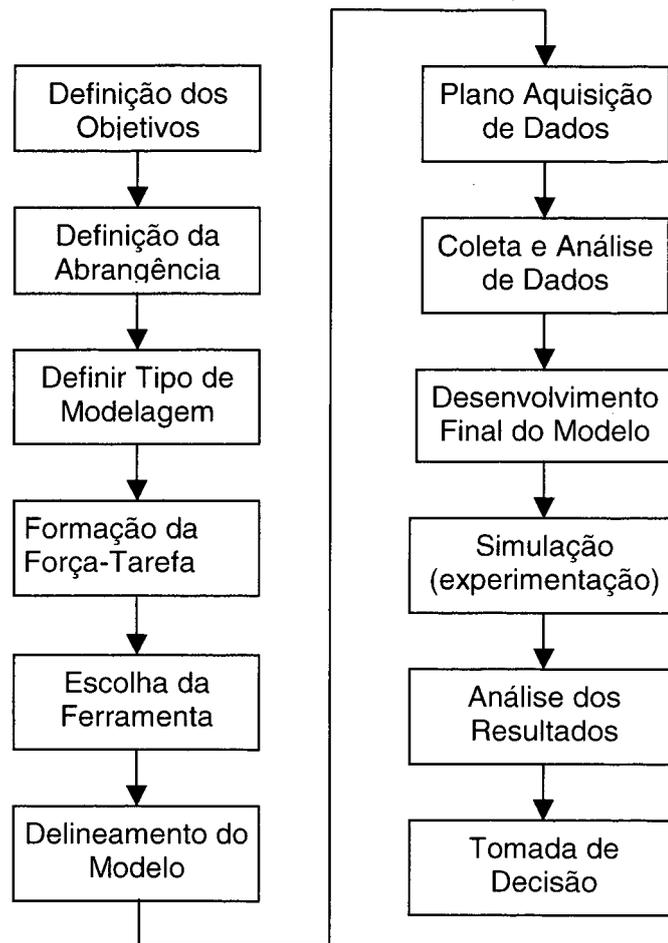
- Selecione métricas: É necessário escolher critérios, métricas, para comparar a performance. Em geral, essas métricas estão relacionadas com a velocidade, tempo de resposta, disponibilidade de recursos, etc. As métricas devem permitir conseguir meios que possibilitem alcançar os objetivos desejados;
- Liste parâmetros: O próximo passo num projeto de avaliação de desempenho é listar todos os parâmetros que afetam a performance do sistema. A lista pode ser dividida em *parâmetros do sistema* e *parâmetros de carga*. Nem todos os parâmetros são facilmente identificados num primeiro estudo.
- Selecione fatores: A lista de parâmetros do passo anterior pode ser dividida também em duas partes: Os parâmetros que variam durante as avaliações e aqueles que permanecem fixos. Os parâmetros que variam durante as avaliações são chamados de fatores, e os seus possíveis valores são chamados de níveis. Os parâmetros que, supõe-se, tenham maior impacto sobre a performance devem ser escolhidos como fatores.
- Selecione uma técnica de avaliação: Há, basicamente, três formas de avaliar o desempenho de um sistema: modelagem analítica, simulação e medição direta. Cada técnica possui características que devem ser cruzadas com o sistema de interesse para verificar sua adequação;
- Selecione a carga de trabalho: A carga de trabalho consiste numa lista de requisições aos serviços providos pelo sistema, e representa o que será 'exigido' do sistema. Para produzir cargas representativas, é necessário medir e caracterizar a carga em sistemas existentes;
- Projete experimentos: Uma vez que se tenha uma lista de fatores e seus níveis, é necessário definir uma seqüência de experimentos que ofereça o máximo de informações com esforço mínimo (menor número de experimentos). Normalmente isso é feito em duas etapas: Na primeira, o número de fatores pode ser grande, mas o número de níveis é pequeno. O objetivo é determinar o efeito relativo de vários

fatores. Na segunda fase, o número de fatores é reduzido e o número de níveis dos fatores mais representativos é aumentado;

- Análise e interprete dados: A interpretação dos resultados de uma análise é fundamental. Deve ser entendido que a análise produz apenas resultados, e não conclusões. Além disso, é importante também compreender que os resultados de medidas e simulações são valores aleatórios e que podem ser diferentes a cada vez que o experimento é repetido. Assim, é necessário realizar uma análise da variância desses resultados;
- Apresente os resultados: O último passo no projeto de avaliação de desempenho é a apresentação dos resultados. Os resultados devem ser apresentados de forma representativa e de fácil entendimento. Para completar o projeto, é necessário repassar por essas etapas várias vezes, refinando os dados e definições.

FREITAS (2000) também cita as etapas principais de um projeto de avaliação de desempenho, sendo coerente e complementar às etapas apresentadas por JAIN (1991). A figura 6.1 ilustra essas etapas.

Figura 6.1 – Etapas de um projeto de avaliação de desempenho



Fonte: (FREITAS 2000)

## 6.2. Métricas de Desempenho

Segundo PATTERSON (1998), algumas das métricas mais importantes de desempenho são as seguintes:

*Throughput*: É a quantidade de tarefas executadas por unidade de tempo. Refere-se à taxa de trabalho realizado num certo tempo.

$$\textit{throughput} = \frac{\textit{tarefas\_executadas}}{\textit{tempo}} \quad (23)$$

MIPS: Milhões de Instruções Por Segundo. Essa métrica considera apenas a quantidade de instruções em ponto fixo realizadas pela máquina por unidade de tempo.

$$MIPS = \frac{\text{quantidade\_de\_instruções\_num\_programa}}{\text{tempo\_execução} \cdot 10^6} \quad (24)$$

MFLOPS: Milhões de operações em ponto flutuante por segundo, popularmente conhecido como “megaflops”. Esse indicador foi criado pois as operações aritméticas em ponto flutuante são muito utilizadas em cálculos científicos.

$$MFLOPS = \frac{\text{quantidade\_operações\_ponto\_flutuante\_num\_programa}}{\text{tempo\_execução} \cdot 10^6} \quad (25)$$

Porém, MFLOPS não é um bom indicador de desempenho. PATTERSON (1996) mostra que, entre outros problemas, MFLOPS “podem variar inversamente com o desempenho”, o que restringe muito sua utilidade.

Tempo de Resposta: “O tempo de resposta é o tempo entre o início e o término da execução de uma tarefa” (PATTERSON 1998), também é conhecido como tempo de execução. Entre os vários possíveis indicadores de desempenho, PATTERSON (1998) avisa que “a única medida completa e confiável de performance de computadores é o tempo”,

Tempo de Execução: O Tempo de Execução é o tempo decorrido desde o momento que o processo é criado até o momento em que termina; é composto pelo tempo em que o processo executa no processador mais o tempo em que o processo gasta com operações de entrada/saída e o tempo de espera pelo sistema e outros processos. Processos *CPU-Bound* têm a maior parte de seu tempo de execução gasto com a CPU enquanto processos *I/O-Bound* (ou *E/S-Bound*) têm seu tempo de execução devido principalmente às operações de E/S.

$$\text{tempo de execução} = \text{tempo CPU} + \text{tempo E/S} + \text{tempo espera} \quad (26)$$

“Deve-se notar a diferença entre *tempo de resposta* ou *tempo de execução* e *tempo de execução de CPU*, que é o tempo em que o processador está trabalhando, e não inclui o tempo de espera por entrada/saída ou executando outros programas. O tempo de execução da CPU pode ainda ser dividido em *tempo do usuário na CPU*, que é o tempo

executando tarefas do usuário e *tempo do sistema na CPU*, que é o tempo gasto pelo sistema operacional [*overhead*]”. (PATTERSON 1998)

Como o desempenho do processador costuma ser significativo no desempenho do sistema, algumas vezes toma-se o desempenho apenas desse componente como sendo o desempenho do sistema como um todo, o que é válido para aplicações *CPU Bound* (compiladores, programas de cálculo, ...). Já aplicações *I/O Bound* (multimídia, impressão de relatórios, ferramentas de disco, ...) têm no tempo de E/S o principal fator de desempenho. Porém, deve-se distinguir entre a *performance do sistema*, que é o tempo de resposta no sistema, e a *performance da CPU*, que é o *tempo do usuário na CPU*.

Em relação à *performance da CPU*, PATTERSON (1998) mostra que o *tempo do usuário na CPU*, ou simplesmente *tempo de execução na CPU*, depende basicamente de três fatores inter-relacionados:

- Quantidade de Instruções por Programa. A quantidade de instruções permite relacionar dois programas ou compiladores diferentes. Por si só, a quantidade de instruções (medida no MIPS, por exemplo) não é suficiente, pois não permite comparar máquinas com conjuntos de instruções diferentes.
- Ciclos de Clock por Instrução (CPI), que é a o número médio de ciclos de *clock* que uma instrução leva para executar, considerando a frequência dinâmica de ocorrência dessas instruções. “O CPI provê um meio de comparar duas implementações diferentes do mesmo conjunto de instruções, uma vez que a quantidade de instruções para um programa será, com certeza, a mesma [pois o conjunto de instruções é o mesmo]” (PATTERSON 1998).
- Período do Clock, que é o tempo entre dois pulsos de *clock* consecutivos. O *período de clock* é o inverso da *frequência do clock*.

Assim, o *tempo de execução na CPU* pode ser escrito como:

$$\text{tempo\_execucao\_CPU} = \frac{\text{Instruções}}{\text{Programa}} \cdot \frac{\text{Ciclos\_Clock}}{\text{Instrução}} \cdot \frac{\text{Segundos}}{\text{Ciclo\_Clock}} \quad (27)$$

ou, mais resumidamente,  $tempo\_execucao\_CPU = I.CPI.T_{CLOCK}$

Esses três fatores representam de maneira completa o tempo de execução, e sua equivalência pode ser verificada na simplificação abaixo.

$$tempo\_execucao\_CPU = \frac{\cancel{Instruções}}{Programa} \cdot \frac{\cancel{Ciclos\_Clock}}{\cancel{Instrução}} \cdot \frac{\cancel{Segundos}}{\cancel{Ciclo\_Clock}} = \frac{Segundos}{Programa}$$

Outros fatores que influenciam o desempenho podem ser incluídos ou abstraídos, dependendo basicamente dos objetivos que devem ser alcançados. Para a quantificação do desempenho de sistemas distribuídos, por exemplo, os fatores relacionados à rede de comunicação (topologia, velocidade, largura de banda, protocolos, etc) devem ser incluídos em detalhes, enquanto outros talvez possam ser ignorados.

Assim, antes de iniciar qualquer avaliação de desempenho, é necessário definir exatamente o que significa “desempenho” (ou performance) no escopo do estudo. Um mesmo sistema pode ter inúmeras métricas de desempenho, dependendo do interesse do avaliador.

Um conceito muito importante é o de Performance: PATTERSON (1998) define performance como o inverso do tempo de execução de uma tarefa numa máquina  $X$ , ou seja:

$$performance_x = \frac{1}{tempo\_execução_x} \quad (28)$$

A Performance Relativa entre duas máquinas e  $Y$  relaciona quantas vezes a máquina  $X$  é mais rápida que a máquina  $Y$ . Dizer que “ $X$  é  $n$  vezes mais rápida que  $Y$ ”, significa:

$$n = \frac{performance_x}{performance_y} = \frac{tempo\_execução_y}{tempo\_execução_x} \quad (29)$$

A Lei de Amdahl estabelece que a mudança de desempenho obtida na mudança do nível de um fator está limitada à fração do tempo relacionada a esse fator. PATTERSON (1998) indica como objetivo de projeto para obter maior desempenho: “Torne rápido o caso freqüente”, significando que o aumento de desempenho será tão mais significativo quanto for a utilização do componente acelerado.

Como exemplo, suponha que se acelere a memória de um computador em 10 (dez) vezes. Só poderíamos esperar um aumento total de desempenho do computador de 10 vezes se os programas passassem 100% do tempo acessando a memória. Como os programas gastam tempo com outros recursos além da memória, o aumento total será, certamente, menor que 10 vezes, e tão menor quanto menos acessos à memória os programas fizerem. Para 50% do tempo acessando a memória, o aumento do desempenho seria de apenas 1,82 vezes.

### **6.3. Carga de Trabalho**

Segundo JAIN (1991), “para se medir a performance de um sistema, é necessária a monitoração do mesmo quando submetido a uma determinada carga de trabalho (*workload*). Ou seja, deve-se examinar o comportamento do sistema durante a execução de um conjunto de tarefas específicas. Essas cargas de trabalho usadas para medir a performance de um sistema, durante um estudo, são conhecidas como *workloads* de teste”.

Uma carga de trabalho (*workload*) pode ser real ou sintética. Uma carga real é aquela apresentada no dia-a-dia do sistema, e não pode ser repetida. Por esse motivo seu uso não é adequado como *workload* de teste. As cargas de trabalho sintéticas são aquelas criadas especificamente para realizar medidas de desempenho. As cargas sintéticas dividem-se em quatro subgrupos.

- Instruções de Adição. É a forma mais simples de se gerar uma *workload*. Nos primeiros computadores, o tempo de execução da instrução de adição era usado como métrica de desempenho. Totalmente limitado para os dias de hoje.
- Misto de Instruções. Esse tipo de carga de trabalho considera o tempo de execução de vários tipos de instrução, baseado em estatísticas sobre a freqüência de

ocorrência dessas instruções. Os resultados são apresentados em MIPS ou MFLOPS. Porém, conforme PATTERSON (1996) essas duas medidas não são bons indicadores de desempenho, “podendo variar inversamente com o desempenho do sistema [grifo meu]”. Elas apresentam resultados muito limitados quanto ao desempenho do sistema todo.

- Kernels. JAIN (1991) diz que “Sua principal característica é que as instruções não são mais examinadas individualmente, mas em conjunto com outras na implementação de algum algoritmo específico.” e complementa que “os algoritmos que aparecem são os mais variados possíveis e tornam-se cada vez mais específicos a certos tipos de aplicações (banco de dados, processamento numérico [, multimídia, redes, etc] )”. Um problema com esse tipo de *workload* que os algoritmos usados normalmente não possuem instruções de entrada/saída, fazendo com que o desempenho dessa importante parte do sistema não apareça nos resultados. Os *kernels* não fazem chamadas ao S.O. e não avaliam outros componentes do sistema senão a CPU.
- Programas Sintéticos. Os programas sintéticos tentam levar em consideração mais componentes do sistema do que qualquer outra *workload* apresentada anteriormente. Normalmente são algoritmos bem simples onde chamadas do sistema (operacional) são executadas repetidamente. “Não só serviços de entrada/saída são requisitados, mas também operações de criação de processos e alocação de memória.” (ibidem).

As cargas reais normalmente são Aplicações de Benchmarks, um conjunto de aplicações típicas de determinadas classes que são usados para representar as demais aplicações dessas classes, generalizando as medições e resultados obtidos sobre o *benchmark*. As aplicações do *benchmark* devem representar os programas e linguagens de programação usados numa determinada época, devendo ser substituídos regularmente.

A seleção correta da carga de trabalho adotada é de fundamental importância sobre os resultados que serão obtidos. Conforme JAIN (1991),

“a escolha das métricas deve refletir o desempenho do serviço a nível de sistema e não de componente. Por

exemplo, MIPS é uma métrica justificável na comparação de duas CPUs [mesmo assim pode não ser uma boa medida], mas não é apropriada para a comparação de dois sistemas de *timesharing*. Para *timesharing* a métrica deveria ser transações por segundo e, neste caso, além da CPU, outros componentes também devem ser adicionados na análise.”

Uma carga de trabalho de teste também deve ser representativa de uma classe de aplicação real. Não existe uma carga de trabalho que seja adequada a todas as aplicações. Para que haja representatividade, é necessário que a carga de trabalho tenha a taxa de chegada, a demanda de recursos e o perfil de utilização de recursos o mais parecido possível com a classe de aplicação que se deseja avaliar. Deve-se, entretanto, tomar cuidado para não tornar demasiadamente específica a carga de trabalho, o que pode comprometer a representatividade do modelo.

Conforme FREITAS (2000), “o desempenho de sistemas distribuídos, com seus inúmeros clientes, servidores e redes depende, e muito, das características de sua carga. Por isso, um ponto de extrema importância em estudos de avaliação de desempenho é a compreensão e caracterização da carga de trabalho a qual o sistema está submetido.”

### 6.3.1. Benchmarks

Como exposto acima, os melhores programas para se avaliar o desempenho são programas reais, tipicamente usados por usuários de certa classe de aplicações. Assim, o conjunto de programas que compõem um *benchmark* reflete apenas o desempenho da máquina quando executando especificamente os programas do *benchmarks* que, por serem programas típicos, conclui-se, por indução, que seu desempenho seja muito semelhante a outros programas da mesma classe.

Deste modo, os *benchmarks* são separados por classe de aplicação, e substituídos periodicamente, pois as características das aplicações vão se alterando ao longo dos anos. Como os usuários comuns não possuem seu próprio *benchmark* para caracterizar sua carga de trabalho, recorrem a *benchmarks* prontos, como os produzidos pela SPEC

(*System Performance Evaluation Cooperative*), uma cooperativa americana de empresas do ramo de fabricação de computadores fundada em 1988.

“O objetivo da SPEC é garantir que o mercado tenha um conjunto de métricas confiável e útil para diferenciar sistemas candidatos. A metodologia básica da SPEC é prover o *benchmarker* com um conjunto de códigos fonte padronizados, baseado em aplicações existentes que já tenham sido portadas para uma ampla variedade de plataformas por seus desenvolvedores. O *benchmarker* então toma o código fonte, compila-o no sistema em questão e então pode ajustar o sistema para obter os melhores resultados. O uso de códigos fonte já aceitos e portáveis reduz significativamente o problema de comparar maçãs com laranjas.” (SPEC 2000)

A SPEC possui um comitê original, o OSG (*Open Systems Group*). Esse grupo está concentrado em *benchmarks* para estações de trabalho high-end e servidores em sistemas abertos. Conforme (SPEC 2000) seus sub-comitês são:

- CPU: *Benchmarks* para caracterização da carga de trabalho sobre o processador. Produziu os SPECmarks e outros CPU benchmarks (SPECint95, SPECint2000, SPECfp95, SPECfp2000, SPECrates2000, etc);
- JAVA: *Benchmarks* para avaliação da máquina virtual Java. Produziu o JVM98;
- SDM: *Benchmarks* para comandos UNIX.
- SFS: *Benchmarks* servidores de arquivos e outros. Produziu o SFS93 (LADDIS) e SFS97;
- WEB: *Benchmarks* para servidores *web*. Produziu o SPECweb96 e o SPECweb99;

### **O Benchmark SPECweb96**

O objetivo do SPECweb é fornecer uma medida de serviços básicos da *web*, sendo que o SPECweb96, última versão disponível no site da SPEC até o presente momento (embora já tenha sido lançado o SPECweb99), é baseado no *log* de muitos servidores. Esses *logs* mostram similaridade nas relações entre tamanhos de arquivos e a frequência de seus acessos. Um razoável número de requisições (35%) é para arquivos muitos pequenos: pequenas imagens, arquivos-texto curtos, etc. A maioria dos acessos

(50%) é para arquivos de alguns KB de tamanho: principalmente arquivos HTML e imagens. À medida que o tamanho aumenta, a frequência de acesso diminui, sendo pequena (14%) para arquivos entre 10 e 100 KB, e apenas ocasionalmente (1%) um acesso a documentos muito grandes ou arquivos multimídia maiores que 100 KB e menores que 1 MB. (SPECWEB 2000)

Os tamanhos foram divididos em 4 classes, sendo que há 9 valores discretos em cada classe, e a probabilidade de acesso dentro de uma classe não é uniforme, mas *Poisson* centrada no ponto médio da classe. A tabela 6.1 apresenta a carga de trabalho conforme a SPECweb96.

Tabela 6.1 – Tamanho de arquivos em cada classe de acessos

	Classe 0	Classe 1	Classe 2	Classe 3
1	102	1024	10,240	102,400
2	204	2,048	20,480	204,800
3	307	3,072	30,720	307,200
4	410	4,096	40,960	409,600
5	512	5,120	51,200	512,000
6	614	6,144	61,440	614,400
7	717	7,168	71,680	716,800
8	819	8,192	81,920	819,200
9	922	9,216	92,160	921,600

A carga de trabalho resultante pode ser imaginada como representando o comportamento de um sistema que disponibilize uma quantidade de *home pages*. É um bom ponto de partida para testar a capacidade de servidores em atender requisições GET básicas, mas ainda possui muitas limitações. Serviços como páginas dinâmicas, *scripts*, FTP, transações seguras, entre outros, são indicados como trabalhos futuros, e possivelmente já incluídos no SPECweb99. (SPECWEB 2000)

#### 6.4. Projeto Experimental

De acordo com JAIN (1991), “o objetivo de um projeto de experimentos é obter o máximo de informações com um número mínimo de experimentos”. Já BARROS et al

(1995) diz que “a essência de um bom planejamento consiste em projetar um experimento de forma que ele seja capaz de fornecer exatamente o tipo de informação que procuramos”.

Os seguintes conceitos são usados freqüentemente na análise de projetos de experimentos:

- Variável de Resposta: A saída de um experimento é chamada variável de resposta. Geralmente a variável de resposta é a medida de desempenho do sistema;
- Fator: Cada variável que afeta a variável de resposta e pode assumir mais de um valor é chamada de fator. Os fatores são também chamados preditores;
- Nível: Os valores que um fator pode assumir são chamados níveis desse fator. Em outras palavras, um nível é uma alternativa para um fator;
- Fator Primário: Os fatores cujo efeito precisam ser quantificados são chamados de fatores primários;
- Fator Secundário: Os fatores que influenciam na performance, mas cujo impacto não é de interesse e não precisam ser quantificados, são chamados fatores secundários;
- Replicação: Replicação é a repetição de todos ou alguns experimentos. É importante repetir o mesmo experimento mais de uma vez para possibilitar a análise da variância;
- Projeto: Um projeto de experimento consiste em especificar o número de experimentos, as combinações de fatores e níveis para cada experimento e o número de replicações de cada experimento;
- Unidade Experimental: Qualquer entidade que é usada pelo experimento é chamada de uma unidade experimental;
- Interação: Dois fatores A e B têm interação se o efeito de um depende do nível do outro. A interação entre fatores é um aspecto muito relevante.

“Os dois critérios chaves para seleção dos parâmetros [fatores] são seus impactos sobre a performance ou sua variância. Parâmetros que não influenciam a performance do sistema devem ser omitidos.” (FREITAS 2000).

Existem vários tipos de projetos de experimentos. Eles incluem projetos de um fator e principalmente projetos fatoriais, completos ou fracionários, cuja utilidade depende da quantidade de fatores, níveis, e replicações usados, além de características do próprio sistema, como os efeitos da interação entre os fatores.

Segundo BARROS (1995), “o número de efeitos de interação de ordem alta aumenta dramaticamente com o número de fatores. [...] Muitos deles têm efeitos muito pequenos [...]”. Além disso quando a quantidade de fatores cresce, aumenta também as chances de um fator ou mais fatores não afetarem significativamente a variável de resposta. Nesses casos, o custo para a realização de um projeto fatorial completo (que exige  $2^k$  experimentos para  $k$  fatores) pode ser muito caro, e a utilização de um projeto fatorial fracionário (que exige menos experimentos) pode ser uma boa solução.

Os projetos experimentais fatoriais fracionários, designados normalmente como  $2^{k-p}$ , têm como propriedade principal a de reduzir o número de experimentos para avaliar os efeitos de  $k$  fatores de  $2^k$  experimentos para  $2^{k-p}$  experimentos, onde  $p$  é um fator inteiro ajustável de redução.  $p \in \mathbb{Z}^+ \mid p < k$ . Quanto maior o valor de  $p$ , menos experimentos são necessários.

Obviamente, há um preço por esse ganho. Tal preço é a ‘mistura’ (*confounding*) de efeitos de interações, sendo impossível isola-los. Quanto maior o valor de  $p$ , maior é a ‘mistura’ de efeitos. Esse preço, indesejável, pode ser tolerado se as interações entre os fatores forem pequenas.

Após os experimentos terem sido realizados, conforme o projeto experimental adotado, normalmente segue-se a uma fase de otimização, que visa maximizar ou minimizar a variável de resposta. Vários métodos de otimização foram desenvolvidos e estão disponíveis, entre eles, o método da superfície de resposta e o método Simplex, com suas variantes (ROSS 1988). A otimização não será tratada neste trabalho.

A seguir são apresentadas brevemente as características de alguns projetos experimentais.

### 6.4.1. Projeto com Um Fator

O projeto de experimentos com um único fator é usado para comparar as várias alternativas (níveis) de um único parâmetro (fator). Assume-se que o resultado produzido por cada nível seja independente dos demais. As equações utilizadas nesse tipo de Projeto são apresentadas na tabela 6.2.

Nessas equações, (30) a (35),  $y_{ij}$  é o resultado do experimento com o (único) fator (fator A) no nível  $i$  na  $j$ -ésima replicação;  $\mu$  é um parâmetro comum a todos os experimentos chamado de média geral;  $\alpha_j$  é o efeito do fator,  $\varepsilon_{ij}$  é o componente de erro aleatório em cada experimento,  $\bar{y}_{..}$  é a média dos resultados dos experimentos de todos os níveis do fator e de todas as replicações, ou seja, a média geral;  $\bar{y}_{.j}$  é a média dos resultados dos experimentos de todos os níveis do fator na  $j$ -ésima replicação;  $a$  é o número de níveis do fator;  $r$  é o número de replicações.

Tabela 6.2 – Equações para o projeto experimental com um fator

Modelo	$y_{ij} = \mu + \alpha_j + e_{ij}$	(30)
Efeitos	$\mu = \bar{y}_{..}; \alpha = \bar{y}_{.j} - \bar{y}_{..}$	(31)
Alocação da variação	$\sum_{ij} y_{ij}^2 = \sum_{i,j} \mu^2 + \sum_{i,j} \alpha_j^2 + \sum_{i,j} e_{i,j}^2$	(32)
Graus de liberdade	$ar = 1 + (a-1) + a(r-1)$	(33)
Médias quadráticas	$MSA = \frac{r \sum \alpha_i^2}{a-1}; MSE = \frac{\sum y_{ij}^2 - ar\mu^2 - r \sum \alpha_i^2}{a(r-1)}$	(34)
Análise da variância	$\frac{MSA}{MSE} > F_{[1-\alpha; a-1, a(r-1)]}$	(35)

### 6.4.2. Projeto Fatorial $2^k$

Conforme JAIN (1991), um projeto de experimentos  $2^k$  é usado para determinar o efeito de  $k$  fatores, cada um deles tendo dois níveis. O estudo de apenas dois níveis por fator (normalmente os níveis extremos) é justificado pelo fato que a performance aumenta continuamente ou diminui continuamente enquanto o fator varia do nível mínimo para o nível máximo.

Dados  $k$  fatores com 2 níveis, são necessários  $2^k$  experimentos. O projeto fatorial  $2^k$  consiste na construção da matriz de sinais ( $2^k \times 2^k$ ) para guiar na determinação dos níveis de cada fator em cada experimento e na determinação dos efeitos dos fatores no desempenho obtido em cada experimento. Em seguida é necessária uma verificação de cada uma das suposições feitas inicialmente. O modelo matemático para esse projeto experimental é apresentado na tabela 6.3. Nas equações (36) a (41),  $\beta_i$  é o efeito do segundo fator (fator B) e  $b$  é o número de níveis do segundo fator.

Tabela 6.3 – Equações para o projeto experimental  $2^k$

Modelo	$y_{ij} = \mu + \alpha_j + \beta_i + e_{ij}$ (36)
Efeitos	$\mu = \bar{y}_{..}; \alpha = \bar{y}_{.j} - \bar{y}_{..}; \beta = \bar{y}_{i.} - \bar{y}_{..}$ (37)
Alocação da variação	$\sum_{ij} y_{ij}^2 = ab\mu^2 + b\sum_j \alpha_j^2 + a\sum_i \beta_i^2 + \sum_{ijk} e_{ijk}^2$ (38)
Graus de liberdade	$ab = 1 + (a-1) + (b-1) + (a-1)(b-1)$ (39)
Médias quadráticas	$MSA = \frac{b\sum_j \alpha_j^2}{a-1}; MSB = \frac{a\sum_i \beta_i^2}{b-1}; MSE = \frac{\sum_{ijk} e_{ijk}^2}{(a-1)(b-1)}$ (40)
Análise da variância	$\frac{MSA}{MSE} > F_{[1-\alpha; a-1, (a-1)(b-1)]}; \frac{MSB}{MSE} > F_{[1-\alpha; b-1, (a-1)(b-1)]}$ (41)

### 6.4.3. Projeto Fatorial $2^{k_r}$

Um dos problemas do projeto  $2^k$  é que os experimentos são aleatórios, e podem fornecer resultados diferentes se forem realizados novamente, sob exatamente as mesmas condições. Assim, é interessante poder avaliar o erro associado aos experimentos para obter uma resposta melhor. Ao contrário do  $2^k$ , o projeto  $2^{k_r}$  repete (replica) cada experimento  $r$  vezes, permitindo avaliar esse erro e o efeito na performance através da variação dos resultados dos experimentos. O projeto  $2^{k_r}$  exige que sejam realizados  $2^k \cdot r$  experimentos. O modelo matemático para esse tipo de projeto é apresentado na tabela 6.4.

Tabela 6.4 – Equações para o projeto experimental  $2^{k_r}$

Modelo	$y_{ij} = \mu + \alpha_j + \beta_i + \gamma_{ij} + e_{ijk}$ (42)
Efeitos	$\mu = \bar{y}_{...}; \alpha_j = \bar{y}_{.j} - \bar{y}_{...}; \beta_i = \bar{y}_{i.} - \bar{y}_{...}; \gamma_{ij} = \bar{y}_{ij.} - \bar{y}_{i.} - \bar{y}_{.j} + \bar{y}_{...}$ (43)
Alocação da variação	$\sum_{ijk} y_{ijk}^2 = abr\mu^2 + br \sum_j \alpha_j^2 + ar \sum_i \beta_i^2 + r \sum_{ij} \gamma_{ij}^2 + \sum_{ijk} e_{ijk}^2$ (44)
Graus de liberdade	$abr = 1 + (a-1) + (b-1) + (a-1)(b-1) + ab(r-1)$ (45)
Médias quadráticas	$MSA = \frac{br \sum_j \alpha_j^2}{a-1}; MSB = \frac{ar \sum_i \beta_i^2}{b-1};$ $MSAB = \frac{r \sum_{ij} \gamma_{ij}^2}{(a-1)(b-1)}; MSE = \frac{\sum_{ijk} e_{ijk}^2}{ab(r-1)}$ (46)
Análise da variância	$\frac{MSA}{MSE} > F_{[1-\alpha; a-1, ab(r-1)]}; \frac{MSB}{MSE} > F_{[1-\alpha; b-1, ab(r-1)]}$ $\frac{MSAB}{MSE} > F_{[1-\alpha; (a-1)(b-1), ab(r-1)]}$ (47)

É impossível dar alguma resposta em relação a um evento estocástico (aleatório) sem ter uma estimativa da incerteza associada a essa resposta. Assim, sempre que se chega a um resultado, produto de um modelo não determinístico, faz-se necessário

apresentar o intervalo de confiança da resposta, dado um nível de certeza. A análise da variância permite esse procedimento.

## 7. O Ambiente Multicomputador CRUX

O ambiente CRUX representa um ambiente completo para programação paralela, sendo composto pelo multicomputador CRUX e pelo Sistema Operacional  $\alpha$ -CRUX, desenvolvidos no Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina, pelo grupo de pesquisa em Computação Paralela e Distribuída.

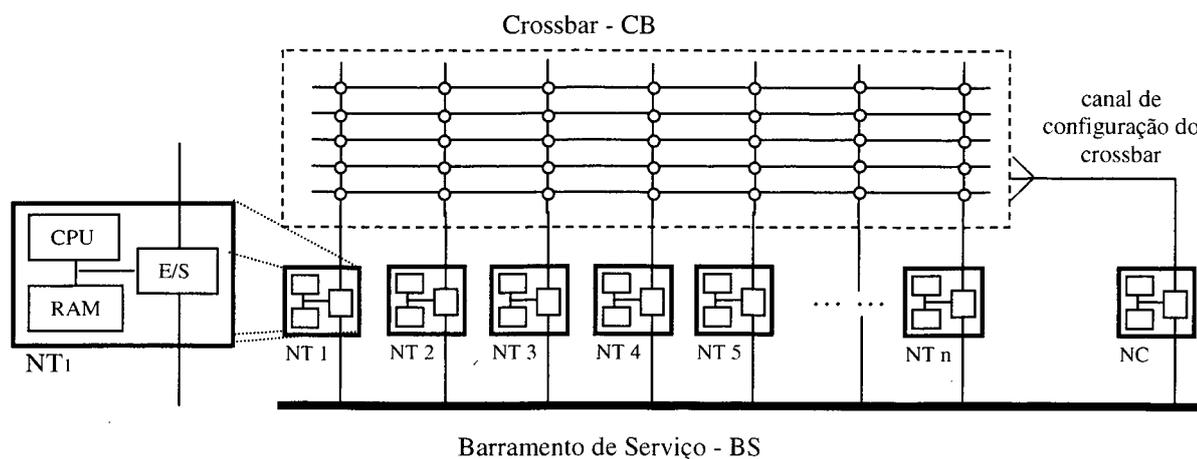
### 7.1. A Arquitetura do CRUX

O multicomputador CRUX é composto de um conjunto de Nós de Trabalho (NT) ligados por meio de um comutador de conexões e um barramento compartilhado. O comutador de conexões (*crossbar*) é manipulado durante o funcionamento normal da máquina pelo Nó de Controle (NC), que por sua vez utiliza um Barramento de Serviço (BS) e um canal de configuração para determinar as necessidades do sistema e definir dinamicamente a estrutura da rede comunicação. A figura 7.1 apresenta o ambiente CRUX.

O nó de controle (NC) utiliza o barramento para realizar uma pesquisa seqüencial de modo a determinar os pedidos nos nós de trabalho. Esta pesquisa é feita através do questionamento pelo envio de um comando ao nó inquirido. Caso este Nó não deseje nenhum serviço, o NC questiona o próximo nó de trabalho. No entanto, o nó questionado desejar algum serviço, ele então envia uma requisição ao NC. Após atender cada pedido, o NC volta ao questionamento seqüencial dos nós de trabalho (NT). Se o pedido por de conexão com outro NT, o NC envia uma nova configuração ao comutador de conexões. Se o nó solicitado já estiver conectado, o pedido é colocado numa fila e o Nó de Controle prossegue com o questionamento seqüencial.

Os NTs são computadores pessoais, com processador, microcódigo, memória própria e dispositivos de E/S, incluindo a interface com o comutador de conexões, com 4 canais físicos e a interface com o barramento de serviço.

Figura 7.1 – Arquitetura do multicomputador CRUX



O NC é responsável pelo controle do comutador de conexões (*crossbar*), através de um canal de configuração. Todos os demais nós podem requisitar conexões ponto-a-ponto com outro nó de trabalho, comunicando com o nó de controle através do barramento de serviço.

De forma semelhante a um NT, o NC é constituído de processador, memória RAM e iria ROM com o seu microcódigo. Entretanto, a capacidade de comandar o comutador de conexões o distingue dos demais nós. Devido a essa importante atribuição, ele é dedicado exclusivamente às tarefas relacionadas com essas funções, e não utiliza o *crossbar* para se comunicar com outros nós.

O comutador de conexões é do tipo *crossbar*, que estabelece uma rede de interconexão dinâmica entre os nós processadores. Todo nó tem conexão direta com qualquer outro nó, estabelecida dinamicamente, e toda comunicação é ponto-a-ponto.

O barramento de serviço é o canal de comunicação entre os NT e o NC, e é utilizado apenas para troca de mensagens curtas, com requisições e respostas pré-estabelecidas. O NC verifica, seqüencialmente, se há algum NT que deseje se comunicar com ele, quando então recebe uma mensagem de requisição do NT interessado. O NC permanece nesse *polling* indefinidamente.

## 7.2. O Sistema Operacional $\alpha$ -CRUX

O sistema operacional  $\alpha$ -CRUX, utilizado no ambiente CRUX é constituído de um *microkernel* distribuído, uma biblioteca de funções para acesso às primitivas do sistema e de um servidor. O gerenciamento de memória não utiliza memória virtual, o que introduziria indeterminismo no tempo de execução de uma atividade. A comunicação entre processos é síncrona, feita ponto-a-ponto e com endereçamento direto (CORSO 1996). Dois processos servidores realizam as primitivas do sistema: servido de processos e servidor de arquivos. “Os processos de aplicação acessam estes serviços através de uma cama de *stubs*, que provê uma interface de programação compatível com a do ambiente UNIX” (CORRÊA 1998).

### Estrutura do *Microkernel*

O sistema operacional CRUX é composto de cinco camadas, mais a camada das aplicações, conforme a figura 7.2.

“A camada de chamada do sistema fornece serviços equivalentes aos serviços providos pelo sistema UNIX, acrescidos de serviços de comunicação do modelo das redes de processos comunicantes. A camada das comunicações de alto nível oferece serviços para o transporte de mensagens volumosas entre os nós de trabalho através da rede de comunicação. A camada das chamadas de núcleo fornece os serviços para o estabelecimento e o cancelamento de conexões no *crossbar*, além de alocações e da liberação de nós de trabalho. A camada das comunicações de baixo nível fornece serviços para a troca de mensagens de serviços entr os nós de trabalho e o nó de controle. A camada das aplicações refere-se a aplicações desenvolvidas sobre o sistema e a da máquina real refere-se aos recursos materiais do multicomputador CRUX”. (CORRÊA 1998)

Quadro 7.1 – Estrutura em camada do sistema operacional  $\alpha$ -CRUX

Camada das Aplicações
Camada das Chamadas do Sistema
Camada das Comunicações de Alto Nível
Camada das Chamadas do Núcleo
Camada das Comunicações de Baixo Nível
Camada da Máquina Real

O *microkernel* do sistema  $\alpha$ -CRUX fornece serviços de comunicação entre processadores e alocação de processadores. O *microkernel* está estruturado em três camadas e cada nó processador tem um *microkernel* em memória ROM. Os serviços tratam de gerência do barramento de serviço, gerência de nós de conexões e gerência e comunicação de processos.

### Suporte a Tempo Real

CORRÊA (1998) propõe um conjunto mínimo de primitivas de suporte a tempo real para o ambiente do CRUX baseado no padrão POSIX da IEEE e considerando o modelo de processos *multithread*, onde um processo pode conter mais de uma entidade de execução (*thread*).

## 8. Metodologia de Pesquisa

### 8.1. Introdução

Este capítulo apresenta a metodologia utilizada na realização deste estudo de simulação e avaliação de desempenho.

### 8.2. Projeto de Avaliação de Desempenho

A metodologia para avaliação de desempenho segue as dez etapas básicas sugeridas por JAIN (1991). Cada etapa é refinada diversas vezes, num ciclo de *software* conhecido como '*espiral*'. As etapas para a avaliação de desempenho do CRUX, objeto deste estudo, são apresentadas a seguir.

#### 8.2.1. Definição dos Objetivos e do Sistema

Objetivos Gerais: Avaliar o desempenho do ambiente CRUX em relação à execução de atividades de tempo-real, principalmente considerando o algoritmo de escalonamento utilizado.

Objetivos Específicos: Mensurar quantitativamente as seguintes métricas de desempenho:

- Tempo médio de resposta das tarefas *soft-vbr*;
- Variação da taxa de execução de tarefas *soft-cbr*;
- Tempo médio de conexão de todas as tarefas.

e ainda determinar o efeito cada fator de interesse (apresentados mais abaixo), e também suas interações, para cada uma das métricas de desempenho avaliadas.

Sistema: Entende-se por sistema o ambiente CRUX descrito anteriormente, composto dos nós de trabalho, o nó de controle, o barramento de serviço e a rede de

interconexão *crossbar*. São entidades desse sistema as tarefas que devem ser executadas e as mensagens e requisições transmitidas.

### 8.2.2. Relação dos Serviços e Saídas

Cada elemento do Sistema fornece certo conjunto de serviços e exibe certo conjunto de saídas desejáveis ou não, sendo produto das entradas e do serviço realizado. O quadro 8.1 apresenta rapidamente os principais serviços providos por cada um dos elementos do sistema.

Quadro 8.1 – Breve descrição dos serviços oferecidos pelo sistema

Elemento do Sistema	Descrição dos Serviços
Nós de Trabalho (NTs)	- Recebe uma tarefa que deve ser escalonada segundo critérios de prioridade do modelo. A tarefa é executada, e após isso pode ser finalizada, ser criado um pedido de conexão/desconexão ou o envio de uma mensagem pelo <i>crossbar</i> . Essa mensagem pode ser comunicação entre processos ou transferência de arquivo.
Nó de Controle (NC)	- Recebe um pedido de conexão/desconexão de canais do <i>crossbar</i> , verifica se a mensagem pode ser transmitida sem perder a <i>deadline</i> , aloca os canais necessários (dinamicamente) e sinaliza a transmissão (ou não) da mensagem
Barramento de Serviço (BS)	- Fica constantemente fazendo um <i>pooling</i> nos NTs verificando se há algum pedido pendente, e os envia ao NC
Rede <i>Crossbar</i> (CB)	- Simplesmente recebe uma mensagem para transmitir, gasta um tempo transmitindo, e entrega uma atividade para o NT destino da transmissão

### 8.2.3. Seleção das Métricas

As métricas são utilizadas para avaliar e comparar a performance do sistema. As métricas de desempenho neste estudo são:

- Tempo médio de resposta das tarefas *soft-vbr* ( $\mu$ s);
- Variação da taxa de execução das tarefas *soft-cbr* ( $\mu$ s);
- Tempo médio de conexão de todas as tarefas ( $\mu$ s).

### 8.2.4. Relação dos Parâmetros

Parâmetros são as variáveis que influenciam o desempenho. Foram identificados os seguintes parâmetros:

- Parâmetros do Sistema:
  - Quantidade de nós processadores (NTs);
  - Quantidade de canais físicos no *crossbar*;
  - Velocidade de acesso de um canal físico do *crossbar*;
  - Velocidade de acesso do barramento de serviço;
  - Algoritmo de escalonamento usado pelo NC;
  - Algoritmo de escalonamento usado pelos NTs;
- Parâmetros de Carga:
  - Taxa de chegada de atividades;
  - Tamanho das mensagens;
  - Tamanho dos pedidos de conexão/desconexão;
  - Tempo de processamento no NC;
  - Tempo de processamento nos NTs;
  - *Quantum* de tempo para preempção;
  - Quantidade de comunicações por atividade (uso da rede de interconexão);

“A velocidade de acesso refere-se à frequência na qual os bits podem ser enviados e recebidos na interface de rede. Esta frequência é normalmente definida pela tecnologia usada pela rede, ou pela subscrição de um cliente feito a um serviço de rede. Por exemplo, a velocidade de acesso de uma LAN Ethernet convencional é de 10 Mbps” (WILLRICH 2000).

#### 8.2.5. **Seleção dos Fatores**

Dos parâmetros relacionados anteriormente, e que influenciam no desempenho do sistema, alguns foram considerados importantes, sendo então chamados de fatores. Os fatores terão seus valores alterados durante as simulações, visando estabelecer sua influência nas métricas selecionadas. Todos os demais parâmetros são mantidos fixos. Os fatores do sistema são:

- **Fator A**: Quantidade de nós processadores (NTs): (8, 16, 24, 32);
- **Fator B**: Quantidade de canais físicos no *Crossbar*: (1, 2, 4);
- **Fator C**: Algoritmo de escalonamento usado pelos NTs: (EDF, SMF, RM, DM, BS, PS, DS, SS, DSS, CBS);
- **Fator D**: Fator de Utilização da CPU: (80%, 90%, 100%, 110%);
- **Fator E**: Velocidade das redes *Crossbar* e BS: (1, 10, 100) Mb/s

#### 8.2.6. **Técnica de Avaliação**

Basicamente, três técnicas de avaliação podem ser utilizadas:

- **Modelagem Analítica**: Representação matemática do sistema. Possui limites quanto à complexidade do sistema representado devido aos cálculos envolvidos.
- **Medição Efetiva**: Coletar os dados diretamente do sistema real. Exige que o sistema exista e que os dados possam ser coletados.

- Simulação: Modelo abstrato representa as características relevantes do sistema. Exige um computador com grande capacidade de processamento para realizar as simulações.

Considerando que ambiente CRUX, sistema de interesse, é complexo, e que não é possível realizar medições sobre esse sistema, e considerando ainda a facilidade que a técnica de simulação de sistemas apresenta na modelagem de sistemas complexos, permitindo a representação de sistemas ainda inexistentes, o refinamento progressivo do modelo, e teste de hipóteses sobre o sistema, entre outras vantagens, fez com que recaísse sobre a *Simulação* a opção de representação e modelagem do CRUX.

#### 8.2.7. Seleção da Carga de Trabalho

A carga de trabalho tenta representar o comportamento de uma aplicação multimídia, sendo composta por tarefas aperiódicas que representam a acesso a *home pages* na internet, conforme o *benchmark* SPECweb96; tarefas *soft* que representam a transmissão de uma mídia contínua, como vídeo ou áudio; juntamente com um conjunto de tarefas *hard* que devem ser garantidas *a priori* e podem representar qualquer atividade ou sensoramento crítico.

As atividades *web* serão compostas pela requisição a 36 arquivos (4 classes x 9 tamanhos) de uma página HTML, e as atividades multimídia representarão a “transmissão de áudio qualidade telefone, onde amostras de 8 bits são produzidas a intervalos fixos de 125  $\mu$ s (mais ou menos uma pequena variação)” (WILLRICH 2000).

Conforme o SPECweb96, a probabilidade de acessos dentro de uma classe segue uma distribuição *Poisson* com média no ponto central da classe. O gráfico 8.1 apresenta essa distribuição para a classe 0 ( 0 a 1 KB) e o gráfico 8.2 para a classe 1 (1 a 10 KB).

Figura 8.1 – Distribuição *Poisson* de acesso aos arquivos da classe 0

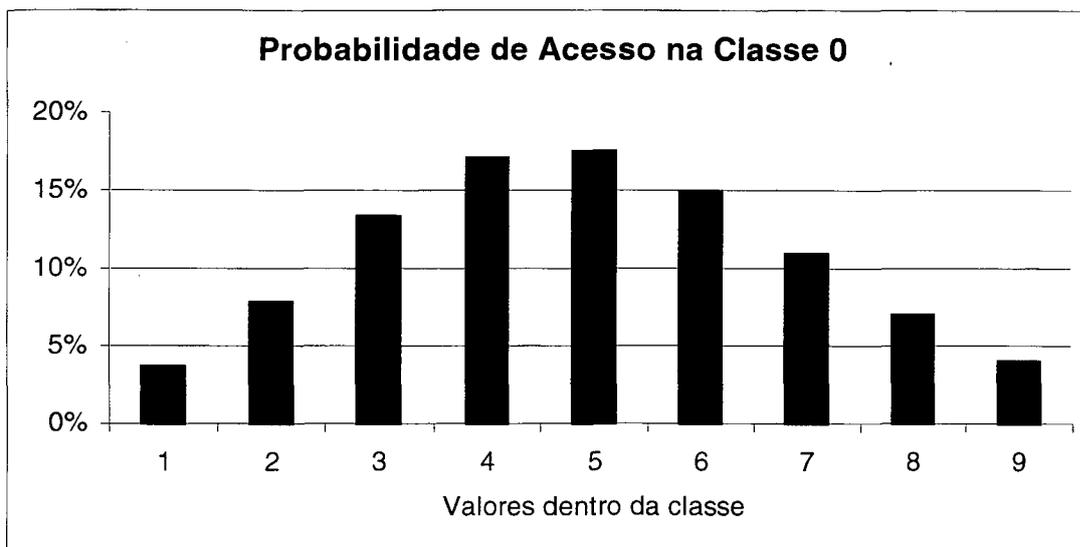
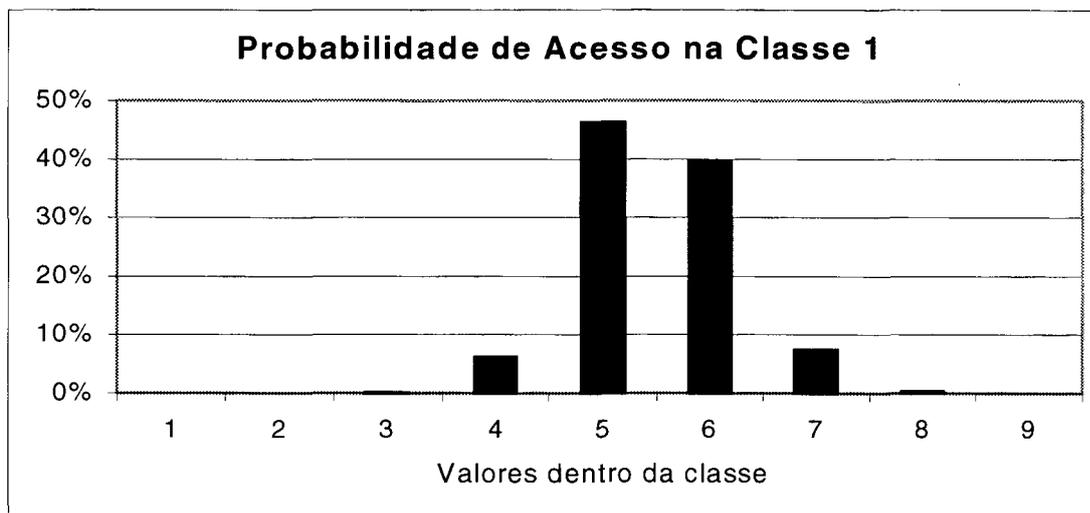


Figura 8.2 – Distribuição *Poisson* de acessos aos arquivos da classe 1



#### 8.2.8. Projeto de Experimentos

A quantidade de experimentos necessários para avaliar o efeito de todos os parâmetros pode ser muito grande, dependendo da quantidade de fatores e níveis. Para os cinco parâmetros escolhidos, A, B, C, D e E considerando que os parâmetros têm, respectivamente, 4, 3, 10, 4 e 3 níveis, o total de experimentos necessários, sem replicações, seria:  $4 \times 3 \times 10 \times 4 \times 3 = 1.440$ . Supondo agora 3

replicações para permitir avaliar os erros dos experimentos, temos  $1.440 \times 3 = 4.220$  experimentos. Experimentos anteriores num Pentium 166MHz 32MB RAM mostram que o tempo total de simulação para esses experimentos seria de pouco mais de 3 meses ininterruptos.

Assim, para diminuir a quantidade de experimentos necessários para avaliar inicialmente quais são os fatores mais representativos das métricas de desempenho escolhidas, sem perder tempo com tantos experimentos, os resultados serão baseados em dois conjuntos de experimentos:

- Inicialmente, será feito um Projeto Fatorial com Um Fator, fazendo variar apenas o algoritmo de escalonamento utilizado, para verificar as diferenças de desempenho dos algoritmos quando executando em um sistema onde todos os demais parâmetros são fixados em seu valor médio;
- No segundo conjunto de experimentos todos os 5 fatores serão avaliados através de seus dois níveis extremos com um projeto fatorial  $2^k$ . Assim, na segunda bateria teremos  $2^5 = 32$  experimentos. Os dois níveis extremos do fator C (algoritmo de escalonamento), serão os dois níveis que apresentarem o pior e o melhor desempenho na primeira bateria experimental;

#### 8.2.9. Análise e Interpretação dos Resultados

De acordo com JAIN (1991), cinco suposições são inicialmente feitas, e devem ser verificadas (confirmadas) após realizados os experimentos. São elas:

- Os erros do modelo são estatisticamente independentes: pode ser verificado através de um gráfico de resíduos. Se os resíduos forem uma ou mais ordens de magnitude menores que a métrica de desempenho, então os erros podem ser considerados independentes, o caso em contrário indica covariação entre os fatores;
- Os erros estão normalmente distribuídos: pode ser verificado através de um gráfico *quantile-quantile*;

- Os erros têm um desvio padrão  $\delta$  constante: pode ser verificado através de um gráfico com o desvio-padrão de cada experimento;
- Os efeitos dos fatores são aditivos: O projetista deve avaliar se os fatores são realmente aditivos. Caso não sejam, uma transformação de variáveis é necessária.

A interpretação dos resultados de uma análise é fundamental. Deve ser entendido que a análise produz apenas resultados, e não conclusões. Além disso, é importante também compreender que os resultados de medidas e simulações são valores aleatórios e que podem ser diferentes a cada vez que o experimento é repetido. Assim, é necessário realizar uma análise da variância desses resultados;

#### 8.2.10. Apresentação dos Resultados

Os dados serão apresentados em forma de tabelas e gráficos, resumizando de forma clara os resultados obtidos.

## 9. Desenvolvimento do Modelo de Simulação

### 9.1. Introdução

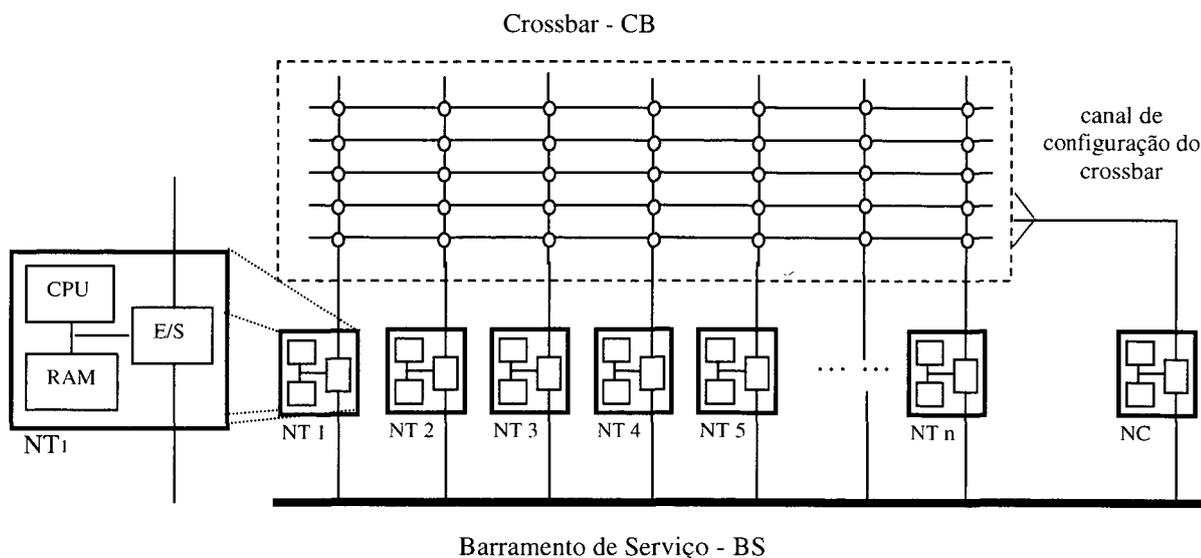
Com o objetivo de avaliar o comportamento de algoritmos de escalonamento de tempo real para o Ambiente CRUX, foi desenvolvido um modelo computacional de simulação e que representa o comportamento de tal Ambiente no que se refere à execução de processos em tempo real, considerando ainda fatores de interconexão e outros.

Como ferramenta computacional para modelagem do sistema, optou-se pelo software Arena, da *Systems Modeling Corporation*, em sua versão 3.0c, pelo seu potencial e quantidade de recursos, boa interface com o usuário, analisador de dados de entrada e de resultados, integração com a linguagem de programação VBA (*Visual Basic for Applications*), além de sua disponibilidade para uso.

O modelo desenvolvido no *software* Arena representa o ambiente CRUX em suas características que dizem respeito ao processamento e escalonamento de *threads* nos diversos Nós de Trabalhos (NT), aos pedidos de conexão e desconexão de canais do *Crossbar* ao Nó de Controle (NC) através do Barramento de Serviço (BS), e ao envio de mensagens entre NTs cliente-servidor através do *Crossbar*.

O modelo criado classifica-se, de acordo com os critérios apresentados no Capítulo 5, como um modelo simbólico, uma vez que é representado por entes simbólicos, estocástico, pois representa fenômenos aleatórios, dinâmico, pois seu estado varia com o tempo, discreto, pois a mudança de estado ocorre em intervalos discretos de tempo, aberto, pois possui entradas externas ao modelo e é instável, pois não tende a um estado final.

Figura 9.1 – Elementos do modelo do multicomputador CRUX



Os elementos básicos do modelo computacional e da arquitetura do CRUX se confundem. A figura 7.1 mostra tal modelo, sendo rerepresentado na figura 9.1 por conveniência. Quatro elementos principais estão representados:

- Nós de Trabalho (NT): Executam tarefas e recebem/enviam mensagens/arquivos a outros NTs através do *Crossbar*; e enviam pedidos de conexão/desconexão ao NC através do Barramento de Serviço.
- Nó de Controle (NC): Recebe pedidos de conexão/desconexão entre dois NTs distintos e aloca/desaloca os canais necessários a troca de mensagens entre os NTs.
- Barramento de Serviço (BS): Pesquisa os NTs seqüencialmente (*pooling*) verificando se há pedidos pendentes e os remetendo ao NC.
- Crossbar: Simplesmente permite a transmissão de mensagens entre os NTs que têm canais alocados para comunicação. As mensagens que correspondem a pedidos de URL ou transmissão de arquivos (*soft VBR*) sofrem um atraso

Um dos objetivos almejados na concepção do modelo foi o de reusabilidade. O modelo apresenta diversos parâmetros que alteram as características e o comportamento do sistema, sem a necessidade de alterar a estrutura do modelo, o que facilita a realização de futuros estudos. A integração da ferramenta Arena com o VBA foi explorada e o resultado foi um modelo controlado por uma interface personalizada e de fácil uso. Os algoritmos de escalonamentos também foram escritos parte em VBA, o que aumentou muito seu poder de especificação algorítmica e também a possibilidade de ser expandido e/ou refinado.

## **9.2. Descrição do Comportamento do Modelo**

Em cada experimento aleatório, o modelo simula a execução das atividades especificadas pelo tempo também determinado pelo projetista. Durante a simulação, todas as métricas de desempenho (apresentadas na seção 9.6) são coletadas, sendo que a unidade básica do tempo simulado no modelo é o microssegundo ( $\mu s$ ).

Para uma caracterização mais real da carga de trabalho ao qual o modelo do CRUX será submetido, optou-se pela representação de aplicações multimídia. O modelo permite que outros tipos de aplicações, como robótica ou controle industrial, por exemplo, sejam também simulados, bastando alterar os parâmetros do modelo, e apenas eventualmente sua estrutura.

### **9.2.1. Comportamento Geral**

O modelo representa a execução de atividades de tempo real (TR) executando no multicomputador CRUX. Dos  $n$  Nós de Trabalho representados, um grupo de aproximadamente 20% (25% a 15,625%) dos processadores compõe um *pool* de NTs servidores, enquanto os restantes são considerados NTs clientes. A equação (43) expressa a quantidade de servidores em função do total de NTs.

$$eNumServidores = \left\lfloor \frac{vMaxNodos}{10} \right\rfloor + 2 \quad (43)$$

onde:  $eNumServidores$ : quantidade de NTs servidores;

$vMaxNodos$ : quantidade total de Nós de Trabalho.

Cada atividade tem uma *deadline* relativa (atributo *aDeadline* no modelo) e é composta por tarefas e compartilham esse valor. As tarefas são as verdadeiras entidades escalonáveis. As atividades podem ser *soft-vbr*, *soft-cbr* ou *hard* (atributo *aTipo*), periódicas ou aperiódicas. O modelo escolhe aleatoriamente um NT como cliente para iniciar a execução das atividades *soft* (não há balanceamento dinâmico de carga). Já as atividades *hard* são distribuídas igualmente entre todos os NTs clientes de forma a manter constante o Fator de Utilização (equação 9) especificado. O comportamento das atividades varia conforme o seu tipo:

As atividades *hard* (*real-time*) são periódicas e devem ser garantidas *a priori*, pois representam algum processamento crítico para a aplicação em questão. As atividades *hard* são compostas por uma quantidade infinita de tarefas que são ativadas periodicamente nos NTs. Deve-se garantir que uma tarefa *hard* irá executar completamente antes da próxima ativação (período igual à *deadline*), sendo que tarefas *hard* não trocam mensagens com tarefas em outros NTs, ou seja, executam apenas localmente.

As aplicações *soft* representam dois comportamentos multimídia distintos:

- Atividades com tráfego variável (VBR) e transmissão assíncrona, representadas pelo acesso a páginas *web* HTML (*home pages*). Cada atividade *web* gera o pedido a 36 arquivos, sendo que o tamanho dos arquivos é dado pela frequência de acesso conforme o *benchmark* SPECweb96, apresentado no Capítulo 6. Qualquer NT do *pool* de servidores é considerado um servidor de *web*.
- Atividades com tráfego constante (CBR) e transmissão síncrona, representadas pela transmissão de áudio digital sem compactação. Cada atividade gera *n* tarefas por segundo, que devem executar a uma taxa constante. Essas tarefas também geram tráfego constante no *crossbar*. Apesar de serem aperiódicas, o período no qual é desejável que as tarefas executem é especificado.

“No caso de uma transferência assíncrona, o usuário final da informação terá que aguardar a transferência completa do arquivo da fonte para o destino, além de exigir uma capacidade de armazenamento no destino suficiente para armazenar todo o arquivo. [...] No caso

de uma transferência síncrona, o usuário final não aguardará a carga completa do arquivo, a informação deve ser transferida da fonte em uma taxa muito próxima a de apresentação e, após um pequeno atraso de transmissão, deve ser apresentada no destino.” (WILLRICH 2000)

As atividades *soft* são aperiódicas e possuem grande variação no tempo de execução, assim, o atendimento das *deadlines* não é garantido *a priori*. Depois de executar inicialmente num NT cliente, a tarefa *soft* pode solicitar um serviço num dos NTs do *pool* de servidores (NT servidor) enviando uma mensagem através do *Crossbar*. Para enviar uma mensagem, a tarefa deve requisitar ao NC uma conexão com o outro NT. Para fazer tal requisição, as tarefas *soft* usam o Barramento de Serviço. Uma vez feita a conexão, a mensagem é transferida ao NT destino.

No caso de atividades *soft VBR*, depois de chegar ao NT destino (servidor), a conexão é desfeita e uma nova tarefa (pertencente à mesma atividade) é colocada na fila *ready* do processador para iniciar sua execução. Depois de terminar sua execução e deixar o processador, outra mensagem (resposta) é gerada e enviada de volta ao NT cliente seguindo o mesmo processo. Quando essa mensagem (resposta) chega ao NT cliente, a conexão é desfeita e a atividade continua sua execução, podendo executar e terminar, ou executar e fazer um novo pedido ao mesmo ou a outro NT servidor.

No caso de atividades *soft CBR*, a atividade representa a transmissão do áudio, e cada tarefa representa uma pequena amostra de voz (a cada 62,5  $\mu$ s). A conexão é estabelecida para o envio da primeira mensagem (primeira amostra de voz). Após isso, muitas outras mensagens são transmitidas utilizando a conexão já estabelecida, e apenas após o término da transmissão do áudio (após a última amostra) é que a conexão é desfeita.

As atividades de transmissão de áudio são iniciadas num NT cliente, e nesse caso, há apenas uma mensagem do cliente para o servidor, que solicita o áudio e estabelece a conexão. Durante a conexão, o NT servidor envia mensagens do servidor ao cliente num intervalo de 62,5  $\mu$ s, e que devem executar a uma taxa constante no cliente. Após a transmissão de todo o áudio, o cliente envia um pedido de desconexão ao NC. Deste modo, o cliente não precisa esperar por toda a transferência para iniciar a

execução. Basta um pequeno retardo inicial (a transmissão da primeira amostra) e a execução da mídia já pode iniciar.

### 9.2.2. Comportamento de Atividades *Hard Real-Time*

Uma atividade *hard* é uma atividade periódica, com tempo de execução igual a 200  $\mu$ s, e período que depende do fator de utilização estabelecido, conforme a seguinte equação:

$$ePeriodoHard = \frac{vTempoExecucaoHard}{vFatorUHard} \quad (44)$$

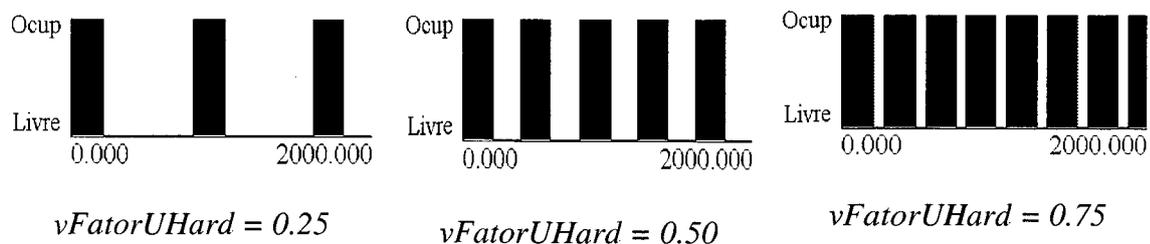
onde: *ePeriodoHard*: período de ativação das atividades *hard real-time*;

*vTempoExecucaoHard*: tempo de execução da atividade *hard*;

*vFatorHard*: fator de utilização do processador por atividades *hard*.

A figura 9.2 apresenta a ocupação do processador de um NT durante 2 ms (2.000  $\mu$ s), enquanto executa uma atividade *hard real-time*, para diferentes valores de Utilização da CPU.

Figura 9.2 – Ocupação do NT por uma atividade *hard real-time*



A *deadline* de uma tarefa *hard* é igual ao seu período de ativação. Se alguma tarefa *hard* perder sua *deadline*, ela é imediatamente retirada do sistema, com exceção se estiver sendo utilizado o algoritmo de escalonamento RED, que as mantêm numa fila

separada. Entretanto, a escalonabilidade dessas tarefas pode ser garantida *a priori* sempre que  $vFatorUHard \leq 1$ . Todas as atividades *hard* criadas possuem o mesmo período de ativação e tempo de execução, tal que:

$$\sum_{i=1, i \text{ é } hard}^n \frac{C_i}{T_i} = vFatorUHard \quad (45)$$

onde:  $n$ : quantidade de atividades;

$C_i$ : tempo de computação da atividade  $i$ ;

$T_i$ : período de ativação da atividade  $i$ ;

$vFatorUHard$ : fator de utilização do processador por tarefas *hard*.

A atividade *hard real-time* executa no NT e termina, sem gerar tráfego na rede. Sua execução é gerenciada pelo escalonador de processos escolhido. O modelo incorpora os seguintes algoritmos de escalonamento de tempo-real: SMF (*Smallest Message First*), RM (*Rate Monotonic*), EDF (*Earliest Deadline First*), DM (*Deadline Monotonic*), BS (*Background Scheduling*), PS (*Pooling Server*), DS (*Deferrable Server*), SS (*Sporadic Server*), DSS (*Dynamic Sporadic Server*) e CBS (*Constant Bandwidth Reservation*).

Alguns dos escalonadores de tempo-real listados acima, os do tipo “*Server*”, criam um outro tipo de tarefa periódica *hard real-time*. Essas tarefas são criadas especificamente para atender as tarefas aperiódicas, conforme apresentado no capítulo 4, que trata dos algoritmos de escalonamento de tempo-real.

Quando há tarefas aperiódicas pendentes (VBR ou CBR), o escalonador, em algum instante, cria uma tarefa “servidora”, que é tratada pelo modelo quase exatamente como outra tarefa *hard real-time* qualquer, sendo que sua prioridade depende do tipo do algoritmo:

- Servidores de prioridade fixa: Esses servidores têm sua prioridade primária conforme o algoritmo RM. Assim, a prioridade é constante e igual ao seu período de ativação. O período de ativação das tarefas servidoras foi

mantido constante e igual ao tempo de execução das demais tarefas *hard* ( $vTempoExecHard$ ), que é de  $200\mu s$ .

- Servidores de prioridade dinâmica: Esses algoritmos baseiam sua prioridade primária no algoritmo EDF. Assim, sua prioridade muda ao longo do tempo, dependendo de sua *deadline* absoluta. A *deadline* relativa de uma tarefa servidora” é igual ao seu período de ativação.

Apesar de que tanto servidores de prioridade fixa quanto de prioridade dinâmica se basearem no período de ativação (e a *deadline* ser igual ao período de ativação) não significa suas prioridades ou comportamentos sejam os mesmos. O seguinte exemplo ilustra essa diferença: Uma tarefa *hard*  $\tau_1$ , com período de ativação  $T_1=300$  e *deadline* relativa  $D_1 = 300$  é iniciada no instante  $t = 0$  e escalonada. No instante  $t = 200$ , outra tarefa *hard*  $\tau_2$  é iniciada com período de ativação  $T_2=200$  e *deadline* relativa  $D_2 = 200$ . No instante de ativação de  $\tau_2$ , algoritmos baseados em RM preemptariam  $\tau_1$  e escalonariam  $\tau_2$ , pois a última tem período menor. Os algoritmos baseados em EDF, entretanto, continuariam a executar  $\tau_1$ , pois sua *deadline* absoluta  $d_1=0+D_1=300$ , é menor que a *deadline* absoluta de  $\tau_2$ ,  $d_2=200+D_2=400$ .

A diferença entre tarefas “servidoras” *hard*, e as demais tarefas *hard* do sistema, está no fato de que as tarefas servidoras recebem vários atributos da tarefa aperiódica a que estão servindo, como o tempo de execução e parâmetros para determinação dos indicadores de desempenho das tarefas aperiódicas. Enquanto a tarefa “servidora” executa, a tarefa aperiódica sendo servida é mantida na fila de tarefas aperiódicas. Quando a execução termina, a tarefa “servidora” é destruída, e tarefa aperiódica (CBR

ou VBR) é retirada da fila e prossegue pelo sistema, como se tivesse executado normalmente.

Os escalonadores do tipo “*Server*” possuem como parâmetros: o Período de ativação da tarefa “servidora” e a Capacidade máxima da “servidora”. No modelo criado, o Período de ativação da tarefa “servidora” é igual ao tempo de execução de uma tarefa *hard* ( $vTempoExecHard$ ), que é de  $200\mu s$ . A Capacidade máxima é estabelecida conforme a taxa de utilização das tarefas *soft*, de acordo com a equação a seguir:

$$CapacidadeMax = vTempoExecHard \cdot vFatorUSoft \quad (46)$$

### 9.2.3. Comportamento de Atividades *Soft Real-Time VBR*

Uma atividade *soft-vbr* é representada pelo acesso a páginas HTML através de um servidor *web*. A atividade inicia com uma tarefa executando num NT cliente. O tempo de computação no cliente é desprezível, sendo composto basicamente pelo tempo de montar o pedido de uma página HTML a um NT servidor. A quantidade de arquivos solicitados por uma atividade *soft\_vbr* é informada pelo parâmetro  $vMaxComunicVBR$ .

Para enviar uma mensagem, a tarefa *soft* deve requisitar uma conexão ao NC através do Barramento de Serviço (BS). O NC faz constantemente um *pooling* aos NTs para verificar se há requisições pendentes. O tempo e transmissão de uma requisição pelo BS, em microssegundos ( $\mu s$ ), desconsiderando inicialmente a espera pelo *pooling*, é de:

$$TempoBS = \frac{aTamRequisicao}{\frac{1.10^6}{1.10^6} \cdot vVelocBarramento} = \frac{aTamRequisicao}{vVelocBarramento} \quad (47)$$

onde:  $aTamRequisicao$ : tamanho da requisição de (des)conexão em bytes;

$vVelocBarramento$ : velocidade de acesso do BS em  $\frac{MB}{s}$ ;

E o tempo médio de envio de uma requisição, já considerando o atraso devido ao *pooling*, é dado por:

$$TMTBS = TempoBS + vPeriodoPooling \cdot \frac{vMaxNodos}{2} \quad (48)$$

onde: *TMTBS*: Tempo Médio de Transmissão pelo BS em *ms*;

*vPeríodoPooling*: intervalo de tempo em  $\mu s$  entre duas verificações por requisições pendentes num NT;

*vMaxNodos*: quantidade de NTs.

Assim, o tempo gasto entre montar um pedido de conexão/desconexão num NT e esse pedido chegar à fila de requisições no NC, o que envolve o *pooling* e a transmissão propriamente dita, tem seu limite superior dado por:

$$PTTBS \leq \frac{aTamRequisicao}{vVelocBarramento} + vPeriodoPooling \cdot vMaxNodos \quad (49)$$

onde *PTTBS* é o pior tempo de transmissão (em  $\mu s$ ) de um pedido de conexão/desconexão envolvendo o BS.

Patterson (1998) lembra que no sistema de E/S, as unidades são medidas na base 10, ao contrário das demais unidades utilizadas pelo computador, como o tamanho das mensagens, por exemplo. Deste modo, uma mensagem de 1MB ( $2^{10}$  bytes) não leva 1 s para ser transmitida numa rede de 1MB/s ( $10^3$  bytes/s), mas na realidade 1,048576 s.

O algoritmo de escalonamento das requisições de conexão/desconexão é o mesmo dos NTs, com a ressalva que as requisições de desconexão sempre devem possuir maior prioridade que qualquer pedido de conexão, de modo que não ocorra impasse na alocação dos canais físicos.

O tempo de transmissão (em  $\mu s$ ) de uma mensagem através do *Crossbar*, utilizando um único canal físico, é dado por:

$$TempoCrossbar = \frac{aTamMensagem}{\frac{1.10^6}{1.10^6} \cdot vVelocCrossbar} = \frac{aTamMensagem}{vVelocCrossbar} \quad (50)$$

onde:  $vTamMensagem$ : tamanho da mensagem em bytes;

$vVelocCrossbar$ : velocidade de acesso do *crossbar* em  $MB/s$ ;

Após a mensagem ser recebida pelo NT servidor, o arquivo solicitado é enviado de volta ao NT cliente pelo *Crossbar*. O tamanho do arquivo solicitado segue as distribuições de probabilidade apresentadas no SPECweb96 e seu tempo de transmissão a equação (50). Quando o arquivo é recebido no NT origem, a solicitação para um novo arquivo pode ser feita ou a atividade pode ser encerrada.

Cada NT pode alocar até  $n$  canais físicos no *Crossbar*, que podem ser usados para enviar uma mensagem/arquivo, sendo  $n$  limitado pela quantidade de canais existentes e disponíveis. Cada canal adicionado duplica a capacidade do *Crossbar*. No modelo criado é usado apenas o número necessário de canais de modo que a mensagem atenda sua *deadline*. Uma vez a conexão sendo requisitada, o NC verifica a disponibilidade dos canais de comunicação tanto no NT origem quanto no NT destino e também verifica quantos canais (atributo *aNumCanais*) são necessários para enviar a mensagem de tal forma que a tarefa possa atender a *deadline*.

Se há canais de comunicação disponíveis suficientes, então a conexão é estabelecida e o tempo de comunicação é calculado segundo a equação (50). Se o número disponível de canais de comunicação no *Crossbar* não é suficiente para que a tarefa atenda à *deadline*, então o total de canais disponível é alocado e a conexão é estabelecida de qualquer maneira.

Uma mensagem pode ser enviada pelo *crossbar* e atender sua *deadline* se e somente se as seguintes inequações forem satisfeitas:

$$\text{tempo\_atual} + \frac{\text{tempoCrossbar}}{a\text{NumCanais}} \leq di \quad (51)$$

$$0 < a\_NumCanais \leq vMaxCanais \quad (52)$$

onde:  $di$ : *deadline* absoluta da tarefa a que pertence a mensagem;

$\text{tempo\_atual}$ : tempo atual da simulação ( $t_{now}$ ) em  $\mu\text{s}$ ;

$a\text{NumCanais}$ : quantidade de canais físicos alocados para a transmissão da mensagem;

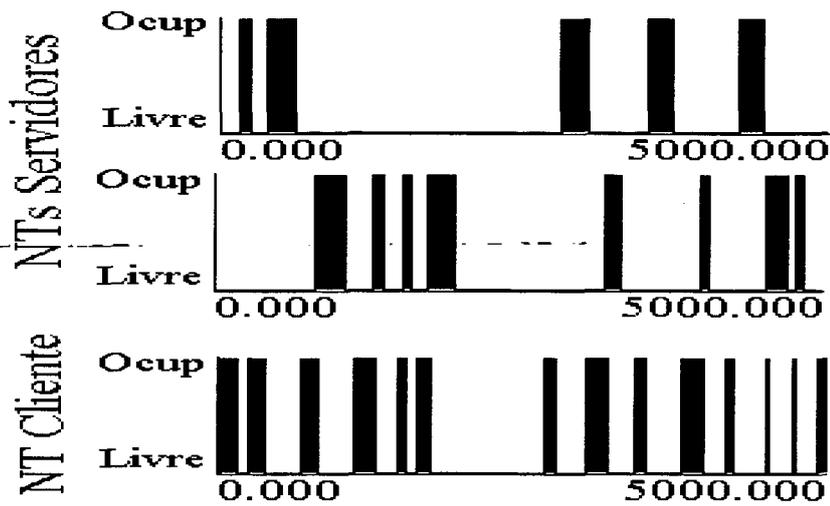
$vMaxCanais$ : total de canais físicos existentes no *crossbar*;

Se não há pelo menos um canal disponível, tanto no NT origem quanto no NT destino, então o pedido de conexão volta à fila *ready* do NC para escalonamento. A mensagem pode perder sua *deadline* durante a transmissão, mas não é descartada, apenas marcada. A *deadline* escolhida para uma atividade *soft-vbr* varia uniformemente de um valor mínimo a um valor máximo, conforme será apresentado posteriormente.

Se uma tarefa *soft* perde sua *deadline* enquanto executando ou enviando uma mensagem, ela não é descartada imediatamente, mas é marcada como '*perdida*' e continua normalmente. Apenas quando todas as tarefas pertencentes a uma única atividade terminam, então a atividade é classificada. Se pelo menos uma de suas tarefas perdeu a *deadline*, então toda a atividade perdeu a *deadline*. De outro modo, a atividade cumpriu a *deadline*.

A figura 9.3 apresenta a ocupação do processador de dois NTs servidores e de um NT cliente quando executando uma atividade *soft real-time VBR* por um período de 5 ms. Nesse trecho a atividade solicita 5 arquivos de um NT servidor e 8 de outro. Os intervalos em que ambos os servidores e clientes se encontram livres correspondem aos tempos de estabelecimento de conexão e transmissão de mensagens.

Figura 9.3 – Ocupação de NTs servidores e cliente por uma atividade *soft VBR*

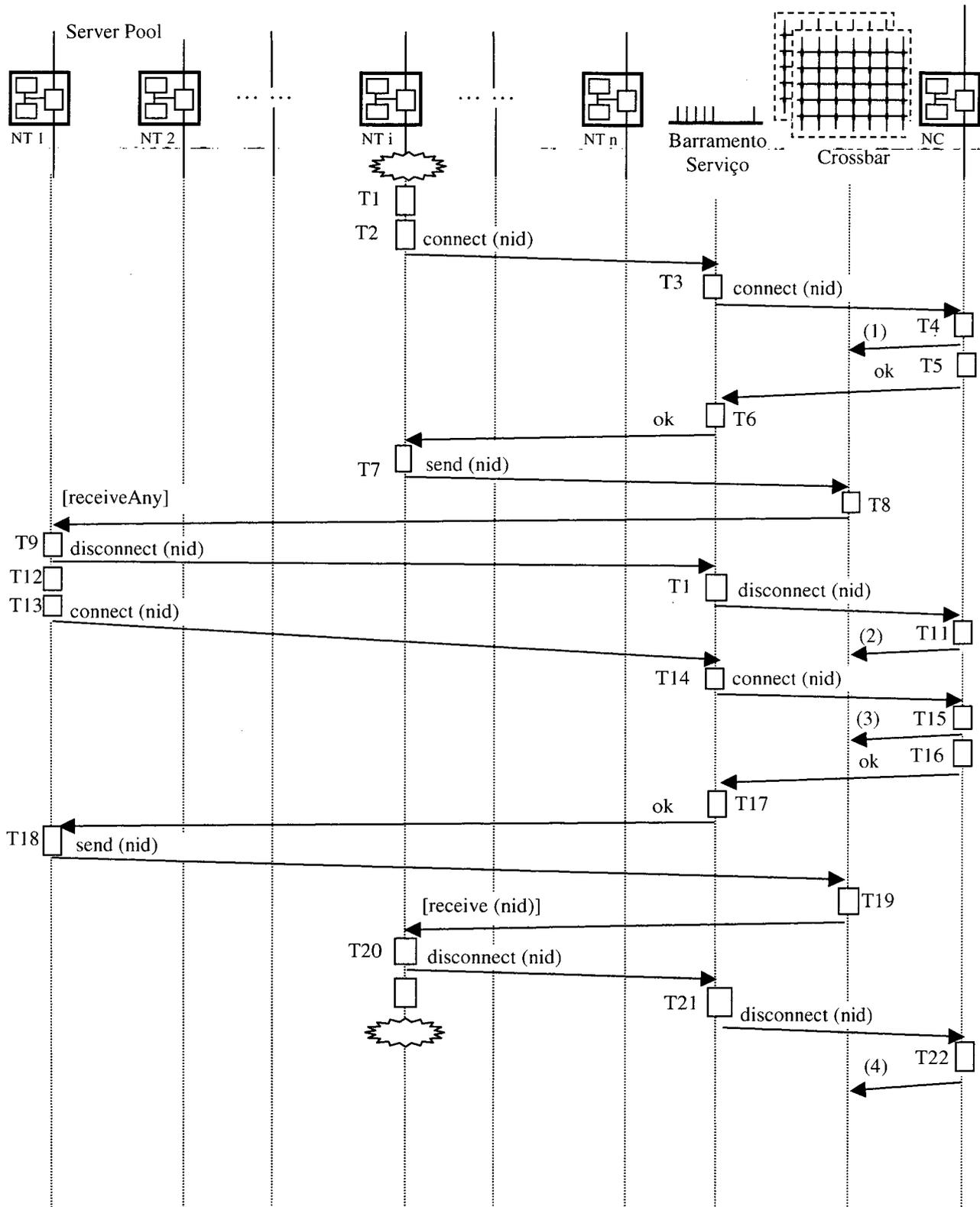


A figura 9.4 é um diagrama de seqüência (estilo UML) que representa com mais detalhes o comportamento descrito para as atividades *soft VBR*. As primitivas trocadas entre os componentes são apresentadas como rótulos das mensagens e os tempos de processamento, representados pelas caixas quando o componente tem o foco (rótulo Tx) são detalhados no quadro 9.1.

Quadro 9.1 – Tempos envolvidos no processamento de atividade *soft real-time VBR*

Tempo	Descrição	Tempo	Descrição
	A atividade é criada no NT i	<b>T13</b>	Montagem do pedido de conexão c/ NT i
<b>T1</b>	Tempo de execução da tarefa no NT i	<b>T14</b>	Tempo de transmissão (pooling) no Barramento de Serviço
<b>T2</b>	Montagem do pedido de conexão c/ NT servidor	<b>T15</b>	Tempo de execução no NC (conexão entre NT servidor e NT i)
<b>T3</b>	Tempo de transmissão (pooling) no BS	<b>T16</b>	Montagem da confirmação da conexão
<b>T4</b>	Tempo de execução no NC (conexão entre NT i e NT servidor)	<b>T17</b>	Tempo de transmissão do reconhecimento pelo Barramento
<b>T5</b>	Montagem da confirmação da conexão	<b>T18</b>	Montagem mensagem resposta para NT i
<b>T6</b>	Tempo de transmissão do reconhecimento pelo Barramento	<b>T19</b>	Tempo de Transmissão da mensagem pelo Crossbar
<b>T7</b>	Montagem da mensagem para NT servidor	<b>T20</b>	Montagem do pedido de desconexão
<b>T8</b>	Tempo de transmissão da mensagem pelo Crossbar	<b>T21</b>	Tempo de transmissão (pooling) no BS
<b>T9</b>	Montagem do pedido de desconexão	<b>T22</b>	Tempo de execução no NC
<b>T10</b>	Tempo de transmissão (pooling) no BS	<b>T23</b>	Tempo de execução da tarefa no NT i
<b>T11</b>	Tempo de execução no NC		A atividade é finalizada no NT i
<b>T12</b>	Tempo de execução da tarefa no NT i		

Figura 9.4 – Seqüência de eventos de uma atividade *soft real-time VBR*



#### 9.2.4. Comportamento de Atividades *Soft Real-Time CBR*

Uma atividade *soft-cbr* produz mensagens que devem ser transmitidas a uma taxa de bits constante, e o usuário final não deve esperar pela carga completa do arquivo (WILLRICH 2000). A atividade inicia num NT cliente, processa por um tempo e em seguida gera um pedido de arquivo de áudio a um dos NT servidores. Para que tal mensagem chegue ao NT servidor escolhido, é necessário solicitar ao NC uma conexão com esse NT. Esse processo é o mesmo explicado acima para as atividades *soft-vbr*.

Ao contrário de atividades *soft-vbr*, uma vez que o pedido de um arquivo de áudio multimídia (*soft-cbr*) tenha chegado ao NT servidor, a conexão não é desfeita e será utilizada para transmitir tal arquivo em várias amostras consecutivas. Assim, ao receber esse pedido, o NT servidor dispara periodicamente, a cada  $62,5 \mu\text{s}$  ( $v\text{PeriodoCBR}$ ), uma mensagem para o NT cliente através do *crossbar*, contendo uma amostra de áudio.

A quantidade de amostras enviadas depende da taxa de amostragem e da resolução do sinal, e neste estudo a quantidade de amostras geradas foi mantida constante e igual a 100 amostras por arquivo solicitado.

Outra diferença apresentada pelas atividades *soft-cbr*, é o fato de alocarem um único canal físico do *crossbar* para transmissão das amostras de áudio. Uma vez que a conexão será estabelecida por um tempo relativamente longo ( $\approx 62,5\mu\text{s} \times 100$  amostras), é importante que outros canais existentes no *crossbar* permaneçam livres para possíveis conexões nesse ínterim.

Assim que uma amostra chega ao NT cliente, ela se converte numa tarefa que deve executar no processador a uma taxa constante. Apesar das amostras serem transmitidas periodicamente pelo NT servidor, outras tarefas podem estar executando no NT clientes, o que significa que tais amostras podem executar numa taxa diferente. Alguns algoritmos de escalonamento específicos buscam diminuir essa diferença, embora a maioria ignore essa restrição. Algumas amostras podem ser descartadas por esse motivo.

Uma vez que o NT cliente tenha recebido a última amostra de áudio, é gerado um pedido de desconexão com o NT servidor, que é enviada ao NC através do Barramento de Serviço.

A métrica escolhida para indicar o cumprimento da taxa de execução das tarefas *soft-cbr*, representa a variação do instante de início de execução da amostra  $i$  em relação ao instante desejado para sua execução, o que é determinado pelo início da tarefa  $i-1$ , conforme descrito a seguir:

$$\text{variacaoCBR} = \sqrt{\frac{\sum_{i=2}^n |(a_i - a_{i-1}) - v\text{PeriodoCBR}|^2}{n}} \quad (53)$$

onde:  $v\text{PeriodoCBR}$ : período desejado entre execuções de tarefas CBR

$a_i$ : instante de início de execução da tarefa  $i$ ;

$n$ : quantidade de amostras de áudio coletadas;

Essa métrica, expressa em  $\mu s$ , corresponde à variância dos tempos de execução das amostras em torno da taxa de execução desejada, e deve ser minimizada. O coeficiente de variação da taxa de execução das atividades *soft-cbr*, é, então, dado por:

$$\text{coefVariacao} = \frac{v\text{PeriodoCBR}}{\text{variacaoCBR}} \quad (54)$$

A figura 9.5 apresenta a ocupação dos processadores de um NT servidor e um NT cliente durante os 2 ms iniciais da execução de uma atividade *soft real-time CBR* através do algoritmo EDF. A taxa de execução foi constante para todas as amostras.

Já a figura 9.6 apresenta praticamente a mesma situação da figura 9.5, ou seja, a ocupação dos processadores de um NT servidor e um NT cliente durante os 2 ms iniciais da execução de uma atividade *soft real-time CBR*. Entretanto, a utilização de outro escalonador (PS) permitiu a ocorrência de um fenômeno conhecido em transmissões multimídia onde algoritmos inadequados são utilizados.

Esse fenômeno indesejado, ilustrado nas figuras 9.6 e 9.7, possui períodos de tempo em que nenhuma amostra de áudio (ou vídeo, dependendo do caso) é apresentada, e, logo em seguida, períodos onde amostras são apresentadas numa taxa bastante alta, voltando depois ao normal.

Figura 9.5 – Ocupação de NTs servidor e cliente por uma atividade *soft CBR*

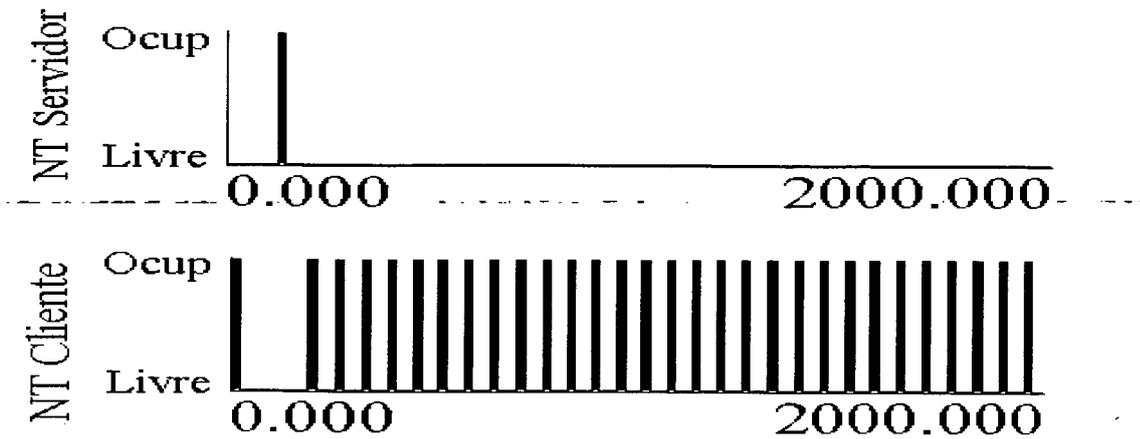


Figura 9.6 – Ocupação de NTs servidor e cliente por uma atividade *soft CBR*

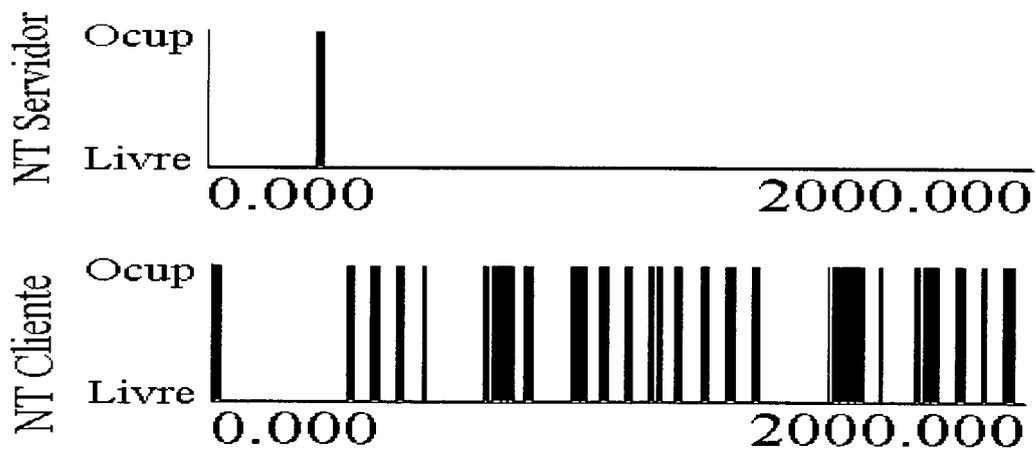
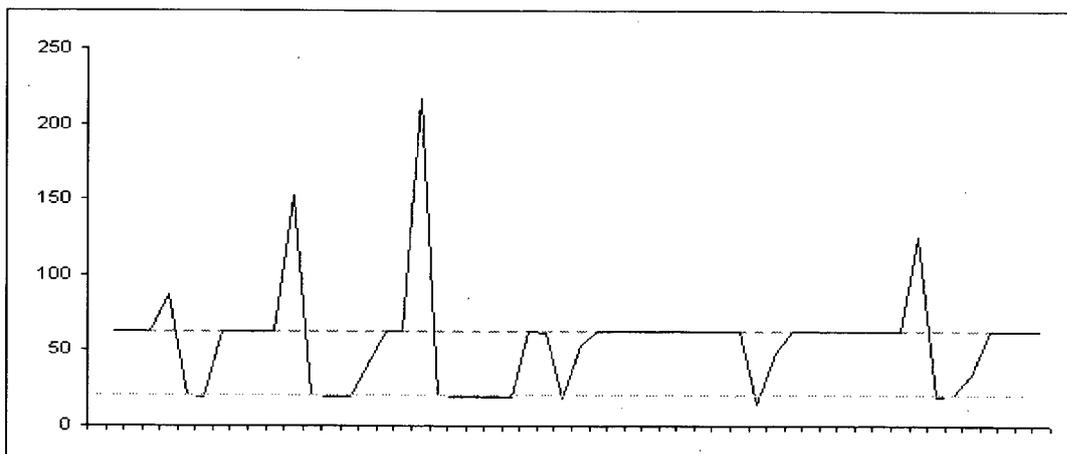


Figura 9.7 – Variação da taxa de execução de uma atividade *soft CBR*



A figura 9.7 apresenta, para cada amostra de áudio  $i$  ( $i=2,n$ ) de uma tarefa *soft* CBR, a diferença, em  $\mu s$ , do início de execução dessa amostra  $i$  com a amostra anterior  $i-1$  ( $a_i - a_{i-1}$ ). A linha tracejada em  $62,5 \mu s$  representa o período desejado entre execuções, cuja variação deve ser minimizada. A linha tracejada em  $20 \mu s$  representa o tempo de execução de uma única amostra, sendo o limite inferior para o período entre execuções de amostras CBR.

Quadro 9.2 – Tempos envolvidos no processamento de atividade *soft real-time CBR*

Tempo	Descrição	Tempo	Descrição
	A atividade é criada no NT $i$	<b>T13</b>	Montagem do pedido de conexão c/ NT $i$
<b>T1</b>	Tempo de execução da tarefa no NT $i$	<b>T14</b>	Tempo de transmissão (pooling) no Barramento de Serviço
<b>T2</b>	Montagem do pedido de conexão c/ NT servidor	<b>T15</b>	Tempo de execução no NC (conexão entre NT servidor e NT $i$ )
<b>T3</b>	Tempo de transmissão (pooling) no Barramento de Serviço	<b>T16</b>	Montagem da confirmação da conexão
<b>T4</b>	Tempo de execução no NC (conexão entre NT $i$ e NT servidor)	<b>T17</b>	Tempo de transmissão do reconhecimento pelo Barramento
<b>T5</b>	Montagem da confirmação da conexão	<b>T18</b>	Montagem da mensagem resposta para NT $i$
<b>T6</b>	Tempo de transmissão do reconhecimento pelo Barramento	<b>T19</b>	Tempo de Transmissão da mensagem pelo Crossbar
<b>T7</b>	Montagem da mensagem para NT servidor	<b>T20</b>	Montagem do pedido de desconexão
<b>T8</b>	Tempo de transmissão da mensagem pelo Crossbar	<b>T21</b>	Tempo de transmissão (pooling) no Barramento de Serviço



### 9.2.5. Comportamento dos Elementos do Sistema

A seguir são apresentados os principais elementos do modelo desenvolvido. Cada elemento é uma estação (*Station*) do Arena, o que facilita a transferência de entidades.

- Nós de Trabalho (NT): Executam tarefas e recebem/enviam mensagens/arquivos a outros NTs através do *Crossbar*; e enviam pedidos de conexão/desconexão ao NC através do Barramento de Serviço. Apenas uma parte do modelo do Nó de Trabalho é implementada através dos componentes do Arena.

A lógica mais complexa do modelo, que envolve principalmente os algoritmos de escalonamento, foi implementada em sua maioria usando VBA. Vários blocos VBA, utilizados para disparar procedimentos nessa linguagem, estão misturados aos demais blocos lógicos do Arena. O modelo do NT representado na ferramenta Arena é apresentado no ANEXO 1, sendo que o código-fonte VBA, que completa o modelo, é apresentado nos ANEXOS 5 a 8. Basicamente, o NT é responsável por:

- Disparar entidades de controle para simular a interrupção do *timer* (tipo *vTimer*) e acionar o escalonador de processos;
- Colocar tarefas pendentes em filas separadas (periódicas e aperiódicas), e as retirar para alocação da CPU quando são escalonadas
- Gerenciar a execução das “tarefas servidoras”, que são tarefas periódicas usadas por alguns escalonadores para servir tarefas aperiódicas;
- Coletar estatísticas sobre a execução das tarefas (as métricas desejadas).

O modelo do NT no Arena foi construído de forma que outros algoritmos de escalonamento possam ser definidos apenas através da programação em VBA, sem alterações na estrutura de blocos do modelo.

- Nó de Controle (NC): Recebe pedidos de conexão/desconexão entre dois NTs distintos e aloca/desaloca os canais necessários a troca de mensagens entre os NTs. O ANEXO 2 apresenta o modelo do Nó de Controle (juntamente com outros

elementos do sistema). O funcionamento do NC é basicamente o seguinte: Ao receber uma requisição, aloca a CPU, gasta o tempo de processamento ( $vQuantum$ ) e verifica se é um pedido de conexão ou desconexão. Se for desconexão, simplesmente libera a CPU do NC, desaloca os canais físicos do *crossbar* e libera a requisição.

Um pedido de conexão é mais complexo: Inicialmente o modelo calcula quantos canais seriam necessários alocar para que a mensagem cumpra a *deadline*. Se essa quantidade for maior que a quantidade existente, então a *deadline* não poderá ser atendida. Se essa quantidade for no máximo igual, então o modelo verifica quantos desses canais estão livres, tanto no NT origem quanto no NT destino. Se o menor destes valores for suficiente então a *deadline* será atendida, os canais do *crossbar* são alocados, a CPU do NC é liberada e a mensagem é transmitida. Se os canais disponíveis não forem suficientes, então os canais disponíveis são alocados e a mensagem é transmitida, mesmo com a tarefa perdendo a *deadline*.

Também no NC, a parte mais complexa da lógica de controle foi implementada em VBA.

- Barramento de Serviço: Pesquisa os NTs seqüencialmente (*pooling*) verificando se há pedidos pendentes e os remetendo ao NC. Também recebe respostas do NC e as envia ao NT que havia solicitado o pedido. O ANEXO 2 apresenta o modelo do barramento de serviço. A entidade verifica em cada NT se há algum pedido de conexão ou desconexão pendente. Se houver, um 'sinal' é enviado para liberar tal pedido. O tempo de transmissão do pedido pelo barramento é dado pela equação (46).
- Crossbar: Simplesmente permite a transmissão de mensagens entre os NTs que têm canais alocados para comunicação. O modelo da rede *crossbar* é ainda apresentado no ANEXO 2. As mensagens que correspondem a pedidos de URL ou transmissão de arquivos (*soft VBR*) sofrem um atraso da transmissão de toda a mensagem, conforme a equação (49). Já as amostras de áudio *stream (soft CBR)* sofrem apenas o retardo inicial, e após isso já podem iniciar sua apresentação no NT cliente.

### 9.3. Parâmetros do Modelo

O modelo desenvolvido possui 16 parâmetros que podem ser ajustados diretamente de uma interface VBA, sem a necessidade de alterar ou sequer visualizar a estrutura desse modelo. O Quadro 9.3 apresenta esses parâmetros, com o nome da variável ou expressão usada no Arena e uma lista dos valores mais comuns.

Quadro 9.3 – Parâmetros do modelo

Parâmetros do Sistema	Variável	Valores Comuns
Quantidade de Nós de Trabalho	<i>vMaxNodos</i>	8; 16; 24; 32
Quantidade de canais físicos do <i>Crossbar</i>	<i>vMaxCanais</i>	1; 2; 4
Velocidade de acesso de um canal físico do <i>Crossbar</i> (Mb/s)	<i>vVelocCrossbar</i>	1; 10; 100
Velocidade de acesso do Barramento de Serviço (Mb/s)	<i>vVelocBarramento</i>	1; 10; 100
Tamanho do pedido de conexão (bytes)	<i>vTamRequisicao</i>	46; 64; 128; 256; ...
Algoritmo de Escalonamento	<i>vAlgoritmo</i>	EDF; SS; CBS; ...
de Carga		
Fator de utilização das tarefas <i>hard</i>	<i>vFatorUHard</i>	0.2; 0.3; 0.4; 0.5; ...
Fator de utilização das tarefas <i>soft</i>	<i>vFatorUSoft</i>	0.1; 0.2; 0.3; 0.4; ...
Quantidade de arquivos por atividade VBR	<i>vMaxComunicVBR</i>	9; 15; 18; 27; 36; ...
Quantidade de amostras por atividade CBR	<i>vMaxComunicCBR</i>	50; 100; 200; ...
Tamanho de uma mensagem de URL (bytes)	<i>vTamMensagem</i>	(SPEC <sub>web</sub> 96)
Período de uma amostra de som ( $\mu$ s)	<i>vPeriodoCBR</i>	0.625; 0.125
Tempo de processamento no NC	<i>vTmpProcNC</i>	5; 10; 20;
do Projeto		
Tempo de Simulação ( $\mu$ s)	-	$10^4$ ; $10^5$ ; $10^6$ ; ...
Quantidade de Replicações	-	1; 2; 3; 4; 5; ...
Tempo de Aquecimento ( $\mu$ s)	-	$10^2$ ; $10^3$ ; $10^4$ ; ...

Esses parâmetros definem ainda, indiretamente, mais algumas expressões “secundárias” do modelo que são, então, calculadas automaticamente a partir dos parâmetros principais. As expressões “secundárias” não podem ser alteradas através da interface VBA, mas apenas diretamente sobre o modelo Arena.

Além dos parâmetros já mencionados, há ainda os parâmetros para os algoritmos de escalonamento de tempo-real. Algoritmos como o PS, DS, SS e DSS exigem a especificação da capacidade e do período da tarefa servidora, e os algoritmos CBR, ABS e RED exigem também outros parâmetros, conforme apresentado no capítulo 4.

#### **9.4. Descrição da Interface VBA**

O Arena possui integração com o VBA, que permite criar aplicações escritas em VBA que interagem com os objetos do ARENA e assim fornecer muito mais flexibilidade ao modelo. Utilizando esse recurso disponível, foi criada uma aplicação VBA completa que interage diretamente com o modelo desenvolvido. Essa aplicação tem os seguintes objetivos:

- Permitir controlar todos os fatores envolvidos na simulação sem se preocupar com o modelo;
- Prover maior poder algorítmico para implementar os algoritmos de escalonamento de tempo-real;
- Automatizar a coleta de resultados, gerando automaticamente uma planilha no Excel;
- Fornecer uma interface mais fácil e amigável com o usuário final;
- Facilitar a expansão do modelo.

A aplicação possui uma tela principal, composta por várias páginas, onde é possível controlar os parâmetros do modelo. A figura 9.9 apresenta a página '*Parâmetros Projeto*' da tela principal.

O VBA interage com o modelo Arena basicamente de duas maneiras:

- Blocos VBA: A ferramenta Arena possui o bloco VBA, que pode ser incorporado ao modelo. Cada bloco VBA tem um número identificador único (1, 2, 3, ...). Sempre que uma entidade do modelo alcança esse bloco,

um procedimento relacionado ao bloco é executado. O modelo desenvolvido para o CRUX possui 48 blocos VBA.

- **Eventos:** A ocorrência de eventos na ferramenta Arena (como a abertura de um modelo, o início, pausa ou término de uma replicação ou de toda a simulação, etc) pode estar associada a procedimentos em VBA, promovendo grande integração.

Assim, o uso da integração com o VBA não se limitou à construção de uma interface, mas principalmente foi meio para a implementação de grande parte da lógica do comportamento do modelo. Essa integração teve como vantagens principais: a diminuição significativa do tamanho do modelo Arena (quantidade de blocos e declarações de recursos, filas e variáveis), e a facilidade de alteração do comportamento do modelo, uma vez que deve ser mais fácil alterar código-fonte em VB do que centenas de blocos na ferramenta Arena.

Figura 9.9 – Tela da interface VBA criada para o modelo



## 9.6. Verificação e Validação do Modelo

A Verificação de um modelo refere-se a análise que permite estabelecer se a implementação computacional do modelo mantém as características definidas na especificação conceitual do sistema. Um modelo é dito ‘verificado’ se seu comportamento é aquele descrito conceitualmente.

Para a verificação do modelo foram utilizadas, em várias etapas do desenvolvimento do modelo, as seguintes técnicas:

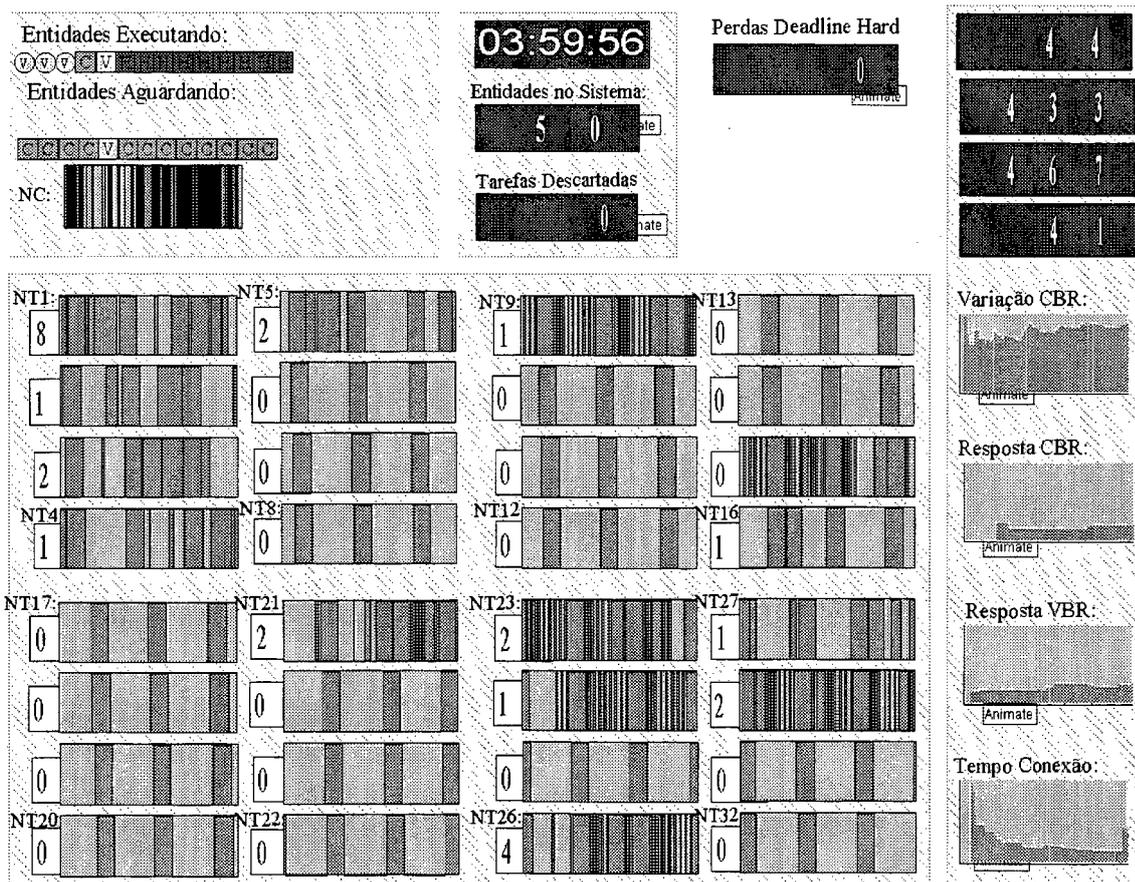
- Uso de Modelos Determinísticos: As distribuições de probabilidade envolvidas no modelo foram substituídas por constantes numéricas, eliminando sua aleatoriedade. Foi feita a monitoração das métricas de desempenho em várias simulações com fatores em diferentes níveis;
- Variação dos Dados de Entrada: Uma mudança num dado de entrada deve promover um resultado coerente. Foram feitas várias simulações, alterando os parâmetros de entrada para níveis subsequentes e então avaliadas as alterações nas métricas de saída. Por exemplo, a diminuição da quantidade de canais no *crossbar* levava a um aumento no tempo de conexão das tarefas *soft*;
- Refinamento Sucessivo: O comportamento do sistema foi incorporado ao modelo através de refinamento sucessivo, conforme o ciclo de *software* em ‘espiral’. Assim, o modelo inicialmente correspondia a casos simplificados, que tinham seu comportamento verificado. Em seguida novos refinamentos eram incluídos e nova verificação era aplicada;
- Rotinas de Depuração: Em casos específicos foram utilizadas as rotinas de depuração da ferramenta, que auxiliaram na resolução de problemas e posterior comprovação de funcionamento.

Os ANEXOS 9 a 12 apresentam exemplos de arquivos de *log* gerados pelo modelo durante testes de verificação. Os anexos 9 a 11 apresentam a execução exclusivamente de atividades *hard*, *soft-vbr* e *soft-cbr*, respectivamente. O ANEXO 12 apresenta a execução de todas as atividades no mesmo cenário. Os eventos constantes nos arquivos de *log* são consistentes com a especificação do comportamento do sistema.

A figura 9.10 apresenta a tela de animação, durante a execução de uma simulação, e foi criada também com o objetivo de verificação do modelo. Nela é possível visualizar, na parte direita, as métricas de desempenho de interesse, tanto na forma numérica (em cima) quanto gráfica (embaixo). Na parte central estão gráficos com a ocupação dos até 32 NTs representáveis, e a quantidade de canais físicos alocados em cada um (dígito à esquerda do gráfico). Na parte superior central estão o tempo atual da simulação e outras informações. E na parte superior esquerda é possível visualizar o tipo de cada entidade executando ou nas filas de tarefas periódicas e aperiódicas, e também é a ocupação do NC e sua fila de requisições pendentes. O ANEXO 4 também mostra essa parte do modelo.

Por não se tratar de modelo matemático, não é possível fornecer prova cabal de funcionamento. Apenas pela coerência dos resultados apresentados pelo modelo convenceu-se que a implementação do mesmo está conforme a especificação.

Figura 9.10 – Tela de animação para verificação do modelo



A Validação de um modelo refere-se à análise que permite estabelecer se a implementação do modelo realmente imita as características do sistema real de interesse. Um modelo é ‘validado’ se seu comportamento é semelhante ao que se verificaria no sistema real sob as mesmas condições simuladas.

Conforme JAIN [1991], “a validação refere-se a assegurar-se que as suposições usadas na concepção do modelo são aceitáveis, isto é, se corretamente implementadas, o modelo deveria produzir resultados próximos aos observados no sistema real”.

A Validação pode ser feita com uma comparação entre os resultados do modelo e os resultados coletados do sistema real. Entretanto, para o presente estudo, o sistema real ainda não foi construído, sendo que um de seus objetivos é justamente prever o desempenho de tal sistema. Assim, a validação direta não é possível. A comparação com o comportamento de sistemas ‘semelhantes’ mostrou coerência aceitável.

Outras técnicas incluem a comparação com modelos teóricos, como o emprego da Teoria das Filas ou Redes de Petri. O emprego da experiência e o conhecimento de especialistas na área também podem ser utilizados para validar um modelo. Apesar da busca por “indícios suficientes”, a validação nunca é demonstrada de forma definitiva. JAIN [1991] afirma que “a validação completa de um modelo é um mito”.

## 10. Análise dos Resultados

### 10.1. Introdução

Os experimentos planejados para a avaliação de desempenho de tempo-real do CRUX foram realizados utilizando o modelo descrito no capítulo anterior, sendo que dois projetos experimentais foram utilizados.

O primeiro conjunto de experimentos tem por objetivo fazer uma análise inicial da diferença das contribuições dos algoritmos de escalonamento sobre as métricas de desempenho escolhidas, desconsiderando outros fatores. O segundo conjunto de experimentos analisa simultaneamente a variação de 2 níveis dos 5 fatores e determina as contribuições de cada um sobre as métricas de desempenho.

### 10.1. Primeiro Conjunto de Experimentos

No primeiro conjunto de experimentos, composto por 24 simulações (experimentos), são avaliadas as diferenças nas 3 métricas de desempenho escolhidas para cada um dos 8 escalonadores de tempo-real modelados.

Através de simulações prévias obteve-se intervalos de confiança para as métricas de desempenho usadas, o que permitiu estipular uma nova duração para cada simulação. O tempo de simulação estipulado foi longo o suficiente para que o semi-intervalo de confiança da métrica “variação da taxa de execução de tarefas *soft-cbr*” fosse inferior a 10% de sua média, fornecendo, assim, um resultado estatístico preciso.

Deste modo, o tempo estipulado para cada simulação, que foi mantido constante em todas as simulações realizadas posteriormente, é de 0,5 s, ou seja  $5 \cdot 10^5 \mu\text{s}$ , que é a unidade de tempo fundamental do sistema. Uma análise visual prévia de gráficos das métricas de desempenho permitiu determinar o *tempo de aquecimento* em 2 ms ( $2 \cdot 10^3 \mu\text{s}$ ), período inicial da simulação no qual as estatísticas coletadas são desconsideradas.

No primeiro conjunto de experimentos, a carga de trabalho sobre o ambiente foi mantida fixa, variando-se apenas o algoritmo de escalonamento de tempo-real utilizado. Os parâmetros foram fixados nos valores apresentados na tabela 10.1.

Tabela 10.1 - Parâmetros fixos no primeiro conjunto de experimentos

<b>Parâmetros do Sistema</b>	<b>Valor</b>
Quantidade de Nós de Trabalho	16
Quantidade de canais físicos do <i>Crossbar</i>	2
Velocidade de acesso de um canal físico do <i>Crossbar</i> (Mb/s)	50
Velocidade de acesso do Barramento de Serviço (Mb/s)	10
Tamanho do pedido de conexão (bytes)	64
<b>de Carga</b>	
Fator de utilização das tarefas <i>hard</i>	0.45
Fator de utilização das tarefas <i>soft</i>	0.45
Quantidade de arquivos por atividade VBR	18
Quantidade de amostras por atividade CBR	100
Tamanho de uma mensagem de URL (bytes)	(SPEC <sub>web96</sub> )
Período de uma amostra de som ( $\mu$ s)	62,5
Tempo de processamento no NC	4
<b>do Projeto</b>	
Tempo de Simulação ( $\mu$ s)	$2 \cdot 10^6$
Quantidade de Replicações	1
Tempo de Aquecimento ( $\mu$ s)	$2 \cdot 10^3$

Para os algoritmos de escalonamento tipo “Servidor” (PS, DS, SS, DSS, ...), os parâmetros adotados foram:

- Período de ativação: É igual ao tempo de execução de uma tarefa *hard* ( $vTempoExecucaoHard$ ).
- Capacidade máxima: A capacidade das tarefas periódicas servidoras em atender tarefas aperiódicas segue a equação (46).

- **Prioridade:** Igual ao Período de ativação, para os servidores estáticos, e igual à *deadline* para os servidores dinâmicos.

Para o projeto experimental de um fator, cada um dos 8 algoritmos foi simulado 3 vezes (replicações), fornecendo 24 simulações, cujos resultados são apresentados na tabela 10.2.

Tabela 10.2 - Resultados das simulações do projeto experimental de um fator

Exec.	Seq	Fator	Resp. 1	Resp. 2	Resp. 3
1	16	SS	1895,1	42,515	1556,2
2	19	EDF	1796,1	36,450	1631,6
3	8	BS	1921,2	48,271	1726,5
4	3	EDF	2237,1	36,826	1473,6
5	7	BS	2551,6	42,814	1980,3
6	18	SS	2093,4	35,34	1799,0
7	1	EDF	2215,7	36,693	1512,4
8	10	PS	2418,5	94,73	1895,4
9	20	DSS	2017,3	34,156	1630,1
10	22	CBR	1725,0	29,912	1771,4
11	13	DS	1993,7	83,986	1628,5
12	23	CBR	1958,0	28,425	1645,5
13	12	PS	2898,1	91,952	1912,5
14	9	BS	2138,5	43,653	1602,7
15	2	EDF	1979,0	37,461	1632,4
16	15	DS	2193,4	83,146	1651,3
17	14	DS	2185,3	82,357	1795,2
18	4	RM	1946,6	48,748	1834,6
19	6	RM	1970,4	41,539	1935,0
20	11	PS	2728,3	92,129	1802,7
21	21	DSS	2031,5	34,951	1434,5
22	5	RM	2078,4	44,023	1760,2
23	17	SS	1960,1	39,333	1618,4
24	24	CBR	2029,0	30,123	1548,4

A coluna 'Seq' indica a ordem seqüencial na formação dos valores dos fatores. É recomendado não realizar os experimentos nessa ordem natural, mas sim aleatoriamente. A ferramenta *Expert Design* fornece então uma ordem alternativa para a realização dos experimentos, que corresponde à ordem apresentada nesse quadro, conforme a coluna 'Exec'. Os resultados propriamente ditos, para as 3 métricas de desempenho escolhidas, são apresentados nas 3 últimas colunas: 'Resp. 1', 'Resp. 2' e 'Resp. 3', que correspondem ao tempo de resposta *VBR*, à variação de execução *CBR* e ao tempo de conexão, respectivamente.

A seguir são apresentadas as análises dos resultados do primeiro conjunto de experimentos para cada uma dessas métricas.

#### 10.1.1. Tempo de Resposta das Tarefas *soft VBR*

Alguns resultados da ANOVA (*analysis of Variance*) sobre o tempo médio de resposta das tarefas *soft VBR*, ou simplesmente tempo de resposta *VBR*, são apresentados na tabela 10.3. As mudanças nos resultados dos diferentes escalonadores explicam 71,96% da variação dessa métrica. O valor F do algoritmo (5,87) implica que o modelo e o termo são significativos.

Tabela 10.3 – Resultados da ANOVA sobre o tempo de resposta *VBR*

Termo	Graus de Liberdade	Soma dos Quadrados	Quadrado Médio	Valor de F	Prob>F	Contribuição %
Algoritmo	7	1,298E+6	1,854E+5	5,87	0,0017	71,96
Erro	16	5,057E+5	31605,52			28,4

A média do tempo de resposta *VBR* foi de 2123,39  $\mu$ s, com desvio-padrão de 177,78  $\mu$ s e coeficiente de variação de 8,37%. A precisão foi de 7,576, sendo que é

desejável um valor superior a 4. A média do tempo de resposta *VBR* para cada algoritmo de escalonamento simulado é apresentada na tabela 10.4

Tabela 10.4 – Média do tempo de resposta *VBR* para os algoritmos de escalonamento

Algoritmo	Média	Erro Padrão
1 - EDF	2143,93	102,64
2 - RM	1998,47	102,64
3 - BS	2203,77	102,64
4 - PS	2681,63	102,64
5 - DS	2124,13	102,64
6 - SS	1982,87	102,64
7 - DSS	1948,30	102,64
8 - CBR	1904,00	102,64

Os resultados da ANOVA sobre a diferença das médias entre os escalonadores são apresentados na tabela 10.5. Os resultados dessa análise mostram que, ao nível de confiança de 95%, são significativas ( $Prob > |t|$  menor que 0,050) apenas as diferenças entre os seguintes pares de escalonadores: EDF e PS, RM e PS, BS e PS, BS e DSS, PS e DS, PS e SS, PS e DSS, PS e CBR. As diferenças entre os algoritmos podem ser melhor visualizadas na figura 10.1. Nela, percebe-se que os escalonadores cujos intervalos de confiança não se sobrepõem têm diferença significativa. Basicamente, verifica-se que o algoritmo PS tem diferença significativa em relação aos demais (é pior).

Tabela 10.5 – Resultados das diferenças entre as médias de tempo de resposta *VBR*

Algoritmos	Diferença Média	Graus de Liberdade	Erro Padrão	Valor de t para $H_0$	Prob >  t
EDF / RM	125,47	1	145,16	1,00	0,3312
EDF / BS	-59,83	1	145,16	-0,41	0,6857
EDF / PS	-537,70	1	145,16	-3,70	0,0019
EDF / DS	19,80	1	145,16	0,14	0,8932

Algoritmos	Diferença Média	Graus de Liberdade	Erro Padrão	Valor de t para $H_0$	Prob> t
EDF / SS	161,07	1	145,16	1,11	0,2836
EDF / DSS	195,63	1	145,16	1,35	0,1965
EDF / CBR	239,93	1	145,16	1,65	0,1178
RM / BS	-205,30	1	145,16	-1,41	0,1764
RM / PS	-683,17	1	145,16	-4,71	0,0002
RM / DS	-125,67	1	145,16	-0,87	0,3994
RM / SS	15,60	1	145,16	0,11	0,9158
RM / DSS	50,17	1	145,16	0,35	0,7341
RM / CBR	94,47	1	145,16	0,65	0,5244
BS / PS	-477,87	1	145,16	-3,29	0,0046
BS / DS	79,63	1	145,16	0,55	0,5909
BS / SS	220,90	1	145,16	1,52	0,1476
BS / DSS	255,47	1	145,16	1,76	0,0975
BS / CBR	299,77	1	145,16	2,07	0,0555
PS / DS	557,50	1	145,16	3,84	0,0014
PS / SS	698,77	1	145,16	4,81	0,0002
PS / DSS	733,33	1	145,16	5,05	0,0001
PS / CBR	777,63	1	145,16	5,36	< 0,0001
DS / SS	141,27	1	145,16	0,97	0,3449
DS / DSS	175,83	1	145,16	1,21	0,2434
DS / CBR	220,13	1	145,16	1,52	0,1489
SS / DSS	34,57	1	145,16	0,24	0,8148
SS / CBR	78,87	1	145,16	0,54	0,5944
DSS / CBR	44,30	1	145,16	0,31	0,7642

A análise estatística realizada faz algumas suposições acerca dos dados, conforme foi apresentado no item 8.2.9. Assim, a figura 10.2 apresenta dois gráficos: um gráfico de resíduos (a) e um gráfico *quantile-quantile* (b), que permitem comprovar que os erros são estatisticamente independentes (sem tendências) e normalmente distribuídos (forma de “S”, concentrado em torno de  $y=50$ ).

Figura 10.1 - Gráfico das diferenças entre as médias de tempo de resposta VBR

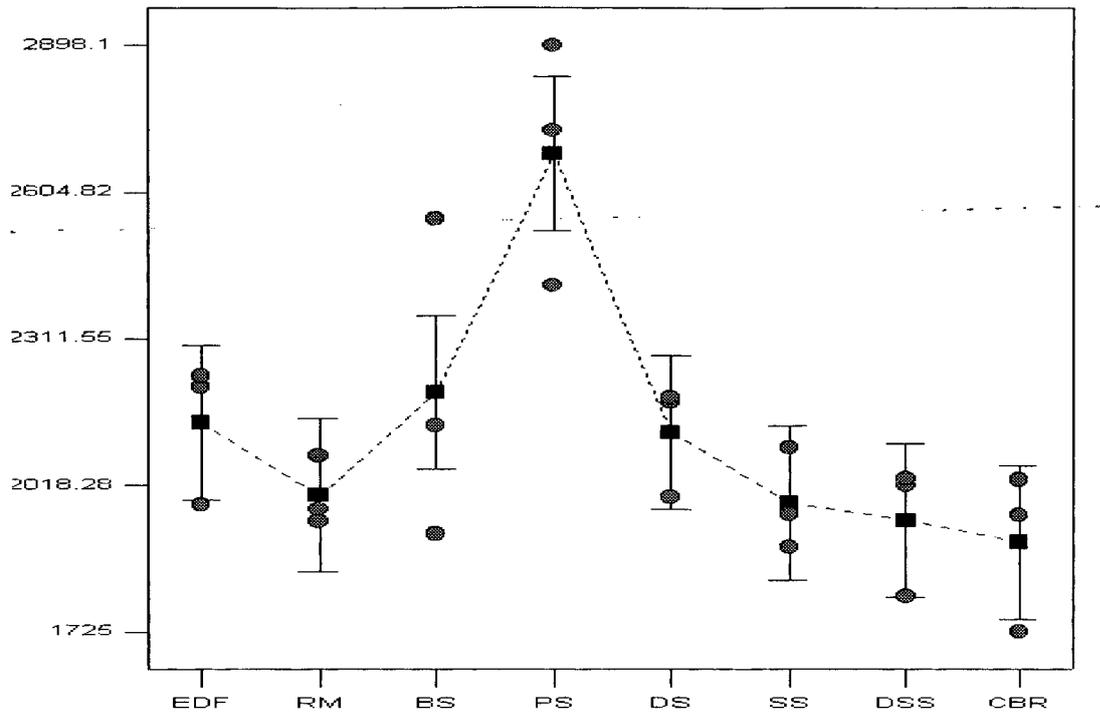
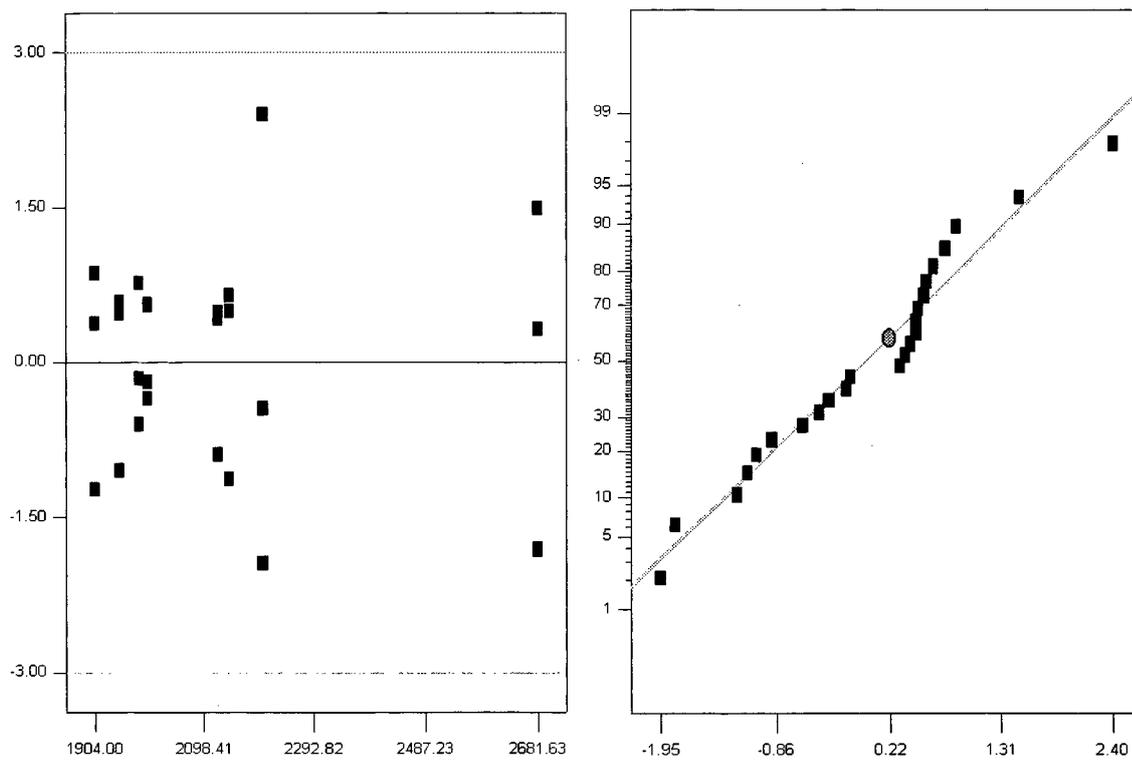


Figura 10.2 – Gráficos de resíduo e *quantile-quantile* do Tempo de resposta VBR



(a) Resíduos

(b) *Quantile*

### 10.1.2. Taxa de Variação de Execução das Tarefas *soft CBR*

Alguns resultados da ANOVA sobre a taxa de variação de execução das tarefas *soft CBR*, ou simplesmente taxa de variação *CBR*, são apresentados na tabela 10.6. As mudanças nos resultados dos diferentes escalonadores explicam 99,32% da variação dessa métrica. O valor F do algoritmo (332,43) implica que o modelo e o termo são significativos.

Tabela 10.6 – Resultados da ANOVA sobre a taxa de variação CBR

Termo	Graus de Liberdade	Soma dos Quadrados	Quadrado Médio	Valor de F	Prob>F	Contribuição %
Algoritmo	7	11861,10	1680,16	332,42	<0,0001	99,32
Erro	16	80,87	5,05			0,68

A média da taxa de variação *CBR* foi de 50,81  $\mu$ s, com desvio-padrão de 2,25  $\mu$ s e coeficiente de variação de 4,42%. A precisão foi de 48,883, sendo que valores superiores 4 são desejáveis. A média da taxa de variação *CBR* para cada algoritmo de escalonamento simulado é apresentada na tabela 10.7

Tabela 10.7 – Média da taxa de variação CBR para os algoritmos de escalonamento

Algoritmo	Média	Erro Padrão
1 - EDF	36,99	1,30
2 - RM	44,77	1,30
3 - BS	44,91	1,30
4 - PS	92,94	1,30
5 - DS	83,16	1,30
6 - SS	39,06	1,30
7 - DSS	35,19	1,30
8 - CBR	29,49	1,30

Os resultados da ANOVA sobre a diferença das médias entre os escalonadores são apresentados na tabela 10.8. Os resultados dessa análise mostram que, ao nível de confiança de 95%, são insignificantes ( $Prob>|t|$  maior que 0,050) apenas as diferenças entre os algoritmos EDF e SS, EDF e DSS, RM e BS. As diferenças entre os algoritmos podem ser melhor visualizadas na figura 10.3. Nela, percebe-se que são significativos os escalonadores cujos intervalos de confiança não se sobrepõem.

Tabela 10.8 – Resultados das diferenças entre as médias de variação de execução CBR

Algoritmos	Diferença Média	Graus de Liberdade	Erro Padrão	Valor de t para $H_0$	$Prob> t $
EDF / RM	-7,78	1	1,84	-4,24	0,0006
EDF / BS	-7,92	1	1,84	-4,31	0,0005
EDF / PS	-55,94	1	1,84	-30,48	< 0,0001
EDF / DS	-46,17	1	1,84	-25,15	< 0,0001
EDF / SS	-2,07	1	1,84	-1,13	0,2762
EDF / DSS	1,81	1	1,84	0,98	0,3394
EDF / CBR	7,51	1	1,84	4,09	0,0009
RM / BS	-0,14	1	1,84	-0,078	0,9390
RM / PS	-48,17	1	1,84	-26,24	< 0,0001
RM / DS	-38,39	1	1,84	-20,92	< 0,0001
RM / SS	5,71	1	1,84	3,11	0,0067
RM / DSS	9,58	1	1,84	5,22	< 0,0001
RM / CBR	15,28	1	1,84	8,33	< 0,0001
BS / PS	-48,02	1	1,84	-26,16	< 0,0001
BS / DS	-38,25	1	1,84	-20,84	< 0,0001
BS / SS	5,85	1	1,84	3,19	0,0057
BS / DSS	9,73	1	1,84	5,30	< 0,0001
BS / CBR	15,43	1	1,84	8,40	< 0,0001
PS / DS	9,77	1	1,84	5,32	< 0,0001
PS / SS	53,87	1	1,84	29,35	< 0,0001
PS / DSS	57,75	1	1,84	31,46	< 0,0001

PS / CBR	63,45	1	1,84	34,57	< 0,0001
DS / SS	44,10	1	1,84	24,02	< 0,0001
DS / DSS	47,98	1	1,84	26,14	< 0,0001
DS / CBR	53,68	1	1,84	29,24	< 0,0001
SS / DSS	3,88	1	1,84	2,11	0,0508
SS / CBR	9,58	1	1,84	5,22	< 0,0001
DSS / CBR	5,70	1	1,84	3,10	0,0068

A análise estatística realizada faz algumas suposições acerca dos dados. Assim, a figura 10.4 apresenta dois gráficos: um gráfico de resíduos (a) e um gráfico *quantile-quantile* (b), que permitem comprovar que os erros são estatisticamente independentes e normalmente distribuídos.

Figura 10.3 - Gráfico das diferenças entre as médias de variação de execução CBR

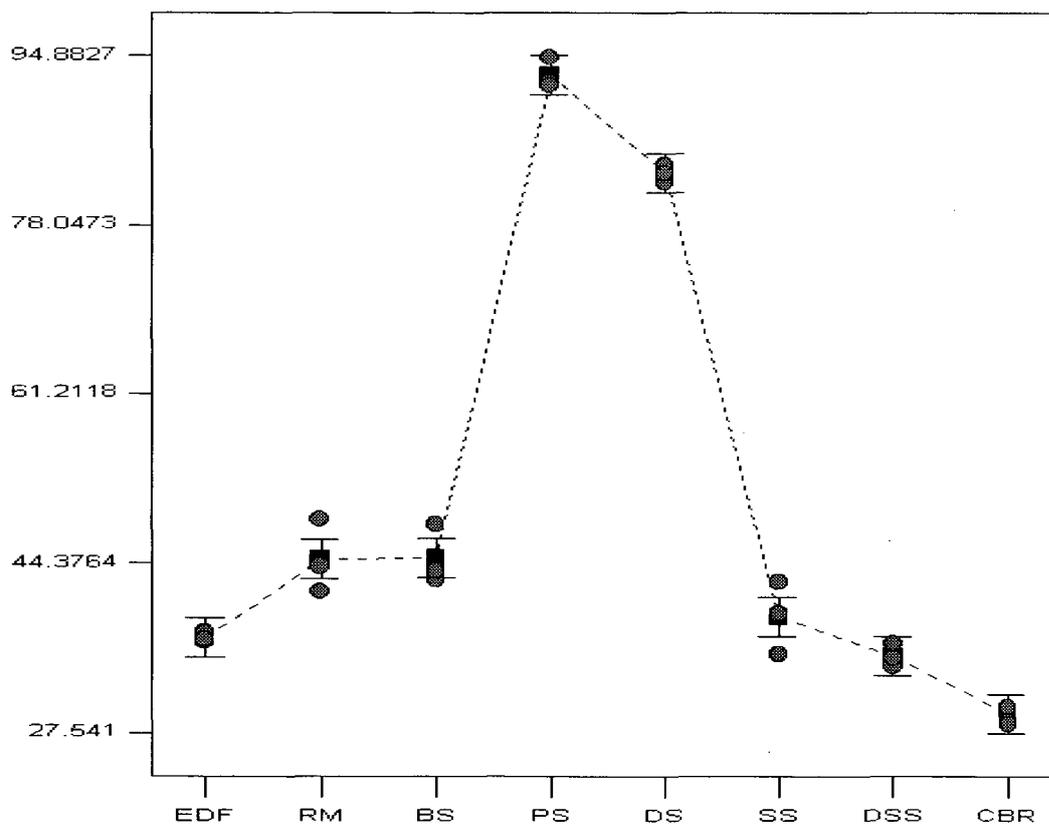
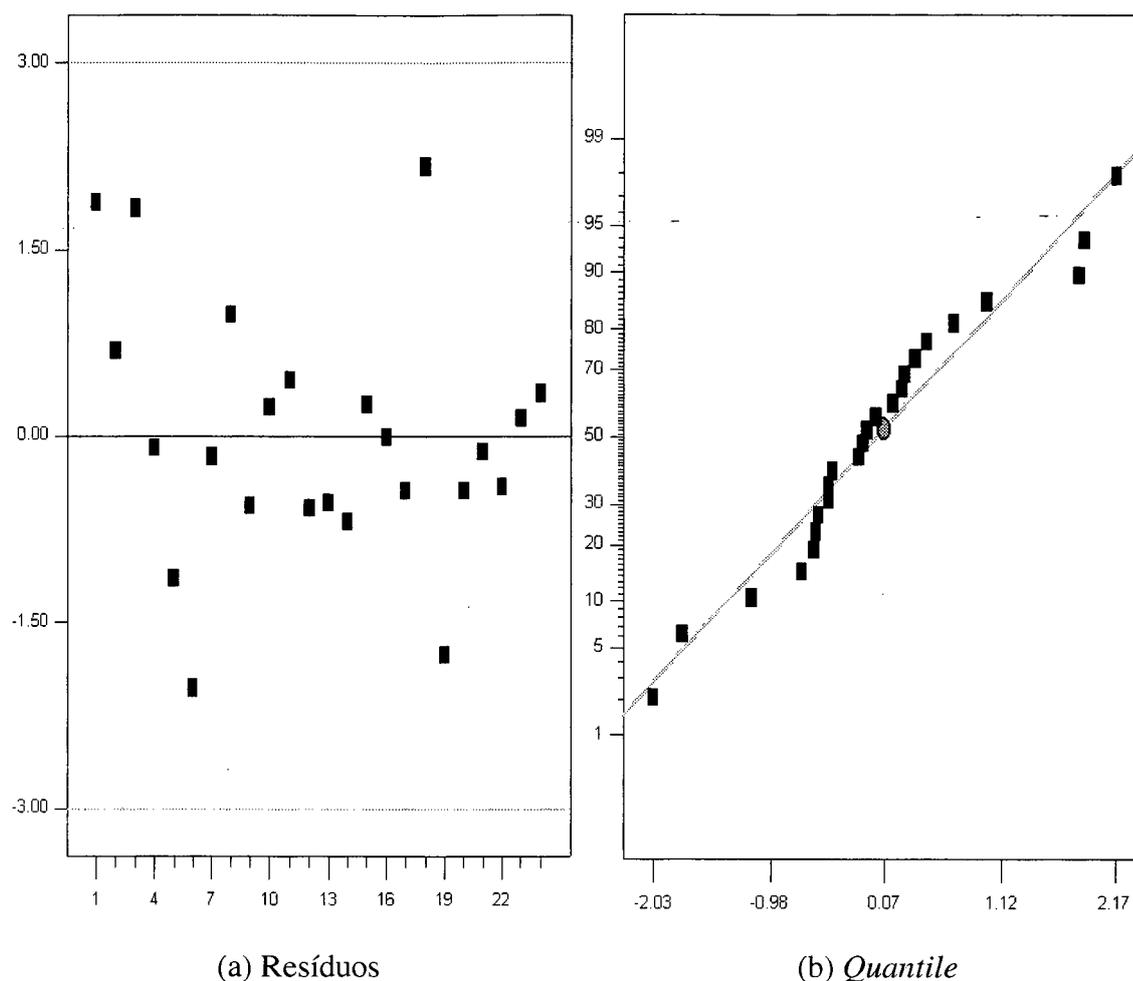


Figura 10.4 – Gráficos de resíduo e *quantile-quantile* da variação de execução *CBR*



### 10.1.2. Tempo de Conexão das Tarefas *soft*

Alguns resultados da ANOVA sobre o tempo de conexão das tarefas *soft VBR* ou *CBR*, ou simplesmente tempo de conexão, é apresentado na tabela 10.9. As mudanças nos resultados dos diferentes escalonadores explicam 58,87% da variação dessa métrica. O valor F do algoritmo (3,27) implica que o algoritmo ainda é significativo, mesmo não tendo muita influência.

Tabela 10.9 – Resultados da ANOVA sobre o tempo de conexão

<b>Termo</b>	<b>Graus de Liberdade</b>	<b>Soma dos Quadrados</b>	<b>Quadrado Médio</b>	<b>Valor de F</b>	<b>Prob&gt;F</b>	<b>Contribuição %</b>
Algoritmo	7	3,093E+5	44178,60	3,27	0,0235	58,87
Erro	16	2,160E+5	13502,81			41,13

O tempo médio de conexão foi de 1697,85  $\mu$ s, com desvio-padrão de 116,20  $\mu$ s e coeficiente de variação de 6,84%. A precisão foi de 4,93, sendo que valores maiores que 4 são desejáveis. O tempo médio de conexão para cada algoritmo de escalonamento simulado é apresentada na tabela 10.10

Tabela 10.10 – Tempo médio de conexão para os algoritmos de escalonamento

<b>Algoritmo</b>	<b>Média</b>	<b>Erro Padrão</b>
1 - EDF	1539,47	67,09
2 - RM	1843,27	67,09
3 - BS	1769,83	67,09
4 - PS	1870,20	67,09
5 - DS	1691,67	67,09
6 - SS	1657,87	67,09
7 - DSS	1565,40	67,09
8 - CBR	1645,07	67,09

Os resultados da ANOVA sobre a diferença das médias entre os escalonadores são apresentados na tabela 10.11. Os resultados dessa análise mostram que, ao nível de confiança de 95%, são significantes ( $'Prob>|t|'$  menor que 0,050) apenas as diferenças entre os algoritmos EDF e RM, EDF e BS, EDF e PS, RM e DSS, BS e DSS, PS e SS, PS e DSS, PS e CBR. As diferenças entre os algoritmos podem ser melhor visualizadas na figura 10.5. Nela, percebe-se que os escalonadores cujos intervalos de confiança não se sobrepõem têm diferença significativa.

Tabela 10.11 – Resultados das diferenças entre as médias de tempo de conexão

Algoritmos	Diferença Média	Graus de Liberdade	Erro Padrão	Valor de t para $H_0$	Prob> t
EDF / RM	-303,80	1	94,88	-3,20	0,0056
EDF / BS	-230,37	1	94,88	-2,43	0,0273
EDF / PS	-330,73	1	94,88	-3,49	0,0031
EDF / DS	-152,20	1	94,88	-1,60	0,1282
EDF / SS	-118,40	1	94,88	-1,25	0,2300
EDF / DSS	-25,93	1	94,88	-0,27	0,7881
EDF / CBR	-105,60	1	94,88	-1,11	0,2821
RM / BS	73,43	1	94,88	0,77	0,4502
RM / PS	-26,93	1	94,88	-0,28	0,7801
RM / DS	151,60	1	94,88	1,60	0,1296
RM / SS	185,40	1	94,88	1,95	0,0684
RM / DSS	277,87	1	94,88	2,93	0,0098
RM / CBR	198,20	1	94,88	2,09	0,0530
BS / PS	-100,37	1	94,88	-1,06	0,3058
BS / DS	78,17	1	94,88	0,82	0,4221
BS / SS	111,97	1	94,88	1,18	0,2552
BS / DSS	204,43	1	94,88	2,15	0,0468
BS / CBR	124,77	1	94,88	1,32	0,2070
PS / DS	178,53	1	94,88	1,88	0,0782
PS / SS	212,33	1	94,88	2,24	0,0398
PS / DSS	304,80	1	94,88	3,21	0,0054
PS / CBR	225,13	1	94,88	2,37	0,0305
DS / SS	33,80	1	94,88	0,36	0,7263
DS / DSS	126,27	1	94,88	1,33	0,2019
DS / CBR	46,60	1	94,88	0,49	0,6300
SS / DSS	92,47	1	94,88	0,97	0,3443
SS / CBR	12,80	1	94,88	0,13	0,8944
DSS / CBR	-79,67	1	94,88	-0,84	0,4135

A análise estatística realizada faz algumas suposições acerca dos dados. Assim, a figura 10.6 apresenta dois gráficos: um gráfico de resíduos (a) e um gráfico *quantile-quantile* (b), que permitem comprovar que os erros são estatisticamente independentes e normalmente distribuídos.

Figura 10.5 - Gráfico das diferenças entre as médias de tempo de conexão

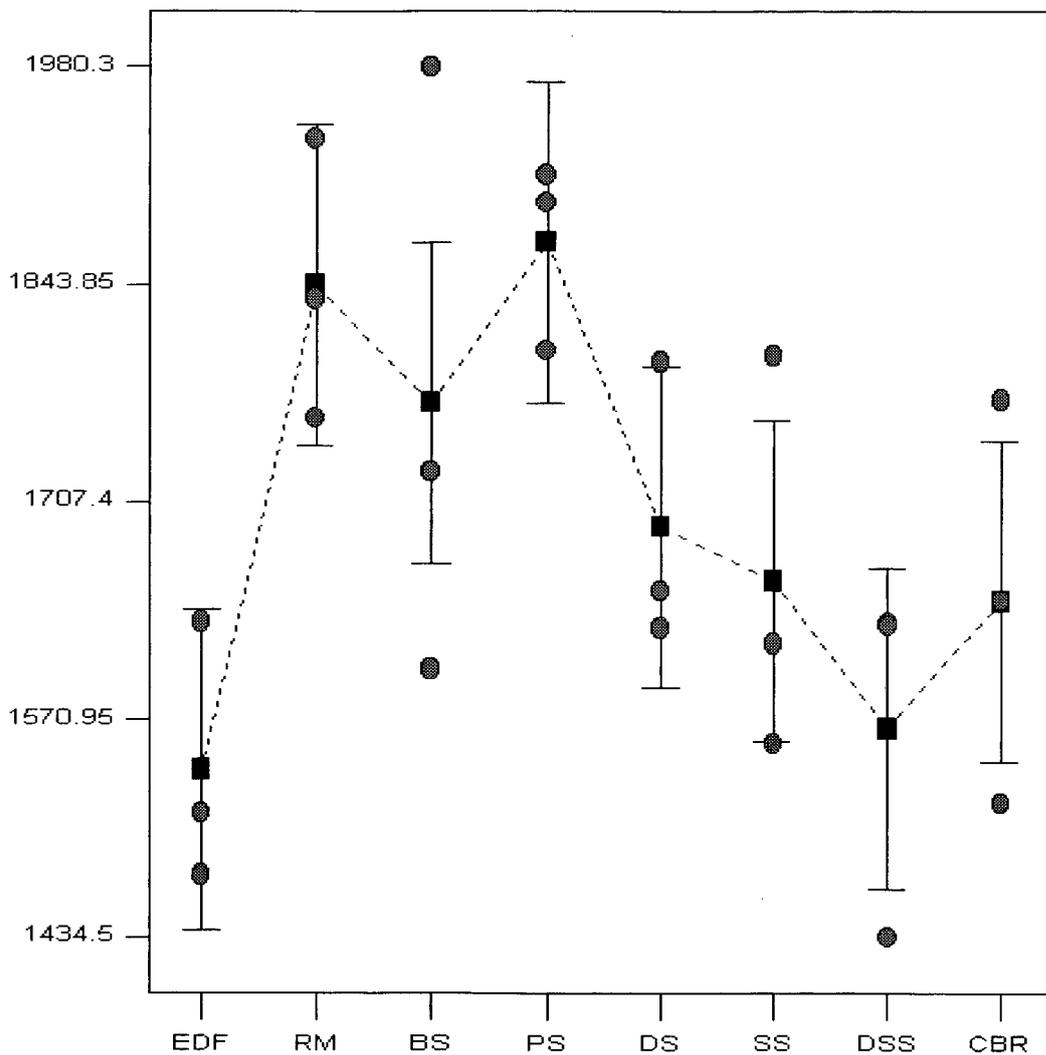
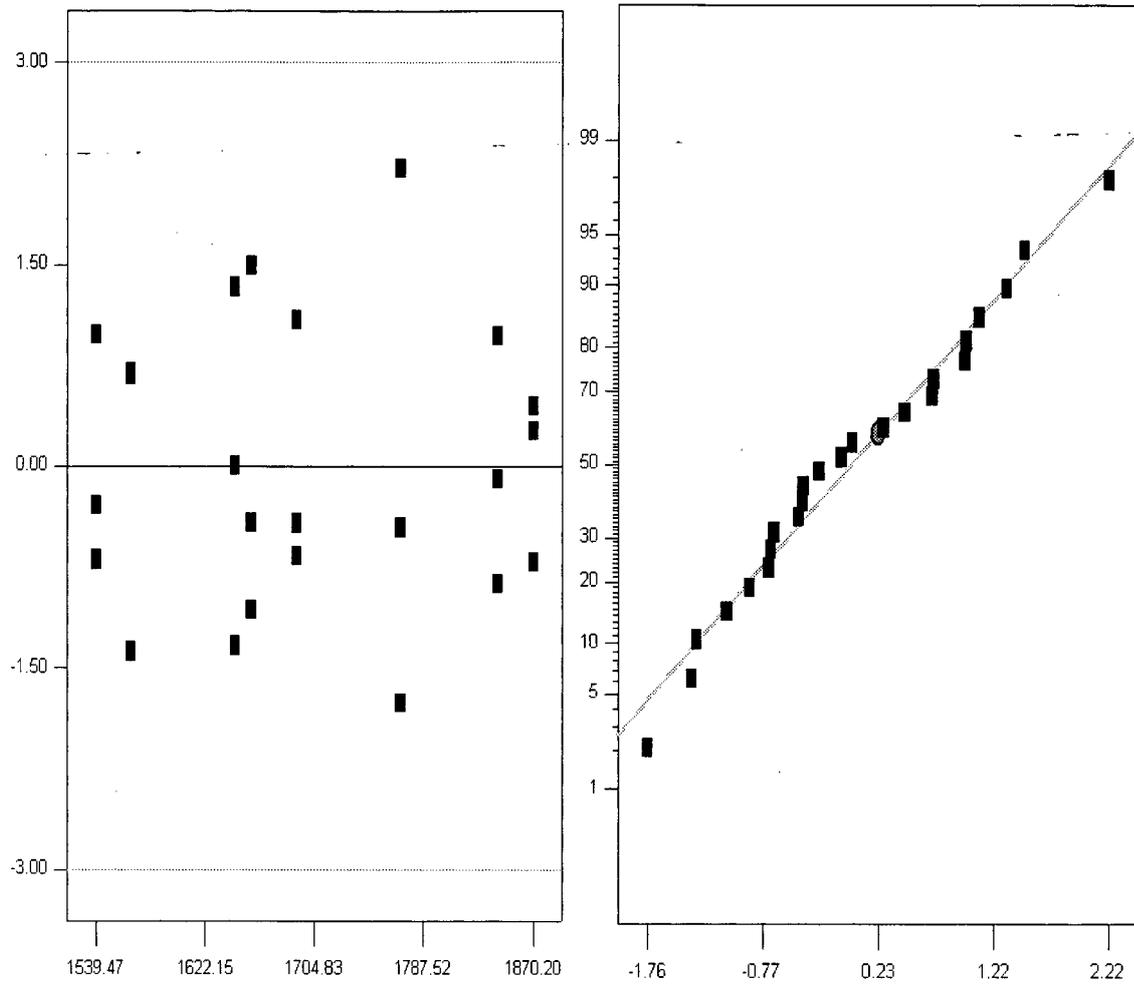


Figura 10.6 – Gráficos de resíduo e *quantile-quantile* do tempo de conexão

(a) Resíduos

(b) *Quantile*

## 10.2. Segundo Conjunto de Experimentos

O projeto experimental fatorial completo  $2^5$  foi também realizado com o auxílio da ferramenta estatística *Design Expert 6.0*. As características dos fatores utilizados nesse projeto experimentais são apresentadas na tabela 10.12

Tabela 10.12 – Fatores e suas características do projeto experimental fatorial  $2^5$

Fator	Nome	Unidade	Tipo	Mín. (-1)	Máx. (+1)
A	Nós de Trabalho	Processadores	Numérico	8	32
B	Canais Físicos	Canais	Numérico	1	4
C	Algoritmo de Escalonamento		Catagórico	PS	CBR
D	Fator de Utilização da CPU	%	Numérico	80	110
E	Velocidade do <i>Crossbar</i>	Mb/s	Numérico	1	100

Os resultados das 32 simulações ( $2^5=32$ ) são apresentados na tabela 10.13. A coluna 'Seq' indica a ordem seqüencial na formação dos valores dos fatores. É recomendado não realizar os experimentos nessa ordem natural, mas sim aleatoriamente. A ferramenta *Expert Design* fornece então uma ordem alternativa para a realização dos experimentos, que corresponde à ordem apresentada nesse quadro, conforme a coluna 'Exec'. Os resultados propriamente ditos, para as 3 métricas de desempenho escolhidas, são apresentados nas 3 últimas colunas: 'Resp. 1', 'Resp. 2' e 'Resp. 3', que correspondem ao tempo de resposta VBR, à variação de execução CBR e ao tempo de conexão, respectivamente.

Tabela 10.13 – Resultados das simulações do projeto experimental fatorial completo 2<sup>5</sup>.

Exec.	Seq	Fator A	Fator B	Fator C	Fator D	Fator E	Resp. 1	Resp. 2	Resp. 3
1	18	32	1	PS	80	100	5909,5	134,45	9396,8
2	23	8	4	CBR	80	100	1498,5	42,817	249,5
3	8	32	4	CBR	80	1	5711,8	17,289	3920,0
4	1	8	1	PS	80	1	14967,0	45,062	6139,5
5	7	8	4	CBR	80	1	3772,17	37,94	2007,5
6	31	8	4	CBR	110	100	3293,64	37,605	229,27
7	22	32	1	CBR	80	100	6501,15	15,188	8767,2
8	2	32	1	PS	80	1	11992	127,02	9537,8
9	29	8	1	CBR	110	100	3069,5	35,37	1819,1
10	17	8	1	PS	80	100	3107,5	61,23	2251,1
11	6	32	1	CBR	80	1	15056,0	14,588	12886
12	30	32	1	CBR	110	100	2338,5	29,816	3027,7
13	11	8	4	PS	110	1	5829,8	36,225	3727,9
14	5	8	1	CBR	80	1	5975,7	31,276	5000,0
15	10	32	1	PS	110	1	13087,0	138,3	11456,0
16	32	32	4	CBR	110	100	2057,0	31,709	252,25
17	16	32	4	CBR	110	1	6326,2	151,216	7991,3
18	3	8	4	PS	80	1	4573,4	64,019	3575,2
19	12	32	4	PS	110	1	7085,5	153,333	8182,8
20	26	32	1	PS	110	100	2665,0	98,847	4181,8
21	14	32	1	CBR	110	1	2052,4	50,661	3990,1
22	24	32	4	CBR	80	100	2877,6	28,656	827,07
23	27	8	4	PS	110	100	3150,4	49,001	299,52
24	19	8	4	PS	80	100	1936,6	83,513	254,9
25	13	8	1	CBR	110	1	7692,5	34,684	3292,6
26	15	8	4	CBR	110	1	5245,0	36,159	5336,9
27	25	8	1	PS	110	100	5168,4	36,459	2765,7
28	9	8	1	PS	110	1	10770,0	32,629	5219,9
29	21	8	1	CBR	80	100	3385,7	22,989	2347,3
30	4	32	4	PS	80	1	3664,0	181,428	5554,4
31	28	32	4	PS	110	100	1625,0	133,037	452,88
32	20	32	4	PS	80	100	3681,6	183,424	1041,45

A seguir são apresentadas as análises dos resultados do segundo conjunto de experimentos para cada uma dessas métricas.

### 10.2.1. Análise do Tempo de Resposta das Atividades *Soft Real Time VBR*.

As contribuições dos fatores sobre o Tempo de Resposta das Atividades *Soft Real Time VBR*, ou simplesmente Tempo de Resposta *VBR*, são apresentadas na tabela 10.14. Foram escolhidos como fatores significantes aqueles com contribuição igual ou superior a 3%, ou seja, os fatores B, C, e E, e as interações entre os fatores AD, BC, BD, BE, ACD e ACDE. Esses fatores, juntos, explicam 78,74% da variação dessa métrica.

Além dos fatores considerados relevantes, apresentados acima, os fatores e as interações A, D, AE, CD, CE, DE, ACE, ADE, CDE foram acrescidos ao modelo para que esse se torne hierárquico, mesmo não tendo contribuição significativa.

Tabela 10.14 – Contribuição dos fatores sobre o tempo de resposta *soft VBR*

<b>Termo</b>	<b>Efeito</b>	<b>Contribuição (%)</b>
A	574,65	0,59
B	-3213,10	18,45
C	-1397,46	3,49
D	-822,15	1,21
E	-4470,93	35,73
AB	-108,50	0,021
AC	548,84	0,54
AD	-1447,48	3,75
AE	-194,01	0,067
BC	1301,91	3,03
BD	1684,26	5,07
BE	1709,99	5,23
CD	-765,84	1,05
CE	1119,66	2,24
DE	130,81	0,031

Termo	Efeito	Contribuição (%)
ABC	-224,17	0,09
ABD	875,05	1,37
ABE	-181,62	0,059
ACD	-1307,65	3,06
ACE	-297,75	0,16
ADE	-432,27	0,33
BCD	669,17	0,80
BCE	-1190,82	2,53
BDE	-959,98	1,65
CDE	581,10	0,60
ACBD	1011,54	1,83
ABCE	-46,21	0,0038
ABDE	-466,83	0,39
ACDE	1571,85	4,42
BCDE	-30,10	0,0016
ABCDE	-1112,07	2,21

A análise da variância ANOVA (*Analysis of Variance*) é apresentada nas tabelas 10.15 e 10.16. O valor da estatística F do modelo (5,13) indica que este é significativo. Os termos B, E, BD, BE e ACDE são considerados significantes (*Prob>F* menor que 0,05). Dos termos escolhidos inicialmente, com contribuição igual ou superior a 3%, nenhum foi considerado insignificante (*Prob>F* maior que 0,10).

O algoritmo de escalonamento, fator C, mostrou-se significativo, embora com uma contribuição ao tempo de resposta VBR pequeno, com 3,49%. A velocidade da rede *crossbar* (fator E), como era esperado, foi o termo de maior importância (35,73%), seguido pela quantidade de Canais Físicos (fator B, 18,45%), termo intimamente relacionado ao primeiro. O terceiro termo com maior contribuição foi justamente a interação entre esses dois fatores, BE (não adiante ter um *crossbar* muito rápido e apenas um único canal). Esses resultados confirmam o que já era esperado. Apesar do escalonamento ser significativo, a velocidade da rede contribui muito mais para a determinação do tempo de resposta das atividades *soft real-time VBR*.

Os valores das colunas '*R-Squared Pred*' e '*R-Squared Ajust*'. não estão tão próximos quanto desejados, o que talvez poderia ser melhorado com uma redução do modelo (menos termos) ou com uma verificação de *outliers*. Entretanto, essa diferença não é significativa a ponto de refazer o estudo. Isso pode ser visto pelo valor da coluna '*Precisão Adeq.*', que mede a "taxa de ruído" (erro), e tem valor 8,415, sendo desejados valores superiores a 4.

Tabela 10.15 – Resultados da ANOVA sobre o tempo de resposta *VBR*

Fonte	Soma dos Quadrados	Graus de Liberdade	Quadrado Médio	Valor de F	Prob>F
Modelo	3,985E+8	19		5,13	0,0029
A	2,642E+6	1	2,642E+6	0,65	0,4370
B	8,259E+7	1	8,259E+7	20,21	0,0007
C	1,562E+7	1	1,562E+7	3,82	0,0742
D	5,407E+6	1	5,407E+6	1,32	0,2724
E	1,599E+8	1	1,599E+8	39,14	< 0,0001
AC	2,410E+6	1	2,410E+6	0,59	0,4574
AD	1,676E+7	1	1,676E+7	4,10	0,0657
AE	3,011E+5	1	3,011E+5	0,074	0,7906
BC	1,356E+7	1	1,356E+7	3,32	0,0935
BD	2,269E+7	1	2,269E+7	5,55	0,0363
BE	2,339E+7	1	2,339E+7	5,73	0,0340
CD	4,692E+6	1	4,692E+6	1,15	0,3050
CE	1,003E+7	1	1,003E+7	2,45	0,1432
DE	1,369E+5	1	1,369E+5	0,034	0,8578
ACD	1,368E+7	1	1,368E+7	3,35	0,0922
ACE	7,092E+5	1	7,092E+5	0,17	0,6843
ADE	1,495E+6	1	1,495E+6	0,37	0,5565
CDE	2,701E+6	1	2,701E+6	0,66	0,4320
ACDE	1,977E+7	1	1,977E+7	4,84	0,0482
Resíduo	4,903E+7	12	4,086E+6		
Corr. Total	4,475E+8	31			

Tabela 10.16 – Estatísticas da ANOVA sobre o tempo de resposta *VBR*

Estatística	Valor
Média	5502,06
Desvio-padrão	2021,39
Coef. Variação	36,74
R-Squared Ajust.	0,7170
R-Squared Pred.	0,2209
Precisão Adeq.	8,415

A equação resultante de uma regressão linear sobre o modelo, usada para prever os valores do Tempo de Resposta *VBR*, é apresentada a seguir. Essa equação assume que os níveis dos fatores (variáveis da equação) estão codificados, ou seja, são sempre  $-1$  (nível mínimo) ou  $+1$  (nível máximo).

$$\begin{aligned}
 \text{Tempo Resposta } VBR = & 5502,06 + 287,33 * A - 1606,55 * B + \\
 & - 698,73 * C - 411,07 * D - 2235,47 * E + 274,42 * A * C + \\
 & - 723,74 * A * D - 97,01 * A * E + 650,96 * B * C + 842,12 * B * D + \\
 & + 854,99 * B * E - 382,92 * C * D + 559,83 * C * E + 65,41 * D * E + \\
 & - 653,82 * A * C * D - 148,88 * A * C * E - 216,13 * A * D * E + \\
 & + 290,55 * C * D * E + 785,92 * A * C * D * E
 \end{aligned} \tag{56}$$

Para prever o Tempo de Resposta *VBR* utilizando outros níveis desses fatores, além dos níveis mínimo e máximo adotados, como por exemplo, com 12 NTs e 3 Canais Físicos, a equação (56) não é adequada. Nesses casos, deve-se utilizar as equações (57) e (58), que aceitam quaisquer valores para os fatores (com exceção do algoritmo de escalonamento). São necessárias duas equações, pois o fator C, o algoritmo de escalonamento, é categórico (qualitativo), necessitando de uma equação para cada algoritmo.

Para o algoritmo de escalonamento PS:

$$\begin{aligned}
 \text{Tempo Resposta VBR} = & \quad (57) \\
 & +32001,18496 + \\
 & -505,97976 * \text{Nós de Trabalho} + \\
 & -5642,17021 * \text{Canais Físicos} + \\
 & -185,95426 * \text{Fator Utilização} + \\
 & -271,88045 * \text{Velocidade Crossbar} + \\
 & +5,29101 * \text{Nós de Trabalho} * \text{Fator Utilização} + \\
 & +10,77142 * \text{Nós de Trabalho} * \text{Velocidade Crossbar} + \\
 & +37,42794 * \text{Canais Físicos} * \text{Fator Utilização} + \\
 & +11,51508 * \text{Canais Físicos} * \text{Velocidade Crossbar} + \\
 & +1,94606 * \text{Fator Utilização} * \text{Velocidade Crossbar} + \\
 & -0,11246 * \text{Nós de Trabalho} * \text{Fator Utilização} * \text{Velocidade Crossbar}
 \end{aligned}$$

Para o algoritmo de escalonamento CBR:

$$\begin{aligned}
 \text{Tempo Resposta VBR} = & \quad (58) \\
 & +3745,70232 + \\
 & +1101,56158 * \text{Nós de Trabalho} + \\
 & -4774,23021 * \text{Canais Físicos} + \\
 & +46,93929 * \text{Fator Utilização} + \\
 & +21,60108 * \text{Velocidade Crossbar} + \\
 & -10,88258 * \text{Nós de Trabalho} * \text{Fator Utilização} + \\
 & -6,48915 * \text{Nós de Trabalho} * \text{Velocidade Crossbar} + \\
 & +37,42794 * \text{Canais Físicos} * \text{Fator Utilização} + \\
 & +11,51508 * \text{Canais Físicos} * \text{Velocidade Crossbar} + \\
 & -0,79959 * \text{Fator Utilização} * \text{Velocidade Crossbar} + \\
 & +0,063949 * \text{Nós de Trabalho} * \text{Fator Utilização} * \text{Velocidade Crossbar}
 \end{aligned}$$

Os coeficientes da equação (56), estimados a partir de regressão linear do modelo, são o ponto central de uma estimativa intervalar, com nível de significância de 5% (erro-padrão de 357,33). O intervalo de confiança para cada um desses coeficientes

é apresentado no ANEXO 13. Esse mesmo anexo apresenta também uma comparação (resíduos) dos resultados da simulação com os valores preditos pela equação. A figura 10.7 permite uma melhor visualização da relação entre os valores atuais (resultados da simulação) e a reta de valores preditos.

A análise estatística realizada faz algumas suposições acerca dos dados, conforme foi apresentado no item 8.2.9. Assim, a figura 10.8 apresenta dois gráficos: um gráfico de resíduos (a) e um gráfico *quantile-quantile* (b), que permitem comprovar que os erros são estatisticamente independentes (sem tendências) e normalmente distribuídos (forma de “S”, concentrado em torno de  $y=50$ ).

Figura 10.7 – Valores atuais x Valores preditos do Tempo de Resposta VBR

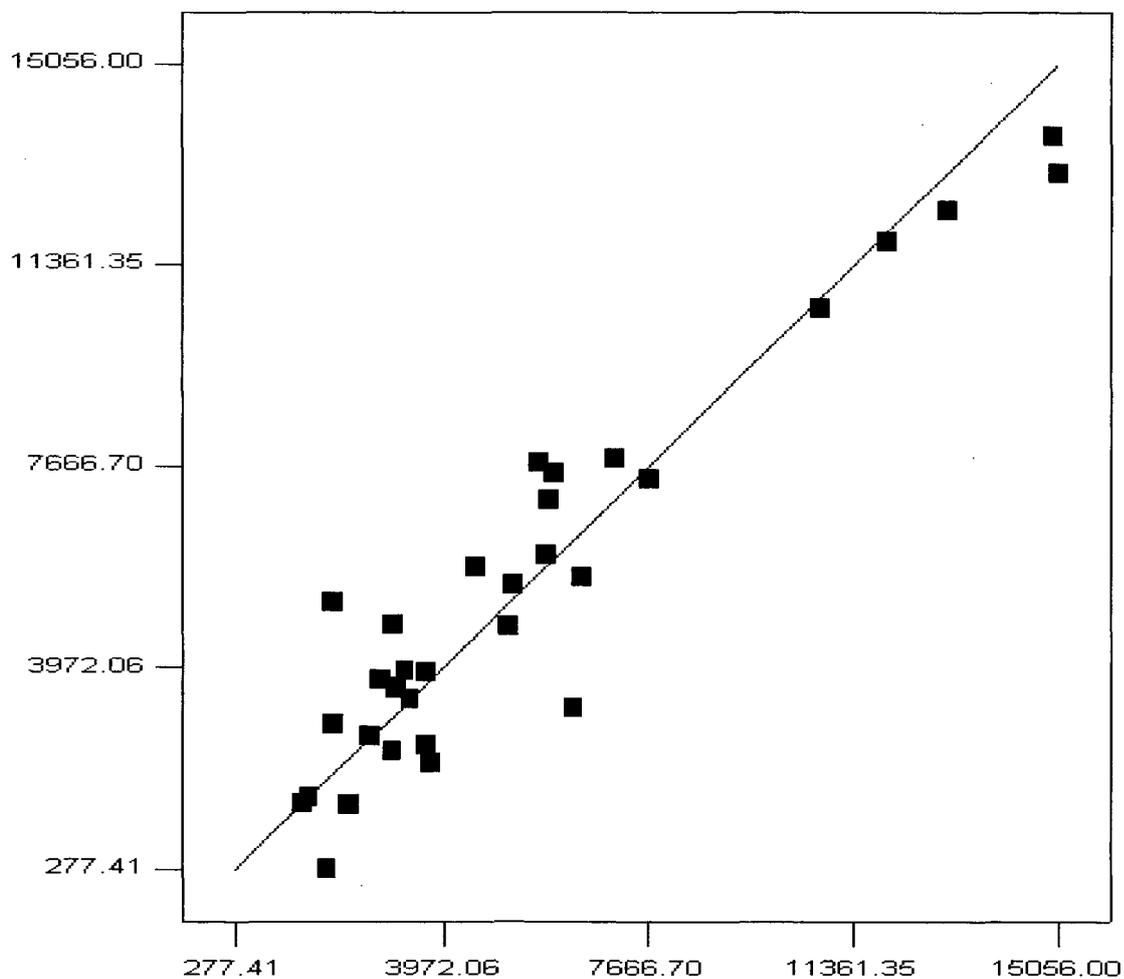
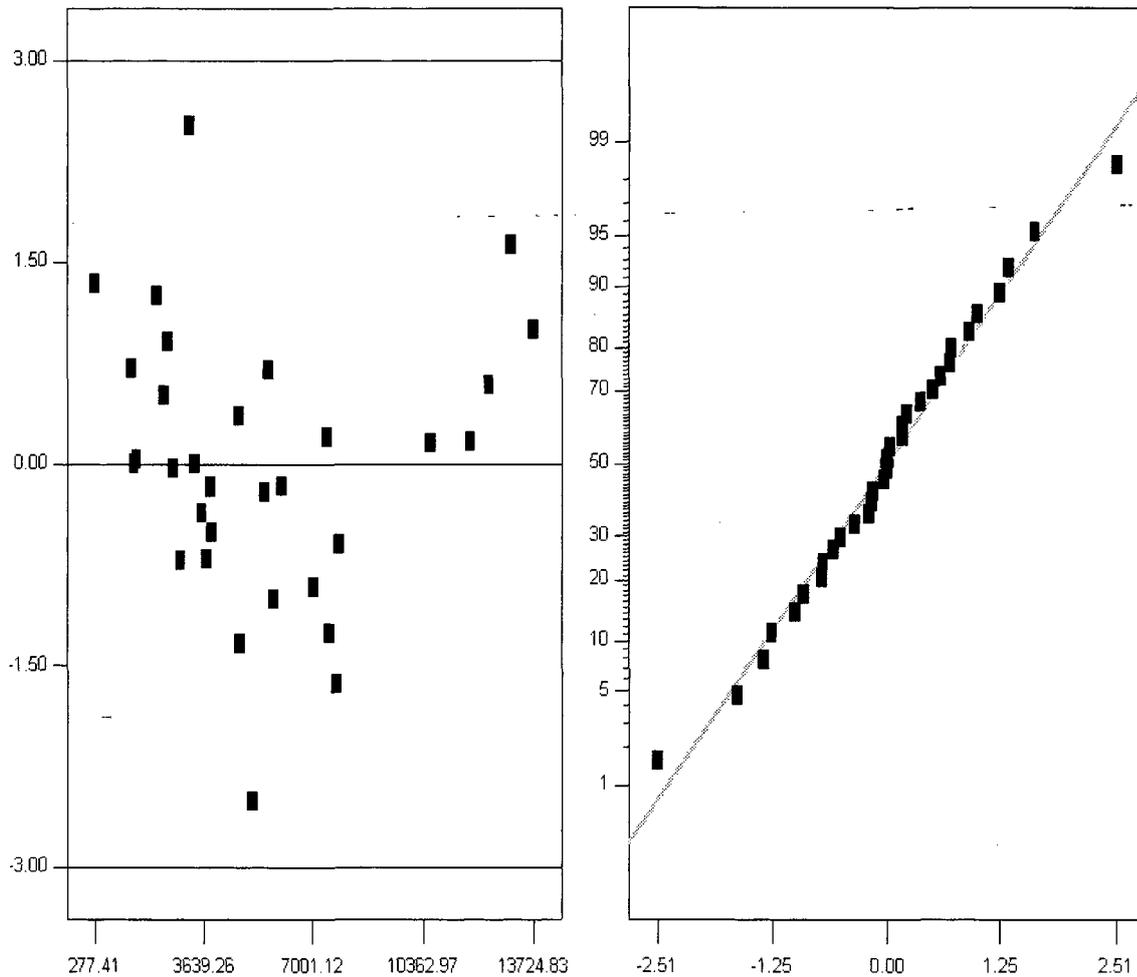


Figura 10.8 – Gráficos de resíduo e *quantile-quantile* do Tempo de resposta *VBR*



(a) Resíduos

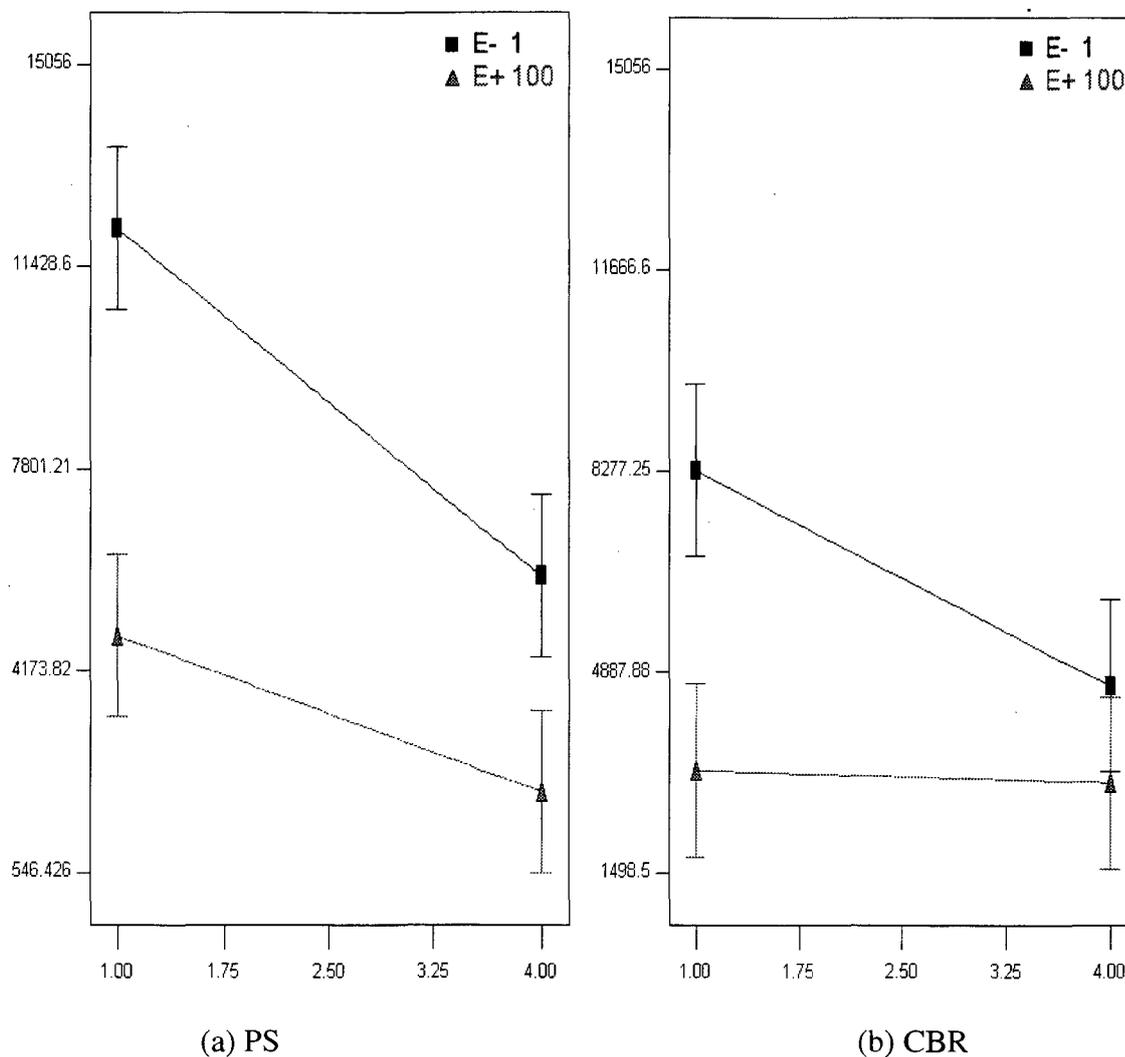
(b) *Quantile*

Para uma melhor compreensão e visualização da influência dos fatores sobre o tempo de resposta das atividades *soft VBR*, são apresentados dois gráficos de interação na figura 10.9. Esses gráficos apresentam o tempo de resposta *VBR* no eixo das ordenadas (y), em relação sempre a dois fatores, um representado no eixo das abscissas (x) e outro pelas duas curvas do gráfico, uma para cada nível desse segundo fator. Cada gráfico dessa figura ilustra um nível do escalonamento (PS ou CBR), permitindo analisar também a variação desse terceiro fator.

Na figura 10.6, a quantidade de NTs (fator A) foi mantida em 20 e o fator de utilização (fator D) foi mantido em 95% (pontos médios desses fatores). Os três fatores principais são então variados: A quantidade de canais físicos (fator B) no eixo das

abscissas e a velocidade do *crossbar* (fator E) pelas duas curvas no gráfico, conforme legenda na própria figura. O gráfico (a) apresenta essa relação quando escalonamento (fator C) é o PS e o gráfico (b) quando o escalonamento é o CBR.

Figura 10.9 – Gráficos de interação do tempo de resposta VBR



Através desses gráficos pode-se notar que a velocidade do *Crossbar* (fator E) tem 4 mais influência quando há poucos canais físicos (fator B), para ambos os níveis do escalonador (distância das curvas à esquerda e à direita de cada gráfico). Também o próprio escalonador influencia mais o tempo de resposta quando há poucos canais físicos e a velocidade do *crossbar* é lenta (fator E no mínimo: 1Mb/s), chegando a ser insignificante no caso dos fatores B e E estarem no nível máximo.

### 10.2.2. Análise da Variação da Taxa de Execução das Atividades Soft Real Time CBR.

A faixa de resposta dos experimentos foi de 12,5736 (14,588 a 183,424). Faixas superiores a 10 normalmente indicam que uma transformação é necessária. Após selecionar os fatores significantes (contribuição igual ou superior a 3%) e prosseguir com a análise dos resultados, com a visualização do gráfico *Box-Cox*, verificou-se que uma transformação logarítmica produziria um modelo mais adequado (sugestão da própria ferramenta estatística *Expert Design*).

Deste modo, a análise foi descartada e os resultados dos experimentos passaram pela transformação logarítmica (com  $k=0$ ), cuja equação é apresentada a seguir:

$$y' = \ln(y+k) \quad (59)$$

Após a transformação, uma nova análise foi realizada. As contribuições dos fatores sobre a Variação da Taxa de Execução de Atividades *soft real-time CBR*, ou simplesmente Variação de Execução *CBR*, são apresentadas na tabela 10.17. Foram escolhidos como fatores significantes aqueles com contribuição igual ou superior a 3%, ou seja, os fatores A e C e as interações entre os fatores AC, AD e CD. Esses fatores, juntos, explicam 82,23% da variação dessa métrica.

Além dos fatores considerados relevantes, apresentados acima, o fator D foi adicionado ao modelo para que esse se torne hierárquico, mesmo não tendo contribuição significativa.

Tabela 10.17 – Contribuição dos fatores sobre a variação de execução *CBR*

<b>Termo</b>	<b>Efeito</b>	<b>Contribuição (%)</b>
A	0,48	10,68
B	0,31	4,59
C	-0,94	41,06
D	0,12	0,65
E	-0,069	0,22
AB	0,065	0,20
AC	-0,59	16,29
AD	0,32	4,70
AE	-0,18	1,45
BC	0,047	0,10
BD	-0,056	0,15
BE	0,011	0,005
CD	0,45	9,50
CE	-0,14	0,93
DE	-0,19	1,76
ABC	0,065	0,20
ABD	0,066	0,20
ABE	-0,064	0,19
ACD	0,16	1,24
ACE	-0,001	0,00
ADE	-0,20	1,96
BCD	0,010	0,005
BCE	-0,026	0,032
BDE	-0,090	0,38
CDE	-0,11	0,52
ACBD	0,067	0,21
ABCE	-0,063	0,19
ABDE	-0,067	0,21
ACDE	-0,16	1,14
BCDE	-0,15	1,04
ABCDE	-0,068	0,22

A análise da variância ANOVA é apresentada nas tabelas 10.19 e 10.19. O valor da estatística F do modelo (20,16) indica que este é significativo. Os termos A, C, AC, AD e CD são considerados significantes (*'Prob>F'* menor que 0,05). Dos termos escolhidos inicialmente, com contribuição igual ou superior a 3%, nenhum foi considerado insignificante (*'Prob>F'* maior que 0,10). Os valores das colunas '*R-Squared Ajust.*', '*R-Squared Pred.*' e '*Precisão Adeq.*' mostram que o modelo é adequado.

O algoritmo de escalonamento, fator C, mostrou-se significativo, sendo o termo de maior contribuição para a Taxa de Variação de Execução CBR, com 41,06%. O segundo termo de maior importância (contribuição), foi a quantidade de NTs, fator A, com 10,68%, e o terceiro termo com maior contribuição foi a interação entre o escalonamento e o fator de utilização da CPU (CD), com 9,50%. A rede de interconexão (fatores B e E) não se mostrou muito significativa.

Esses resultados também são compreensíveis e eram, aproximadamente, esperados. Apenas algoritmos desenvolvidos especificamente com essa finalidade são capazes de manter constante a taxa de execução de atividades multimídia CBR (*Constant Bandwidth Rate*). Assim, o escalonador tem papel crucial no desempenho dessas atividades. Quanto maior a carga de trabalho (quantidade de processadores – NTs – e a utilização dessas CPUs) mais diferença faz o escalonador, o que justifica os outros termos de maior importância.

Já a rede de interconexão faz com que mensagens sejam transferidas mais rápida ou lentamente, mas não mantém sua taxa de execução constante. Obviamente, uma rede lenta pode fazer com que a taxa de execução desejada não seja alcançada, e uma rede rápida pode permitir a execução de atividades CBR numa taxa mais elevada que a desejada (se o escalonador não cuidar disso), o que justifica a contribuição de 4,59% para o *crossbar*, considerado estatisticamente significativo.

Tabela 10.18 – Resultados da ANOVA sobre a variação de execução *CBR*

Fonte	Soma dos Quadrados	Graus de Liberdade	Quadrado Médio	Valor de F	Prob>F
Modelo	14,14	6		20,16	< 0,0001
A	1,82	1	1,82	15,59	0,0006
C	7,01	1	7,01	59,93	< 0,0001
D	0,11	1	0,11	0,95	0,3379
AC	2,78	1	2,78	23,77	< 0,0001
AD	0,80	1	0,80	6,86	0,0148
CD	1,62	1	1,62	13,86	0,0010
Resíduo	2,92	25			
Corr. Total	17,07	31			

Tabela 10.19 – Estatísticas da ANOVA sobre a variação de execução *CBR*

Estatística	Valor
Média	3,95
Desvio-padrão	0,34
Coef. Variação	8,66
R-Squared Ajust.	0,7876
R-Squared Pred.	0,7194
Precisão Adeq.	12,351

A equação resultante da regressão linear (60), usada para prever os valores da Taxa de Variação de Execução *CBR*, é apresentada a seguir. Essa equação assume que os níveis dos fatores (variáveis da equação) estão codificados, ou seja, são sempre  $-1$  (nível mínimo) ou  $+1$  (nível máximo).

$$\begin{aligned} \ln(\text{VariacaoExecucaoCBR}) = & +3,95 + 0,24*A - 0,47*C + \\ & + 0,059*D - 0,29*A*C + 0,16*A*D + 0,23*C*D \end{aligned} \quad (60)$$

Para prever o Tempo de Resposta VBR utilizando outros níveis desses fatores, além dos níveis mínimo e máximo adotados, deve-se utilizar as equações (61) e (62), que aceitam quaisquer valores para os fatores (com exceção do algoritmo de escalonamento). São necessárias duas equações, pois o fator C, o algoritmo de escalonamento, é categórico, necessitando de uma equação para cada algoritmo.

Para o algoritmo de escalonamento PS:

$$\begin{aligned} \ln(\text{Variacao Execucao CBR}) = & \quad (61) \\ & +6,25010 + \\ & -0,039085 * \text{Nós de Trabalho} + \\ & -0,028652 * \text{Fator Utilização} + \\ & +8,79322E-004 * \text{Nós de Trabalho} * \text{Fator Utilização} \end{aligned}$$

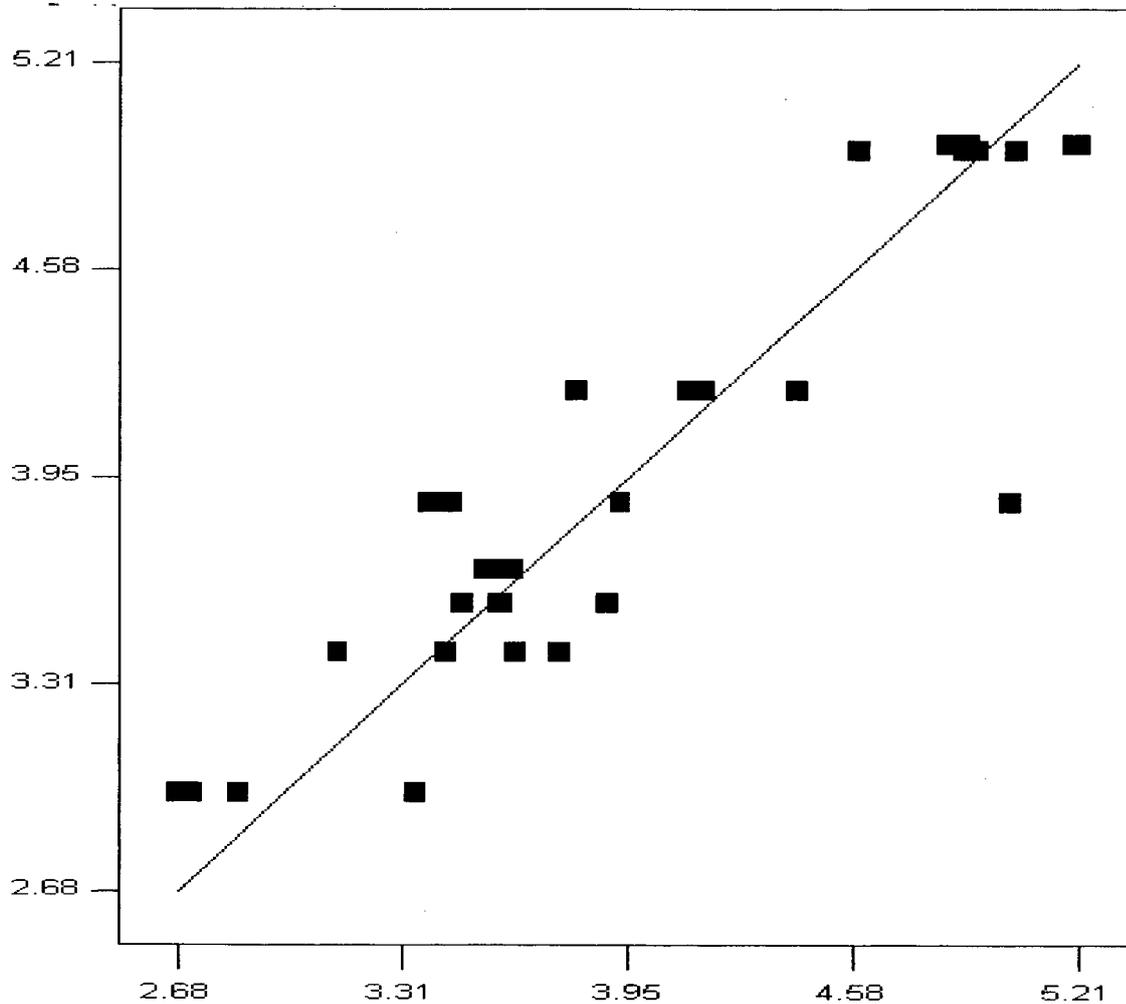
Para o algoritmo de escalonamento CBR:

$$\begin{aligned} \ln(\text{Variacao Execucao CBR}) = & \quad (62) \\ & +3,44594 + \\ & -0,088204 * \text{Nós de Trabalho} + \\ & +1,35427E-003 * \text{Fator Utilização} + \\ & +8,79322E-004 * \text{Nós de Trabalho} * \text{Fator Utilização} \end{aligned}$$

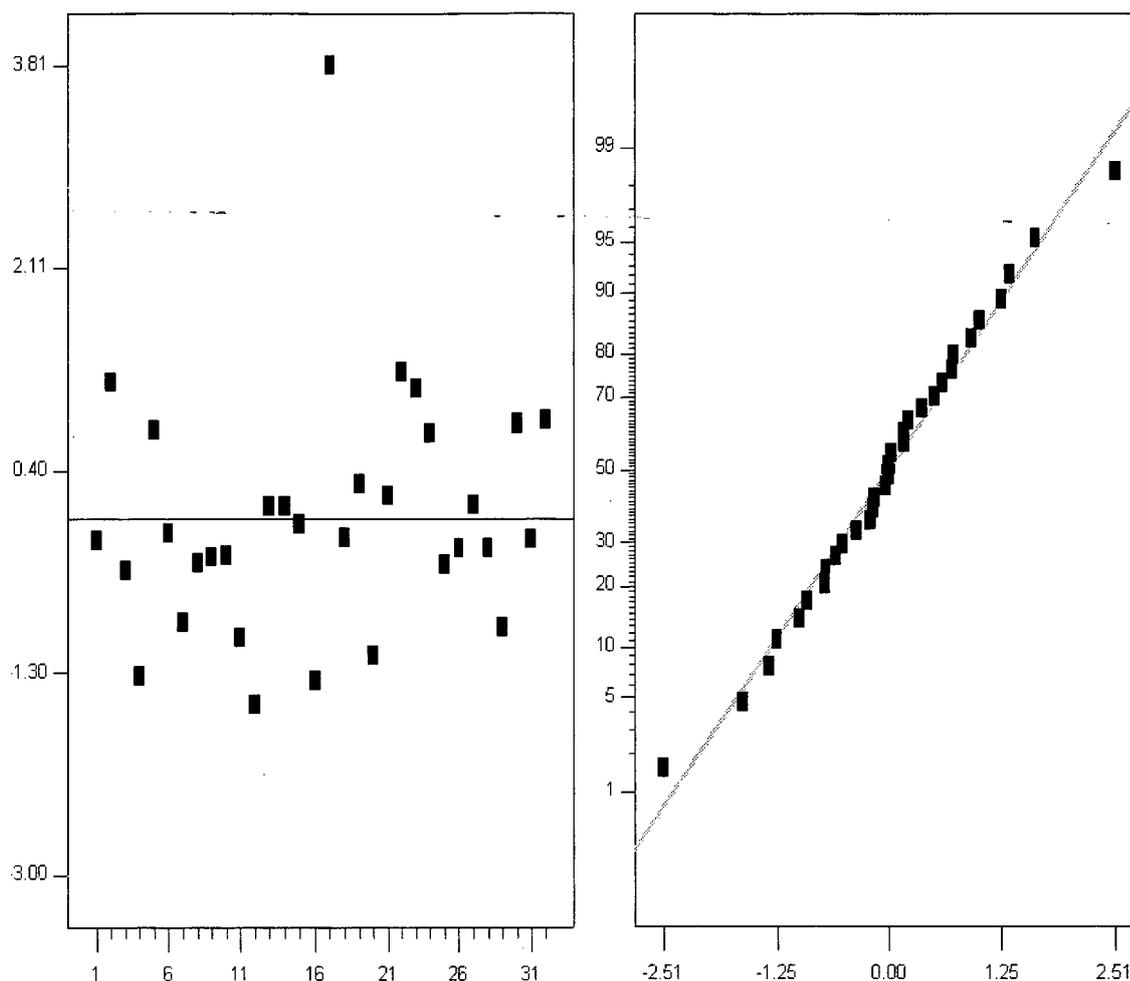
Os coeficientes da equação (60), estimados a partir de regressão linear, são o ponto central de uma estimativa intervalar, com nível de significância de 5% (erro-padrão de 0,060). O intervalo de confiança para cada um desses coeficientes é apresentado no ANEXO 14. Esse mesmo anexo apresenta também uma comparação (resíduos) dos resultados da simulação com os valores preditos pela equação. A figura

10.10 permite uma melhor visualização da relação entre os valores atuais (resultados da simulação) e a reta de valores preditos.

Figura 10.10 –Valores atuais x Valores preditos para a variação de execução *CBR*



A análise estatística realizada faz algumas suposições acerca dos dados, conforme foi apresentado no item 8.2.9. Assim, a figura 10.11 apresenta dois gráficos: um gráfico de resíduos (a) e um gráfico *quantile-quantile* (b), que permitem comprovar que os erros são estatisticamente independentes (sem tendências) e normalmente distribuídos (forma de “S”, concentrado em torno de  $y=50$ ).

Figura 10.11 – Gráficos de resíduo e *quantile-quantile* para variação de execução CBR

(a) Resíduos

(b) *Quantile*

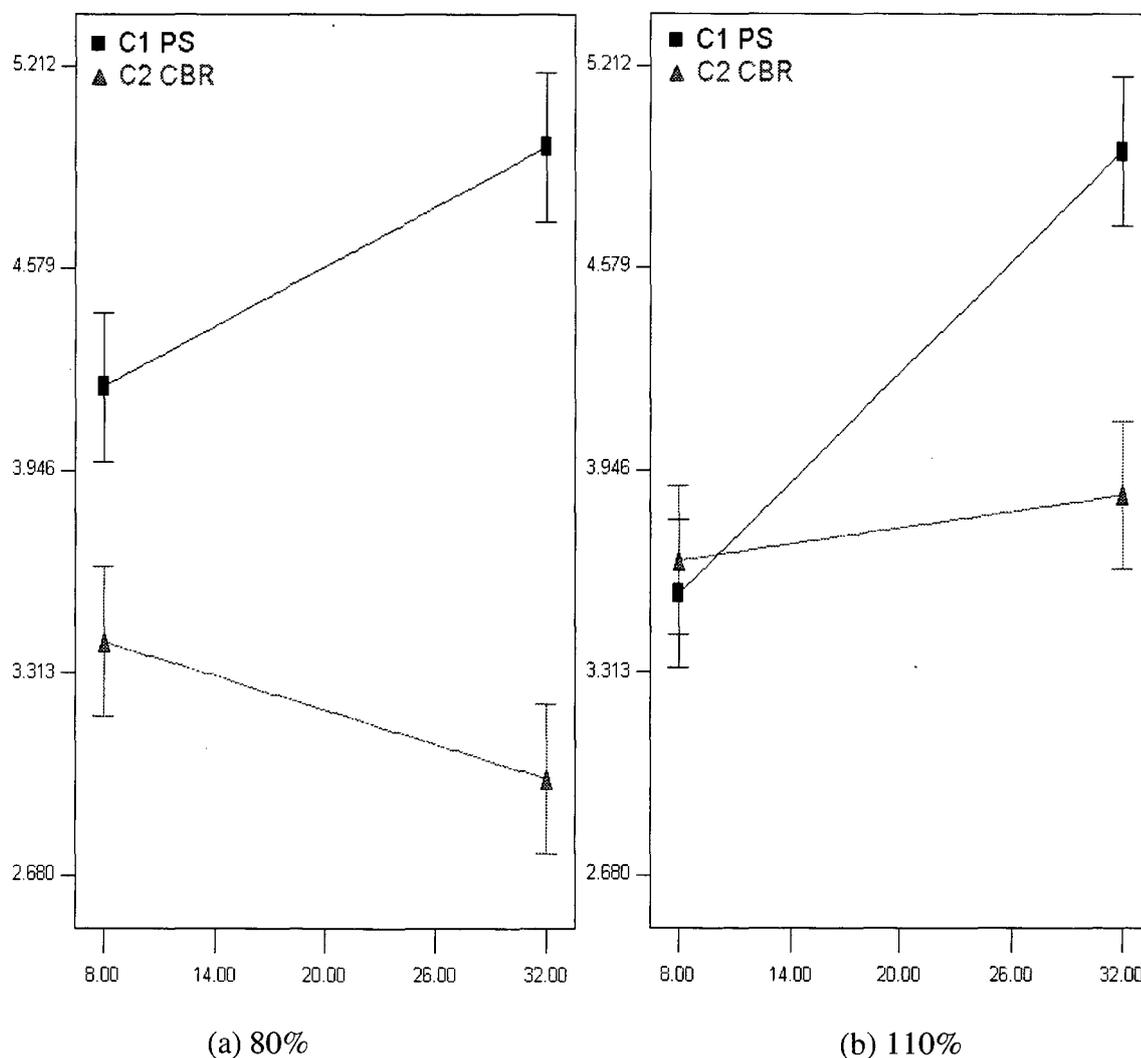
Na figura 10.12 são apresentados dois gráficos de interação, que apresentam o logaritmo natural da taxa de variação de execução de atividades *soft CBR* (devido à transformação realizada) no eixo das ordenadas em relação aos dois fatores de maior influência sobre essa métrica: a quantidade de NTs (fator A) no eixo das abscissas e o algoritmo de escalonamento (fator C) pelas duas curvas no gráfico, conforme a legenda apresentada. Cada gráfico dessa figura, (a) e (b), ilustra um fator de utilização da CPU. Os demais fatores foram mantidos em seu ponto médio.

Pode-se notar que o algoritmo CBR fornece resultados melhores para qualquer combinação de quantidade de NTs e utilização da CPU, exceto aparentemente para 8 NTs e utilização de 110% (à esquerda do gráfico b). Entretanto, como seus Intervalos de

Confiança estão praticamente sobrepostos, estatisticamente não se pode afirmar que o PS foi melhor que o CBR.

Também é perceptível que o algoritmo PS piora seu desempenho à medida que aumenta a quantidade de NTs, e piora com maior intensidade quando a utilização da CPU é maior (b). Já o escalonamento CBR manteve praticamente constante seu desempenho com alta utilização da CPU, independente da quantidade de NTs. Um efeito curioso foi a melhora do desempenho (menos variação) do escalonamento CBR, para uma utilização de 80% da CPU, à medida que se aumenta a quantidade de NTs. Isso pode ser devido à diferença da carga individual de cada processador nessas simulações.

Figura 10.12 – Gráficos de interação da variação de execução CBR



### 10.2.3. Análise do Tempo de Conexão de Todas as Atividades *Soft Real Time*.

A faixa de resposta dos experimentos foi de 56,2045 (229,27 a 12886). Faixas superiores a 10 normalmente indicam que uma transformação é necessária. Após selecionar os fatores significantes (contribuição igual ou superior a 3%) e prosseguir com a análise dos resultados, com a visualização do gráfico *Box-Cox*, verificou-se que uma transformação do tipo raiz quadrada (*square root*) produziria um modelo mais adequado.

Deste modo, a análise foi descartada e os resultados dos experimentos passaram pela transformação raiz quadrada (com  $k=0$ ), cuja equação é apresentada a seguir:

$$y' = \sqrt{y + k} \quad (63)$$

Após a transformação, uma nova análise foi realizada. As contribuições dos fatores sobre o Tempo de Conexão de todas as atividades são apresentadas na tabela 10.20. Foram escolhidos como fatores significantes aqueles com contribuição igual ou superior a 3%, ou seja, os fatores A, B e E, e as interações entre os fatores BD e BE. Esses fatores, juntos, explicam 85,75% da variação dessa métrica.

Além dos fatores considerados relevantes, apresentados acima, o fator D foi adicionado ao modelo para que esse se torne hierárquico, mesmo não tendo contribuição significativa.

Tabela 10.20 – Contribuição dos fatores e suas interações sobre o tempo de conexão

<b>Termo</b>	<b>Efeito</b>	<b>Contribuição (%)</b>
A	20,69	13,42
B	-27,90	24,39
C	-6,04	1,14
D	-5,37	0,90
E	-34,31	36,89
AB	-6,66	1,39
AC	-1,82	0,10
AD	-6,26	1,23
AE	-1,93	0,12
BC	2,64	0,22
BD	11,50	4,14
BE	-14,85	6,91
CD	-3,33	0,35
CE	2,27	0,16
DE	-6,85	1,47
ABC	-0,25	0,002
ABD	4,84	0,73
ABE	-3,61	0,41
ACD	-3,81	0,45
ACE	0,12	0,0004
ADE	-5,21	0,85
BCD	7,57	1,80
BCE	-1,68	0,089
BDE	-5,08	0,81
CDE	0,31	0,003
ACBD	1,80	0,10
ABCE	0,31	0,003
ABDE	0,50	0,007
ACDE	4,77	0,71
BCDE	-5,52	0,96
ABCDE	-2,74	0,24

A análise da variância ANOVA é apresentada nas tabelas 10.21 e 10.22. O valor da estatística F do modelo (27,06) indica que este é significativo. Os termos A, B, E, BD e BE são considerados significantes (*'Prob>F'* menor que 0,05). Dos termos escolhidos inicialmente, com contribuição igual ou superior a 3%, nenhum foi considerado insignificante (*'Prob>F'* maior que 0,10). Os valores das colunas '*R-Squared Ajust.*', '*R-Squared Pred.*' e '*Precisão Adeq.*' mostram que o modelo é adequado.

O algoritmo de escalonamento, fator C, mostrou-se estatisticamente insignificante, com contribuição de apenas 1,14% para a determinação dessa métrica, tendo sido portanto, desconsiderado no modelo. A velocidade da rede *crossbar* (fator E) mostrou-se o fator mais influente, com 36,89%, seguido pela quantidade de Canais Físicos (fator B, 24,39%), um fator intimamente relacionado com o primeiro. O terceiro fator que mais contribui com a determinação do tempo de conexão é a quantidade de NTs (fator A, 13,42%).

Tabela 10.21 – Resultados da ANOVA sobre o tempo de conexão

Fonte	Soma dos Quadrados	Graus de Liberdade	Quadrado Médio	Valor de F	Prob>F
Modelo	2212,59	6	3687,10	27,06	< 0,0001
A	3425,27	1	3425,27	25,14	< 0,0001
B	6226,49	1	6226,49	45,70	< 0,0001
D	230,60	1	230,60	1,69	0,2051
E	9417,74	1	9417,74	69,13	< 0,0001
BD	1057,97	1	1057,97	7,77	0,0100
BE	1764,52	1	1764,52	12,95	0,0014
Resíduo	3405,89	25	136,24		
Corr. Total	25528,47	31			

Tabela 10.22 – Estatísticas da ANOVA sobre o tempo de conexão

Estatística	Valor
Média	58,83
Desvio-padrão	11,67
Coef. Variação	19,84
R-Squared Ajust.	0,8346
R-Squared Pred.	0,7814
Precisão Adeq.	17,292

A equação resultante da regressão linear (64), usada para prever os valores do Tempo de Conexão, é apresentada a seguir. Essa equação assume que os níveis dos fatores (variáveis da equação) estão codificados, ou seja, são sempre  $-1$  (nível mínimo) ou  $+1$  (nível máximo).

$$\begin{aligned} \text{Sqrt}(\text{TempoConexao}) = & +58,83 + 10,35*A - 13,95*B + \\ & -2,68*D - 17,16*E + 5,75*B*D - 7,43*B*E \end{aligned} \quad (64)$$

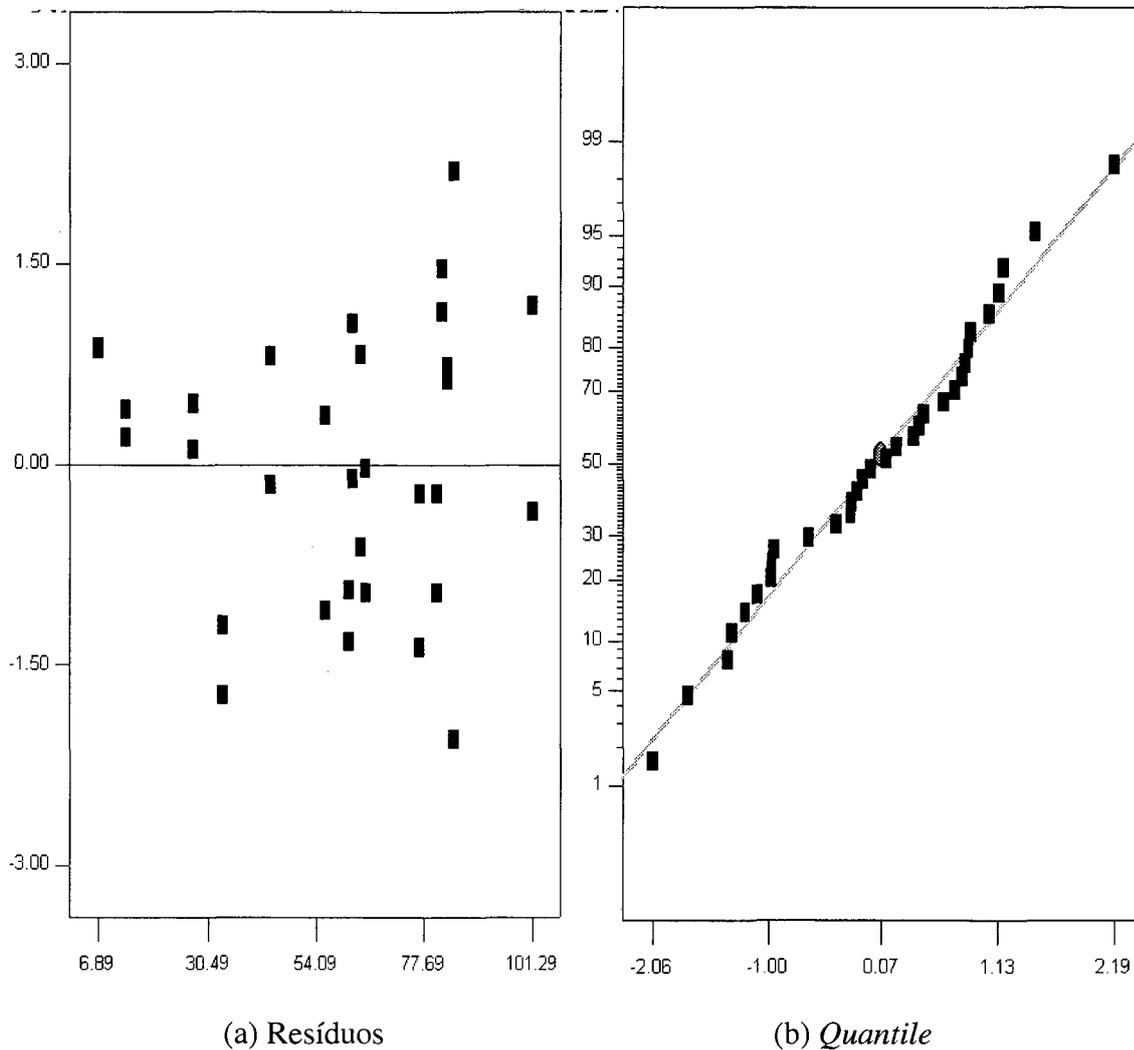
Para prever o Tempo de Conexão utilizando outros níveis desses fatores, além dos níveis mínimo e máximo adotados deve-se utilizar a equação (65), que aceita quaisquer valores para os fatores.

$$\begin{aligned} \text{Sqrt}(\text{Tempo Conexao}) = & \\ & +147,40643 + \\ & +0,86217 * \text{Nós de Trabalho} + \\ & -28,52638 * \text{Canais Físicos} + \\ & -0,81784 * \text{Fator Utilização} + \\ & -0,096548 * \text{Velocidade Crossbar} + \\ & +0,25555 * \text{Canais Físicos} * \text{Fator Utilização} + \\ & -0,10001 * \text{Canais Físicos} * \text{Velocidade Crossbar} \end{aligned} \quad (65)$$



*quantile* (b), que permitem comprovar que os erros são estatisticamente independentes e normalmente distribuídos.

Figura 10.14 – Gráficos de resíduo e *quantile-quantile* para o tempo de conexão

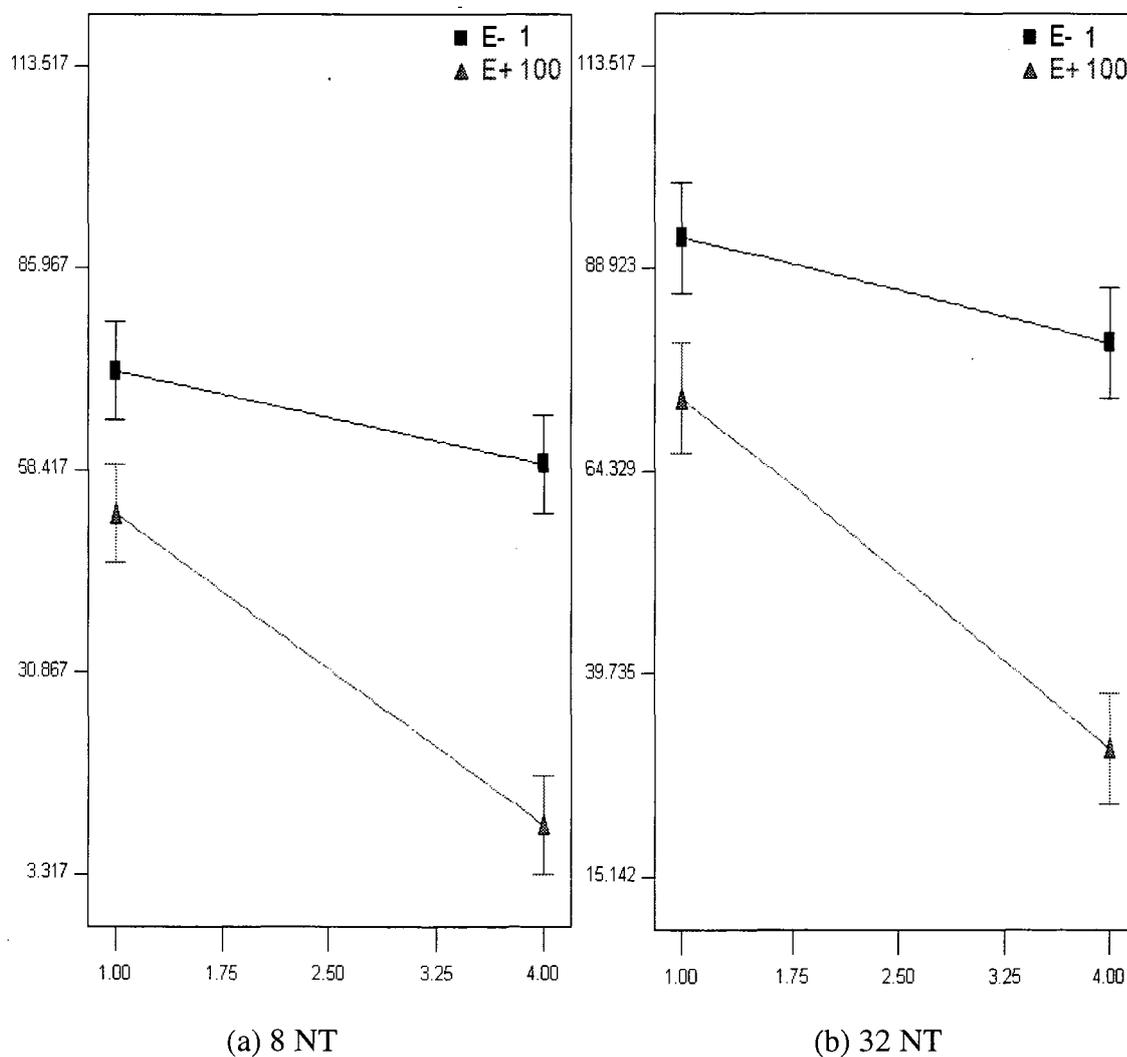


Na figura 10.15 são apresentados dois gráficos de interação, que apresentam a raiz quadrada do tempo de conexão de todas as atividades *soft* (devido à transformação realizada) no eixo das ordenadas em relação aos dois fatores de maior influência sobre essa métrica: a quantidade de canais físicos (fator B) no eixo das abscissas e a velocidade do *crossbar* (fator E) pelas duas curvas no gráfico, conforme a legenda

apresentada. Cada gráfico dessa figura, (a) e (b), ilustra uma quantidade de NTs. Os demais fatores foram mantidos em seu ponto médio.

Percebe-se, inicialmente, que o aumento na quantidade de NTs causa um aumento no tempo de conexão, sem, entretanto, modificar as tendências dos outros fatores, ou seja, sua interação com eles é insignificante. Conforme se aumenta a quantidade de canais físicos, também se aumenta proporcionalmente a influência da velocidade desses canais (do *crossbar*). Essa proporção está, obviamente, relacionada com a velocidade (inclinação das curvas em cada gráfico), uma vez que esses fatores estão intimamente relacionados.

Figura 10.15 – Gráficos de interação do o tempo de conexão



## 11. Conclusões

Este trabalho envolveu as áreas de simulação de sistemas, sistemas operacionais, sistemas de tempo real, estatística e avaliação de desempenho, sendo a quinta dissertação sobre simulação do ambiente CRUX desenvolvida no CPGCC. Todos os pontos levantados nos trabalhos predecessores, relacionados com o CRUX, e listados na Introdução, foram considerados e implementados no presente trabalho, representando um avanço do projeto deste ambiente paralelo.

O trabalho envolveu um estudo sobre a arquitetura do ambiente CRUX e seu funcionamento, bem como a caracterização da carga de trabalho para este ambiente, sendo escolhidas aplicações multimídia.

Envolveu também um estudo sobre escalonadores de tempo-real encontrados na bibliografia e principalmente em publicações em congressos e revistas científicos relevantes. Foram constatadas a variedade de abordagens existentes e a especificidade dos mesmos, pois têm seu escopo muito reduzido. Esse estudo permitiu a escolha dos mais relevantes algoritmos para que fossem então modelados e simulados.

O estudo a respeito da análise estatística e da avaliação de desempenho permitiu uma validação das restrições estatísticas impostas (distribuição dos erros, independência entre amostras, etc), a obtenção de intervalos de confiança para estimativa dos parâmetros de desempenho do sistema com 95% de certeza e um semi-intervalo menor que 10% de sua média, o que fornece uma precisão adequada aos resultados. Também permitiu a escolha de um projeto de experimentos adequado que permitiu a extração do modelo de todos os resultados dele desejados.

O modelo desenvolvido representou o ambiente CRUX com um nível de detalhamento consideravelmente superior a todos os modelos anteriores, uma vez que considerou aspectos da rede de interconexão e da rede de serviço, do sistema operacional, de sincronismo e de tempo-real.

O modelo cumpriu seu objetivo por ser flexível e reusável, sendo interfaceado por uma aplicação VBA que permite a configuração da simulação e o ajuste dos parâmetros do modelo sem a necessidade de acessar o modelo na ferramenta Arena.

Além da interface, boa parte da lógica também foi implementada em VBA, o que tornou o modelo mais simples e também permite a alteração diretamente no código-fonte, sendo possível alterar o modo de alocação dos canais físicos, os algoritmos de escalonamento e outros comportamentos sem alterações no modelo Arena.

O algoritmo de escalonamento no NC tem algum efeito no desempenho do sistema apenas quando há a formação de filas. Isso ocorre quando há muita comunicação através da rede de interconexão e a mesma não possui vazão suficiente. Nas demais simulações, o efeito do escalonamento foi perceptível apenas nos Nós de Trabalho.

Foi verificada a influência do algoritmo de escalonamento de tempo-real sobre a taxa de variação de execução de atividades de tempo-real *soft* CBR, característica de aplicações multimídia, e representadas nesse estudo pela transmissão de áudio sem compactação. O escalonamento mostrou ser, sob as condições simuladas, o fator de maior influência sobre essa métrica, seguido pela quantidade de processadores o fator de utilização dos mesmos.

A influência do escalonamento e tempo-real mostrou-se ainda significativa para a determinação do tempo de resposta das atividades *soft* VBR, representadas pelo acesso a páginas *web*. Essa influência, entretanto, revelou-se pequena, sendo as características da rede de interconexão (velocidade e quantidade de canais físicos) os fatores mais significativos.

Em relação à terceira métrica de desempenho avaliada neste estudo, o tempo de conexão de atividades *soft* (tanto VBR quanto CBR), sob as condições simuladas, o algoritmo de escalonamento mostrou ser estatisticamente insignificante. Novamente, as características da rede de interconexão foram os fatores mais influentes.

Foi comprovado o efeito de inversão de prioridade, que é provocado por critérios inadequados de alocação de recursos para um sistema de tempo-real. Tal efeito foi o principal motivo para a representação de tarefas *hard* que executam apenas localmente, sem utilizar a rede de interconexão, conforme explicado no capítulo 9. Essa representação eliminou a inversão de prioridade pela eliminação de requisições com prioridades diferentes (*hard* e *soft*) para alocação dos recursos denominados canais físicos.

Isso significa que as condições necessárias para causar o problema não existem, mas o problema não foi eliminado. O modelo não representa nenhum algoritmo de alocação de recursos adequado às garantias de tempo-real, uma vez que não fazia parte do escopo deste trabalho. Esse efeito indesejável pode se manifestar caso futuras simulações utilizando esse modelo sejam feitas incluindo o uso da rede de interconexão também para atividades *hard real-time*.

## 12. Considerações Finais

Algoritmos com *feedback* e realimentação, inspirados na teoria de controle, fornecem ainda uma vasta fonte de estudos, uma vez que vários indicadores de desempenho podem ser usados como critério para mudanças nos parâmetros do escalonador, e ainda podem ser usadas diversas funções de controle que avaliem os indicadores escolhidos.

Uma dificuldade significativa foi a indisponibilidade periódica da ferramenta de simulação Arena. Na época da realização deste trabalho houve problemas com a chave de *hardware* da versão 3.0 da ferramenta Arena, a mudança para a nova versão 4.0 e remodelagem do sistema. Já com a versão 4.0, a necessidade de utilizar a ferramenta exclusivamente no Laboratório do CPGCC (causada pelo novo método de Licença em rede dessa versão) foi outro empecilho.

Ouros fatores, como a localização da máquina que contem a Licença na rede, dificultaram a utilização da ferramenta. Devido a essas dificuldades, optou-se, como última escolha, desenvolver o modelo novamente na versão 3.0 da ferramenta Arena, pois tendo restrições mais flexíveis que a versão 4.0, possibilitou seu uso mesmo sem a Licença.

Uma série de pequenos problemas conjuntos, envolvendo a lógica do Arena e a programação VBA se revelaram apenas em longas simulações, com muitas entidades de todos os tipos, quando a concorrência e uso dos recursos era intensa. Vários problemas de modelagem desse tipo nunca ocorreriam num sistema com *workload* baixo, e não foram detectados inicialmente, durante as simulações de testes. Tornou-se necessário, então, em vários momentos, novos refinamentos do modelo, aumentando um pouco sua complexidade.

Uma última dificuldade foi o aumento indesejado da quantidade de experimentos necessários previsto inicialmente pelo projeto experimental, devido a ajustes no tempo de simulação e na quantidade de replicações de tal forma a obter intervalos de confiança aceitáveis e resultados interessantes. A execução exaustiva de

testes e ajustes dos valores dos parâmetros também demandou tempo excedente em simulações.

Propõe-se uma avaliação de outras restrições de tempo-real, como a alocação de recursos e o tratamento de interrupções, que não foram incorporados nesse estudo. O principal recurso envolvido no gerenciamento de comunicação no ambiente estudado é a rede de interconexão, e métodos especiais para sua alocação devem ser acrescentados. As interrupções são eventos inesperados, que levam a indeterminismos no tempo de computação. Assim, o tratamento de interrupções, quando simulado, deve ser agregado ao conjunto de tarefas e alocado conforme o escalonamento de tempo-real utilizado.

Outras propostas para a continuação desse projeto não envolvem mais o uso de simulação, mas sim a construção física do ambiente CRUX. Os resultados desse e de outros estudos já realizados pelo CPGCC da UFSC dão base teórica provavelmente suficiente para que tal ambiente seja implementado com sucesso, sendo ele próprio, base para futuros projetos e aplicações paralelas e distribuídas.

### 13. Referência Bibliográfica

1. ABENI, Luca; BUTTAZZO, Giorgio. Integrating Multimedia Applications in Hard Real-Time Systems. Madrid, Spain: *Proceedings of the IEEE Real-Time Systems Symposium, December 1998*.
2. \_\_\_\_\_; BUTTAZZO, Giorgio. Adaptive Bandwidth Reservation for Multimedia Computing. Hong Kong, China: *Proceedings of the IEEE Real-Time Systems Symposium, December 1999*.
3. ANCILIOTTI, P.; BUTTAZZO, G.C.; NATALE, M.; BIZZARRI, M. A Flexible Tool Kit for Development of Real-Time Applications. *Proceedings of the IEEE Real-Time Technology and Application Symposium, Brookline, pp. 260-262, June 1996*.
4. ATHAS, William C.; Seitz, Charles L. Multicomputers: Message-Passing Concurrent Computers. *IEEE Computer, August 1988*.
5. BARBETTA, P. A. Estatística Aplicada às Ciências Sociais. Florianópolis – SC: Editora da UFSC, 1994.
6. BARROS Neto; Benício, SCARMINIO; Ieda S., BRUNS, Roy E. Planejamento e Otimização de Experimentos. 2ª ed. Campinas – SP: Editora da Unicamp, 1995.
7. BUSSAB, Wilton O.; MORETTIN, Pedro A. Estatística Básica. 4ª ed. São Paulo – SP: Atual Editora Ltda, 1995.
8. BUTTAZZO, Giorgio; SENSINI, Fabrizio. Optimal Deadline Assignment for Scheduling Soft Aperiodic Tasks In Hard Real-Time Environments. *IEEE Transactions on Computers, Vol. 48, No. 10, October 1999*.
9. \_\_\_\_\_. Hard Real-Time Computing Systems. Dordrecht, Netherlands: Kluwer Academic Publishers, 1997.

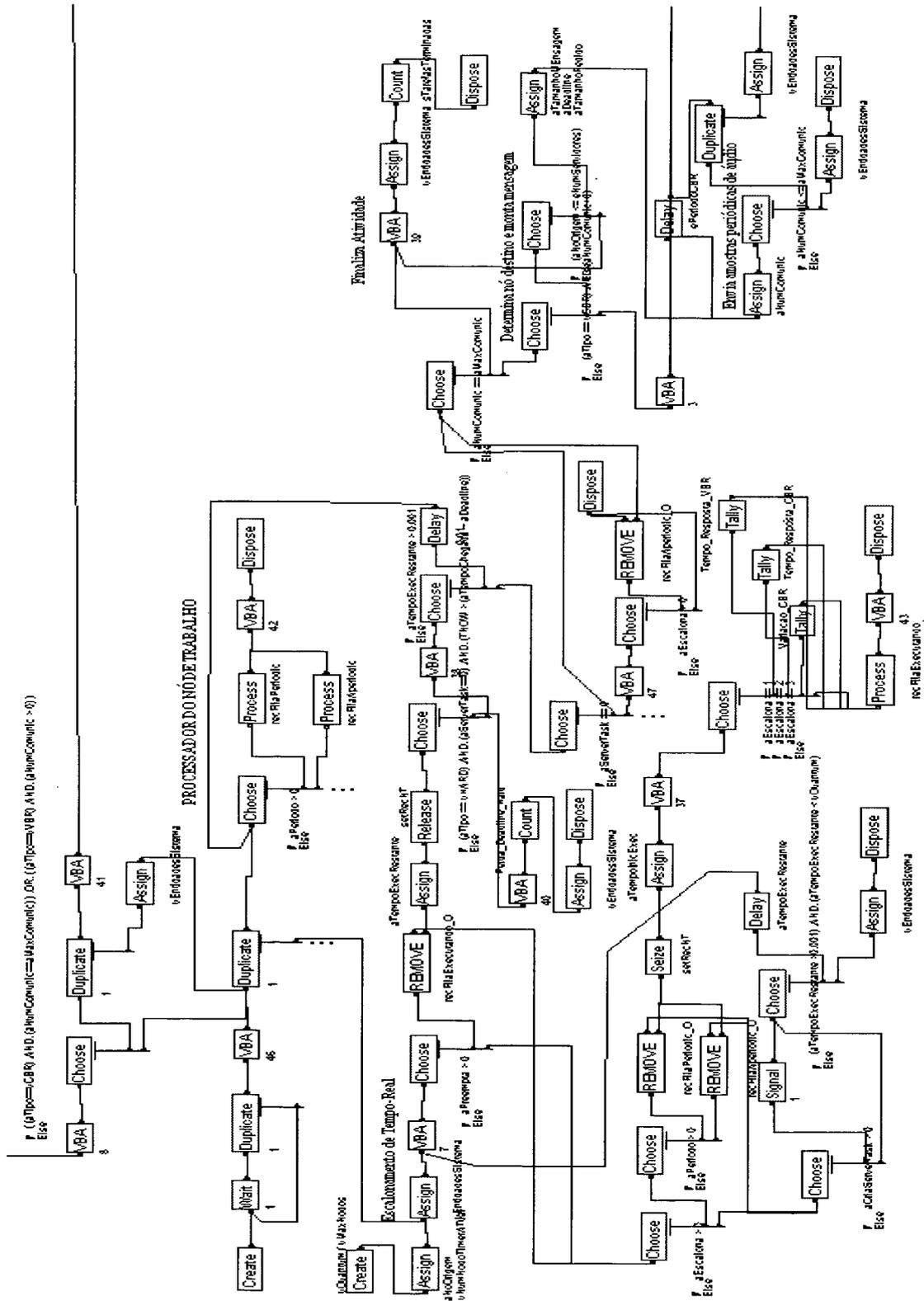
10. \_\_\_\_\_.; STANKOVIC, John A. Adding Robustness in Dynamic Preemptive Scheduling. in “*Responsive Computer Systems: Steps Toward Fault-Tolerant Real-Time Systems*”, edited by D. S. Fussell and M. Malek, Kluwer Academic Publishers, Boston, 1995.
11. CORRÊA, Edgard de Faria. Aplicações de Tempo Real em um Ambiente Baseado em Multicomputador: Serviços de Suporte e Avaliação de Desempenho. Florianópolis – SC: Dissertação de Mestrado, CPGCC, UFSC, 1998.
12. CORSO, Thadeu B., Ambiente para Programação Paralela em Multicomputador. Florianópolis – SC: Relatório técnico n.1, INE-UFSC, Novembro de 1993.
13. COULOURIS, George, DOLLIMORE, Jean, KINDBERG, Tim. Distributed Systems – Concepts and Design. 4th impression. Harlow – England: Addison Wesley Longman Limited, 1995.
14. FENG, Tse-yun. A Survey of Interconnection Networks. *IEEE Computer*, v .12 n. 14, December 1981; pp 12-27
15. FREITAS Filho, Paulo J. Modelagem, Projeto e Avaliação de Desempenho de Redes. Florianópolis – SC: Instituto de Estudos Superiores - CPGCC – UFSC, Abril de 2000.
16. JAIN, Raj. The Art of Computer Systems Performance Analysis. New York, US: John Wiley & Sons, Inc, 1991.
17. KELTON, W.; SADOWSKI, Randall; SADOWSKI, Deborah. Simulation with Arena. WCB/McGraw-Hill, 1998
18. LEVINE, David; BERENSON, Mark; STEPHAN, David. Estatística: Teoria e Aplicações. Rio de Janeiro – RJ: LTC – Livros Técnicos e Científicos Editora S.A, 2000.
19. LIPARI, Giuseppe; BUTTAZZO, Giorgio; ABENI, Luca. A Bandwidth Reservation Algorithmic for Multi-Application Systems. Hiroshima, Japan: *Proceedings of the IEEE Conf. on Real-Time Computer Systems and Application*, October 1998.

20. MENDENHALL, William. Probabilidade e Estatística. Rio de Janeiro – RJ: Editora Campus Ltda, 1985.
21. MONTGOMERY, Douglas C. Design and Analysis of Experiments. Tempe, Arizona - USA: 4<sup>th</sup> edition, John Wiley & Sons, Arizona State University, 1997.
22. NATALE, M.; ANCIOTTI, P.; BUTTAZZO, G.; BIZZARRI, M. The MORRIS Control System. L'Aquila, Italy: *Proceedings of the 8<sup>th</sup> IEEE Euromicro Workshop on Real-Time Systems*, pp 77-82, June 1996.
23. NAYLOR, Thomas H.; BALITFY, Joseph L.; BURDICK, Donald S.; et al. Técnicas de Simulação em Computadores. São Paulo: Editora da Universidade de São Paulo, 1966.
24. PATTERSON, David A.; HENNESSY, John L. Computer Architecture. A Quantitative Approach. San Francisco, California, US: Morgan Kaufmann Publishers, Inc, 1996.
25. \_\_\_\_\_.; HENNESSY, John L. Computer Organization & Design. The Hardware / Software Interface. San Francisco, California, US: Morgan Kaufmann Publishers, Inc, 1998.
26. PEGDEN, D. C. et al. Introduction to Simulation Using Siman. Second Edition, WcGraw-Hill, 1995.
27. REIS, Marcelo Menezes. Estatística para Ciências Biológicas: Notas de Aula. Florianópolis – SC: Apostila da disciplina INE5117. Depto INE – UFSC, 1998.
28. ROSS, P.J. Tagushi Techniques for quality engineering. New York, Mcgraw-Hill, 1988.
29. SENSINI, Fabrizio; BUTTAZZO, Giorgio; ANCILOTTI, Paolo. GHOST: A Tool for Simulation and Analysis of Real-Time Scheduling Algorithms. Montreal, Canada: *Proceedings of the IEEE Real-Time Educational Workshop (RTEW'97)*, pp. 42-49, June 1997.

30. SHA, L., RAJKUMAR, R. e LEHOCSKY, J. P. Priority Inheritance Protocol: An approach to real-time systems. *IEEE Transactions on Computers*, 39(9), september, 1990.
31. SILBERSCHATZ, Abraham; GALVIN, Peter B. Sistemas Operacionais – conceitos. São Paulo, SP: Ed. Prentice-Hall do Brasil LTDA, 2000.
32. (SPEC 2000) Disponível na Internet em: <http://www.specbench.org/spec/> ; Capturado em 18/08/2000, 02:12:08
33. (SPECWEB2000). Disponível na Internet no endereço eletrônico: <http://open.spec.org/osg/web96/workload.html> ; Capturado em 16/07/2000, 23:45:57
34. SPIEGEL, Murray Ralph. Schaum's Outline of Theory and Problems of Probability and Statistics. São Paulo – SP: McGraw-Hill do Brasil, 1977.
35. SPURI, Marco; BUTTAZZO, Giorgio; SENSINI Fabrizio. Robust Aperiodic Scheduling under Dynamic Priority Systems. *Pisa, Italy: Proceedings of the 16<sup>th</sup> IEEE Real-Time System Symposium (RTSS 95)*, pp. 210-219, December 1995.
36. STANKOVIC, John A. What is a Real-Time System. *Amherst, Massachusetts: Department of Computer Science, University of Massachusetts, April 16, 1992*.
37. \_\_\_\_\_ . Distributed Real-Time Computing: The Next Generation. *Amherst, Massachusetts: Department of Computer Science, University of Massachusetts, January 3, 1992*.
38. TAKUS, David A.; PROFOZICH, David M. Arena Software Tutorial. *Proceedings of the 1997 Winter Simulation Conference: Ed. S Andradottir, K. Healy, D. Withers and B. Nelson, 1997*.
39. TANENBAUM, Andrew S. Sistemas Operacionais Modernos. Travessa do Ouvidor, 11, Rio de Janeiro - RJ: Editora Prentice-Hall do Brasil LTDA, 1995.

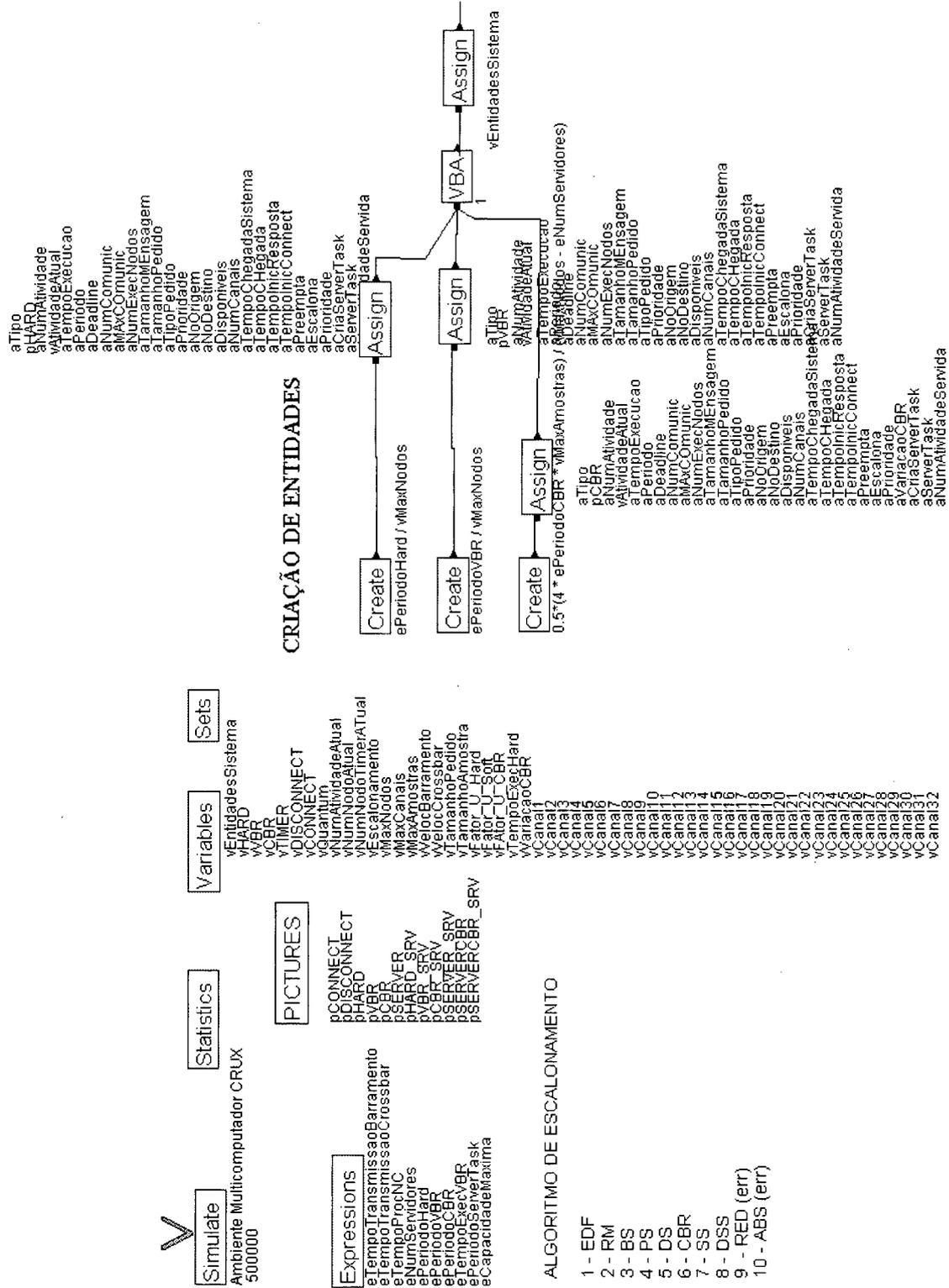
40. WILLRICH, Roberto. Sistemas Multimídia. Florianópolis, SC: INE - UFSC, Junho de 2000.

# ANEXO 1 – Modelo Arena dos Nós de Trabalho (NTs).

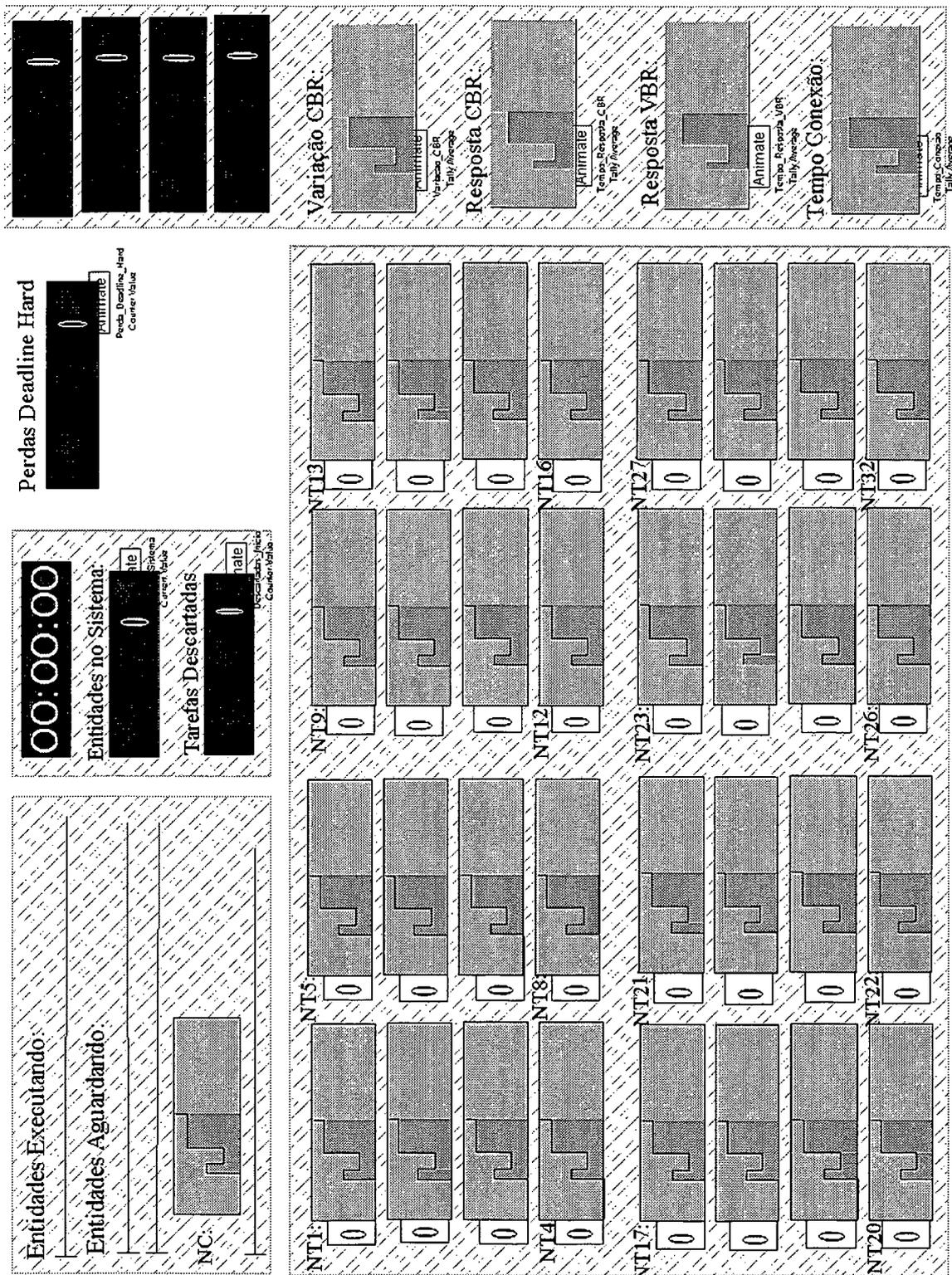




# ANEXO 3 – Modelo Arena para Criação de Entidades e Declarações.



# ANEXO 4 – Animação do Modelo Arena para Verificação Visual.



## ANEXO 5 – Código-fonte VBA, parte integrante do modelo.

### Arquivo "ThisDocument.cls"

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1 'True
END
Attribute VB_Name = "ThisDocument"
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = True
Option Explicit

Private Sub ModelLogic_RunBeginSimulation()
  'inicializa modelo

  'inicializa objetos do modelo
  Set M = ThisDocument.Model
  Set S = M.SIMAN

  'inicializa novo arquivo de log
  Open "C:\CRUX_LOG.TXT" For Output As #1
  Print #1, "MODELO DO MULTICOMPUTADOR CRUX - CPGCC - UFSC"
  Print #1, "Avaliação de Desempenho de Algoritmos de Escalonamento de Tempo Real"
  Print #1, ""
  'Print #1, ""
  Close #1

  Open "C:\DIFERENCAS_CBR.TXT" For Output As #1
  Print #1, "MODELO DO MULTICOMPUTADOR CRUX - CPGCC - UFSC"
  Print #1, "Avaliação de Desempenho de Algoritmos de Escalonamento de Tempo Real"
  Print #1, "Diferenças entre instantes de início de execução de amostars soft CBR"
  Close #1

  'inicializa variáveis gerais
  Dim i As Integer, j As Integer, escalonamento As Double, Cmax As Double
  escalonamento = Get_Variavel("vEscalonamento")
  Cmax = Get_Expressao("eCapacidadeMaxima")
  For i = 1 To 32
    For j = 1 To 8
      gCanalFisico(i, j) = 0
      gUltimaExecucaoCBR(i, j, 1) = 0
      gUltimaExecucaoCBR(i, j, 2) = 0
    Next j
    gUltimaAtivacao(i) = -65535 'um número negativo para ativar o servidor de tarefas aperiódicas já no
tempo 0
    If escalonamento = 7 Then
      'escalonamento SS começa com a Capacidade Maxima
      gCapacidadeServidor(i) = Cmax
    Else
      gCapacidadeServidor(i) = 0
    End If
  Next i
End Sub
```

```

gUltimoConsumo(i) = 0
gIDAperiodicServida(i) = 0
gServerTaskAssociada(i) = 0
gConsumido(i) = 0
gDSSAtivo(i) = 0
For j = 1 To 16
    gRT(i, j) = 0
    gRA(i, j) = 0
Next j
Next i
gCabecalho = True
End Sub

```

```

Private Sub VBA_Block_1_Fire()
    Dim servidores As Double, nodoAtual As Double, maxNodos As Double
    Dim atividade As Double

```

```

' INICIALIZAÇÃO DA TAREFA

```

```

'pega valores do modelo

```

```

servidores = Get_Expressao("eNumServidores")

```

```

maxNodos = Get_Variavel("vMaxNodos")

```

```

If Get_Atributo("aTipo") = Get_Variavel("vHARD") Then

```

```

    'cria tarefa HARD sequencialmente em todos os nós

```

```

    nodoAtual = Get_Variavel("vNumNodoAtual")

```

```

    Set_Atributo "aNoOrigem", nodoAtual

```

```

    If nodoAtual < maxNodos Then

```

```

        nodoAtual = nodoAtual + 1

```

```

    Else

```

```

        nodoAtual = 1

```

```

    End If

```

```

    Set_Variavel "vNumNodoAtual", nodoAtual

```

```

Else

```

```

    'escolhe aleatoriamente um nó cliente para a tarefa SOFT

```

```

    Randomize

```

```

    nodoAtual = 0

```

```

    While (nodoAtual <= servidores) Or (nodoAtual > maxNodos)

```

```

        nodoAtual = Int(Rnd() * maxNodos) + 1

```

```

    Wend

```

```

    Set_Atributo "aNoOrigem", nodoAtual

```

```

End If

```

```

'atualiza tempo de chegada no sistema

```

```

Set_Atributo "aTempoChegadaSistema", S.RunCurrentTime

```

```

'incrementa número da atividade

```

```

atividade = Get_Variavel("vAtividadeAtual")

```

```

Set_Atributo "aNumAtividade", atividade

```

```

Set_Variavel "vAtividadeAtual", (atividade + 1)

```

```

    Registra_Log True, "Inicializa Tarefa. NT" + Str(Get_Atributo("aNoOrigem")) + ", Deadline " +
    Str(Get_Atributo("aDeadline"))
End Sub

```

```

Private Sub VBA_Block_3_Fire()

```

```

    Dim numComunic As Double, numServidores As Double

```

```

    Dim noDestino As Double, noOrigem As Double

```

```

    Dim aux As Double, tipo As Double, classe As Double

```

```

    Dim tamanhoMensagem As Double, tamanhoPedido As Double

```

## ' TAREFA MONTA PEDIDO DE CONEXÃO

Randomize

'pega valores

noOrigem = Get\_Atributo("aNoOrigem")

numServidores = Get\_Expressao("eNumServidores")

numComunic = Get\_Atributo("aNumComunic")

noDestino = Get\_Atributo("aNoDestino")

tipo = Get\_Atributo("aTipo")

tamanhoPedido = Get\_Variavel("vTamanhoPedido")

If noOrigem > numServidores Then

'Está executando no cliente

'Gera um nome número de NT destino (um servidor)

Do

noDestino = Int(Rnd() \* numServidores) + 1

Loop While (noDestino < 1) Or (noDestino > numServidores) Or (noDestino = noOrigem)

Set\_Atributo "aNoDestino", noDestino

'se está executando no cliente então é uma solicitação de página web (VBR) ou de uma amostra de áudio (CBR)

'ajusta o tamanho que a mensagem deve ter

If tipo = Get\_Variavel("vVBR") Then

'requisição de página web

tamanhoMensagem = tamanhoPedido

Else

'requisição de amostra de áudio

tamanhoMensagem = tamanhoPedido

End If

Else

'está executando no servidor

'é uma atividade VBR ou CBR executando no servidor

'o NT destino é o cliente que enviou esse pedido. Inverte Origem com Destino

'se está executando no servidor, então vai enviar uma mensagem de resposta ao cliente

If tipo = Get\_Variavel("vVBR") Then

'está devolvendo uma página da web (tamanho deve seguir SPECweb96)

'as distribuicoes abaixo geram o tamanho das páginas web (conforme SPECweb96)

Dim prob(1 To 8) As Double

prob(1) = 0.35 '35% para arquivos muito pequenos < 1KB

prob(2) = 3

prob(3) = prob(1) + 0.5 '50% para arquivos de 1KB a 10KB

prob(4) = 2

prob(5) = prob(3) + 0.14 '14% para arquivos entre 10KB e 100KB

prob(6) = 1

prob(7) = 1 'O restante (1%) para arquivos entre 100KB e 1MB

prob(8) = 0

'faz sorteio da classe do tamanho do arquivo que será solicitado

classe = S.SampleDiscreteEmpirical(prob, 8, 2)

'nova probabilidade para um tamanho dentro da classe (SPECweb96)

Dim clprob(1 To 18) As Double

clprob(1) = 0.04 '4%

clprob(2) = 102400

clprob(3) = clprob(1) + 0.08 '8%

clprob(4) = 204800

clprob(5) = clprob(3) + 0.13 '13%

clprob(6) = 307200

clprob(7) = clprob(5) + 0.17 '17%

clprob(8) = 409600

clprob(9) = clprob(7) + 0.18 '18%

clprob(10) = 512000

```

    clprob(11) = clprob(9) + 0.15 '15%
    clprob(12) = 614400
    clprob(13) = clprob(11) + 0.11 '11%
    clprob(14) = 716800
    clprob(15) = clprob(13) + 0.07 '7%
    clprob(16) = 819200
    clprob(17) = 1 '4%
    clprob(18) = 921600
    'faz sorteio do tamanho do arquivo
    aux = S.SampleDiscreteEmpirical(clprob, 18, 2)
    tamanhoMensagem = Int(aux / (10 ^ classe)) 'tamanho do arquivo em bytes
Else
    'está devolvendo uma amostra de áudio
    tamanhoMensagem = Get_Variavel("vTamanhoAmostra")
End If
End If
'monta pedido de conexão
Set_Atributo "aNumComunic", (numComunic + 1)
Set_Atributo "aTipoPedido", Get_Variavel("vCONNECT")
Set_Atributo "aTamanhoPedido", tamanhoPedido
Set_Atributo "aTamanhoMensagem", tamanhoMensagem
'marca o tempo em que começa a esperar pela resposta
'VBR em transmissões ímpares e CBR na primeira transmissão
If ((tipo = Get_Variavel("vVBR")) And ((numComunic Mod 2) > 0)) Or ((tipo =
Get_Variavel("vCBR")) And (numComunic = 0)) Then
    Set_Atributo "aTempoInicResposta", S.RunCurrentTime
End If

Registra_Log True, "Monta Pedido de Conexão de" + Str(tamanhoPedido) + " bytes de NT" +
Str(noOrigem) + " para NT" + Str(noDestino) + ". Tamanho da Mensagem" + Str(tamanhoMensagem) + "
bytes"
End Sub

```

```

Private Sub VBA_Block_33_Fire()
    'TRANSMISSÃO DE PEDIDO PELO BARRAMENTO

    If Get_Atributo("aTipoPedido") = Get_Variavel("vCONNECT") Then
        Registra_Log True, "Inicia Transmissão de Pedido de Conexão pelo BS"
    Else
        Registra_Log True, "Inicia Transmissão de Pedido de Desconexão pelo BS"
    End If
End Sub

```

```

Private Sub VBA_Block_34_Fire()
    Dim pedidoConexao As Boolean, prioridade As Double
    Dim NTS As String

    'PEDIDO DE CONEXÃO/DESCONEXÃO CHEGA AO NÓ DE CONTROLE

    pedidoConexao = (Get_Atributo("aTipoPedido") = Get_Variavel("vCONNECT"))
    'ajusta a prioridade do pedido
    If pedidoConexao Then
        'muda conforme o algoritmo do NC
        Set_Picture "pCONNECT"
    Else
        'pedido de desconexão deve ser o MAIS PRIORITÁRIO possível
        Set_Picture "pDISCONNECT"
    End If

```

```

Set_Atributo "aTempoInicConnect", S.RunCurrentTime

NTS = "NT" + Str(Get_Atributo("aNoOrigem")) + " e NT" + Str(Get_Atributo("aNoDestino"))
If pedidoConexao Then
    Registra_Log False, "Pedido de Conexão entre " + NTS + " chega ao NC com Prioridade" +
Str(prioridade)
Else
    Registra_Log False, "Pedido de Desconexão entre " + NTS + " chega ao NC com Prioridade" +
Str(prioridade)
End If
End Sub

```

```

Private Sub VBA_Block_35_Fire()
    Dim aux As Double

    ' MENSAGEM TERMINA DE SER TRANSMITIDA PELO CROSSBAR

```

```

'mensagem chegou ao NT destino. Inverte atributos origem e destino
aux = Get_Atributo("aNoOrigem")
Set_Atributo "aNoOrigem", Get_Atributo("aNoDestino")
Set_Atributo "aNoDestino", aux

    Registra_Log True, "Mensagem Termina Tansferência pelo Crossbar"
End Sub

```

```

Private Sub VBA_Block_36_Fire()
    ' MENSAGEM COMEÇA A SER TRANSMITIDA PELO CROSSBAR

```

```

    Registra_Log False, "Inicia Transmissão de Mensagem de" +
Str(Get_Atributo("aTamanhoMensagem")) + " bytes pelo Crossbar. Tempo de Transmissão:" +
Str(Get_Expressao("eTempoTransmissaoCrossbar")) + ". Velocidade da Rede:" +
Str((Get_Atributo("aTamanhoMensagem") / (Get_Expressao("eTempoTransmissaoCrossbar") *
0.000001)) / 1000000) + " MB/s"
End Sub

```

```

Private Sub Ajusta_Figura_Entidade()
'muda a figura da entidade
    Dim tipo As Double
    Dim CBR As Double, VBR As Double, HARD As Double, SERVER As Double

```

```

    CBR = Get_Variavel("vCBR")
    VBR = Get_Variavel("vVBR")
    HARD = Get_Variavel("vHARD")
    SERVER = Get_Atributo("aServerTask")

```

```

    tipo = Get_Atributo("aTipo")
    If Get_Atributo("aNoOrigem") > Get_Expressao("eNumServidores") Then
        'está executando no cliente
        If tipo = HARD Then
            If SERVER > 0 Then
                If SERVER = VBR Then
                    Set_Picture "pSERVER"
                Else
                    Set_Picture "pSERVERCBR"
                End If
            Else
                Set_Picture "pHARD"
            End If
        End If
    End If

```

```

ElseIf tipo = VBR Then
    Set_Picture "pVBR"
Else
    Set_Picture "pCBR"
End If
Else
    'está executando no servidor
    If tipo = HARD Then
        If SERVER > 0 Then
            If SERVER = VBR Then
                Set_Picture "pSERVER_SRV"
            Else
                Set_Picture "pSERVERCBR_SRV"
            End If
        Else
            Set_Picture "pHARD_SRV"
        End If
    ElseIf tipo = VBR Then
        Set_Picture "pVBR_SRV"
    Else
        Set_Picture "pCBR_SRV"
    End If
End If

```

End Sub

Private Sub VBA\_Block\_37\_Fire()

```

Dim comunic As Double, tipo As Double, noOrigem As Double, atividade As Double
Dim i As Integer, agora As Double, variacao As Double
Dim achou As Boolean

```

' TAREFA ACABA DE SER ESCALONADA PARA EXECUTAR NUM NÓ DE CONTROLE

```

Dim CBR As Double, VBR As Double, HARD As Double, SERVER As Double

```

```

CBR = Get_Variavel("vCBR")
VBR = Get_Variavel("vVBR")
HARD = Get_Variavel("vHARD")
SERVER = Get_Atributo("aServerTask")

```

Ajusta\_Figura\_Entidade

'pega atributos da entidade

```

tipo = Get_Atributo("aTipo")

```

```

comunic = Get_Atributo("aNumComunic")

```

```

noOrigem = Get_Atributo("aNoOrigem")

```

```

atividade = Get_Atributo("aNumAtividade")

```

```

agora = S.RunCurrentTime

```

'registra log

```

Registra_Log True, "Tarefa Inicia Execução com Prioridade" + Str(Get_Atributo("aPrioridade"))

```

'deve verificar se é uma server task e registrar o indicador correto

```

'servertask = Get_Atributo("aServerTask")

```

```

' If servertask > 0 Then

```

```

' 'é uma tarefa hard servindo tarefas soft aperiódicas.

```

```

' 'Deve contabilizar o indicador da tarefa que está servindo

```

```

' 'Pega o tipo da tarefa que está servindo

```

```

' tipo = Get_Indice_Tarefa_Servida

```

```

' tipo = Get_Atributo_Fila("recFilaAperiodic_Q", (tipo), "aTipo")

```

```

' server task também poderá fazer registro pelas tarefas que atende
' End If
' faz registros conforme o tipo da tarefa
If ((tipo = VBR) Or (SERVER = VBR)) And ((comunic Mod 2) = 0) And (comunic > 0) Then
  '(aTipo==vVBR) .AND. (AMOD(aNumComunic,2)==0) .AND. (aNumComunic>0)
  'Tempo de Resposta VBR
  If noOrigem <= Get_Expressao("eNumServidores") Then
    MsgBox "OPS. Executando num servidor.... Debug!" 'verifica se realmente está num cliente
  End If
  Registra_Log True, "Tempo de Resposta dessa Tarefa:" + Str(S.RunCurrentTime -
Get_Atributo("aTempoInicResposta"))
  Set_Atributo "aEscalona", 1 'flag para "Resposta VBR"

ElseIf ((tipo = CBR) Or (SERVER = CBR)) And (comunic = 2) Then
  '(aTipo == vCBR) .AND. (aNumComunic == 2)
  'Tempo de Resposta CBR
  If noOrigem <= Get_Expressao("eNumServidores") Then
    MsgBox "OPS. Executando num servidor.... Debug!" 'verifica se realmente está num cliente
  End If
  'é a primeira amostra recebida. Guarda o início desta execução
  'acha uma posição vazia
  i = 1
  achou = False
  While (i <= 8) And Not achou
    achou = (gUltimaExecucaoCBR(noOrigem, i, 1) = 0)
    i = i + 1
  Wend
  If Not achou Then
    'caso incomum. Deve liberar as informações de outra tarefa
    i = i - 1
  Else
    i = i - 1
  End If
  gUltimaExecucaoCBR(noOrigem, i, 1) = atividade 'id da atividade
  gUltimaExecucaoCBR(noOrigem, i, 2) = agora 'tempo de início da tarefa i
  Registra_Log True, "Tempo de Resposta dessa Tarefa:" + Str(S.RunCurrentTime -
Get_Atributo("aTempoInicResposta"))
  Set_Atributo "aEscalona", 2 'flag para "Resposta CBR"

ElseIf ((tipo = CBR) Or (SERVER = CBR)) And (comunic > 2) Then 'não precisa verificar onde
executa. Sempre é num NT cliente (comunic>2)
  '(aTipo == vCBR) .AND. (aNoOrigem > eNumServidores) .AND. (aNumComunic > 2)
  'Variação de execução CBR
  If noOrigem <= Get_Expressao("eNumServidores") Then
    MsgBox "OPS. Executando num servidor.... Debug!" 'verifica se realmente está num cliente
  End If
  'é uma amostra executando no cliente (exceto a primeira).
  'Determina a diferença entre os períodos de execução desta e da tarefa anterior
  'procura pelo tempo de início de execução da tarefa anterior (i-1)
  i = 1
  achou = False
  While (i <= 8) And Not achou
    achou = (gUltimaExecucaoCBR(noOrigem, i, 1) = atividade)
    i = i + 1
  Wend 'sempre deve achar.
  If Not achou Then
    'caso incomum. Pode ter sido retirado por outra tarefa (código acima)
    'não pode comparar com início da amostra anterior: não pode coletar variação

```

```

'poderia salvar o tempo início desta amostra para a seguinte
i = i - 1
Else
i = i - 1
End If
'calcula a variação na taxa de execução
variacao = agora - gUltimaExecucaoCBR(noOrigem, i, 2) 'período da última amostra
variacao = Sqr(Abs(variacao - Get_Expressao("ePeriodoCBR")) ^ 2) 'raiz quadrada do quadrado do
módulo da diferença com o desejado
'Guarda essa métrica no atributo da entidade
Set_Atributo "aVariacaoCBR", variacao

"Salva num arquivo as diferenças originais
Open "C:\DIFERENCAS_CBR.TXT" For Append As #1
Print #1, Str(agora - gUltimaExecucaoCBR(noOrigem, i, 2)) + " =" + Str(agora) + " - " +
Str(gUltimaExecucaoCBR(noOrigem, i, 2))
Close #1

'Atualiza tempo de início de execução da amostra atual
gUltimaExecucaoCBR(noOrigem, i, 2) = agora
Registra_Log True, "Variação da Taxa de Execução dessa Amostra:" + Str(variacao)
Set_Atributo "aEscalona", 3 'flag para "Variação CBR"
Else
'Não precisa registrar nada
End If
End Sub

```

```

Private Sub VBA_Block_38_Fire()
Dim restante As Double

```

TAREFA DEIXA O PROCESSADOR DO NÓ DE TRABALHO

```

restante = Get_Atributo("aTempoExecRestante")
If restante <= 0.001 Then
Registra_Log True, "Tarefa Termina Execução"
Else
Registra_Log True, "Tarefa é Preemptada faltando" + Str(restante) + " us. Prioridade" +
Str(Get_Atributo("aPrioridade"))
End If
End Sub

```

```

Private Sub VBA_Block_39_Fire()

```

TAREFA TERMINA SUA EXECUÇÃO E DEIXA O SISTEMA

```

'se for uma amostra de áudio, deve liberar a estrutura usada para cálculo da taxa de execução
If Get_Atributo("aTipo") = Get_Variavel("vCBR") Then
If Get_Atributo("aNumComunic") = Get_Atributo("aMaxComunic") Then
'é a última amostra de áudio. Libera a "gUltimaExecucaoCBR" para guardar novos valores
Dim i As Integer, achou As Boolean, noOrigem As Double, atividade As Double
noOrigem = Get_Atributo("aNoOrigem")
atividade = Get_Atributo("aNumAtividade")
i = 1
achou = False
While (i <= 8) And Not achou
achou = (gUltimaExecucaoCBR(noOrigem, i, 1) = atividade)
i = i + 1

```

```

Wend
If achou Then
    i = i - 1
    gUltimaExecucaoCBR(noOrigem, i, 1) = 0
    gUltimaExecucaoCBR(noOrigem, i, 2) = 0
End If
End If
End If

Registra_Log True, "Tarefa Deixa o Sistema Normalmente. " + Str(Get_Atributo("aNumComunic")) +
" comunicações"
End Sub

Private Sub VBA_Block_4_Fire()
    on error GoTo VBA_Block_4_Fire_Err

    Dim noOrigem As Double, noDestino As Double, numCanais As Double
    Dim alocados As Double
    Dim i As Integer

    ' NÓ DE CONTROLE LIBERA CANAIS FÍSICOS DO CROSSBAR

    'pega atributos
    noOrigem = Get_Atributo("aNoOrigem")
    noDestino = Get_Atributo("aNoDestino")
    numCanais = Get_Atributo("aNumCanais")
    'desaloca canais na origem
    i = 1
    alocados = numCanais
    While alocados > 0
        If gCanalFisico(noOrigem, i) = noDestino Then
            'desaloca o canal
            gCanalFisico(noOrigem, i) = 0
            'atualiza variável vCanal[] no modelo
            Set_Variavel "vCanal" + Trim(Str(noOrigem)), (Get_Variavel("vCanal" + Trim(Str(noOrigem))) -
1)
            alocados = alocados - 1
        End If
        i = i + 1
    Wend
    'desaloca canais no destino
    i = 1
    alocados = numCanais
    While alocados > 0
        If gCanalFisico(noDestino, i) = noOrigem Then
            'desaloca o canal
            gCanalFisico(noDestino, i) = 0
            'atualiza variável vCanal[] no modelo
            Set_Variavel "vCanal" + Trim(Str(noDestino)), (Get_Variavel("vCanal" + Trim(Str(noDestino))) -
1)
            alocados = alocados - 1
        End If
        i = i + 1
    Wend

    Registra_Log False, "Libera" + Str(Get_Atributo("aNumCanais")) + " Canais Físicos do Crossbar entre
NT" + Str(noOrigem) + " e NT" + Str(noDestino)
    Exit Sub

```

```
VBA_Block_4_Fire_Err:
  MsgBox ("Erro ao liberar canais físicos do crossbar")
  Resume Next
End Sub
```

```
Private Sub VBA_Block_40_Fire()
  Dim atraso As Double

  ' TAREFA HARD PERDE A DEADLINE

  atraso = S.RunCurrentTime - (Get_Atributo("aTempoChegada") + Get_Atributo("aDeadline"))
  Registra_Log True, "Tarefa deixa o Sistema Após Perder Deadline por" + Str(atraso) + " us"
End Sub
```

```
Private Sub VBA_Block_41_Fire()

  'MONTA PEDIDO DE DESCONEXÃO

  Set_Atributo "aTipoPedido", Get_Variavel("vDISCONNECT")
  Registra_Log True, "Monta Pedido de Desconexão"
End Sub
```

```
Private Sub VBA_Block_42_Fire()
  MsgBox "Não poderia estar passando por aqui. Debug!" 'NUNCA DEVE SER EXECUCADO.
  APENAS PARA VERIFICAÇÃO DO MODELO
End Sub
```

```
Private Sub VBA_Block_43_Fire()
  MsgBox "Não poderia estar passando por aqui. Debug!" 'NUNCA DEVE SER EXECUCADO.
  APENAS PARA VERIFICAÇÃO DO MODELO
End Sub
```

```
Private Sub VBA_Block_45_Fire()
  Dim pedidoConexao As Boolean, prioridade As Double
  Dim NTS As String

  'PEDIDO CONEXÃO/DESCONEXÃO CHEGA AO NÓ DE CONTROLE

  pedidoConexao = (Get_Atributo("aTipoPedido") = Get_Variavel("vCONNECT"))
  'ajusta a prioridade do pedido
  If pedidoConexao Then
    'muda conforme o algoritmo do NC
    prioridade = S.RunCurrentTime 'fifo
  Else
    'pedido de desconexão deve ser o MAIS PRIORITÁRIO possível (A menos que se use round-Robin)
    'se pedido de desconexão for menos prioritário que pedido de conexão dos mesmos canais, pode
    ocorrer impasse (deadlock).
    prioridade = -1 'todas as outras prioridades são positivas
  End If
  Set_Atributo "aPrioridade", prioridade
End Sub
```

```
Private Sub VBA_Block_46_Fire()

  ' INICIALIZA TAREFA SERVIDORA (PS, DS, SS, DSS, CBR)
```

'usa gIDaperiodicServida[] e gServerTaskAssociada[] como parâmetros de entrada

Dim esteNo As Double, aux As Double, servida As Double

'acha uma tarefa aperiódica que deve ser derivada mas não tem a server task associada

Dim i As Integer, achou As Boolean, maxNodos As Double, indice As Integer

i = 1

achou = False

maxNodos = Get\_Variavel("vMaxNodos")

While (i <= maxNodos) And Not achou

    achou = (gIDaperiodicServida(i) > 0) And (gServerTaskAssociada(i) = 0)

    i = i + 1

Wend

If achou Then

    'sempre deve achar, pois a entidade server task foi disparada (este procedimento está executando!)

    i = i - 1

Else

    MsgBox "Não foi encontrada uma tarefa aperiódica para servir. Debug!"

End If

indice = Procura\_Atividade\_Fila("recFilaAperiodic\_Q", "aNumAtividade", gIDaperiodicServida(i))

If indice <= 0 Then

    MsgBox "Não foi encontrada uma tarefa aperiódica para servir. Debug!"

End If

'incrementa número da atividade

Dim atividade As Double

atividade = Get\_Variavel("vAtividadeAtual")

Set\_Atributo "aNumAtividade", atividade

Set\_Variavel "vAtividadeAtual", (atividade + 1)

'informa que a tarefa aperiódica será atendida por esta task server

gServerTaskAssociada(i) = atividade

'seta valores iniciais

esteNo = Get\_Atributo\_Fila("recFilaAperiodic\_Q", (indice), "aNoOrigem")

Set\_Atributo "aTipo", Get\_Variavel("vHARD") 'tarefa hard periódica para atender tarefas soft

aperiódicas

aux = Get\_Atributo\_Fila("recFilaAperiodic\_Q", (indice), "aTipo")

Set\_Atributo "aServerTask", aux 'tipo da tarefa servida

Set\_Atributo "aPeriodo", Get\_Expressao("ePeriodoServerTask")

Set\_Atributo "aDeadline", Get\_Expressao("ePeriodoServerTask")

Set\_Atributo "aNoOrigem", esteNo

servida = Get\_Atributo\_Fila("recFilaAperiodic\_Q", (indice), "aNumAtividade")

Set\_Atributo "aNumAtividadeServida", servida 'num. da atividade servida

'Set\_Atributo "aCapacidade", gCapacidadeServidor(esteNo)

aux = Get\_Atributo\_Fila("recFilaAperiodic\_Q", (indice), "aTempoExecucao")

Set\_Atributo "aTempoExecucao", aux

Set\_Atributo "aTempoChegadaSistema", S.RunCurrentTime

'Set\_Atributo "aNumComunic", 0

'Set\_Atributo "aMaxComunic", 0

'copia atributos da tarefa servida para coletar estatísticas por ela

aux = Get\_Atributo\_Fila("recFilaAperiodic\_Q", (indice), "aTempoInicResposta")

Set\_Atributo "aTempoInicResposta", aux

aux = Get\_Atributo\_Fila("recFilaAperiodic\_Q", (indice), "aNumComunic")

Set\_Atributo "aNumComunic", aux

aux = Get\_Atributo\_Fila("recFilaAperiodic\_Q", (indice), "aMaxComunic")

Set\_Atributo "aMaxComunic", aux

'invoca procedimento de inicialização dos atributos quando chega ao NT

VBA\_Block\_8\_Fire

```
Registra_Log True, "Inicializa Servidor Aperiódico (" + Trim(Str(servida)) + "). Capacidade " +  
Str(gCapacidadeServidor(esteNo))  
End Sub
```

```
Private Function Get_Indice_Tarefa_Servida() As Double  
Dim i As Integer, achou As Boolean  
Dim numServida As Double, numElementos As Double, esteNo As Double  
  
numServida = Get_Atributo("aNumAtividadeServida")  
i = 1  
achou = False  
esteNo = Get_Atributo("aNoOrigem")  
numElementos = Get_Num_Elementos_Fila("recFilaAperiodic_Q")  
While (i <= numElementos) And Not achou  
achou = (Get_Atributo_Fila("recFilaAperiodic_Q", i, "aNumAtividade") = numServida)  
If achou Then  
achou = (Get_Atributo_Fila("recFilaAperiodic_Q", i, "aNoOrigem") = esteNo)  
If Not achou Then  
MsgBox "Teria retirado a tarefa aperiódica de outro nó. Agora ok."  
End If  
End If  
i = i + 1  
Wend  
If achou Then  
'sempre deve achar  
Get_Indice_Tarefa_Servida = i - 1  
Else  
'Nunca deve executar aqui. Apenas para verificação do modelo  
MsgBox "Tarefa Aperiódica Servida Não Foi Encontrada na Fila. Debug!"  
Get_Indice_Tarefa_Servida = 0  
  
'Não achou a tarefa que esteja sendo atendida. Compara com todas na fila periódica  
Dim texto1 As String, texto2 As String  
numElementos = Get_Num_Elementos_Fila("recFila_Aperiodic_Q")  
texto2 = "ID:" + Str(Get_Atributo("aNumAtividade")) + " - NT:" + Str(Get_Atributo("aNoOrigem"))  
+ " - Servindo:" + Str(Get_Atributo("aNumAtividadeServida"))  
For i = 1 To numElementos  
texto2 = "ID:" + Str(Get_Atributo_Fila("recFila_Aperiodic_Q", i, "aNumAtividade")) + " - NT:" +  
Str(Get_Atributo_Fila("recFila_Aperiodic_Q", i, "aNoOrigem")) + " - Servindo:" +  
Str(Get_Atributo_Fila("recFila_Aperiodic_Q", i, "aNumAtividadeServida"))  
MsgBox texto1 + Chr(13) + texto2, [vbOKOnly]  
Next i  
End If  
End Function
```

```
Private Sub VBA_Block_47_Fire()
```

```
DETERMINA POSIÇÃO DA ATIVIDADE APERIÓDICA QUE ACABA DE SER TOTALMENTE  
SERVIDA
```

```
Dim indice As Double, esteNo As Double  
indice = Get_Indice_Tarefa_Servida  
Set_Atributo "aEscalona", indice 'ranking da entidade na fila aperiódica. Será retirada pois já foi  
atendida completamente  
'libera estrutura da server task  
esteNo = Get_Atributo("aNoOrigem")
```

```
Registra_Log True, "Tarefa Aperiódica (" + Trim(Str(gIDAperiodicServida(estesteNo))) + ") Termina de ser Atendida por Server Task (" + Trim(Str(gServerTaskAssociada(estesteNo))) + ")"
```

```
'após o log para apresentar os valores que foram liberados  
gIDAperiodicServida(estesteNo) = 0  
gServerTaskAssociada(estesteNo) = 0  
End Sub
```

```
Private Sub VBA_Block_48_Fire()  
Registra_Log True, "Sinal Enviado Liberando Server Task"  
End Sub
```

```
Private Sub VBA_Block_49_Fire()  
Registra_Log True, "Server Task Aguarda Sinal de Liberação"  
End Sub
```

```
Private Sub VBA_Block_5_Fire()  
on error GoTo VBA_Block_5_Fire_Err  
Dim numCanais As Double, tempoTransmissao As Double  
Dim DispOrigem As Integer, dispDestino As Integer, maxCanais As Double  
Dim noOrigem As Double, noDestino As Double  
Dim tamanho As Double, velocidade As Double  
Dim disponiveis As Double, deadline As Double, chegada As Double  
Dim agora As Double, i As Integer
```

' NÓ DE CONTROLE VERIFICA A QUANTIDADE DE CANAIS FÍSICOS DISPONÍVEIS NO CROSSBAR

```
'pega informações do modelo  
agora = S.RunCurrentTime  
chegada = Get_Atributo("aTempoChegada")  
deadline = Get_Atributo("aDeadline")
```

```
maxCanais = Get_Variavel("vMaxCanais")  
noOrigem = Get_Atributo("aNoOrigem")  
noDestino = Get_Atributo("aNoDestino")  
tamanho = Get_Atributo("aTamanhoMensagem")  
velocidade = Get_Variavel("vVelocCrossbar")  
'verifica quantos canais seriam necessários para transmitir a mensagem sem perda de deadline  
numCanais = 0
```

```
Do  
numCanais = numCanais + 1  
tempoTransmissao = ((tamanho / velocidade) / 1.048576) / numCanais  
Loop While (numCanais <= 8) And ((agora + tempoTransmissao) > (chegada + deadline))  
'verifica se há canais existentes suficiente  
If numCanais > maxCanais Then  
'precisaria de mais que o máximo existente. Vai perder a deadline de qualquer modo  
'tenta alocar o máximo possível  
numCanais = maxCanais  
End If  
If Get_Atributo("aTipo") = Get_Variavel("vCBR") Then  
numCanais = 1  
End If
```

```
'atualiza atributo  
Set_Atributo "aNumCanais", numCanais
```

```
'verifica quantos canais estão disponíveis na Origem e no Destino
```

```

DispOrigem = 0
For i = 1 To maxCanais
    If gCanalFisico(noOrigem, i) = 0 Then
        DispOrigem = DispOrigem + 1
    End If
Next i
dispDestino = 0
For i = 1 To maxCanais
    If gCanalFisico(noDestino, i) = 0 Then
        dispDestino = dispDestino + 1
    End If
Next i
'pega o menor dos dois, pois ambos devem estar disponíveis
If DispOrigem < dispDestino Then
    disponiveis = DispOrigem
Else
    disponiveis = dispDestino
End If
Set_Atributo "aDisponiveis", disponiveis

If disponiveis = 0 Then
    "Registra_Log True, "Não Há Canais Físicos Disponíveis entre NT" + Str(noOrigem) + " e NT" +
Str(noDestino)
Else
    Registra_Log False, "Determina Existência de" + Str(disponiveis) + " Canais Físicos Disponíveis
entre NT" + Str(noOrigem) + " e NT" + Str(noDestino)
End If
Exit Sub

```

```

VBA_Block_5_Fire_Err:
    MsgBox ("Erro ao determinar Canais Físicos disponíveis")
    Resume Next
End Sub

```

```

Private Sub VBA_Block_6_Fire()
    on error GoTo VBA_Block_6_Fire_Err
    Dim numCanais As Double, disponiveis As Double
    Dim noOrigemas As Double, noOrigem As Double, noDestino As Double
    Dim i As Integer, alocados As Integer

```

' NÓ DE CONTROLE ALOCA CANAIS FÍSICOS NO CROSSBAR

```

'pega atributos
disponiveis = Get_Atributo("aDisponiveis")
numCanais = Get_Atributo("aNumCanais")
noOrigem = Get_Atributo("aNoOrigem")
noDestino = Get_Atributo("aNoDestino")
'verifica se os canais necessários estão disponíveis
If disponiveis < numCanais Then
    'aloca só os disponíveis
    numCanais = disponiveis
    Set_Atributo "aNumCanais", numCanais
End If
'aloca canais na origem
i = 1
alocados = 0
While alocados < numCanais
    If gCanalFisico(noOrigem, i) = 0 Then

```

```

'canal livre. Vai alocar
gCanalFisico(noOrigem, i) = noDestino
'atualiza variável vCanal[] no modelo
Set_Variavel "vCanal" + Trim(Str(noOrigem)), (Get_Variavel("vCanal" + Trim(Str(noOrigem)))) +
1)
    alocados = alocados + 1
End If
i = i + 1
Wend
'aloca canais no nó destino
i = 1
alocados = 0
While alocados < numCanais
    If gCanalFisico(noDestino, i) = 0 Then
        'canal livre. Vai alocar
        gCanalFisico(noDestino, i) = noOrigem
        'atualiza variável vCanal[] no modelo
        Set_Variavel "vCanal" + Trim(Str(noDestino)), (Get_Variavel("vCanal" + Trim(Str(noDestino)))) +
1)
            alocados = alocados + 1
        End If
        i = i + 1
    Wend

    Registra_Log False, "Aloca" + Str(numCanais) + " Canais Físicos Disponíveis. Tempo de conexão:" +
Str(S.RunCurrentTime - Get_Atributo("aTempoInicConnect")) + ". Deadline:" +
Str(Get_Atributo("aTempoChegada") + Get_Atributo("aDeadline"))
    Exit Sub

```

```

VBA_Block_6_Fire_Err:
    MsgBox ("Erro ao alocar canais físicos do crossbar")
    Resume Next
End Sub

```

```

Private Sub Verifica_Filas(esteno As Double, prioridadeFilaPeriodic As Double, prioridadeFilaAperiodic
As Double, prioridadeExecutando As Double, indicePeriodic As Integer, indiceAperiodic As Integer,
indiceExecutando As Integer)

```

```

    Dim achou As Boolean, i As Integer, numElementos As Long
    Dim noFila As Double

```

```

' VERIFICA AS FILAS "PERIÓDICA", "APERIÓDICA" E A CPU DO NT ATUAL POR
ENTIDADES

```

```

'Fila de tarefas periódicas
numElementos = Get_Num_Elementos_Fila("recFilaPeriodic_Q")
If numElementos > 0 Then
    'há entidades na fila. Verifica se alguma esta neste NT (aNoOrigem)
    i = 1
    achou = False
    While (i <= numElementos) And Not achou
        noFila = Get_Atributo_Fila("recFilaPeriodic_Q", i, "aNoOrigem") 'pega o NT onde a tarefa da fila
aguarda
        achou = (noFila = esteNo)
        If achou Then
            prioridadeFilaPeriodic = Get_Atributo_Fila("recFilaPeriodic_Q", i, "aPrioridade") 'pega a
prioridade da tarefa na fila
        Else

```

```

        i = i + 1
    End If
Wend
End If
indicePeriodic = i

'Fila de tarefas aperiódicas
numElementos = Get_Num_Elementos_Fila("recFilaAperiodic_Q")
If numElementos > 0 Then
    'há entidades na fila. Verifica se alguma esta neste NT (aNoOrigem)
    i = 1
    achou = False
    While (i <= numElementos) And Not achou
        noFila = Get_Atributo_Fila("recFilaAperiodic_Q", i, "aNoOrigem") 'pega o NT onde a tarefa da fila
        aguarda
        achou = (noFila = esteNo)
        If achou Then
            prioridadeFilaAperiodic = Get_Atributo_Fila("recFilaAperiodic_Q", i, "aPrioridade") 'pega a
            prioridade da tarefa na fila
        Else
            i = i + 1
        End If
    Wend
End If
indiceAperiodic = i

```

```

'Fila de tarefas executando
numElementos = Get_Num_Elementos_Fila("recFilaExecutando_Q")
If numElementos > 0 Then
    'há entidades na fila. Verifica se alguma esta neste NT (aNoOrigem)
    i = 1
    achou = False
    While (i <= numElementos) And Not achou
        noFila = Get_Atributo_Fila("recFilaExecutando_Q", i, "aNoOrigem") 'pega o NT onde a tarefa da
        fila aguarda
        achou = (noFila = esteNo)
        If achou Then
            prioridadeExecutando = Get_Atributo_Fila("recFilaExecutando_Q", i, "aPrioridade") 'pega a
            prioridade da tarefa na fila
        Else
            i = i + 1
        End If
    Wend
End If
indiceExecutando = i
End Sub

```

```

Private Sub VBA_Block_7_Fire()
    on error GoTo VBA_Block_7_Fire_Err

```

'INTERRUPÇÃO DO TIMER OU EVENTO DE ESCALONAMENTO (CRIAÇÃO OU TÉRMINO DE TAREFAS)

'DEVE FAZER O ESCALONAMENTO DE PROCESSOS

```

Dim prioridadeFilaPeriodic As Double
Dim prioridadeFilaAperiodic As Double
Dim prioridadeExecutando As Double
Dim esteNo As Double

```

```
Dim quantum As Double, algoritmo As Double
Dim indicePeriodic As Integer, indiceAperiodic As Integer, indiceExecutando As Integer
Dim escalona As Double, preempta As Double, periodo As Double, criaserver As Double
Dim tmpRestante As Double
```

```
'valores iniciais
esteNo = Get_Atributo("aNoOrigem")
quantum = Get_Variavel("vQuantum")
'ninguém em nenhuma fila
prioridadeFilaPeriodic = -1
prioridadeFilaAperiodic = -1
prioridadeExecutando = -1
```

```
'Verifica as tarefas prontas para executar,
'que estão esperando nas filas recPeriodic_Q e recAperiodic_Q
Verifica_Filas esteNo, prioridadeFilaPeriodic, prioridadeFilaAperiodic, prioridadeExecutando,
indicePeriodic, indiceAperiodic, indiceExecutando
```

```
'Escalonamento de Tempo-Real
tmpRestante = 0
periodo = 0 'ninguém será escalonado
escalona = 0
preempta = 0
criaserver = 0
'verifica se preempta a tarefa executando por tempo de execucao
If prioridadeExecutando > -1 Then
    'há uma tarefa executando nesse NT
    'Verifica seu tempo de execucao restante
    tmpRestante = Get_Atributo_Fila("recFilaExecutando_Q", indiceExecutando,
"aTempoExecRestante") - (.RunCurrentTime - Get_Atributo_Fila("recFilaExecutando_Q",
indiceExecutando, "aTempoInicExec"))
    If tmpRestante <= 0.001 Then 'era para ser 0, mas qualquer valor muito pequeno pode evitar que o
tempo passe.
        'vai preemptar a tarefa que está executando
        preempta = indiceExecutando
    End If
End If
```

```
'chama o escalonador
algoritmo = Get_Variavel("vEscalonamento")
If algoritmo = 1 Then 'EDF
    Escalonamento_EDF prioridadeFilaPeriodic, prioridadeFilaAperiodic, prioridadeExecutando,
indicePeriodic, indiceAperiodic, indiceExecutando, tmpRestante, preempta, escalona, periodo, criaserver
```

```
ElseIf algoritmo = 2 Then 'RM
    Escalonamento_RM prioridadeFilaPeriodic, prioridadeFilaAperiodic, prioridadeExecutando,
indicePeriodic, indiceAperiodic, indiceExecutando, tmpRestante, preempta, escalona, periodo, criaserver
```

```
ElseIf algoritmo = 3 Then 'BS
    Escalonamento_BS prioridadeFilaPeriodic, prioridadeFilaAperiodic, prioridadeExecutando,
indicePeriodic, indiceAperiodic, indiceExecutando, tmpRestante, preempta, escalona, periodo, criaserver
```

```
ElseIf algoritmo = 4 Then 'PS
    Escalonamento_PS prioridadeFilaPeriodic, prioridadeFilaAperiodic, prioridadeExecutando,
indicePeriodic, indiceAperiodic, indiceExecutando, tmpRestante, preempta, escalona, periodo, criaserver
```

```
ElseIf algoritmo = 5 Then 'DS
```

```
Escalonamento_DS prioridadeFilaPeriodic, prioridadeFilaAperiodic, prioridadeExecutando,
indicePeriodic, indiceAperiodic, indiceExecutando, tmpRestante, preempta, escalona, periodo, criaserver
```

```
ElseIf algoritmo = 6 Then 'CBR
```

```
Escalonamento_CBR prioridadeFilaPeriodic, prioridadeFilaAperiodic, prioridadeExecutando,
indicePeriodic, indiceAperiodic, indiceExecutando, tmpRestante, preempta, escalona, periodo, criaserver
```

```
ElseIf algoritmo = 7 Then 'SS
```

```
Escalonamento_SS prioridadeFilaPeriodic, prioridadeFilaAperiodic, prioridadeExecutando,
indicePeriodic, indiceAperiodic, indiceExecutando, tmpRestante, preempta, escalona, periodo, criaserver
```

```
ElseIf algoritmo = 8 Then 'DSS
```

```
Escalonamento_DSS prioridadeFilaPeriodic, prioridadeFilaAperiodic, prioridadeExecutando,
indicePeriodic, indiceAperiodic, indiceExecutando, tmpRestante, preempta, escalona, periodo, criaserver
```

```
ElseIf algoritmo = 9 Then 'RED
```

```
Escalonamento_RED prioridadeFilaPeriodic, prioridadeFilaAperiodic, prioridadeExecutando,
indicePeriodic, indiceAperiodic, indiceExecutando, tmpRestante, preempta, escalona, periodo, criaserver
```

```
ElseIf algoritmo = 10 Then 'ABS
```

```
Escalonamento_ABS prioridadeFilaPeriodic, prioridadeFilaAperiodic, prioridadeExecutando,
indicePeriodic, indiceAperiodic, indiceExecutando, tmpRestante, preempta, escalona, periodo, criaserver
End If
```

```
'Depois do escalonamento
```

```
'O tempo da próxima entidade de controle depende da tarefa executando
```

```
If preempta Then
```

```
'A tarefa atual está sendo preemptada. Próxima ent. controle em um vQuantum de tempo
```

```
Set_Atributo "aTempoExecRestante", quantum
```

```
'If tmpRestante > 0.001 Then
```

```
' 'preemptou mesmo
```

```
'End If
```

```
Else
```

```
'A tarefa não será preemptada (talvez nenhuma esteja executando)
```

```
If (tmpRestante = 0) Or (tmpRestante >= quantum) Then
```

```
'ninguém está executando ou vai executar por mais de um quantum -> Próxima ent. controle em um
vQuantum
```

```
Set_Atributo "aTempoExecRestante", quantum
```

```
Else
```

```
'a tarefa executando vai terminar antes do quantum. Próxima ent. de controle assim que a tarefa
terminar.
```

```
Set_Atributo "aTempoExecRestante", tmpRestante
```

```
End If
```

```
End If
```

```
'verifica se alguém será escalonado e marca a próxima entidade de controle
```

```
If escalona > 0 Then 'if periodo <>0 then 'mesmo efeito
```

```
'Alguém será escalonado nesse momento.
```

```
'Verifica quando deve vir a próxima entidade de controle
```

```
'Normalmente após vQuantum de tempo, a não ser que a tarefa termine antes
```

```
If periodo > 0 Then
```

```
'a tarefa periódica será escalonada
```

```
tmpRestante = Get_Atributo_Fila("recFilaPeriodic_Q", indicePeriodic, "aTempoExecRestante")
```

```
Else
```

```
'a tarefa aperiódica será escalonada
```

```
tmpRestante = Get_Atributo_Fila("recFilaAperiodic_Q", indiceAperiodic, "aTempoExecRestante")
```

```
End If
```

```

    If tmpRestante >= quantum Then
        'próxima ent. controle em um Quantum de tempo
        Set_Atributo "aTempoExecRestante", quantum
    Else
        'a tarefa que será escalonada termina antes. Próxima ent. controle em tmpRestante unidades de
tempo
        Set_Atributo "aTempoExecRestante", tmpRestante
    End If
End If

'Grava valores de resposta para o modelo
Set_Atributo "aEscalona", escalona
Set_Atributo "aPreempta", preempta
Set_Atributo "aPeriodo", periodo
Set_Atributo "aCriaServerTask", criaserver

'Registra_Log True, "Quantum - NT" + Str(esteNo) + " - E:" + Str(escalona) + " - Pr:" + Str(preempta)
+ " - Pe:" + Str(periodo) + " - Rst:" + Str(tmpRestante)
Exit Sub

```

```

VBA_Block_7_Fire_Err:
    MsgBox ("Erro no Escalonamento de Tarefas")
    Resume Next
End Sub

```

```

Private Sub VBA_Block_8_Fire()
    on error GoTo VBA_Block_8_Fire_Err
    Dim agora As Double, deadline As Double, tempoexec As Double, noOrigem As Double
    Dim algoritmo As Double, fatorsoft As Double, tipo As Double

    'TAREFA CHEGA AO NÓ DE TRABALHO

    'pega atributos
    agora = S.RunCurrentTime
    noOrigem = Get_Atributo("aNoOrigem")
    tipo = Get_Atributo("aTipo")
    If tipo = Get_Variavel("vVBR") Then
        'define um novo tempo de execução para tarefa soft-vbr
        tempoexec = Get_Expressao("eTempoExecVBR")
        If noOrigem <= Get_Expressao("eNumServidores") Then
            'Deve ter exec. tal que o Fator de Utilizacao no cliente seja aproximadamente U_Fator_Soft
            fatorsoft = Get_Variavel("vFator_U_Soft")
            tempoexec = 2 * tempoexec * (1 - fatorsoft)
        End If
        Set_Atributo "aTempoExecucao", tempoexec
        deadline = Get_Atributo("aTempoExecucao") * (1 / Get_Variavel("vFator_U_Soft"))
        Set_Atributo "aDeadline", deadline
    End If
    'atualiza atributos
    Set_Atributo "aTempoChegada", agora
    Set_Atributo "aTempoExecRestante", Get_Atributo("aTempoExecucao")
    Set_Atributo "aPreempta", 0
    Set_Atributo "aEscalona", 0
    Set_Atributo "aCriaServerTask", 0

    Ajusta_Figura_Entidade

    'muda conforme o escalonador

```

```

algoritmo = Get_Variavel("vEscalonamento")
If algoritmo = 1 Then    'EDF
    'Fila ordenada por Deadline Absoluto
    Set_Atributo "aPrioridade", (Get_Atributo("aTempoChegada") + Get_Atributo("aDeadline"))
ElseIf algoritmo = 2 Then    'RM
    'Fila ordenada por período de ativação
    Set_Atributo "aPrioridade", (Get_Atributo("aPeriodo"))
ElseIf (algoritmo >= 3) And (algoritmo <= 10) Then    'BS, PS, DS, em diante
    If tipo = Get_Variavel("vHARD") Then
        'Fila periódica ordenada por período de ativação - RM
        Set_Atributo "aPrioridade", (Get_Atributo("aPeriodo"))
    Else
        'Fila aperiódica FIFO ordenada por tempo de chegada
        Set_Atributo "aPrioridade", (Get_Atributo("aTempoChegada"))
    End If
Else
    'EDF
    'Fila ordenada por Deadline Absoluto
    Set_Atributo "aPrioridade", (Get_Atributo("aTempoChegada") + Get_Atributo("aDeadline"))
End If

Registra_Log True, "Tarefa Chega com Prioridade" + Str(Get_Atributo("aPrioridade")) + " e" +
Str(Get_Atributo("aNumComunic")) + " comunicações"
Exit Sub

```

```

VBA_Block_8_Fire_Err:
    MsgBox ("Erro inicializando tarefa no NT")
    Resume Next
End Sub

```

```

Public Function Procura_Atividade_Fila(fila As String, atributo As String, valor As Double) As Integer
    '0 se não encontrou, >0 ranking da entidade na fila
    Dim i As Integer, achou As Boolean, numElementos As Double

    numElementos = Get_Num_Elementos_Fila(fila)
    i = 1
    achou = False
    While (i <= numElementos) And Not achou
        achou = (Get_Atributo_Fila(fila, i, atributo) = valor)
        i = i + 1
    Wend
    If achou Then
        Procura_Atividade_Fila = i - 1
    Else
        Procura_Atividade_Fila = 0 'não achou
    End If
End Function

```

## ANEXO 6 – Código-fonte VBA, parte integrante do modelo.

### Arquivo “mEscalonadores.bas”

```
Public Sub Escalonamento_RM(prioridadeFilaPeriodic As Double, prioridadeFilaAperiodic As Double,
prioridadeExecutando As Double, indicePeriodic%, indiceAperiodic%, indiceExecutando%,
tmpRestante#, preempta#, escalona#, periodo#, criaserver#)
```

```
'RATE MONOTONIC (2)
```

```
'
If (prioridadeFilaPeriodic > -1) Or (prioridadeFilaAperiodic > -1) Then
'Há pelo menos uma tarefa aguardando para executar
If (prioridadeFilaPeriodic > -1) Then
'Se há periódicas, então são atendidas primeiro
If (prioridadeFilaPeriodic < prioridadeExecutando) Or (prioridadeExecutando = -1) Then
escalona = indicePeriodic
periodo = 1 'informa que é da fila periódica
If prioridadeExecutando > -1 Then
preempta = indiceExecutando
End If
End If
ElseIf (prioridadeFilaAperiodic > -1) Then
'Há apenas tarefas aperiódicas. Escalona apenas quando a CPU estiver ociosa (FIFO)
If (prioridadeExecutando = -1) Then
escalona = indiceAperiodic
periodo = -1 'informa que é da fila aperiódica
End If
End If
End If
End Sub
```

```
Public Sub Escalonamento_BS(prioridadeFilaPeriodic As Double, prioridadeFilaAperiodic As Double,
prioridadeExecutando As Double, indicePeriodic%, indiceAperiodic%, indiceExecutando%,
tmpRestante#, preempta#, escalona#, periodo#, criaserver#)
```

```
'BACKGROUND SCHEDULING (3)
```

```
'
If (prioridadeFilaPeriodic > -1) Or (prioridadeFilaAperiodic > -1) Then
'Há pelo menos uma tarefa aguardando para executar
If (prioridadeFilaPeriodic > -1) Then
'Se há periódicas, então escalona.
If (prioridadeFilaPeriodic < prioridadeExecutando) Or (prioridadeExecutando = -1) Then
escalona = indicePeriodic
periodo = 1 'informa que é da fila periódica
If prioridadeExecutando > -1 Then
preempta = indiceExecutando
End If
End If
ElseIf (prioridadeFilaAperiodic > -1) Then ""preempta > 0
'Há apenas tarefas aperiódicas. Escalona apenas quando a CPU ficar livre
If (prioridadeExecutando = -1) Then
```

```

        escalona = indiceAperiodic
        periodo = -1 'informa que é da fila aperiódica
    End If
End If
End If
End Sub

```

```

Public Sub Escalonamento_EDF(prioridadeFilaPeriodic As Double, prioridadeFilaAperiodic As Double,
prioridadeExecutando As Double, indicePeriodic%, indiceAperiodic%, indiceExecutando%,
tmpRestante#, preempta#, escalona#, periodo#, criaserver#)

```

```

'EARLIST DEADLINE FIRST (1)

```

```

If (prioridadeFilaPeriodic > -1) Or (prioridadeFilaAperiodic > -1) Then
'Há pelo menos uma tarefa aguardando para executar
If (prioridadeFilaPeriodic > -1) And (prioridadeFilaAperiodic = -1) Then
'Há apenas tarefas periódicas. Escalona a primeira
If (prioridadeFilaPeriodic < prioridadeExecutando) Or (prioridadeExecutando = -1) Then
    escalona = indicePeriodic
    periodo = 1 'informa que é da fila periódica
    If prioridadeExecutando > -1 Then
        preempta = indiceExecutando
    End If
End If
ElseIf (prioridadeFilaPeriodic = -1) And (prioridadeFilaAperiodic > -1) Then
'Há apenas tarefas aperiódicas. Escalona a primeira
If (prioridadeFilaAperiodic < prioridadeExecutando) Or (prioridadeExecutando = -1) Then
    escalona = indiceAperiodic
    periodo = -1 'informa que é da fila aperiódica
    If prioridadeExecutando > -1 Then
        preempta = indiceExecutando
    End If
End If
Else
'Há tarefas nas duas filas. Escalona aquela com maior prioridade (menor número)
If prioridadeFilaPeriodic < prioridadeFilaAperiodic Then
'A tarefa da fila periódica é mais prioritária
If (prioridadeFilaPeriodic < prioridadeExecutando) Or (prioridadeExecutando = -1) Then
    escalona = indicePeriodic
    periodo = 1 'informa que é da fila periódica
End If
Else
'A tarefa da fila aperiódica é mais prioritária
If (prioridadeFilaAperiodic < prioridadeExecutando) Or (prioridadeExecutando = -1) Then
    escalona = indiceAperiodic
    periodo = -1 'informa que é da fila aperiódica
End If
End If
If prioridadeExecutando > -1 Then
    preempta = indiceExecutando
End If
End If
End If
End Sub

```

```
Public Sub Escalonamento_PS(prioridadeFilaPeriodic As Double, prioridadeFilaAperiodic As Double,
prioridadeExecutando As Double, indicePeriodic%, indiceAperiodic%, indiceExecutando%,
tmpRestante#, preempta#, escalona#, periodo#, criaserver)
```

```
'POOLING SERVER (4)
```

```
Dim agora As Double, periodoServ As Double
Dim posPeriodic As Double, posAperiodic As Double, posExec As Double
Dim esteNo As Double
```

```
'pega informações do modelo
agora = S.RunCurrentTime
periodoServ = Get_Expressao("ePeriodoServerTask")
esteNo = Get_Atributo("aNoOrigem")
```

```
'verifica se é tempo de ativação da Server Task
```

```
If (gUltimaAtivacao(esteNo) + Int(periodoServ)) <= Int(agora) Then
```

```
'início de um novo período da server task
```

```
'procura a server task no sistema
```

```
Procura_Task_Server posPeriodic, posAperiodic, posExec
```

```
If (posPeriodic <= 0) And (posAperiodic <= 0) And (posExec <= 0) Then
```

```
'não há nenhuma server task no sistema.
```

```
'Se houver tarefas aperiódicas pendentes, então manda criar uma entidade Task Server
```

```
If prioridadeFilaAperiodic > -1 Then
```

```
'Set_Atributo "aCriaServerTask", 1 '>0
```

```
criaserver = 1
```

```
'Não faz mais nada no momento.
```

```
'Uma entidade tipo vHARD e atributo aServerTask > 0 será criada e este procedimento será  
chamado novamente (no mesmo instante de tempo simulado, sem delay)
```

```
'DEVE INICIALIZAR OS ATRIBUTOS DA TAREFA QUANDO FOR CRIADA
```

```
'passa o parâmetro necessário para inicializar os atributos (ID da tarefa aperiódica que irá atender)
```

```
gIDAperiodicServida(esteNo) = Get_Atributo_Fila("recFilaAperiodic_Q", indiceAperiodic,  
"aNumAtividade")
```

```
gServerTaskAssociada(esteNo) = 0 'será atualizado quando a server task for inicializada
```

```
'reestabelece a capacidade ao máximo
```

```
gCapacidadeServidor(esteNo) = Get_Expressao("eCapacidadeMaxima")
```

```
Else
```

```
'Não há nenhuma tarefa aperiódica no instante de ativação de PS. Só poderá atender no próximo  
período
```

```
'Não cria server task
```

```
End If
```

```
Else
```

```
'no sistema há uma "server task" periódica que atende tarefas aperiódicas
```

```
'deve ter sua capacidade restabelecida
```

```
gCapacidadeServidor(esteNo) = Get_Expressao("eCapacidadeMaxima")
```

```
'atualiza a ultima ativacao
```

```
End If
```

```
'atualiza o instante de ativação
```

```
gUltimaAtivacao(esteNo) = agora
```

```
Else
```

```
'verifica caso da server task ter terminado de atender uma tarefa aperiódica,
```

```
'ter sobrado capacidade e ter outras tarefas aperiódicas esperando.
```

```
'Nesse caso deve recriar a server task até a capacidade esgotar ou terminar de atender as aperiódicas
```

```
If (gCapacidadeServidor(esteNo) > 0) Then
```

```
'sobrou capacidade da última tarefa aperiódica atendida nesta ativação
```

```
If (prioridadeFilaAperiodic > -1) Then
```

```

'Há capacidade e há tarefas esperando. Portanto continua atendendo
Procura_Task_Server posPeriodic, posAperiodic, posExec
If (posPeriodic <= 0) And (posAperiodic <= 0) And (posExec <= 0) Then
'e não há nenhuma server task no sistema. É o caso apresentado. Deve criar novamente a server
task com a capacidade que lhe resta
'Set_Atributo "aCriaServerTask", 1 '>0
criaserver = 1
'Não faz mais nada no momento.
'Uma entidade tipo vHARD e atributo aServerTask > 0 será criada e este procedimento será
chamado novamente (no mesmo instante de tempo simulado, sem delay)
'DEVE INICIALIZAR OS ATRIBUTOS DA TAREFA QUANDO FOR CRIADA
'passa o parâmetro necessário para inicializar os atributos (ID da tarefa aperiódica que irá atender)
gIDAperiodicServida(esteNo) = Get_Atributo_Fila("recFilaAperiodic_Q", indiceAperiodic,
"aNumAtividade")
gServerTaskAssociada(esteNo) = 0 'será atualizado quando a server task for inicializada
'a capacidade continua a mesma
End If
Else
'não há mais tarefas aperiódicas para servir. Torna o servidor inativo até a próxima ativação
gCapacidadeServidor(esteNo) = 0
End If
End If
End If

'verifica se deve consumir capacidade da server task
If prioridadeExecutando > -1 Then
'verifica se a tarefa que está executando é a server task
Dim SERVER As Double
SERVER = Get_Atributo_Fila("recFilaExecutando_Q", indiceExecutando, "aServerTask")
If SERVER > 0 Then
'a server task está executando. Consume capacidade e diminui o tempo de execucao restante da
tarefa aperiódica
Dim inicioExec As Double, faltaExecutar As Double
inicioExec = Get_Atributo_Fila("recFilaExecutando_Q", indiceExecutando, "aTempoInicExec")
If gUltimoConsumo(esteNo) < inicioExec Then
gUltimoConsumo(esteNo) = inicioExec
End If
gCapacidadeServidor(esteNo) = gCapacidadeServidor(esteNo) - (agora -
gUltimoConsumo(esteNo))
gUltimoConsumo(esteNo) = agora
'verifica se acabou a capacidade ou se terminou de executar a tarefa aperiódica
faltaExecutar = Get_Atributo_Fila("recFilaExecutando_Q", indiceExecutando,
"aTempoExecRestante") - (agora - inicioExec)
If (gCapacidadeServidor(esteNo) <= 0.01) Or (faltaExecutar < 0.01) Then
'preempta a server task
gCapacidadeServidor(esteNo) = 0
preempta = indiceExecutando 'retira Server Task da CPU
Else
'verifica se deve preemptar a server task antes do próximo quantum
Dim quantum As Double
quantum = Get_Variavel("vQuantum")
If (faltaExecutar < quantum) Or (gCapacidadeServidor(esteNo) < quantum) Then
'próxima entidade de controle antes do próximo quantum. Seleciona o evento mais próximo
If faltaExecutar < gCapacidadeServidor(esteNo) Then
tmpRestante = faltaExecutar
Else
tmpRestante = gCapacidadeServidor(esteNo)
End If

```

```

        End If
    End If
End If
End If

'verifica se deve preemptar e escalonar outra de maior prioridade
If prioridadeFilaPeriodic > -1 Then
    If (prioridadePeriodic < prioridadeExecutando) And (prioridadeExecutando > -1) Then
        'há tarefa executando e deve ser preemptada
        preempta = indiceExecutando
    Else
        'não há tarefa executando. Simplesmente escalona a tarefa que espera
        If Get_Atributo_Fila("recFilaPeriodic_Q", indicePeriodic, "aServerTask") > 0 Then
            'é a server task. Só pode escalonar se ela tiver capacidade > 0
            If gCapacidadeServidor(esteNo) > 0 Then
                escalona = indicePeriodic
                periodo = 1 'sempre escalona tarefa periódica
            End If
        Else
            escalona = indicePeriodic
            periodo = 1 'sempre escalona tarefa periódica
        End If
    End If
End If
End If
End Sub

```

```

Public Sub Escalonamento_DS(prioridadeFilaPeriodic As Double, prioridadeFilaAperiodic As Double,
prioridadeExecutando As Double, indicePeriodic%, indiceAperiodic%, indiceExecutando%,
tmpRestante#, preempta#, escalona#, periodo#, criaserver)

```

```

'DEFERREBLE SERVER (5)

```

```

Dim agora As Double, periodoServ As Double
Dim posPeriodic As Double, posAperiodic As Double, posExec As Double
Dim esteNo As Double

```

```

'verifica se é tempo de ativação da Server Task
'pega informações do modelo
agora = S.RunCurrentTime
periodoServ = Get_Expressao("ePeriodoServerTask")
esteNo = Get_Atributo("aNoOrigem")

```

```

'verifica se é tempo de ativação da Server Task
If (gUltimaAtivacao(esteNo) + Int(periodoServ)) <= Int(agora) Then
    'início de um novo período da server task
    'procura a server task no sistema
    Procura_Task_Server posPeriodic, posAperiodic, posExec
    If (posPeriodic <= 0) And (posAperiodic <= 0) And (posExec <= 0) Then
        'não há nenhuma server task no sistema.
        'Se houver tarefas aperiódicas pendentes, então manda criar uma entidade Task Server
        If prioridadeFilaAperiodic > -1 Then
            'Set_Atributo "aCriaServerTask", 1 '>0
            criaserver = 1
            'Não faz mais nada no momento.
            'Uma entidade tipo vHARD e atributo aServerTask > 0 será criada e este procedimento será
            chamado novamente (no mesmo instante de tempo simulado, sem delay)
            'DEVE INICIALIZAR OS ATRIBUTOS DA TAREFA QUANDO FOR CRIADA

```

```

'passa o parâmetro necessário para inicializar os atributos (ID da tarefa aperiódica que irá atender)
gIDAperiodicServida(esteNo) = Get_Atributo_Fila("recFilaAperiodic_Q", indiceAperiodic,
"aNumAtividade")
gServerTaskAssociada(esteNo) = 0 'será atualizado quando a server task for inicializada
'reestabelece a capacidade ao máximo
gCapacidadeServidor(esteNo) = Get_Expressao("eCapacidadeMaxima")
Else
'Não há nenhuma tarefa aperiódica no instante de ativação de PS. Só poderá atender no próximo
período
'Não cria server task
End If
Else
'no sistema há uma "server task" periódica que atende tarefas aperiódicas
'deve ter sua capacidade resstabelecida
gCapacidadeServidor(esteNo) = Get_Expressao("eCapacidadeMaxima")
'atualiza a ultima ativacao
End If
'atualiza o instante de ativação
gUltimaAtivacao(esteNo) = agora
Else
'verifica caso da server task ter terminado de atender uma tarefa aperiódica,
'ter sobrado capacidade e ter outras tarefas aperiódicas esperando.
'Nesse caso deve recriar a server task até a capacidade esgotar ou terminar de atender as aperiódicas
If (gCapacidadeServidor(esteNo) > 0) Then
'sobrou capacidade da última tarefa aperiódica atendida nesta ativação
If (prioridadeFilaAperiodic > -1) Then
'Há capacidade e há tarefas esperando. Portanto continua atendendo
Procura_Task_Server posPeriodic, posAperiodic, posExec
If (posPeriodic <= 0) And (posAperiodic <= 0) And (posExec <= 0) Then
'e não há nenhuma server task no sistema. É o caso apresentado. Deve criar novamente a server
task com a capacidade que lhe resta
'Set_Atributo "aCriaServerTask", 1 '>0
criaserver = 1
'Não faz mais nada no momento.
'Uma entidade tipo vHARD e atributo aServerTask > 0 será criada e este procedimento será
chamado novamente (no mesmo instante de tempo simulado, sem delay)
'DEVE INICIALIZAR OS ATRIBUTOS DA TAREFA QUANDO FOR CRIADA
'passa o parâmetro necessário para inicializar os atributos (ID da tarefa aperiódica que irá atender)
gIDAperiodicServida(esteNo) = Get_Atributo_Fila("recFilaAperiodic_Q", indiceAperiodic,
"aNumAtividade")
gServerTaskAssociada(esteNo) = 0 'será atualizado quando a server task for inicializada
'a capacidade continua a mesma
Else
'o servidor ainda está no sistema
End If
Else
'não há mais tarefas aperiódicas para servir.
'mas o servidor retém sua capacidade para atender possíveis tarefas aperiódicas que venham a
iniciar durante esse período
End If
End If
End If

'verifica se deve consumir capacidade da server task
If prioridadeExecutando > -1 Then
'verifica se a tarefa que está executando é a server task
Dim SERVER As Double
SERVER = Get_Atributo_Fila("recFilaExecutando_Q", indiceExecutando, "aServerTask")

```

```

If SERVER > 0 Then
  'a server task está executando. Consome capacidade e diminui o tempo de execucao restante da
  tarefa aperiódica
  Dim inicioExec As Double, faltaExecutar As Double
  inicioExec = Get_Atributo_Fila("recFilaExecutando_Q", indiceExecutando, "aTempoInicExec")
  If gUltimoConsumo(esteNo) < inicioExec Then
    gUltimoConsumo(esteNo) = inicioExec
  End If
  gCapacidadeServidor(esteNo) = gCapacidadeServidor(esteNo) - (agora -
gUltimoConsumo(esteNo))
  gUltimoConsumo(esteNo) = agora
  'verifica se acabou a capacidade ou se terminou de executar a tarefa aperiódica
  faltaExecutar = Get_Atributo_Fila("recFilaExecutando_Q", indiceExecutando,
"aTempoExecRestante") - (agora - inicioExec)
  If (gCapacidadeServidor(esteNo) <= 0.01) Or (faltaExecutar < 0.01) Then
    'preempta a server task
    gCapacidadeServidor(esteNo) = 0
    preempta = indiceExecutando 'retira Server Task da CPU
  Else
    'verifica se deve preemptar a server task antes do próximo quantum
    Dim quantum As Double
    quantum = Get_Variavel("vQuantum")
    If (faltaExecutar < quantum) Or (gCapacidadeServidor(esteNo) < quantum) Then
      'próxima entidade de controle antes do próximo quantum. Seleciona o evento mais próximo
      If faltaExecutar < gCapacidadeServidor(esteNo) Then
        tmpRestante = faltaExecutar
      Else
        tmpRestante = gCapacidadeServidor(esteNo)
      End If
    End If
  End If
End If
End If
End If
End If

'verifica se deve preemptar e escalonar outra de maior prioridade
If prioridadeFilaPeriodic > -1 Then
  If (prioridadePeriodic < prioridadeExecutando) And (prioridadeExecutando > -1) Then
    'há tarefa executando e deve ser preemptada
    preempta = indiceExecutando
  Else
    'não há tarefa executando. Simplesmente escalona a tarefa que espera
    If Get_Atributo_Fila("recFilaPeriodic_Q", indicePeriodic, "aServerTask") > 0 Then
      'é a server task. Só pode escalonar se ela tiver capacidade > 0
      If gCapacidadeServidor(esteNo) > 0 Then
        escalona = indicePeriodic
        periodo = 1 'sempre escalona tarefa periódica
      End If
    Else
      escalona = indicePeriodic
      periodo = 1 'sempre escalona tarefa periódica
    End If
  End If
End If
End If
End Sub

```

```
Public Sub Escalonamento_SS(prioridadeFilaPeriodic As Double, prioridadeFilaAperiodic As Double,
prioridadeExecutando As Double, indicePeriodic%, indiceAperiodic%, indiceExecutando%,
tmpRestante#, preempta#, escalona#, periodo#, criaserver)
```

```
'SPORADIC SERVER (7)
```

```
Dim agora As Double, periodoServ As Double
Dim posPeriodic As Double, posAperiodic As Double, posExec As Double
Dim esteNo As Double, estavaAtivo As Boolean, capacidadeAntiga As Double
Dim priorServer As Double, i As Integer, achou As Boolean
```

```
'pega informações do modelo
agora = S.RunCurrentTime
periodoServ = Get_Expressao("ePeriodoServerTask")
esteNo = Get_Atributo("aNoOrigem")
```

```
Procura_Task_Server posPeriodic, posAperiodic, posExec
If (posPeriodic <= 0) And (posAperiodic <= 0) And (posExec <= 0) Then
'nao ha server task neste NT
priorServer = -1 "
```

```
Else
'pega a prioridade da server task
If posPeriodic > 0 Then
priorServer = Get_Atributo_Fila("recFilaPeriodic_Q", (posPeriodic), "aPrioridade")
ElseIf posAperiodic > 0 Then
priorServer = Get_Atributo_Fila("recFilaAperiodic_Q", (posAperiodic), "aPrioridade")
Else
priorServer = Get_Atributo_Fila("recFilaExecutando_Q", (posExec), "aPrioridade")
End If
End If
```

```
estavaAtivo = gDSSAtivo(esteNo)
gDSSAtivo(esteNo) = (prioridadeExecutando <= priorServer) And (priorServer > -1) 'está ativo se
quem executa é pelo menos tao prioritário quanto o servidor
gDSSAtivo(esteNo) = gDSSAtivo(esteNo) Or ((prioridadeExecutando > -1) And
(prioridadeExecutando <= Get_Expressao("ePeriodoServerTask"))) And (priorServer = -1))
If (priorServer = -1) And (prioridadeFilaAperiodic > -1) Then
'nao há uma task server e há tarefas aperiódicas pendentes.
```

```
'cria uma server task
criaserver = 1
'Torna-se ativo
gDSSAtivo(esteNo) = True
'Não faz mais nada no momento.
'Uma entidade tipo vHARD e atributo aServerTask > 0 será criada e este procedimento será chamado
novamente (no mesmo instante de tempo simulado, sem delay)
'DEVE INICIALIZAR OS ATRIBUTOS DA TAREFA QUANDO FOR CRIADA
'passa o parâmetro necessário para inicializar os atributos (ID da tarefa aperiódica que irá atender)
gIDAperiodicServida(esteNo) = Get_Atributo_Fila("recFilaAperiodic_Q", indiceAperiodic,
"aNumAtividade")
gServerTaskAssociada(esteNo) = 0 'será atualizado quando a server task for inicializada
'reestabelece a capacidade ao máximo
'gCapacidadeServidor(esteNo) = Get_Expressao("eCapacidadeMaxima")
Registra_Log True, "CRIA SERVER TASK"
End If
```

```
If (Not estavaAtivo And gDSSAtivo(esteNo)) And (gCapacidadeServidor(esteNo) > 0) Then
'recem se tornou ativo, e CS > 0, portanto TNOW = ta
'ajusta o instante de reestabelecimento RT
```

```

'procura uma entrada vazia na lista de RTs
i = 1
achou = False
While (i <= 16) And Not achou
  achou = (gRT(esteNo, i) = 0)
  i = i + 1
Wend
If achou Then
  'caso normal. Inclui o instante de reestabelecimento na lista (nao está em ordem cronológica)
  i = i - 1
Else
  i = 1 'lista cheia. substitui o primeiro elemento.
  MsgBox ("Lista de RTs cheia. Debug!!")
End If
gRT(esteNo, i) = agora + Get_Expressao("ePeriodoServerTask")
Registra_Log True, "ATUALIZA RT" + Trim(Str(i)) + "=" + Str(gRT(esteNo, i))
End If

'verifica se deve consumir capacidade da server task
Dim SERVER As Double
capacidadeAntiga = gCapacidadeServidor(esteNo)
tmpRestante = Get_Variavel("vQuantum")
If prioridadeExecutando > -1 Then
  'verifica se a tarefa que está executando é a server task
  SERVER = Get_Atributo_Fila("recFilaExecutando_Q", indiceExecutando, "aServerTask")
  If SERVER > 0 Then
    'a server task está executando. Consome capacidade e diminui o tempo de execucao restante da
    tarefa aperiódica
    Dim inicioExec As Double, faltaExecutar As Double
    inicioExec = Get_Atributo_Fila("recFilaExecutando_Q", indiceExecutando, "aTempoInicExec")
    If gUltimoConsumo(esteNo) < inicioExec Then
      gUltimoConsumo(esteNo) = inicioExec
    End If
    gCapacidadeServidor(esteNo) = gCapacidadeServidor(esteNo) - (agora -
    gUltimoConsumo(esteNo))
    gConsumido(esteNo) = gConsumido(esteNo) + (agora - gUltimoConsumo(esteNo))
    gUltimoConsumo(esteNo) = agora
    Registra_Log True, "CONSOME CAPACIDADE SERVER TASK. CONSUMIDO=" +
    Str(gConsumido(esteNo)) + " CAPACIDADE=" + Str(gCapacidadeServidor(esteNo))
    If gCapacidadeServidor(esteNo) <= 0.01 Then
      'tarefa esgotou a capacidade do servidor
      preempta = indiceExecutando
      'está inativo
      gDSSAtivo(esteNo) = False
      Registra_Log True, "PREEMPTA TASK"
    ElseIf gCapacidadeServidor(esteNo) < Get_Variavel("vQuantum") Then
      'vai consumir toda a capacidade antes do próximo quantum.
      'determina que uma entidade de controle deve ser criada nesse instante
      tmpRestante = gCapacidadeServidor(esteNo)
    End If
  End If
End If

If (estavaAtivo And Not gDSSAtivo(esteNo)) Or ((capacidadeAntiga > 0) And
(gCapacidadeServidor(esteNo) = 0)) Then
  'se tornou inativo ou recém consumiu a capacidade do servidor, portanto TNOW = ti
  'ajusta o total de capacidade a ser reestabelecido (RA)
  'procura uma entrada vazia na lista de RAs

```

```

i = 1
achou = False
While (i <= 16) And Not achou
    achou = (gRA(esteNo, i) = 0)
    i = i + 1
Wend
If achou Then
    'caso normal. Inclui o total de reestabelecimento na lista (nao está em ordem cronológica)
    i = i - 1
Else
    i = 1 'lista cheia. substitui o primeiro elemento.
    MsgBox ("Lista de RAs cheia. Debug!!")
End If
gRA(esteNo, i) = gConsumido(esteNo)
Registra_Log True, "Atualzia RA" + Trim(Str(i)) + "=" + Str(gRA(esteNo, i))
End If

```

'verifica se há algum esse instante é RT, o instante de reestabelecimento da capacidade  
'procura um RT na lista que seja agora

```

i = 1
achou = False
While (i <= 16) And Not achou
    achou = (Abs(gRT(esteNo, i) - agora) <= 0.01) 'o reestabelecimento é agora
    i = i + 1
Wend
If achou Then
    i = i - 1
    'Tempo de reestabelecimento. Reestabelece a capacidade do servidor
    gCapacidadeServidor(esteNo) = gCapacidadeServidor(esteNo) + gRA(esteNo, i)
    Registra_Log True, "RT: RESTABELECE CAPACIDADE SERVER TASK. CAPACIDADE=" +
Str(gCapacidadeServidor(esteNo))
    'limpa RT e RA
    gRT(esteNo, i) = 0
    gRA(esteNo, i) = 0
    Registra_Log True, "LIMPA RT" + Trim(Str(i)) + " e RT" + Trim(Str(i))
End If

```

If Not achou Then

'agora nao é RT, então verifica se algum RT deve acontecer antes do proximo quantum,  
'para que uma entidade de controle seja criada exatamente nesse instante

```

i = 1
achou = 0
While (i <= 16)
    If (gRT(esteNo, i) > agora) And (gRT(esteNo, i) - agora < tmpRestante) Then
        tmpRestante = gRT(esteNo, i) - agora
    End If
    i = i + 1
Wend
'pega o menor RT
End If

```

If (prioridadeFilaPeriodic > 0) Then

'Ha tarefas periodicas pendentes

If prioridadeExecutando > -1 Then

If (prioridadeFilaPeriodic < prioridadeExecutando) Then

'preempta a tarefa atual

preempta = indiceExecutando

Registra\_Log True, "PREEMPTA TASK"

```

        'escalona tarefa esperando
        escalona = indicePeriodic
        periodo = 1 'sempre periodica
    End If
Else
    'CPU ociosa. simplesmente escalona tarefa esperando (se nao for uma server task com capacidade 0)
    If Not ((Get_Atributo_Fila("recFilaPeriodic_Q", indicePeriodic, "aServerTask") > 0) And
(gCapacidadeServidor(esteNo) <= 0.01)) Then
        escalona = indicePeriodic
        Registra_Log True, "ESCALONA A TAREFA PERIODICA"
        periodo = 1 'sempre periodica
    End If
End If
End If
End Sub

```

```

Public Sub Escalonamento_RED(prioridadeFilaPeriodic As Double, prioridadeFilaAperiodic As Double,
prioridadeExecutando As Double, indicePeriodic%, indiceAperiodic%, indiceExecutando%,
tmpRestante#, preempta#, escalona#, periodo#, criaserver)

```

```

    'ROBUST EARLIEST DEADLINE (9)

```

```

    MsgBox "INDISPONÍVEL"
End Sub

```

```

Public Sub Escalonamento_ABS(prioridadeFilaPeriodic As Double, prioridadeFilaAperiodic As Double,
prioridadeExecutando As Double, indicePeriodic%, indiceAperiodic%, indiceExecutando%,
tmpRestante#, preempta#, escalona#, periodo#, criaserver)

```

```

    'ADAPTATIVE BANDWIDTH SERVER (10)

```

```

    MsgBox "INDISPONÍVEL"
End Sub

```

```

Private Sub Procura_Task_Server(posPeriodic As Double, posAperiodic As Double, posExecutando As
Double)

```

```

    Dim achou As Boolean, esteNo As Double, numElementos As Double

```

```

    esteNo = Get_Atributo("aNoOrigem")

```

```

    'procura na fila periódica

```

```

    numElementos = Get_Num_Elementos_Fila("recFilaPeriodic_Q")

```

```

    If numElementos = 0 Then

```

```

        posPeriodic = 0 'não está na fila periódica

```

```

    Else

```

```

        'procura entre as entidades da lista

```

```

        posPeriodic = 1

```

```

        achou = False

```

```

        While (posPeriodic <= numElementos) And Not achou

```

```

            achou = (Get_Atributo_Fila("recFilaPeriodic_Q", (posPeriodic), "aNoOrigem") = esteNo) And
(Get_Atributo_Fila("recFilaPeriodic_Q", (posPeriodic), "aServerTask") > 0)

```

```

            posPeriodic = posPeriodic + 1

```

```

        Wend

```

```

        If achou Then

```

```

            posPeriodic = posPeriodic - 1 'ajusta o índice correto da server task

```

```

            'não precisa procurar em nenhum outro lugar. pode terminar agora

```

```

            posAperiodic = 0

```

```

    posExecutando = 0
    Exit Sub
Else
    posPeriodic = 0 'não está na fila de periódicas
End If
End If

'procura na fila aperiódica
numElementos = Get_Num_Elementos_Fila("recFilaAperiodic_Q")
If numElementos = 0 Then
    posAperiodic = 0 'não está na fila aperiódica
Else
    'procura entre as entidades da lista
    posAperiodic = 1
    achou = False
    While (posAperiodic <= numElementos) And Not achou
        achou = (Get_Atributo_Fila("recFilaAperiodic_Q", (posAperiodic), "aNoOrigem") = esteNo) And
(Get_Atributo_Fila("recFilaAperiodic_Q", (posAperiodic), "aServerTask") > 0)
        posAperiodic = posAperiodic + 1
    Wend
    If achou Then
        posAperiodic = posAperiodic - 1 'ajusta o índice correto da server task
        'não precisa procurar em nenhum outro lugar. pode terminar agora
        posPeriodic = 0
        posExecutando = 0
        Exit Sub
    Else
        posAperiodic = 0 'não está na fila de aperiódicas
    End If
End If

'procura na fila executando
numElementos = Get_Num_Elementos_Fila("recFilaExecutando_Q")
If numElementos = 0 Then
    posExecutando = 0 'não está na fila periódica
Else
    'procura entre as entidades da lista
    posExecutando = 1
    achou = False
    While (posExecutando <= numElementos) And Not achou
        achou = (Get_Atributo_Fila("recFilaExecutando_Q", (posExecutando), "aNoOrigem") = esteNo)
And (Get_Atributo_Fila("recFilaExecutando_Q", (posExecutando), "aServerTask") > 0)
        posExecutando = posExecutando + 1
    Wend
    If achou Then
        posExecutando = posExecutando - 1 'ajusta o índice correto da server task
        'não precisa procurar em nenhum outro lugar. pode terminar agora
        posAperiodic = 0
        posPeriodic = 0
        Exit Sub
    Else
        posExecutando = 0 'não está na fila de periódicas
    End If
End If

End Sub

Function Str_Escalonamento(numero As Double) As String

```

```
Select Case numero
Case 1
  Str_Escalonamento = "EDF"
Case 2
  Str_Escalonamento = "RM"
Case 3
  Str_Escalonamento = "BS"
Case 4
  Str_Escalonamento = "PS"
Case 5
  Str_Escalonamento = "DS"
Case 6
  Str_Escalonamento = "CBR"
Case 7
  Str_Escalonamento = "SS"
Case 8
  Str_Escalonamento = "DSS"
Case 9
  Str_Escalonamento = "RED"
Case 10
  Str_Escalonamento = "ABS"
Case Else
  Str_Escalonamento = "unknown"
  MsgBox "Escalonamento desconhecido. Debug!"
End Select
End Function
```

## ANEXO 7 – Código-fonte VBA, parte integrante do modelo.

### Arquivo “mFuncoes.bas”

```
Attribute VB_Name = "mFuncoes"  
Option Explicit
```

```
Public Function Get_Atributo(nome As String) As Double  
    on error GoTo Get_Atributo_Err  
    Dim aux As Long, result As Double  
  
    aux = S.SymbolNumber(nome)  
    result = S.AttributeValue(S.ActiveEntity, aux, 0, 0)  
    Get_Atributo = result  
Exit Function
```

```
Get_Atributo_Err:  
    MsgBox "Não pude buscar o valor do atributo " + nome  
    Resume Next  
End Function
```

```
Public Function Get_Variavel(nome As String) As Double  
    on error GoTo Get_Variavel_Err  
    Dim aux As Long, result As Double  
  
    aux = S.SymbolNumber(nome)  
    result = S.VariableValue(aux, 0, 0)  
    Get_Variavel = result  
Exit Function
```

```
Get_Variavel_Err:  
    MsgBox "Não pude buscar o valor da variável " + nome  
    Resume Next  
End Function
```

```
Public Function Get_VariavelArray(nome As String, posicao As Integer) As Double  
    on error GoTo Get_VariavelArray_Err  
    Dim aux As Long, result As Double  
  
    'que função usar?  
    ' aux = S.SymbolNumber(nome)  
    ' result = S.VariableValue(aux, posicao, 0)  
    Get_VariavelArray = result  
Exit Function
```

```
Get_VariavelArray_Err:  
    MsgBox "Não pude buscar o valor da variável " + nome  
    Resume Next  
End Function
```

```
Public Function Get_Expressao(nome As String) As Double
```

```
on error GoTo Get_Expressao_Err  
Dim aux As Long, result As Double
```

```
aux = S.SymbolNumber(nome)  
result = S.ExpressionValue(aux)  
Get_Expressao = result  
Exit Function
```

```
Get_Expressao_Err:  
MsgBox "Não pude buscar o valor da expressao " + nome  
Resume Next  
End Function
```

```
Public Sub Set_Atributo(nome As String, valor As Double)  
on error GoTo Set_Atributo_Err  
Dim aux As Long, result As Double
```

```
aux = S.SymbolNumber(nome)  
S.EntityAttribute(S.ActiveEntity, aux) = valor  
Exit Sub
```

```
Set_Atributo_Err:  
MsgBox "Não pude alterar o valor do atributo " + nome  
Resume Next  
End Sub
```

```
Public Sub Set_Variavel(nome As String, valor As Double)  
on error GoTo Set_Variavel_Err  
Dim aux As Long, result As Double
```

```
aux = S.SymbolNumber(nome)  
S.VariableArrayValue(aux) = valor  
'S.VariableValue(aux, 0, 0) = valor  
Exit Sub
```

```
Set_Variavel_Err:  
MsgBox "Não pude alterar o valor da variável " + nome  
Resume Next  
End Sub
```

```
Public Sub Set_VariavelArray(nome As String, posicao As Integer, valor As Double)  
on error GoTo Set_VariavelArray_Err  
Dim aux As Long, result As Double
```

```
'aux = S.SymbolNumber(nome)  
" que função usar?  
'S.VariableArrayValue(aux,posicao) = valor  
Exit Sub
```

```
Set_VariavelArray_Err:  
MsgBox "Não pude alterar o valor da variável " + nome  
Resume Next  
End Sub
```

```
Public Sub Set_Picture(nome As String)
    on error GoTo Set_Picture_Err
    Dim aux As Long
```

```
    aux = S.SymbolNumber(nome)
    S.EntitySetPicture S.ActiveEntity, aux
    Exit Sub
```

```
Set_Picture_Err:
    MsgBox "Não pode alterar o número da Figura da Entidade"
    Resume Next
End Sub
```

```
Public Function Get_Atributo_Fila(fila As String, posicao As Integer, nome As String) As Double
    on error GoTo Get_Atributo_Fila_Err
    Dim numFila As Long, numAtrib As Long, result As Double
```

```
    numFila = S.SymbolNumber(fila)
    numAtrib = S.SymbolNumber(nome)
    result = S.QueuedEntityAttribute(numFila, posicao, numAtrib)
    Get_Atributo_Fila = result
    Exit Function
```

```
Get_Atributo_Fila_Err:
    MsgBox "Não pode buscar o valor do atributo " + nome
    Resume Next
End Function
```

```
Public Function Get_Num_Elementos_Fila(nome As String) As Integer
    on error GoTo Get_Num_Elementos_Fila_Err
    Dim aux As Long
```

```
    aux = S.SymbolNumber(nome)
    Get_Num_Elementos_Fila = S.QueueNumberOfEntities(aux)
    Exit Function
```

```
Get_Num_Elementos_Fila_Err:
    MsgBox ("Não pode pegar o número de elementos da fila " + nome)
    Resume Next
End Function
```

```
Sub Registra_Log(informacoes As Boolean, evento As String)
    on error GoTo Registra_Log_Err
```

```
    Dim numEnt As Long, tempo As Double, tipo As Double, numTarefa As Double
    Dim texto As String, strTipo As String
    Dim Nodo As String
    Static lastTime As Double
```

```
    Exit Sub
```

```
    If gCabecalho Then
        'apresenta o cabeçalho
        Open "c:\CRUX_LOG.TXT" For Append As #1
        Print #1, "Parâmetros do Modelo:"
        Print #1, " Parâmetros do Sistema: NTs:" + Str(Get_Variavel("vMaxNodos")) + "; Canais:" +
        Str(Get_Variavel("vMaxCanais"))
```

```

        Print #1, "          Veloc.Crossbar: " + Str(Get_Variavel("vVelocCrossbar")) + "MB/s;
Veloc.BS: " + Str(Get_Variavel("vVelocBarramento")) + "MB/s"
        Print #1, "          Tamanho Req.: " + Str(Get_Variavel("vTamanhoPedido")) + "bytes;
Algoritmo de Escalonamento: " + Str_Escalonamento(Get_Variavel("vEscalonamento"))
        Print #1, " Parâmetros de Carga : FatorHard:" + Str(Get_Variavel("vFator_U_Hard")) + ";
FatorSoft:" + Str(Get_Variavel("vFator_U_Soft")) + "; FatorCBR:" +
Str(Get_Variavel("vFator_U_CBR"))
        Print #1, "          ArquivosVBR:" + Str(36) + "; AmostrasCBR:" +
Str(Get_Variavel("vMaxAmostras"))
        Print #1, "          PeríodoAmostraCBR:" + Str(Get_Expressao("ePeriodoCBR")) + ";
TempoProcNC:" + Str(Get_Expressao("eTempoProcNC"))
        Print #1, " Parâmetros do Projeto: Tempo Simulação:" + Str(S.RunEndTime) + "; Replicações:"
+ Str(S.RunMaximumReplications)
        Print #1, ""
        Close #1
        gCabecalho = False
    Else
        lastTime = 0
    End If

    numEnt = S.EntityNumber(S.ActiveEntity)
    numTarefa = Get_Atributo("aNumAtividade")
    tempo = S.RunCurrentTime
    tipo = Get_Atributo("aTipo")
    If tipo = Get_Variavel("vHARD") Then
        strTipo = "HARD"
    ElseIf tipo = Get_Variavel("vVBR") Then
        strTipo = "VBR"
    ElseIf tipo = Get_Variavel("vCBR") Then
        strTipo = "CBR"
    Else
        strTipo = "QUANTUM"
    End If
    If Get_Atributo("aServerTask") > 0 Then
        strTipo = "SERVER"
    End If
    If informacoes Then
        Nodo = "NT" + Trim(Str(Get_Atributo("aNoOrigem")))
    Else
        Nodo = "NC"
    End If
    texto = "Tempo: " + Format(tempo, "0,000,000.00") + " - Ent:" + Str(numTarefa) + " (" + strTipo + ") -
" + Nodo

    texto = texto + " - " + evento
    Open "c:\CRUX_LOG.TXT" For Append As #1
        If tempo > lastTime Then
            Print #1, ""
        End If
        Print #1, texto
    Close #1
    lastTime = tempo
    Exit Sub

Registra_Log_Err:
    MsgBox ("Erro ao registrar log")
    Resume Next
End Sub

```

## ANEXO 8 – Código-fonte VBA, parte integrante do modelo.

### Arquivo “mGlobal.bas”

Global gCanalFisico(1 To 32, 1 To 8) As Integer 'primeiro índice é o nodo origem. O valor é o nodo destino (0 se canal livre)

Global gUltimaExecucaoCBR(1 To 32, 1 To 8, 1 To 2) As Double 'para o cálculo da variação da taxa de execução CBR  
'nodos , cbr , 1=id,2=ultimaexec

Global gServidorExecutando(1 To 32) As Boolean

Global gCapacidadeServidor(1 To 32) As Double

Global gUltimaAtivacao(1 To 32) As Double 'instante da última ativação da tarefa servidora

Global gUltimoConsumo(1 To 32) As Double 'quantidade consumida durante a última ativação da tarefa servidora

Global gIDAperiodicServida(1 To 32) As Double 'id da tarefa aperiodica servida

Global gServerTaskAssociada(1 To 32) As Double 'id da tarefa que esta' servindo

Global gConsumido(1 To 32) As Double 'capacidade consumida no intervalo (SS, DSS)

Global gRT(1 To 32, 1 To 16) As Double 'instante de reestabelecimento (SS, DSS). Temos uma lista (até 10) de reestabelecimentos

Global gRA(1 To 32, 1 To 16) As Double 'capacidade a ser reestabelecida (SS, DSS). Temos uma lista (até 10) de reestabelecimentos

Global gDSSAtivo(1 To 32) As Double 'informa se o servidor SS ou DSS está ativo.

Global gCabecalho As Boolean 'se deve ou nao imprimir o cabecalho no arquivo lo log (começa true e logo vai para false)

'objetos para acesso ao modelo

Global M As Model

Global S As SIMAN

## ANEXO 9 – Arquivo de Log gerado pelo modelo.

### Registro de atividades *hard real time usando algoritmo DS*

#### MODELO DO MULTICOMPUTADOR CRUX - CPGCC - UFSC

#### Avaliação de Desempenho de Algoritmos de Escalonamento de Tempo Real

##### Parâmetros do Modelo:

Parâmetros do Sistema: NTs: 8; Canais: 2

Veloc.Crossbar: 100MB/s; Veloc.BS: 100MB/s

Tamanho Req.: 64bytes; Algoritmo de Escalonamento: EDF

Parâmetros de Carga : FatorHard: .4; FatorSoft: .4; FatorCBR: .4

ArquivosVBR: 36; AmostrasCBR: 100

PeriodoAmostraCBR: 62.5; TempoProcNC: 4

Parâmetros do Projeto: Tempo Simulação: 2000; Replicações: 1

Tempo: 000.000,00 - Ent: 1 (HARD) - NT1 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 000.000,00 - Ent: 1 (HARD) - NT1 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 000.062,50 - Ent: 3 (HARD) - NT2 - Tarefa Chega com Prioridade 562.5 e 0 comunicações  
Tempo: 000.062,50 - Ent: 3 (HARD) - NT2 - Tarefa Inicia Execução com Prioridade 562.5  
Tempo: 000.125,00 - Ent: 5 (HARD) - NT3 - Tarefa Chega com Prioridade 625 e 0 comunicações  
Tempo: 000.125,00 - Ent: 5 (HARD) - NT3 - Tarefa Inicia Execução com Prioridade 625  
Tempo: 000.187,50 - Ent: 7 (HARD) - NT4 - Tarefa Chega com Prioridade 687.5 e 0 comunicações  
Tempo: 000.187,50 - Ent: 7 (HARD) - NT4 - Tarefa Inicia Execução com Prioridade 687.5  
Tempo: 000.200,00 - Ent: 1 (HARD) - NT1 - Tarefa Termina Execução  
Tempo: 000.200,00 - Ent: 1 (HARD) - NT1 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.250,00 - Ent: 9 (HARD) - NT5 - Tarefa Chega com Prioridade 750 e 0 comunicações  
Tempo: 000.250,00 - Ent: 9 (HARD) - NT5 - Tarefa Inicia Execução com Prioridade 750  
Tempo: 000.262,50 - Ent: 3 (HARD) - NT2 - Tarefa Termina Execução  
Tempo: 000.262,50 - Ent: 3 (HARD) - NT2 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.312,50 - Ent: 11 (HARD) - NT6 - Tarefa Chega com Prioridade 812.5 e 0 comunicações  
Tempo: 000.312,50 - Ent: 11 (HARD) - NT6 - Tarefa Inicia Execução com Prioridade 812.5  
Tempo: 000.325,00 - Ent: 5 (HARD) - NT3 - Tarefa Termina Execução  
Tempo: 000.325,00 - Ent: 5 (HARD) - NT3 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.375,00 - Ent: 13 (HARD) - NT7 - Tarefa Chega com Prioridade 875 e 0 comunicações  
Tempo: 000.375,00 - Ent: 13 (HARD) - NT7 - Tarefa Inicia Execução com Prioridade 875  
Tempo: 000.387,50 - Ent: 7 (HARD) - NT4 - Tarefa Termina Execução  
Tempo: 000.387,50 - Ent: 7 (HARD) - NT4 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.437,50 - Ent: 15 (HARD) - NT8 - Tarefa Chega com Prioridade 937.5 e 0 comunicações  
Tempo: 000.437,50 - Ent: 15 (HARD) - NT8 - Tarefa Inicia Execução com Prioridade 937.5  
Tempo: 000.450,00 - Ent: 9 (HARD) - NT5 - Tarefa Termina Execução  
Tempo: 000.450,00 - Ent: 9 (HARD) - NT5 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.500,00 - Ent: 17 (HARD) - NT1 - Tarefa Chega com Prioridade 1000 e 0 comunicações  
Tempo: 000.500,00 - Ent: 17 (HARD) - NT1 - Tarefa Inicia Execução com Prioridade 1000  
Tempo: 000.512,50 - Ent: 11 (HARD) - NT6 - Tarefa Termina Execução  
Tempo: 000.512,50 - Ent: 11 (HARD) - NT6 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.562,50 - Ent: 19 (HARD) - NT2 - Tarefa Chega com Prioridade 1062.5 e 0 comunicações  
Tempo: 000.562,50 - Ent: 19 (HARD) - NT2 - Tarefa Inicia Execução com Prioridade 1062.5  
Tempo: 000.575,00 - Ent: 13 (HARD) - NT7 - Tarefa Termina Execução  
Tempo: 000.575,00 - Ent: 13 (HARD) - NT7 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.625,00 - Ent: 21 (HARD) - NT3 - Tarefa Chega com Prioridade 1125 e 0 comunicações  
Tempo: 000.625,00 - Ent: 21 (HARD) - NT3 - Tarefa Inicia Execução com Prioridade 1125  
Tempo: 000.637,50 - Ent: 15 (HARD) - NT8 - Tarefa Termina Execução  
Tempo: 000.637,50 - Ent: 15 (HARD) - NT8 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.687,50 - Ent: 23 (HARD) - NT4 - Tarefa Chega com Prioridade 1187.5 e 0 comunicações

Tempo: 000.687,50 - Ent: 23 (HARD) - NT4 - Tarefa Inicia Execução com Prioridade 1187.5  
Tempo: 000.700,00 - Ent: 17 (HARD) - NT1 - Tarefa Termina Execução  
Tempo: 000.700,00 - Ent: 17 (HARD) - NT1 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.750,00 - Ent: 25 (HARD) - NT5 - Tarefa Chega com Prioridade 1250 e 0 comunicações  
Tempo: 000.750,00 - Ent: 25 (HARD) - NT5 - Tarefa Inicia Execução com Prioridade 1250  
Tempo: 000.762,50 - Ent: 19 (HARD) - NT2 - Tarefa Termina Execução  
Tempo: 000.762,50 - Ent: 19 (HARD) - NT2 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.812,50 - Ent: 27 (HARD) - NT6 - Tarefa Chega com Prioridade 1312.5 e 0 comunicações  
Tempo: 000.812,50 - Ent: 27 (HARD) - NT6 - Tarefa Inicia Execução com Prioridade 1312.5  
Tempo: 000.825,00 - Ent: 21 (HARD) - NT3 - Tarefa Termina Execução  
Tempo: 000.825,00 - Ent: 21 (HARD) - NT3 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.875,00 - Ent: 29 (HARD) - NT7 - Tarefa Chega com Prioridade 1375 e 0 comunicações  
Tempo: 000.875,00 - Ent: 29 (HARD) - NT7 - Tarefa Inicia Execução com Prioridade 1375  
Tempo: 000.887,50 - Ent: 23 (HARD) - NT4 - Tarefa Termina Execução  
Tempo: 000.887,50 - Ent: 23 (HARD) - NT4 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.937,50 - Ent: 31 (HARD) - NT8 - Tarefa Chega com Prioridade 1437.5 e 0 comunicações  
Tempo: 000.937,50 - Ent: 31 (HARD) - NT8 - Tarefa Inicia Execução com Prioridade 1437.5  
Tempo: 000.950,00 - Ent: 25 (HARD) - NT5 - Tarefa Termina Execução  
Tempo: 000.950,00 - Ent: 25 (HARD) - NT5 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.000,00 - Ent: 33 (HARD) - NT1 - Tarefa Chega com Prioridade 1500 e 0 comunicações  
Tempo: 001.000,00 - Ent: 33 (HARD) - NT1 - Tarefa Inicia Execução com Prioridade 1500  
Tempo: 001.012,50 - Ent: 27 (HARD) - NT6 - Tarefa Termina Execução  
Tempo: 001.012,50 - Ent: 27 (HARD) - NT6 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.062,50 - Ent: 35 (HARD) - NT2 - Tarefa Chega com Prioridade 1562.5 e 0 comunicações  
Tempo: 001.062,50 - Ent: 35 (HARD) - NT2 - Tarefa Inicia Execução com Prioridade 1562.5  
Tempo: 001.075,00 - Ent: 29 (HARD) - NT7 - Tarefa Termina Execução  
Tempo: 001.075,00 - Ent: 29 (HARD) - NT7 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.125,00 - Ent: 37 (HARD) - NT3 - Tarefa Chega com Prioridade 1625 e 0 comunicações  
Tempo: 001.125,00 - Ent: 37 (HARD) - NT3 - Tarefa Inicia Execução com Prioridade 1625  
Tempo: 001.137,50 - Ent: 31 (HARD) - NT8 - Tarefa Termina Execução  
Tempo: 001.137,50 - Ent: 31 (HARD) - NT8 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.187,50 - Ent: 39 (HARD) - NT4 - Tarefa Chega com Prioridade 1687.5 e 0 comunicações  
Tempo: 001.187,50 - Ent: 39 (HARD) - NT4 - Tarefa Inicia Execução com Prioridade 1687.5  
Tempo: 001.200,00 - Ent: 33 (HARD) - NT1 - Tarefa Termina Execução  
Tempo: 001.200,00 - Ent: 33 (HARD) - NT1 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.250,00 - Ent: 41 (HARD) - NT5 - Tarefa Chega com Prioridade 1750 e 0 comunicações  
Tempo: 001.250,00 - Ent: 41 (HARD) - NT5 - Tarefa Inicia Execução com Prioridade 1750  
Tempo: 001.262,50 - Ent: 35 (HARD) - NT2 - Tarefa Termina Execução  
Tempo: 001.262,50 - Ent: 35 (HARD) - NT2 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.312,50 - Ent: 43 (HARD) - NT6 - Tarefa Chega com Prioridade 1812.5 e 0 comunicações  
Tempo: 001.312,50 - Ent: 43 (HARD) - NT6 - Tarefa Inicia Execução com Prioridade 1812.5  
Tempo: 001.325,00 - Ent: 37 (HARD) - NT3 - Tarefa Termina Execução  
Tempo: 001.325,00 - Ent: 37 (HARD) - NT3 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.375,00 - Ent: 45 (HARD) - NT7 - Tarefa Chega com Prioridade 1875 e 0 comunicações  
Tempo: 001.375,00 - Ent: 45 (HARD) - NT7 - Tarefa Inicia Execução com Prioridade 1875  
Tempo: 001.387,50 - Ent: 39 (HARD) - NT4 - Tarefa Termina Execução  
Tempo: 001.387,50 - Ent: 39 (HARD) - NT4 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.437,50 - Ent: 47 (HARD) - NT8 - Tarefa Chega com Prioridade 1937.5 e 0 comunicações  
Tempo: 001.437,50 - Ent: 47 (HARD) - NT8 - Tarefa Inicia Execução com Prioridade 1937.5  
Tempo: 001.450,00 - Ent: 41 (HARD) - NT5 - Tarefa Termina Execução  
Tempo: 001.450,00 - Ent: 41 (HARD) - NT5 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.500,00 - Ent: 49 (HARD) - NT1 - Tarefa Chega com Prioridade 2000 e 0 comunicações  
Tempo: 001.500,00 - Ent: 49 (HARD) - NT1 - Tarefa Inicia Execução com Prioridade 2000  
Tempo: 001.512,50 - Ent: 43 (HARD) - NT6 - Tarefa Termina Execução  
Tempo: 001.512,50 - Ent: 43 (HARD) - NT6 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.562,50 - Ent: 51 (HARD) - NT2 - Tarefa Chega com Prioridade 2062.5 e 0 comunicações  
Tempo: 001.562,50 - Ent: 51 (HARD) - NT2 - Tarefa Inicia Execução com Prioridade 2062.5  
Tempo: 001.575,00 - Ent: 45 (HARD) - NT7 - Tarefa Termina Execução

Tempo: 001.575,00 - Ent: 45 (HARD) - NT7 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.625,00 - Ent: 53 (HARD) - NT3 - Tarefa Chega com Prioridade 2125 e 0 comunicações  
Tempo: 001.625,00 - Ent: 53 (HARD) - NT3 - Tarefa Inicia Execução com Prioridade 2125  
Tempo: 001.637,50 - Ent: 47 (HARD) - NT8 - Tarefa Termina Execução  
Tempo: 001.637,50 - Ent: 47 (HARD) - NT8 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.687,50 - Ent: 55 (HARD) - NT4 - Tarefa Chega com Prioridade 2187.5 e 0 comunicações  
Tempo: 001.687,50 - Ent: 55 (HARD) - NT4 - Tarefa Inicia Execução com Prioridade 2187.5  
Tempo: 001.700,00 - Ent: 49 (HARD) - NT1 - Tarefa Termina Execução  
Tempo: 001.700,00 - Ent: 49 (HARD) - NT1 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.750,00 - Ent: 57 (HARD) - NT5 - Tarefa Chega com Prioridade 2250 e 0 comunicações  
Tempo: 001.750,00 - Ent: 57 (HARD) - NT5 - Tarefa Inicia Execução com Prioridade 2250  
Tempo: 001.762,50 - Ent: 51 (HARD) - NT2 - Tarefa Termina Execução  
Tempo: 001.762,50 - Ent: 51 (HARD) - NT2 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.812,50 - Ent: 59 (HARD) - NT6 - Tarefa Chega com Prioridade 2312.5 e 0 comunicações  
Tempo: 001.812,50 - Ent: 59 (HARD) - NT6 - Tarefa Inicia Execução com Prioridade 2312.5  
Tempo: 001.825,00 - Ent: 53 (HARD) - NT3 - Tarefa Termina Execução  
Tempo: 001.825,00 - Ent: 53 (HARD) - NT3 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.875,00 - Ent: 61 (HARD) - NT7 - Tarefa Chega com Prioridade 2375 e 0 comunicações  
Tempo: 001.875,00 - Ent: 61 (HARD) - NT7 - Tarefa Inicia Execução com Prioridade 2375  
Tempo: 001.887,50 - Ent: 55 (HARD) - NT4 - Tarefa Termina Execução  
Tempo: 001.887,50 - Ent: 55 (HARD) - NT4 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.937,50 - Ent: 63 (HARD) - NT8 - Tarefa Chega com Prioridade 2437.5 e 0 comunicações  
Tempo: 001.937,50 - Ent: 63 (HARD) - NT8 - Tarefa Inicia Execução com Prioridade 2437.5  
Tempo: 001.950,00 - Ent: 57 (HARD) - NT5 - Tarefa Termina Execução  
Tempo: 001.950,00 - Ent: 57 (HARD) - NT5 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 002.000,00 - Ent: 65 (HARD) - NT1 - Tarefa Chega com Prioridade 2500 e 0 comunicações  
Tempo: 002.000,00 - Ent: 65 (HARD) - NT1 - Tarefa Inicia Execução com Prioridade 2500

## ANEXO 10 – Arquivo de Log gerado pelo modelo.

### Registro de atividades *soft real time* VBR.usando algoritmo DS

MODELO DO MULTICOMPUTADOR CRUX - CPGCC - UFSC  
Avaliação de Desempenho de Algoritmos de Escalonamento de Tempo Real

Parâmetros do Modelo:

Parâmetros do Sistema: NTs: 8; Canais: 2  
Veloc.Crossbar: 100MB/s; Veloc.BS: 100MB/s  
Tamanho Req.: 64bytes; Algoritmo de Escalonamento: DS  
Parâmetros de Carga : FatorHard: .4; FatorSoft: .4; FatorCBR: .4  
ArquivosVBR: 36; AmostrasCBR: 100  
PeriodoAmostraCBR: 62.5; TempoProcNC: 4  
Parâmetros do Projeto: Tempo Simulação: 2000; Replicações: 1

Tempo: 000.001,00 - Ent: 1 (VBR) - NT3 - Tarefa Chega com Prioridade 1 e 0 comunicações  
Tempo: 000.001,00 - Ent: 2 (SERVER) - NT3 - Tarefa Chega com Prioridade 200 e 0 comunicações  
Tempo: 000.001,00 - Ent: 2 (SERVER) - NT3 - Inicializa Servidor Aperiódico (1). Capacidade 80  
Tempo: 000.001,00 - Ent: 2 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.081,00 - Ent: 2 (SERVER) - NT3 - Tarefa é Preemptada faltando 44.7487028189353 us. Prioridade 200  
Tempo: 000.205,00 - Ent: 2 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.249,75 - Ent: 2 (SERVER) - NT3 - Tarefa Termina Execução  
Tempo: 000.249,75 - Ent: 2 (SERVER) - NT3 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (2)  
Tempo: 000.249,75 - Ent: 1 (VBR) - NT3 - Monta Pedido de Conexão de 64 bytes de NT 3 para NT 1. Tamanho da Mensagem 64 bytes  
Tempo: 000.249,75 - Ent: 1 (VBR) - NT3 - Inicia Transmissão de Pedido de Conexão pelo BS  
Tempo: 000.252,31 - Ent: 1 (VBR) - NC - Pedido de Conexão entre NT 3 e NT 1 chega ao NC com Prioridade 0  
Tempo: 000.256,31 - Ent: 1 (VBR) - NC - Determina Existência de 2 Canais Físicos Disponíveis entre NT 3 e NT 1  
Tempo: 000.256,31 - Ent: 1 (VBR) - NC - Aloca 1 Canais Físicos Disponíveis. Tempo de conexão: 4.00000000000003. Deadline: 312.871757047338  
Tempo: 000.256,31 - Ent: 1 (VBR) - NC - Inicia Transmissão de Mensagem de 64 bytes pelo Crossbar. Tempo de Transmissão: .64  
Tempo: 000.256,95 - Ent: 1 (VBR) - NT1 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.256,95 - Ent: 1 (VBR) - NT1 - Tarefa Chega com Prioridade 256.948702818935 e 1 comunicações  
Tempo: 000.256,95 - Ent: 1 (VBR) - NT1 - Monta Pedido de Desconexão  
Tempo: 000.256,95 - Ent: 1 (VBR) - NT1 - Inicia Transmissão de Pedido de Desconexão pelo BS  
Tempo: 000.259,51 - Ent: 1 (VBR) - NC - Pedido de Desconexão entre NT 1 e NT 3 chega ao NC com Prioridade 0  
Tempo: 000.263,51 - Ent: 1 (VBR) - NC - Libera 1 Canais Físicos do Crossbar entre NT 1 e NT 3  
Tempo: 000.420,00 - Ent: 3 (SERVER) - NT1 - Tarefa Chega com Prioridade 200 e 1 comunicações  
Tempo: 000.420,00 - Ent: 3 (SERVER) - NT1 - Inicializa Servidor Aperiódico (1). Capacidade 80  
Tempo: 000.420,00 - Ent: 3 (SERVER) - NT1 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.500,00 - Ent: 3 (SERVER) - NT1 - Tarefa é Preemptada faltando 77.7380449206468 us. Prioridade 200  
Tempo: 000.620,00 - Ent: 3 (SERVER) - NT1 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.697,74 - Ent: 3 (SERVER) - NT1 - Tarefa Termina Execução  
Tempo: 000.697,74 - Ent: 3 (SERVER) - NT1 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (3)  
Tempo: 000.697,74 - Ent: 1 (VBR) - NT1 - Monta Pedido de Conexão de 64 bytes de NT 1 para NT 3. Tamanho da Mensagem 307 bytes  
Tempo: 000.697,74 - Ent: 1 (VBR) - NT1 - Inicia Transmissão de Pedido de Conexão pelo BS  
Tempo: 000.700,30 - Ent: 1 (VBR) - NC - Pedido de Conexão entre NT 1 e NT 3 chega ao NC com Prioridade 0  
Tempo: 000.704,30 - Ent: 1 (VBR) - NC - Determina Existência de 2 Canais Físicos Disponíveis entre NT 1 e NT 3  
Tempo: 000.704,30 - Ent: 1 (VBR) - NC - Aloca 2 Canais Físicos Disponíveis. Tempo de conexão: 4. Deadline: 651.293815120552  
Tempo: 000.704,30 - Ent: 1 (VBR) - NC - Inicia Transmissão de Mensagem de 307 bytes pelo Crossbar. Tempo de Transmissão: 1.535  
Tempo: 000.705,83 - Ent: 1 (VBR) - NT3 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.705,83 - Ent: 1 (VBR) - NT3 - Tarefa Chega com Prioridade 705.833044920647 e 2 comunicações  
Tempo: 000.705,83 - Ent: 1 (VBR) - NT3 - Monta Pedido de Desconexão  
Tempo: 000.705,83 - Ent: 1 (VBR) - NT3 - Inicia Transmissão de Pedido de Desconexão pelo BS  
Tempo: 000.705,83 - Ent: 4 (SERVER) - NT3 - Tarefa Chega com Prioridade 200 e 2 comunicações

Tempo: 000.705,83 - Ent: 4 (SERVER) - NT3 - Inicializa Servidor Aperiódico (1). Capacidade 35.2512971810647  
Tempo: 000.705,83 - Ent: 4 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.705,83 - Ent: 4 (SERVER) - NT3 - Tempo de Resposta dessa Tarefa: 8.094999999999999  
Tempo: 000.708,39 - Ent: 1 (VBR) - NC - Pedido de Desconexão entre NT 3 e NT 1 chega ao NC com Prioridade 0  
Tempo: 000.712,39 - Ent: 1 (VBR) - NC - Libera 2 Canais Físicos do Crossbar entre NT 3 e NT 1  
Tempo: 000.741,08 - Ent: 4 (SERVER) - NT3 - Tarefa é Preemptada faltando 17.516286113266 us. Prioridade 200  
Tempo: 000.805,00 - Ent: 4 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.805,00 - Ent: 4 (SERVER) - NT3 - Tempo de Resposta dessa Tarefa: 107.261955079353  
Tempo: 000.822,52 - Ent: 4 (SERVER) - NT3 - Tarefa Termina Execução  
Tempo: 000.822,52 - Ent: 4 (SERVER) - NT3 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (4)  
Tempo: 000.822,52 - Ent: 1 (VBR) - NT3 - Monta Pedido de Conexão de 64 bytes de NT 3 para NT 1. Tamanho da Mensagem 64 bytes  
Tempo: 000.822,52 - Ent: 1 (VBR) - NT3 - Inicia Transmissão de Pedido de Conexão pelo BS  
Tempo: 000.825,08 - Ent: 1 (VBR) - NC - Pedido de Conexão entre NT 3 e NT 1 chega ao NC com Prioridade 0  
Tempo: 000.829,08 - Ent: 1 (VBR) - NC - Determina Existência de 2 Canais Físicos Disponíveis entre NT 3 e NT 1  
Tempo: 000.829,08 - Ent: 1 (VBR) - NC - Aloca 1 Canais Físicos Disponíveis. Tempo de conexão: 4. Deadline: 837.752003156474  
Tempo: 000.829,08 - Ent: 1 (VBR) - NC - Inicia Transmissão de Mensagem de 64 bytes pelo Crossbar. Tempo de Transmissão: .64  
Tempo: 000.829,72 - Ent: 1 (VBR) - NT1 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.829,72 - Ent: 1 (VBR) - NT1 - Tarefa Chega com Prioridade 829.716286113266 e 3 comunicações  
Tempo: 000.829,72 - Ent: 1 (VBR) - NT1 - Monta Pedido de Desconexão  
Tempo: 000.829,72 - Ent: 1 (VBR) - NT1 - Inicia Transmissão de Pedido de Desconexão pelo BS  
Tempo: 000.829,72 - Ent: 5 (SERVER) - NT1 - Tarefa Chega com Prioridade 200 e 3 comunicações  
Tempo: 000.829,72 - Ent: 5 (SERVER) - NT1 - Inicializa Servidor Aperiódico (1). Capacidade 2.26195507935324  
Tempo: 000.829,72 - Ent: 5 (SERVER) - NT1 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.832,28 - Ent: 1 (VBR) - NC - Pedido de Desconexão entre NT 1 e NT 3 chega ao NC com Prioridade 0  
Tempo: 000.836,28 - Ent: 1 (VBR) - NC - Libera 1 Canais Físicos do Crossbar entre NT 1 e NT 3  
Tempo: 000.840,00 - Ent: 5 (SERVER) - NT1 - Tarefa é Preemptada faltando 139.443234370557 us. Prioridade 200  
Tempo: 001.013,51 - Ent: 7 (VBR) - NT3 - Tarefa Chega com Prioridade 1013.51012510125 e 0 comunicações  
Tempo: 001.013,51 - Ent: 8 (SERVER) - NT3 - Tarefa Chega com Prioridade 200 e 0 comunicações  
Tempo: 001.013,51 - Ent: 8 (SERVER) - NT3 - Inicializa Servidor Aperiódico (7). Capacidade 62.483713886734  
Tempo: 001.013,51 - Ent: 8 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.020,00 - Ent: 5 (SERVER) - NT1 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.075,99 - Ent: 8 (SERVER) - NT3 - Tarefa é Preemptada faltando 64.9022230026973 us. Prioridade 200  
Tempo: 001.085,00 - Ent: 8 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.100,00 - Ent: 5 (SERVER) - NT1 - Tarefa é Preemptada faltando 59.4432343705573 us. Prioridade 200  
Tempo: 001.105,00 - Ent: 8 (SERVER) - NT3 - Tarefa é Preemptada faltando 44.9022230026972 us. Prioridade 200  
Tempo: 001.205,00 - Ent: 8 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.220,00 - Ent: 5 (SERVER) - NT1 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.249,90 - Ent: 8 (SERVER) - NT3 - Tarefa Termina Execução  
Tempo: 001.249,90 - Ent: 8 (SERVER) - NT3 - Tarefa Aperiódica (7) Termina de ser Atendida por Server Task (8)  
Tempo: 001.249,90 - Ent: 7 (VBR) - NT3 - Monta Pedido de Conexão de 64 bytes de NT 3 para NT 2. Tamanho da Mensagem 64 bytes  
Tempo: 001.249,90 - Ent: 7 (VBR) - NT3 - Inicia Transmissão de Pedido de Conexão pelo BS  
Tempo: 001.252,46 - Ent: 7 (VBR) - NC - Pedido de Conexão entre NT 3 e NT 2 chega ao NC com Prioridade 0  
Tempo: 001.256,46 - Ent: 7 (VBR) - NC - Determina Existência de 2 Canais Físicos Disponíveis entre NT 3 e NT 2  
Tempo: 001.256,46 - Ent: 7 (VBR) - NC - Aloca 1 Canais Físicos Disponíveis. Tempo de conexão: 4. Deadline: 1331.97496732483  
Tempo: 001.256,46 - Ent: 7 (VBR) - NC - Inicia Transmissão de Mensagem de 64 bytes pelo Crossbar. Tempo de Transmissão: .64  
Tempo: 001.257,10 - Ent: 7 (VBR) - NT2 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.257,10 - Ent: 7 (VBR) - NT2 - Tarefa Chega com Prioridade 1257.1022230027 e 1 comunicações  
Tempo: 001.257,10 - Ent: 7 (VBR) - NT2 - Monta Pedido de Desconexão  
Tempo: 001.257,10 - Ent: 7 (VBR) - NT2 - Inicia Transmissão de Pedido de Desconexão pelo BS  
Tempo: 001.259,66 - Ent: 7 (VBR) - NC - Pedido de Desconexão entre NT 2 e NT 3 chega ao NC com Prioridade 0  
Tempo: 001.263,66 - Ent: 7 (VBR) - NC - Libera 1 Canais Físicos do Crossbar entre NT 2 e NT 3  
Tempo: 001.279,44 - Ent: 5 (SERVER) - NT1 - Tarefa Termina Execução  
Tempo: 001.279,44 - Ent: 5 (SERVER) - NT1 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (5)  
Tempo: 001.279,44 - Ent: 1 (VBR) - NT1 - Monta Pedido de Conexão de 64 bytes de NT 1 para NT 3. Tamanho da Mensagem 6144 bytes  
Tempo: 001.279,44 - Ent: 1 (VBR) - NT1 - Inicia Transmissão de Pedido de Conexão pelo BS  
Tempo: 001.282,00 - Ent: 1 (VBR) - NC - Pedido de Conexão entre NT 1 e NT 3 chega ao NC com Prioridade 0  
Tempo: 001.286,00 - Ent: 1 (VBR) - NC - Determina Existência de 2 Canais Físicos Disponíveis entre NT 1 e NT 3

Tempo: 001.286,00 - Ent: 1 (VBR) - NC - Aloca 2 Canais Físicos Disponíveis. Tempo de conexão: 4. Deadline: 1204.03365675649

Tempo: 001.286,00 - Ent: 1 (VBR) - NC - Inicia Transmissão de Mensagem de 6144 bytes pelo Crossbar. Tempo de Transmissão: 30.72

Tempo: 001.316,72 - Ent: 1 (VBR) - NT3 - Mensagem Termina Transferência pelo Crossbar

Tempo: 001.316,72 - Ent: 1 (VBR) - NT3 - Tarefa Chega com Prioridade 1316.72323437056 e 4 comunicações

Tempo: 001.316,72 - Ent: 1 (VBR) - NT3 - Monta Pedido de Desconexão

Tempo: 001.316,72 - Ent: 1 (VBR) - NT3 - Inicia Transmissão de Pedido de Desconexão pelo BS

Tempo: 001.316,72 - Ent: 9 (SERVER) - NT3 - Tarefa Chega com Prioridade 200 e 4 comunicações

Tempo: 001.316,72 - Ent: 9 (SERVER) - NT3 - Inicializa Servidor Aperiódico (1). Capacidade 35.0977769973028

Tempo: 001.316,72 - Ent: 9 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.316,72 - Ent: 9 (SERVER) - NT3 - Tempo de Resposta dessa Tarefa: 37.28

Tempo: 001.319,28 - Ent: 1 (VBR) - NC - Pedido de Desconexão entre NT 3 e NT 1 chega ao NC com Prioridade 0

Tempo: 001.323,28 - Ent: 1 (VBR) - NC - Libera 2 Canais Físicos do Crossbar entre NT 3 e NT 1

Tempo: 001.342,50 - Ent: 10 (SERVER) - NT2 - Tarefa Chega com Prioridade 200 e 1 comunicações

Tempo: 001.342,50 - Ent: 10 (SERVER) - NT2 - Inicializa Servidor Aperiódico (7). Capacidade 80

Tempo: 001.342,50 - Ent: 10 (SERVER) - NT2 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.351,82 - Ent: 9 (SERVER) - NT3 - Tarefa é Preemptada faltando 148.443723945946 us. Prioridade 200

Tempo: 001.405,00 - Ent: 9 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.405,00 - Ent: 9 (SERVER) - NT3 - Tempo de Resposta dessa Tarefa: 125.556765629443

Tempo: 001.422,50 - Ent: 10 (SERVER) - NT2 - Tarefa é Preemptada faltando 130.529675113683 us. Prioridade 200

Tempo: 001.485,00 - Ent: 9 (SERVER) - NT3 - Tarefa é Preemptada faltando 68.443723945946 us. Prioridade 200

Tempo: 001.562,50 - Ent: 10 (SERVER) - NT2 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.605,00 - Ent: 9 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.605,00 - Ent: 9 (SERVER) - NT3 - Tempo de Resposta dessa Tarefa: 325.556765629443

Tempo: 001.642,50 - Ent: 10 (SERVER) - NT2 - Tarefa é Preemptada faltando 50.5296751136825 us. Prioridade 200

Tempo: 001.673,44 - Ent: 9 (SERVER) - NT3 - Tarefa Termina Execução

Tempo: 001.673,44 - Ent: 9 (SERVER) - NT3 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (9)

Tempo: 001.673,44 - Ent: 1 (VBR) - NT3 - Monta Pedido de Conexão de 64 bytes de NT 3 para NT 2. Tamanho da Mensagem 64 bytes

Tempo: 001.673,44 - Ent: 1 (VBR) - NT3 - Inicia Transmissão de Pedido de Conexão pelo BS

Tempo: 001.676,00 - Ent: 1 (VBR) - NC - Pedido de Conexão entre NT 3 e NT 2 chega ao NC com Prioridade 0

Tempo: 001.680,00 - Ent: 1 (VBR) - NC - Determina Existência de 2 Canais Físicos Disponíveis entre NT 3 e NT 2

Tempo: 001.680,00 - Ent: 1 (VBR) - NC - Aloca 1 Canais Físicos Disponíveis. Tempo de conexão: 4. Deadline: 1775.57698672868

Tempo: 001.680,00 - Ent: 1 (VBR) - NC - Inicia Transmissão de Mensagem de 64 bytes pelo Crossbar. Tempo de Transmissão: .64

Tempo: 001.680,64 - Ent: 1 (VBR) - NT2 - Mensagem Termina Transferência pelo Crossbar

Tempo: 001.680,64 - Ent: 1 (VBR) - NT2 - Tarefa Chega com Prioridade 1680.64372394595 e 5 comunicações

Tempo: 001.680,64 - Ent: 1 (VBR) - NT2 - Monta Pedido de Desconexão

Tempo: 001.680,64 - Ent: 1 (VBR) - NT2 - Inicia Transmissão de Pedido de Desconexão pelo BS

Tempo: 001.683,20 - Ent: 1 (VBR) - NC - Pedido de Desconexão entre NT 2 e NT 3 chega ao NC com Prioridade 0

Tempo: 001.687,20 - Ent: 1 (VBR) - NC - Libera 1 Canais Físicos do Crossbar entre NT 2 e NT 3

Tempo: 001.782,50 - Ent: 10 (SERVER) - NT2 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.833,03 - Ent: 10 (SERVER) - NT2 - Tarefa Termina Execução

Tempo: 001.833,03 - Ent: 10 (SERVER) - NT2 - Tarefa Aperiódica (7) Termina de ser Atendida por Server Task (10)

Tempo: 001.833,03 - Ent: 7 (VBR) - NT2 - Monta Pedido de Conexão de 64 bytes de NT 2 para NT 3. Tamanho da Mensagem 5120 bytes

Tempo: 001.833,03 - Ent: 7 (VBR) - NT2 - Inicia Transmissão de Pedido de Conexão pelo BS

Tempo: 001.835,59 - Ent: 7 (VBR) - NC - Pedido de Conexão entre NT 2 e NT 3 chega ao NC com Prioridade 0

Tempo: 001.839,59 - Ent: 7 (VBR) - NC - Determina Existência de 2 Canais Físicos Disponíveis entre NT 2 e NT 3

Tempo: 001.839,59 - Ent: 7 (VBR) - NC - Aloca 2 Canais Físicos Disponíveis. Tempo de conexão: 4. Deadline: 1783.4264107869

Tempo: 001.839,59 - Ent: 7 (VBR) - NC - Inicia Transmissão de Mensagem de 5120 bytes pelo Crossbar. Tempo de Transmissão: 25.6

Tempo: 001.842,50 - Ent: 11 (SERVER) - NT2 - Tarefa Chega com Prioridade 200 e 5 comunicações

Tempo: 001.842,50 - Ent: 11 (SERVER) - NT2 - Inicializa Servidor Aperiódico (1). Capacidade 29.4703248863175

Tempo: 001.842,50 - Ent: 11 (SERVER) - NT2 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.865,19 - Ent: 7 (VBR) - NT3 - Mensagem Termina Transferência pelo Crossbar

Tempo: 001.865,19 - Ent: 7 (VBR) - NT3 - Tarefa Chega com Prioridade 1865.18967511368 e 2 comunicações

Tempo: 001.865,19 - Ent: 7 (VBR) - NT3 - Monta Pedido de Desconexão

Tempo: 001.865,19 - Ent: 7 (VBR) - NT3 - Inicia Transmissão de Pedido de Desconexão pelo BS

Tempo: 001.865,19 - Ent: 12 (SERVER) - NT3 - Tarefa Chega com Prioridade 200 e 2 comunicações

Tempo: 001.865,19 - Ent: 12 (SERVER) - NT3 - Inicializa Servidor Aperiódico (7). Capacidade 11.556276054054  
Tempo: 001.865,19 - Ent: 12 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.865,19 - Ent: 12 (SERVER) - NT3 - Tempo de Resposta dessa Tarefa: 32.1599999999999  
Tempo: 001.867,75 - Ent: 7 (VBR) - NC - Pedido de Desconexão entre NT 3 e NT 2 chega ao NC com Prioridade 0  
Tempo: 001.871,75 - Ent: 7 (VBR) - NC - Libera 2 Canais Físicos do Crossbar entre NT 3 e NT 2  
Tempo: 001.871,97 - Ent: 11 (SERVER) - NT2 - Tarefa é Preemptada faltando 44.9374730335919 us. Prioridade 200  
Tempo: 001.885,00 - Ent: 12 (SERVER) - NT3 - Tarefa é Preemptada faltando 43.4060789753605 us. Prioridade 200

## ANEXO 11 – Arquivo de Log gerado pelo modelo.

### Registro de atividades *soft real time CBR*. usando algoritmo DS

MODELO DO MULTICOMPUTADOR CRUX - CPGCC - UFSC  
Avaliação de Desempenho de Algoritmos de Escalonamento de Tempo Real

Parâmetros do Modelo:

Parâmetros do Sistema: NTs: 8; Canais: 2  
Veloc.Crossbar: 100MB/s; Veloc.BS: 100MB/s  
Tamanho Req.: 64bytes; Algoritmo de Escalonamento: DS  
Parâmetros de Carga : FatorHard: .4; FatorSoft: .4; FatorCBR: .4  
ArquivosVBR: 36; AmostrasCBR: 100  
PeriodoAmostraCBR: 62.5; TempoProcNC: 4  
Parâmetros do Projeto: Tempo Simulação: 2000; Replicações: 1

Tempo: 000.001,00 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 1 e 0 comunicações  
Tempo: 000.001,00 - Ent: 2 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 0 comunicações  
Tempo: 000.001,00 - Ent: 2 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 80  
Tempo: 000.001,00 - Ent: 2 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.037,50 - Ent: 2 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 000.037,50 - Ent: 2 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (2)  
Tempo: 000.037,50 - Ent: 1 (CBR) - NT8 - Monta Pedido de Conexão de 64 bytes de NT 8 para NT 1. Tamanho da Mensagem 64 bytes  
Tempo: 000.037,50 - Ent: 1 (CBR) - NT8 - Inicia Transmissão de Pedido de Conexão pelo BS  
Tempo: 000.040,06 - Ent: 1 (CBR) - NC - Pedido de Conexão entre NT 8 e NT 1 chega ao NC com Prioridade 0  
Tempo: 000.044,06 - Ent: 1 (CBR) - NC - Determina Existência de 2 Canais Físicos Disponíveis entre NT 8 e NT 1  
Tempo: 000.044,06 - Ent: 1 (CBR) - NC - Aloca 1 Canais Físicos Disponíveis. Tempo de conexão: 4. Deadline: 76  
Tempo: 000.044,06 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 64 bytes pelo Crossbar. Tempo de Transmissão: .64  
Tempo: 000.044,70 - Ent: 1 (CBR) - NT1 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.044,70 - Ent: 1 (CBR) - NT1 - Tarefa Chega com Prioridade 44.7 e 1 comunicações  
Tempo: 000.220,00 - Ent: 3 (SERVER) - NT1 - Tarefa Chega com Prioridade 200 e 1 comunicações  
Tempo: 000.220,00 - Ent: 3 (SERVER) - NT1 - Inicializa Servidor Aperiódico (1). Capacidade 80  
Tempo: 000.220,00 - Ent: 3 (SERVER) - NT1 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.245,00 - Ent: 3 (SERVER) - NT1 - Tarefa Termina Execução  
Tempo: 000.245,00 - Ent: 3 (SERVER) - NT1 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (3)  
Tempo: 000.245,00 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.250,12 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.250,12 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 250.12 e 2 comunicações  
Tempo: 000.250,12 - Ent: 4 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 2 comunicações  
Tempo: 000.250,12 - Ent: 4 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 43.5  
Tempo: 000.250,12 - Ent: 4 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.250,12 - Ent: 4 (SERVER) - NT8 - Tempo de Resposta dessa Tarefa: 212.62  
Tempo: 000.275,12 - Ent: 4 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 000.275,12 - Ent: 4 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (4)  
Tempo: 000.275,12 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 2 comunicações  
Tempo: 000.307,50 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.312,62 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.312,62 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 312.62 e 3 comunicações  
Tempo: 000.312,62 - Ent: 5 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 3 comunicações  
Tempo: 000.312,62 - Ent: 5 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 18.5  
Tempo: 000.312,62 - Ent: 5 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.312,62 - Ent: 5 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 250.12  
Tempo: 000.331,12 - Ent: 5 (SERVER) - NT8 - Tarefa é Preemptada faltando 6.5 us. Prioridade 200  
Tempo: 000.370,00 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.375,12 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar

Tempo: 000.375,12 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 375.12 e 4 comunicações  
Tempo: 000.432,50 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.437,50 - Ent: 5 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.437,50 - Ent: 5 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 62.38  
Tempo: 000.437,62 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.437,62 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 437.62 e 5 comunicações  
Tempo: 000.444,00 - Ent: 5 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 000.444,00 - Ent: 5 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (5)  
Tempo: 000.444,00 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 3 comunicações  
Tempo: 000.444,00 - Ent: 6 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 4 comunicações  
Tempo: 000.444,00 - Ent: 6 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 73.5  
Tempo: 000.444,00 - Ent: 6 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.444,00 - Ent: 6 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 56  
Tempo: 000.469,00 - Ent: 6 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 000.469,00 - Ent: 6 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (6)  
Tempo: 000.469,00 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 4 comunicações  
Tempo: 000.477,50 - Ent: 7 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 5 comunicações  
Tempo: 000.477,50 - Ent: 7 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 48.5  
Tempo: 000.477,50 - Ent: 7 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.477,50 - Ent: 7 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 29  
Tempo: 000.495,00 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.500,12 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.500,12 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 500.12 e 6 comunicações  
Tempo: 000.502,50 - Ent: 7 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 000.502,50 - Ent: 7 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (7)  
Tempo: 000.502,50 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 5 comunicações  
Tempo: 000.502,50 - Ent: 8 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 6 comunicações  
Tempo: 000.502,50 - Ent: 8 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 23.5  
Tempo: 000.502,50 - Ent: 8 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.502,50 - Ent: 8 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 37.5  
Tempo: 000.526,00 - Ent: 8 (SERVER) - NT8 - Tarefa é Preemptada faltando 1.5 us. Prioridade 200  
Tempo: 000.557,50 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.562,62 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.562,62 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 562.62 e 7 comunicações  
Tempo: 000.620,00 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.625,12 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.625,12 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 625.12 e 8 comunicações  
Tempo: 000.657,50 - Ent: 8 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.657,50 - Ent: 8 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 92.5  
Tempo: 000.659,00 - Ent: 8 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 000.659,00 - Ent: 8 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (8)  
Tempo: 000.659,00 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 6 comunicações  
Tempo: 000.677,50 - Ent: 9 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 7 comunicações  
Tempo: 000.677,50 - Ent: 9 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 78.5  
Tempo: 000.677,50 - Ent: 9 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.677,50 - Ent: 9 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 42.5  
Tempo: 000.682,50 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.687,62 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.687,62 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 687.62 e 9 comunicações  
Tempo: 000.702,50 - Ent: 9 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 000.702,50 - Ent: 9 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (9)  
Tempo: 000.702,50 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 7 comunicações  
Tempo: 000.702,50 - Ent: 10 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 8 comunicações  
Tempo: 000.702,50 - Ent: 10 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 53.5  
Tempo: 000.702,50 - Ent: 10 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.702,50 - Ent: 10 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 37.5  
Tempo: 000.727,50 - Ent: 10 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 000.727,50 - Ent: 10 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (10)

Tempo: 000.727,50 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 8 comunicações  
Tempo: 000.737,50 - Ent: 11 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 9 comunicações  
Tempo: 000.737,50 - Ent: 11 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 28.5  
Tempo: 000.737,50 - Ent: 11 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.737,50 - Ent: 11 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 27.5  
Tempo: 000.745,00 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.750,12 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.750,12 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 750.12 e 10 comunicações  
Tempo: 000.762,50 - Ent: 11 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 000.762,50 - Ent: 11 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (11)  
Tempo: 000.762,50 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 9 comunicações  
Tempo: 000.762,50 - Ent: 12 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 10 comunicações  
Tempo: 000.762,50 - Ent: 12 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 3.5  
Tempo: 000.762,50 - Ent: 12 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.762,50 - Ent: 12 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 37.5  
Tempo: 000.777,50 - Ent: 12 (SERVER) - NT8 - Tarefa é Preemptada faltando 10 us. Prioridade 200  
Tempo: 000.807,50 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.812,62 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.812,62 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 812.62 e 11 comunicações  
Tempo: 000.870,00 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.875,12 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.875,12 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 875.12 e 12 comunicações  
Tempo: 000.875,12 - Ent: 12 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.875,12 - Ent: 12 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 50.12  
Tempo: 000.885,12 - Ent: 12 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 000.885,12 - Ent: 12 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (12)  
Tempo: 000.885,12 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 10 comunicações  
Tempo: 000.885,12 - Ent: 13 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 11 comunicações  
Tempo: 000.885,12 - Ent: 13 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 70  
Tempo: 000.885,12 - Ent: 13 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.885,12 - Ent: 13 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 52.5  
Tempo: 000.910,12 - Ent: 13 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 000.910,12 - Ent: 13 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (13)  
Tempo: 000.910,12 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 11 comunicações  
Tempo: 000.917,50 - Ent: 14 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 12 comunicações  
Tempo: 000.917,50 - Ent: 14 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 45  
Tempo: 000.917,50 - Ent: 14 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.917,50 - Ent: 14 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 30.12  
Tempo: 000.932,50 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.937,62 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.937,62 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 937.62 e 13 comunicações  
Tempo: 000.942,50 - Ent: 14 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 000.942,50 - Ent: 14 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (14)  
Tempo: 000.942,50 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 12 comunicações  
Tempo: 000.942,50 - Ent: 15 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 13 comunicações  
Tempo: 000.942,50 - Ent: 15 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 20  
Tempo: 000.942,50 - Ent: 15 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.942,50 - Ent: 15 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 37.5  
Tempo: 000.962,50 - Ent: 15 (SERVER) - NT8 - Tarefa é Preemptada faltando 5 us. Prioridade 200  
Tempo: 000.995,00 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.000,12 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.000,12 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 1000.12 e 14 comunicações  
Tempo: 001.057,50 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.062,62 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar

Tempo: 001.062,62 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 1062.62 e 15 comunicações  
Tempo: 001.077,50 - Ent: 15 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.077,50 - Ent: 15 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 72.5  
Tempo: 001.082,50 - Ent: 15 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 001.082,50 - Ent: 15 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (15)  
Tempo: 001.082,50 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 13 comunicações  
Tempo: 001.097,50 - Ent: 16 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 14 comunicações  
Tempo: 001.097,50 - Ent: 16 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 75  
Tempo: 001.097,50 - Ent: 16 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.097,50 - Ent: 16 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 42.5  
Tempo: 001.120,00 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.122,50 - Ent: 16 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 001.122,50 - Ent: 16 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (16)  
Tempo: 001.122,50 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 14 comunicações  
Tempo: 001.125,12 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.125,12 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 1125.12 e 16 comunicações  
Tempo: 001.125,12 - Ent: 17 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 15 comunicações  
Tempo: 001.125,12 - Ent: 17 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 50  
Tempo: 001.125,12 - Ent: 17 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.125,12 - Ent: 17 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 34.8800000000001  
Tempo: 001.150,12 - Ent: 17 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 001.150,12 - Ent: 17 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (17)  
Tempo: 001.150,12 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 15 comunicações  
Tempo: 001.157,50 - Ent: 18 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 16 comunicações  
Tempo: 001.157,50 - Ent: 18 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 25  
Tempo: 001.157,50 - Ent: 18 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.157,50 - Ent: 18 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 30.11999999999999  
Tempo: 001.182,50 - Ent: 18 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 001.182,50 - Ent: 18 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (18)  
Tempo: 001.182,50 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 16 comunicações  
Tempo: 001.182,50 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.187,62 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.187,62 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 1187.62 e 17 comunicações  
Tempo: 001.245,00 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.250,12 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.250,12 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 1250.12 e 18 comunicações  
Tempo: 001.297,50 - Ent: 19 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 17 comunicações  
Tempo: 001.297,50 - Ent: 19 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 80  
Tempo: 001.297,50 - Ent: 19 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.297,50 - Ent: 19 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 77.5  
Tempo: 001.307,50 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.312,62 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.312,62 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 1312.62 e 19 comunicações  
Tempo: 001.322,50 - Ent: 19 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 001.322,50 - Ent: 19 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (19)  
Tempo: 001.322,50 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 17 comunicações  
Tempo: 001.322,50 - Ent: 20 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 18 comunicações  
Tempo: 001.322,50 - Ent: 20 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 55  
Tempo: 001.322,50 - Ent: 20 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.322,50 - Ent: 20 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 37.5  
Tempo: 001.347,50 - Ent: 20 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 001.347,50 - Ent: 20 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (20)  
Tempo: 001.347,50 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 18 comunicações  
Tempo: 001.357,50 - Ent: 21 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 19 comunicações

Tempo: 001.357,50 - Ent: 21 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 30  
Tempo: 001.357,50 - Ent: 21 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.357,50 - Ent: 21 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 27.5  
Tempo: 001.370,00 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.375,12 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.375,12 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 1375.12 e 20 comunicações  
Tempo: 001.382,50 - Ent: 21 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 001.382,50 - Ent: 21 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (21)  
Tempo: 001.382,50 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 19 comunicações  
Tempo: 001.382,50 - Ent: 22 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 20 comunicações  
Tempo: 001.382,50 - Ent: 22 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 5  
Tempo: 001.382,50 - Ent: 22 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.382,50 - Ent: 22 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 37.5  
Tempo: 001.397,50 - Ent: 22 (SERVER) - NT8 - Tarefa é Preemptada faltando 10 us. Prioridade 200  
Tempo: 001.432,50 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.437,62 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.437,62 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 1437.62 e 21 comunicações  
Tempo: 001.495,00 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.500,12 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.500,12 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 1500.12 e 22 comunicações  
Tempo: 001.500,12 - Ent: 22 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.500,12 - Ent: 22 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 55.11999999999999  
Tempo: 001.510,12 - Ent: 22 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 001.510,12 - Ent: 22 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (22)  
Tempo: 001.510,12 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 20 comunicações  
Tempo: 001.517,50 - Ent: 23 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 21 comunicações  
Tempo: 001.517,50 - Ent: 23 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 70  
Tempo: 001.517,50 - Ent: 23 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.517,50 - Ent: 23 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 45.11999999999999  
Tempo: 001.542,50 - Ent: 23 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 001.542,50 - Ent: 23 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (23)  
Tempo: 001.542,50 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 21 comunicações  
Tempo: 001.557,50 - Ent: 24 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 22 comunicações  
Tempo: 001.557,50 - Ent: 24 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 45  
Tempo: 001.557,50 - Ent: 24 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.557,50 - Ent: 24 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 22.5  
Tempo: 001.557,50 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.562,62 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.562,62 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 1562.62 e 23 comunicações  
Tempo: 001.582,50 - Ent: 24 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 001.582,50 - Ent: 24 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (24)  
Tempo: 001.582,50 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 22 comunicações  
Tempo: 001.582,50 - Ent: 25 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 23 comunicações  
Tempo: 001.582,50 - Ent: 25 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 20  
Tempo: 001.582,50 - Ent: 25 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.582,50 - Ent: 25 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 37.5  
Tempo: 001.602,50 - Ent: 25 (SERVER) - NT8 - Tarefa é Preemptada faltando 5 us. Prioridade 200  
Tempo: 001.620,00 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.625,12 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.625,12 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 1625.12 e 24 comunicações  
Tempo: 001.682,50 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.687,62 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.687,62 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 1687.62 e 25 comunicações  
Tempo: 001.717,50 - Ent: 25 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.717,50 - Ent: 25 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 72.5  
Tempo: 001.722,50 - Ent: 25 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 001.722,50 - Ent: 25 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (25)  
Tempo: 001.722,50 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 23 comunicações  
Tempo: 001.737,50 - Ent: 26 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 24 comunicações  
Tempo: 001.737,50 - Ent: 26 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 75  
Tempo: 001.737,50 - Ent: 26 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.737,50 - Ent: 26 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 42.5  
Tempo: 001.745,00 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.750,12 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.750,12 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 1750.12 e 26 comunicações  
Tempo: 001.762,50 - Ent: 26 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 001.762,50 - Ent: 26 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (26)  
Tempo: 001.762,50 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 24 comunicações  
Tempo: 001.762,50 - Ent: 27 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 25 comunicações  
Tempo: 001.762,50 - Ent: 27 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 50  
Tempo: 001.762,50 - Ent: 27 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.762,50 - Ent: 27 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 37.5  
Tempo: 001.787,50 - Ent: 27 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 001.787,50 - Ent: 27 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (27)  
Tempo: 001.787,50 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 25 comunicações  
Tempo: 001.797,50 - Ent: 28 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 26 comunicações  
Tempo: 001.797,50 - Ent: 28 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 25  
Tempo: 001.797,50 - Ent: 28 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.797,50 - Ent: 28 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 27.5  
Tempo: 001.807,50 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.812,62 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.812,62 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 1812.62 e 27 comunicações  
Tempo: 001.822,50 - Ent: 28 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 001.822,50 - Ent: 28 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (28)  
Tempo: 001.822,50 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 26 comunicações  
Tempo: 001.870,00 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.875,12 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.875,12 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 1875.12 e 28 comunicações  
Tempo: 001.932,50 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.937,50 - Ent: 29 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 27 comunicações  
Tempo: 001.937,50 - Ent: 29 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 80  
Tempo: 001.937,50 - Ent: 29 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.937,50 - Ent: 29 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 77.5  
Tempo: 001.937,62 - Ent: 1 (CBR) - NT8 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.937,62 - Ent: 1 (CBR) - NT8 - Tarefa Chega com Prioridade 1937.62 e 29 comunicações  
Tempo: 001.962,50 - Ent: 29 (SERVER) - NT8 - Tarefa Termina Execução  
Tempo: 001.962,50 - Ent: 29 (SERVER) - NT8 - Tarefa Aperiódica (1) Termina de ser Atendida por Server Task (29)  
Tempo: 001.962,50 - Ent: 1 (CBR) - NT8 - Tarefa Deixa o Sistema Normalmente. 27 comunicações  
Tempo: 001.977,50 - Ent: 30 (SERVER) - NT8 - Tarefa Chega com Prioridade 200 e 28 comunicações  
Tempo: 001.977,50 - Ent: 30 (SERVER) - NT8 - Inicializa Servidor Aperiódico (1). Capacidade 55  
Tempo: 001.977,50 - Ent: 30 (SERVER) - NT8 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.977,50 - Ent: 30 (SERVER) - NT8 - Variação da Taxa de Execução dessa Amostra: 22.5  
Tempo: 001.995,00 - Ent: 1 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12

## ANEXO 12 – Arquivo de Log gerado pelo modelo.

### Registro de atividades *hard real-time*, *soft real time VBR* e *soft real-time CBR*. usando algoritmo DS

#### MODELO DO MULTICOMPUTADOR CRUX - CPGCC - UFSC Avaliação de Desempenho de Algoritmos de Escalonamento de Tempo Real

##### Parâmetros do Modelo:

Parâmetros do Sistema: NTs: 8; Canais: 2  
Veloc.Crossbar: 100MB/s; Veloc.BS: 100MB/s  
Tamanho Req.: 64bytes; Algoritmo de Escalonamento: DS  
Parâmetros de Carga : FatorHard: .4; FatorSoft: .4; FatorCBR: .4  
ArquivosVBR: 36; AmostrasCBR: 100  
PeriodoAmostraCBR: 62.5; TempoProcNC: 4  
Parâmetros do Projeto: Tempo Simulação: 2000; Replicações: 1

Tempo: 000.000,00 - Ent: 1 (HARD) - NT1 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 000.000,00 - Ent: 1 (HARD) - NT1 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 000.001,00 - Ent: 3 (VBR) - NT3 - Tarefa Chega com Prioridade 1 e 0 comunicações  
Tempo: 000.001,00 - Ent: 4 (SERVER) - NT3 - Tarefa Chega com Prioridade 200 e 0 comunicações  
Tempo: 000.001,00 - Ent: 4 (SERVER) - NT3 - Inicializa Servidor Aperiódico (3). Capacidade 80  
Tempo: 000.001,00 - Ent: 4 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.001,00 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 1 e 0 comunicações  
Tempo: 000.001,00 - Ent: 7 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 0 comunicações  
Tempo: 000.001,00 - Ent: 7 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 80  
Tempo: 000.001,00 - Ent: 7 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.035,00 - Ent: 7 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 000.035,00 - Ent: 7 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (7)  
Tempo: 000.035,00 - Ent: 6 (CBR) - NT7 - Monta Pedido de Conexão de 64 bytes de NT 7 para NT 1. Tamanho da Mensagem 64 bytes  
Tempo: 000.035,00 - Ent: 6 (CBR) - NT7 - Inicia Transmissão de Pedido de Conexão pelo BS  
Tempo: 000.037,56 - Ent: 6 (CBR) - NC - Pedido de Conexão entre NT 7 e NT 1 chega ao NC com Prioridade 0  
Tempo: 000.041,56 - Ent: 6 (CBR) - NC - Determina Existência de 2 Canais Físicos Disponíveis entre NT 7 e NT 1  
Tempo: 000.041,56 - Ent: 6 (CBR) - NC - Aloca 1 Canais Físicos Disponíveis. Tempo de conexão: 4. Deadline: 76  
Tempo: 000.041,56 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 64 bytes pelo Crossbar. Tempo de Transmissão: .64  
Tempo: 000.042,20 - Ent: 6 (CBR) - NT1 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.042,20 - Ent: 6 (CBR) - NT1 - Tarefa Chega com Prioridade 42.2 e 1 comunicações  
Tempo: 000.062,50 - Ent: 9 (HARD) - NT2 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 000.062,50 - Ent: 9 (HARD) - NT2 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 000.081,00 - Ent: 4 (SERVER) - NT3 - Tarefa é Preemptada faltando 44.7487028189353 us. Prioridade 200  
Tempo: 000.125,00 - Ent: 11 (HARD) - NT3 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 000.187,50 - Ent: 13 (HARD) - NT4 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 000.187,50 - Ent: 13 (HARD) - NT4 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 000.200,00 - Ent: 14 (SERVER) - NT1 - Tarefa Chega com Prioridade 200 e 1 comunicações  
Tempo: 000.200,00 - Ent: 14 (SERVER) - NT1 - Inicializa Servidor Aperiódico (6). Capacidade 80  
Tempo: 000.200,00 - Ent: 1 (HARD) - NT1 - Tarefa Termina Execução  
Tempo: 000.200,00 - Ent: 1 (HARD) - NT1 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.200,00 - Ent: 14 (SERVER) - NT1 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.205,00 - Ent: 4 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.225,00 - Ent: 4 (SERVER) - NT3 - Tarefa é Preemptada faltando 24.7487028189353 us. Prioridade 200  
Tempo: 000.225,00 - Ent: 14 (SERVER) - NT1 - Tarefa Termina Execução  
Tempo: 000.225,00 - Ent: 14 (SERVER) - NT1 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (14)  
Tempo: 000.225,00 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.230,12 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.230,12 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 230.12 e 2 comunicações

Tempo: 000.230,12 - Ent: 15 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 2 comunicações  
Tempo: 000.230,12 - Ent: 15 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 46  
Tempo: 000.230,12 - Ent: 15 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.230,12 - Ent: 15 (SERVER) - NT7 - Tempo de Resposta dessa Tarefa: 195.12  
Tempo: 000.245,00 - Ent: 4 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.250,00 - Ent: 17 (HARD) - NT5 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 000.250,00 - Ent: 17 (HARD) - NT5 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 000.255,12 - Ent: 15 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 000.255,12 - Ent: 15 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (15)  
Tempo: 000.255,12 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 2 comunicações  
Tempo: 000.262,50 - Ent: 9 (HARD) - NT2 - Tarefa Termina Execução  
Tempo: 000.262,50 - Ent: 9 (HARD) - NT2 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.265,00 - Ent: 4 (SERVER) - NT3 - Tarefa é Preemptada faltando 4.7487028189353 us. Prioridade 200  
Tempo: 000.285,00 - Ent: 4 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.287,50 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.289,75 - Ent: 4 (SERVER) - NT3 - Tarefa Termina Execução  
Tempo: 000.289,75 - Ent: 4 (SERVER) - NT3 - Tarefa Aperiódica (3) Termina de ser Atendida por Server Task (4)  
Tempo: 000.289,75 - Ent: 3 (VBR) - NT3 - Monta Pedido de Conexão de 64 bytes de NT 3 para NT 2. Tamanho da Mensagem 64 bytes  
Tempo: 000.289,75 - Ent: 3 (VBR) - NT3 - Inicia Transmissão de Pedido de Conexão pelo BS  
Tempo: 000.292,31 - Ent: 3 (VBR) - NC - Pedido de Conexão entre NT 3 e NT 2 chega ao NC com Prioridade 0  
Tempo: 000.292,62 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.292,62 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 292.62 e 3 comunicações  
Tempo: 000.292,62 - Ent: 18 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 3 comunicações  
Tempo: 000.292,62 - Ent: 18 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 21  
Tempo: 000.292,62 - Ent: 18 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.292,62 - Ent: 18 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 230.12  
Tempo: 000.296,31 - Ent: 3 (VBR) - NC - Determina Existência de 2 Canais Físicos Disponíveis entre NT 3 e NT 2  
Tempo: 000.296,31 - Ent: 3 (VBR) - NC - Aloca 1 Canais Físicos Disponíveis. Tempo de conexão: 4. Deadline: 312.871757047338  
Tempo: 000.296,31 - Ent: 3 (VBR) - NC - Inicia Transmissão de Mensagem de 64 bytes pelo Crossbar. Tempo de Transmissão: .64  
Tempo: 000.296,95 - Ent: 3 (VBR) - NT2 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.296,95 - Ent: 3 (VBR) - NT2 - Tarefa Chega com Prioridade 296.948702818935 e 1 comunicações  
Tempo: 000.296,95 - Ent: 3 (VBR) - NT2 - Monta Pedido de Desconexão  
Tempo: 000.296,95 - Ent: 3 (VBR) - NT2 - Inicia Transmissão de Pedido de Desconexão pelo BS  
Tempo: 000.299,51 - Ent: 3 (VBR) - NC - Pedido de Desconexão entre NT 2 e NT 3 chega ao NC com Prioridade 0  
Tempo: 000.303,51 - Ent: 3 (VBR) - NC - Libera 1 Canais Físicos do Crossbar entre NT 2 e NT 3  
Tempo: 000.305,00 - Ent: 11 (HARD) - NT3 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 000.312,50 - Ent: 20 (HARD) - NT6 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 000.312,50 - Ent: 20 (HARD) - NT6 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 000.313,62 - Ent: 18 (SERVER) - NT7 - Tarefa é Preemptada faltando 4 us. Prioridade 200  
Tempo: 000.350,00 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.355,12 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.355,12 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 355.12 e 4 comunicações  
Tempo: 000.375,00 - Ent: 22 (HARD) - NT7 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 000.387,50 - Ent: 13 (HARD) - NT4 - Tarefa Termina Execução  
Tempo: 000.387,50 - Ent: 13 (HARD) - NT4 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.412,50 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.415,00 - Ent: 18 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.415,00 - Ent: 18 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 59.88  
Tempo: 000.417,62 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.417,62 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 417.62 e 5 comunicações  
Tempo: 000.417,62 - Ent: 18 (SERVER) - NT7 - Tarefa é Preemptada faltando 1.38 us. Prioridade 200  
Tempo: 000.419,00 - Ent: 18 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.419,00 - Ent: 18 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 58.5  
Tempo: 000.420,38 - Ent: 18 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 000.420,38 - Ent: 18 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (18)  
Tempo: 000.420,38 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 3 comunicações

Tempo: 000.435,00 - Ent: 23 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 4 comunicações  
Tempo: 000.435,00 - Ent: 23 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 76  
Tempo: 000.435,00 - Ent: 23 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.435,00 - Ent: 23 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 46.5  
Tempo: 000.437,50 - Ent: 25 (HARD) - NT8 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 000.437,50 - Ent: 25 (HARD) - NT8 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 000.450,00 - Ent: 17 (HARD) - NT5 - Tarefa Termina Execução  
Tempo: 000.450,00 - Ent: 17 (HARD) - NT5 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.460,00 - Ent: 23 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 000.460,00 - Ent: 23 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (23)  
Tempo: 000.460,00 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 4 comunicações  
Tempo: 000.460,00 - Ent: 22 (HARD) - NT7 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 000.462,50 - Ent: 26 (SERVER) - NT2 - Tarefa Chega com Prioridade 200 e 1 comunicações  
Tempo: 000.462,50 - Ent: 26 (SERVER) - NT2 - Inicializa Servidor Aperiódico (3). Capacidade 80  
Tempo: 000.462,50 - Ent: 26 (SERVER) - NT2 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.475,00 - Ent: 27 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 5 comunicações  
Tempo: 000.475,00 - Ent: 27 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 51  
Tempo: 000.475,00 - Ent: 22 (HARD) - NT7 - Tarefa é Preemptada faltando 185 us. Prioridade 500  
Tempo: 000.475,00 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.480,12 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.480,12 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 480.12 e 6 comunicações  
Tempo: 000.480,12 - Ent: 27 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.480,12 - Ent: 27 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 17.38  
Tempo: 000.495,00 - Ent: 27 (SERVER) - NT7 - Tarefa é Preemptada faltando 10.12 us. Prioridade 200  
Tempo: 000.500,00 - Ent: 29 (HARD) - NT1 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 000.500,00 - Ent: 29 (HARD) - NT1 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 000.505,00 - Ent: 11 (HARD) - NT3 - Tarefa Termina Execução  
Tempo: 000.505,00 - Ent: 11 (HARD) - NT3 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.512,50 - Ent: 20 (HARD) - NT6 - Tarefa Termina Execução  
Tempo: 000.512,50 - Ent: 20 (HARD) - NT6 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.515,00 - Ent: 27 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.515,00 - Ent: 27 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 27.62  
Tempo: 000.525,12 - Ent: 27 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 000.525,12 - Ent: 27 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (27)  
Tempo: 000.525,12 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 5 comunicações  
Tempo: 000.535,00 - Ent: 30 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 6 comunicações  
Tempo: 000.535,00 - Ent: 30 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 26  
Tempo: 000.535,00 - Ent: 30 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.535,00 - Ent: 30 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 42.5  
Tempo: 000.537,50 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.542,50 - Ent: 26 (SERVER) - NT2 - Tarefa é Preemptada faltando 77.7380449206468 us. Prioridade 200  
Tempo: 000.542,62 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.542,62 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 542.62 e 7 comunicações  
Tempo: 000.560,00 - Ent: 30 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 000.560,00 - Ent: 30 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (30)  
Tempo: 000.560,00 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 6 comunicações  
Tempo: 000.560,00 - Ent: 22 (HARD) - NT7 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 000.560,00 - Ent: 31 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 7 comunicações  
Tempo: 000.560,00 - Ent: 31 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 1  
Tempo: 000.560,00 - Ent: 22 (HARD) - NT7 - Tarefa é Preemptada faltando 185 us. Prioridade 500  
Tempo: 000.562,50 - Ent: 33 (HARD) - NT2 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 000.575,00 - Ent: 31 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.575,00 - Ent: 31 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 22.5  
Tempo: 000.595,00 - Ent: 31 (SERVER) - NT7 - Tarefa é Preemptada faltando 5 us. Prioridade 200  
Tempo: 000.600,00 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.605,12 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.605,12 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 605.12 e 8 comunicações  
Tempo: 000.615,00 - Ent: 31 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200

Tempo: 000.615,00 - Ent: 31 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 22.5  
Tempo: 000.620,00 - Ent: 31 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 000.620,00 - Ent: 31 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (31)  
Tempo: 000.620,00 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 7 comunicações  
Tempo: 000.625,00 - Ent: 35 (HARD) - NT3 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 000.625,00 - Ent: 35 (HARD) - NT3 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 000.635,00 - Ent: 36 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 8 comunicações  
Tempo: 000.635,00 - Ent: 36 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 75  
Tempo: 000.635,00 - Ent: 36 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.635,00 - Ent: 36 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 42.5  
Tempo: 000.637,50 - Ent: 25 (HARD) - NT8 - Tarefa Termina Execução  
Tempo: 000.637,50 - Ent: 25 (HARD) - NT8 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.660,00 - Ent: 36 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 000.660,00 - Ent: 36 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (36)  
Tempo: 000.660,00 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 8 comunicações  
Tempo: 000.660,00 - Ent: 22 (HARD) - NT7 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 000.662,50 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.667,62 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.667,62 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 667.62 e 9 comunicações  
Tempo: 000.667,62 - Ent: 37 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 9 comunicações  
Tempo: 000.667,62 - Ent: 37 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 50  
Tempo: 000.667,62 - Ent: 22 (HARD) - NT7 - Tarefa é Preemptada faltando 177.38 us. Prioridade 500  
Tempo: 000.675,00 - Ent: 37 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.675,00 - Ent: 37 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 22.5  
Tempo: 000.682,50 - Ent: 26 (SERVER) - NT2 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.687,50 - Ent: 39 (HARD) - NT4 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 000.687,50 - Ent: 39 (HARD) - NT4 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 000.695,00 - Ent: 37 (SERVER) - NT7 - Tarefa é Preemptada faltando 5 us. Prioridade 200  
Tempo: 000.700,00 - Ent: 29 (HARD) - NT1 - Tarefa Termina Execução  
Tempo: 000.700,00 - Ent: 29 (HARD) - NT1 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.702,50 - Ent: 26 (SERVER) - NT2 - Tarefa é Preemptada faltando 57.7380449206468 us. Prioridade 200  
Tempo: 000.715,00 - Ent: 37 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.715,00 - Ent: 37 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 22.5  
Tempo: 000.720,00 - Ent: 37 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 000.720,00 - Ent: 37 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (37)  
Tempo: 000.720,00 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 9 comunicações  
Tempo: 000.722,50 - Ent: 26 (SERVER) - NT2 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.725,00 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.730,12 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.730,12 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 730.12 e 10 comunicações  
Tempo: 000.730,12 - Ent: 40 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 10 comunicações  
Tempo: 000.730,12 - Ent: 40 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 25  
Tempo: 000.730,12 - Ent: 40 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.730,12 - Ent: 40 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 47.38  
Tempo: 000.742,50 - Ent: 26 (SERVER) - NT2 - Tarefa é Preemptada faltando 37.7380449206468 us. Prioridade 200  
Tempo: 000.750,00 - Ent: 42 (HARD) - NT5 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 000.750,00 - Ent: 42 (HARD) - NT5 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 000.755,12 - Ent: 40 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 000.755,12 - Ent: 40 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (40)  
Tempo: 000.755,12 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 10 comunicações  
Tempo: 000.755,12 - Ent: 22 (HARD) - NT7 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 000.762,50 - Ent: 26 (SERVER) - NT2 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.782,50 - Ent: 26 (SERVER) - NT2 - Tarefa é Preemptada faltando 17.7380449206468 us. Prioridade 200  
Tempo: 000.787,50 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.792,62 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.792,62 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 792.62 e 11 comunicações  
Tempo: 000.802,50 - Ent: 26 (SERVER) - NT2 - Tarefa Inicia Execução com Prioridade 200

Tempo: 000.812,50 - Ent: 44 (HARD) - NT6 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 000.812,50 - Ent: 44 (HARD) - NT6 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 000.815,00 - Ent: 45 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 11 comunicações  
Tempo: 000.815,00 - Ent: 45 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 80  
Tempo: 000.815,00 - Ent: 22 (HARD) - NT7 - Tarefa é Preemptada faltando 117.5 us. Prioridade 500  
Tempo: 000.820,24 - Ent: 26 (SERVER) - NT2 - Tarefa Termina Execução  
Tempo: 000.820,24 - Ent: 26 (SERVER) - NT2 - Tarefa Aperiódica (3) Termina de ser Atendida por Server Task (26)  
Tempo: 000.820,24 - Ent: 3 (VBR) - NT2 - Monta Pedido de Conexão de 64 bytes de NT 2 para NT 3. Tamanho da Mensagem 307 bytes  
Tempo: 000.820,24 - Ent: 3 (VBR) - NT2 - Inicia Transmissão de Pedido de Conexão pelo BS  
Tempo: 000.822,50 - Ent: 33 (HARD) - NT2 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 000.822,80 - Ent: 3 (VBR) - NC - Pedido de Conexão entre NT 2 e NT 3 chega ao NC com Prioridade 0  
Tempo: 000.825,00 - Ent: 35 (HARD) - NT3 - Tarefa Termina Execução  
Tempo: 000.825,00 - Ent: 35 (HARD) - NT3 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.826,80 - Ent: 3 (VBR) - NC - Determina Existência de 2 Canais Físicos Disponíveis entre NT 2 e NT 3  
Tempo: 000.826,80 - Ent: 3 (VBR) - NC - Aloca 2 Canais Físicos Disponíveis. Tempo de conexão: 4. Deadline: 691.293815120552  
Tempo: 000.826,80 - Ent: 3 (VBR) - NC - Inicia Transmissão de Mensagem de 307 bytes pelo Crossbar. Tempo de Transmissão: 1.535  
Tempo: 000.828,33 - Ent: 3 (VBR) - NT3 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.828,33 - Ent: 3 (VBR) - NT3 - Tarefa Chega com Prioridade 828.333044920647 e 2 comunicações  
Tempo: 000.828,33 - Ent: 3 (VBR) - NT3 - Monta Pedido de Desconexão  
Tempo: 000.828,33 - Ent: 3 (VBR) - NT3 - Inicia Transmissão de Pedido de Desconexão pelo BS  
Tempo: 000.828,33 - Ent: 46 (SERVER) - NT3 - Tarefa Chega com Prioridade 200 e 2 comunicações  
Tempo: 000.828,33 - Ent: 46 (SERVER) - NT3 - Inicializa Servidor Aperiódico (3). Capacidade 35.2512971810647  
Tempo: 000.828,33 - Ent: 46 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.828,33 - Ent: 46 (SERVER) - NT3 - Tempo de Resposta dessa Tarefa: 8.09499999999991  
Tempo: 000.830,89 - Ent: 3 (VBR) - NC - Pedido de Desconexão entre NT 3 e NT 2 chega ao NC com Prioridade 0  
Tempo: 000.834,89 - Ent: 3 (VBR) - NC - Libera 2 Canais Físicos do Crossbar entre NT 3 e NT 2  
Tempo: 000.835,00 - Ent: 45 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.835,00 - Ent: 45 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 42.38  
Tempo: 000.850,00 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.855,00 - Ent: 45 (SERVER) - NT7 - Tarefa é Preemptada faltando 5 us. Prioridade 200  
Tempo: 000.855,12 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.855,12 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 855.12 e 12 comunicações  
Tempo: 000.855,12 - Ent: 45 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.855,12 - Ent: 45 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 42.38  
Tempo: 000.860,12 - Ent: 45 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 000.860,12 - Ent: 45 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (45)  
Tempo: 000.860,12 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 11 comunicações  
Tempo: 000.863,58 - Ent: 46 (SERVER) - NT3 - Tarefa é Preemptada faltando 17.516286113266 us. Prioridade 200  
Tempo: 000.875,00 - Ent: 47 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 12 comunicações  
Tempo: 000.875,00 - Ent: 47 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 55  
Tempo: 000.875,00 - Ent: 47 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.875,00 - Ent: 47 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 42.62  
Tempo: 000.875,00 - Ent: 49 (HARD) - NT7 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 000.875,00 - Ent: 47 (SERVER) - NT7 - Tarefa é Preemptada faltando 25 us. Prioridade 200  
Tempo: 000.875,00 - Ent: 22 (HARD) - NT7 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 000.887,50 - Ent: 39 (HARD) - NT4 - Tarefa Termina Execução  
Tempo: 000.887,50 - Ent: 39 (HARD) - NT4 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.895,00 - Ent: 22 (HARD) - NT7 - Tarefa deixa o Sistema Após Perder Deadline por 20 us  
Tempo: 000.912,50 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.915,00 - Ent: 47 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.915,00 - Ent: 47 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 22.5  
Tempo: 000.917,62 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.917,62 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 917.62 e 13 comunicações  
Tempo: 000.917,62 - Ent: 47 (SERVER) - NT7 - Tarefa é Preemptada faltando 22.38 us. Prioridade 200  
Tempo: 000.935,00 - Ent: 47 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.935,00 - Ent: 47 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 42.5  
Tempo: 000.937,50 - Ent: 51 (HARD) - NT8 - Tarefa Chega com Prioridade 500 e 0 comunicações

Tempo: 000.937,50 - Ent: 51 (HARD) - NT8 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 000.950,00 - Ent: 42 (HARD) - NT5 - Tarefa Termina Execução  
Tempo: 000.950,00 - Ent: 42 (HARD) - NT5 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 000.955,00 - Ent: 47 (SERVER) - NT7 - Tarefa é Preemptada faltando 2.38 us. Prioridade 200  
Tempo: 000.975,00 - Ent: 47 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.975,00 - Ent: 47 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 22.5  
Tempo: 000.975,00 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 000.977,38 - Ent: 47 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 000.977,38 - Ent: 47 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (47)  
Tempo: 000.977,38 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 12 comunicações  
Tempo: 000.980,12 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 000.980,12 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 980.12 e 14 comunicações  
Tempo: 000.980,12 - Ent: 52 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 13 comunicações  
Tempo: 000.980,12 - Ent: 52 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 30  
Tempo: 000.980,12 - Ent: 52 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 000.980,12 - Ent: 52 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 57.38  
Tempo: 001.000,00 - Ent: 54 (HARD) - NT1 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 001.000,00 - Ent: 54 (HARD) - NT1 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 001.005,00 - Ent: 46 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.005,00 - Ent: 46 (SERVER) - NT3 - Tempo de Resposta dessa Tarefa: 184.761955079353  
Tempo: 001.005,12 - Ent: 52 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 001.005,12 - Ent: 52 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (52)  
Tempo: 001.005,12 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 13 comunicações  
Tempo: 001.005,12 - Ent: 49 (HARD) - NT7 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 001.012,50 - Ent: 44 (HARD) - NT6 - Tarefa Termina Execução  
Tempo: 001.012,50 - Ent: 44 (HARD) - NT6 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.013,51 - Ent: 56 (VBR) - NT3 - Tarefa Chega com Prioridade 1013.51012510125 e 0 comunicações  
Tempo: 001.015,00 - Ent: 57 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 14 comunicações  
Tempo: 001.015,00 - Ent: 57 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 80  
Tempo: 001.015,00 - Ent: 49 (HARD) - NT7 - Tarefa é Preemptada faltando 190.12 us. Prioridade 500  
Tempo: 001.022,50 - Ent: 33 (HARD) - NT2 - Tarefa Termina Execução  
Tempo: 001.022,50 - Ent: 33 (HARD) - NT2 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.022,52 - Ent: 46 (SERVER) - NT3 - Tarefa Termina Execução  
Tempo: 001.022,52 - Ent: 46 (SERVER) - NT3 - Tarefa Aperiódica (3) Termina de ser Atendida por Server Task (46)  
Tempo: 001.022,52 - Ent: 3 (VBR) - NT3 - Monta Pedido de Conexão de 64 bytes de NT 3 para NT 1. Tamanho da Mensagem 64 bytes  
Tempo: 001.022,52 - Ent: 3 (VBR) - NT3 - Inicia Transmissão de Pedido de Conexão pelo BS  
Tempo: 001.022,52 - Ent: 58 (SERVER) - NT3 - Tarefa Chega com Prioridade 200 e 0 comunicações  
Tempo: 001.022,52 - Ent: 58 (SERVER) - NT3 - Inicializa Servidor Aperiódico (56). Capacidade 62.483713886734  
Tempo: 001.022,52 - Ent: 58 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.025,08 - Ent: 3 (VBR) - NC - Pedido de Conexão entre NT 3 e NT 1 chega ao NC com Prioridade 0  
Tempo: 001.029,08 - Ent: 3 (VBR) - NC - Determina Existência de 1 Canais Físicos Disponíveis entre NT 3 e NT 1  
Tempo: 001.029,08 - Ent: 3 (VBR) - NC - Aloca 1 Canais Físicos Disponíveis. Tempo de conexão: 4. Deadline: 960.252003156474  
Tempo: 001.029,08 - Ent: 3 (VBR) - NC - Inicia Transmissão de Mensagem de 64 bytes pelo Crossbar. Tempo de Transmissão: .64  
Tempo: 001.029,72 - Ent: 3 (VBR) - NT1 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.029,72 - Ent: 3 (VBR) - NT1 - Tarefa Chega com Prioridade 1029.71628611327 e 3 comunicações  
Tempo: 001.029,72 - Ent: 3 (VBR) - NT1 - Monta Pedido de Desconexão  
Tempo: 001.029,72 - Ent: 3 (VBR) - NT1 - Inicia Transmissão de Pedido de Desconexão pelo BS  
Tempo: 001.029,72 - Ent: 59 (SERVER) - NT1 - Tarefa Chega com Prioridade 200 e 3 comunicações  
Tempo: 001.029,72 - Ent: 59 (SERVER) - NT1 - Inicializa Servidor Aperiódico (3). Capacidade 55  
Tempo: 001.029,72 - Ent: 54 (HARD) - NT1 - Tarefa é Preemptada faltando 170.283713886734 us. Prioridade 500  
Tempo: 001.032,28 - Ent: 3 (VBR) - NC - Pedido de Desconexão entre NT 1 e NT 3 chega ao NC com Prioridade 0  
Tempo: 001.035,00 - Ent: 57 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.035,00 - Ent: 57 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 7.62  
Tempo: 001.036,28 - Ent: 3 (VBR) - NC - Libera 1 Canais Físicos do Crossbar entre NT 1 e NT 3  
Tempo: 001.037,50 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.040,00 - Ent: 59 (SERVER) - NT1 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.042,62 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.042,62 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 1042.62 e 15 comunicações  
Tempo: 001.042,62 - Ent: 57 (SERVER) - NT7 - Tarefa é Preemptada faltando 17.3800000000001 us. Prioridade 200  
Tempo: 001.055,00 - Ent: 57 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.055,00 - Ent: 57 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 42.5  
Tempo: 001.060,00 - Ent: 59 (SERVER) - NT1 - Tarefa é Preemptada faltando 132.863124267317 us. Prioridade 200  
Tempo: 001.062,50 - Ent: 61 (HARD) - NT2 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 001.062,50 - Ent: 61 (HARD) - NT2 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 001.072,38 - Ent: 57 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 001.072,38 - Ent: 57 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (57)  
Tempo: 001.072,38 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 14 comunicações  
Tempo: 001.075,00 - Ent: 62 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 15 comunicações  
Tempo: 001.075,00 - Ent: 62 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 55  
Tempo: 001.075,00 - Ent: 62 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.075,00 - Ent: 62 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 42.5  
Tempo: 001.080,00 - Ent: 59 (SERVER) - NT1 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.085,00 - Ent: 58 (SERVER) - NT3 - Tarefa é Preemptada faltando 108.19917930082 us. Prioridade 200  
Tempo: 001.100,00 - Ent: 59 (SERVER) - NT1 - Tarefa é Preemptada faltando 112.863124267317 us. Prioridade 200  
Tempo: 001.100,00 - Ent: 62 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 001.100,00 - Ent: 62 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (62)  
Tempo: 001.100,00 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 15 comunicações  
Tempo: 001.100,00 - Ent: 49 (HARD) - NT7 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 001.100,00 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.105,12 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.105,12 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 1105.12 e 16 comunicações  
Tempo: 001.105,12 - Ent: 63 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 16 comunicações  
Tempo: 001.105,12 - Ent: 63 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 30  
Tempo: 001.105,12 - Ent: 49 (HARD) - NT7 - Tarefa é Preemptada faltando 185 us. Prioridade 500  
Tempo: 001.115,00 - Ent: 63 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.115,00 - Ent: 63 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 22.5  
Tempo: 001.120,00 - Ent: 59 (SERVER) - NT1 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.125,00 - Ent: 65 (HARD) - NT3 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 001.135,00 - Ent: 63 (SERVER) - NT7 - Tarefa é Preemptada faltando 5 us. Prioridade 200  
Tempo: 001.137,50 - Ent: 51 (HARD) - NT8 - Tarefa Termina Execução  
Tempo: 001.137,50 - Ent: 51 (HARD) - NT8 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.140,00 - Ent: 59 (SERVER) - NT1 - Tarefa é Preemptada faltando 92.8631242673173 us. Prioridade 200  
Tempo: 001.155,00 - Ent: 63 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.155,00 - Ent: 63 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 22.5  
Tempo: 001.160,00 - Ent: 63 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 001.160,00 - Ent: 63 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (63)  
Tempo: 001.160,00 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 16 comunicações  
Tempo: 001.162,50 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.167,62 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.167,62 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 1167.62 e 17 comunicações  
Tempo: 001.167,62 - Ent: 66 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 17 comunicações  
Tempo: 001.167,62 - Ent: 66 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 5  
Tempo: 001.167,62 - Ent: 66 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.167,62 - Ent: 66 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 49.8800000000001  
Tempo: 001.175,00 - Ent: 66 (SERVER) - NT7 - Tarefa é Preemptada faltando 17.6199999999999 us. Prioridade 200  
Tempo: 001.175,00 - Ent: 49 (HARD) - NT7 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 001.187,50 - Ent: 68 (HARD) - NT4 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 001.187,50 - Ent: 68 (HARD) - NT4 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 001.195,00 - Ent: 49 (HARD) - NT7 - Tarefa é Preemptada faltando 165 us. Prioridade 500  
Tempo: 001.200,00 - Ent: 59 (SERVER) - NT1 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.205,00 - Ent: 58 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.215,00 - Ent: 66 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.215,00 - Ent: 66 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 15.1199999999999  
Tempo: 001.220,00 - Ent: 59 (SERVER) - NT1 - Tarefa é Preemptada faltando 72.8631242673173 us. Prioridade 200  
Tempo: 001.225,00 - Ent: 58 (SERVER) - NT3 - Tarefa é Preemptada faltando 88.1991793008199 us. Prioridade 200

Tempo: 001.225,00 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12

Tempo: 001.230,12 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar

Tempo: 001.230,12 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 1230.12 e 18 comunicações

Tempo: 001.230,12 - Ent: 66 (SERVER) - NT7 - Tarefa é Preemptada faltando 2.5 us. Prioridade 200

Tempo: 001.232,62 - Ent: 66 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.232,62 - Ent: 66 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 44.8800000000001

Tempo: 001.235,00 - Ent: 66 (SERVER) - NT7 - Tarefa é Preemptada faltando 119999999999891 us. Prioridade 200

Tempo: 001.235,12 - Ent: 66 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.235,12 - Ent: 66 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 60

Tempo: 001.235,24 - Ent: 66 (SERVER) - NT7 - Tarefa Termina Execução

Tempo: 001.235,24 - Ent: 66 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (66)

Tempo: 001.235,24 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 17 comunicações

Tempo: 001.240,00 - Ent: 59 (SERVER) - NT1 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.245,00 - Ent: 58 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.250,00 - Ent: 70 (HARD) - NT5 - Tarefa Chega com Prioridade 500 e 0 comunicações

Tempo: 001.250,00 - Ent: 70 (HARD) - NT5 - Tarefa Inicia Execução com Prioridade 500

Tempo: 001.255,00 - Ent: 71 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 18 comunicações

Tempo: 001.255,00 - Ent: 71 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 62.3800000000001

Tempo: 001.255,00 - Ent: 71 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.255,00 - Ent: 71 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 42.6199999999999

Tempo: 001.260,00 - Ent: 59 (SERVER) - NT1 - Tarefa é Preemptada faltando 52.8631242673173 us. Prioridade 200

Tempo: 001.262,50 - Ent: 61 (HARD) - NT2 - Tarefa Termina Execução

Tempo: 001.262,50 - Ent: 61 (HARD) - NT2 - Tarefa Deixa o Sistema Normalmente. 0 comunicações

Tempo: 001.265,00 - Ent: 58 (SERVER) - NT3 - Tarefa é Preemptada faltando 68.1991793008199 us. Prioridade 200

Tempo: 001.280,00 - Ent: 59 (SERVER) - NT1 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.280,00 - Ent: 71 (SERVER) - NT7 - Tarefa Termina Execução

Tempo: 001.280,00 - Ent: 71 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (71)

Tempo: 001.280,00 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 18 comunicações

Tempo: 001.280,00 - Ent: 49 (HARD) - NT7 - Tarefa Inicia Execução com Prioridade 500

Tempo: 001.285,00 - Ent: 58 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.287,50 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12

Tempo: 001.292,62 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar

Tempo: 001.292,62 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 1292.62 e 19 comunicações

Tempo: 001.292,62 - Ent: 72 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 19 comunicações

Tempo: 001.292,62 - Ent: 72 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 37.3800000000001

Tempo: 001.292,62 - Ent: 49 (HARD) - NT7 - Tarefa é Preemptada faltando 152.38 us. Prioridade 500

Tempo: 001.295,00 - Ent: 72 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.295,00 - Ent: 72 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 22.5

Tempo: 001.300,00 - Ent: 59 (SERVER) - NT1 - Tarefa é Preemptada faltando 32.8631242673173 us. Prioridade 200

Tempo: 001.305,00 - Ent: 58 (SERVER) - NT3 - Tarefa é Preemptada faltando 48.1991793008199 us. Prioridade 200

Tempo: 001.312,50 - Ent: 74 (HARD) - NT6 - Tarefa Chega com Prioridade 500 e 0 comunicações

Tempo: 001.312,50 - Ent: 74 (HARD) - NT6 - Tarefa Inicia Execução com Prioridade 500

Tempo: 001.315,00 - Ent: 72 (SERVER) - NT7 - Tarefa é Preemptada faltando 5 us. Prioridade 200

Tempo: 001.320,00 - Ent: 59 (SERVER) - NT1 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.325,00 - Ent: 58 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.335,00 - Ent: 72 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.335,00 - Ent: 72 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 22.5

Tempo: 001.340,00 - Ent: 59 (SERVER) - NT1 - Tarefa é Preemptada faltando 12.8631242673173 us. Prioridade 200

Tempo: 001.340,00 - Ent: 72 (SERVER) - NT7 - Tarefa Termina Execução

Tempo: 001.340,00 - Ent: 72 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (72)

Tempo: 001.340,00 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 19 comunicações

Tempo: 001.345,00 - Ent: 58 (SERVER) - NT3 - Tarefa é Preemptada faltando 28.1991793008199 us. Prioridade 200

Tempo: 001.350,00 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12

Tempo: 001.355,00 - Ent: 49 (HARD) - NT7 - Tarefa Inicia Execução com Prioridade 500

Tempo: 001.355,12 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar

Tempo: 001.355,12 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 1355.12 e 20 comunicações

Tempo: 001.355,12 - Ent: 75 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 20 comunicações

Tempo: 001.355,12 - Ent: 75 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 12.3800000000001

Tempo: 001.355,12 - Ent: 49 (HARD) - NT7 - Tarefa é Preemptada faltando 152.26 us. Prioridade 500  
Tempo: 001.375,00 - Ent: 75 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.375,00 - Ent: 75 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 22.5  
Tempo: 001.375,00 - Ent: 77 (HARD) - NT7 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 001.375,00 - Ent: 75 (SERVER) - NT7 - Tarefa é Preemptada faltando 25 us. Prioridade 200  
Tempo: 001.387,50 - Ent: 68 (HARD) - NT4 - Tarefa Termina Execução  
Tempo: 001.387,50 - Ent: 68 (HARD) - NT4 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.395,00 - Ent: 75 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.395,00 - Ent: 75 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 42.5  
Tempo: 001.400,00 - Ent: 59 (SERVER) - NT1 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.405,00 - Ent: 58 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.412,50 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.412,86 - Ent: 59 (SERVER) - NT1 - Tarefa Termina Execução  
Tempo: 001.412,86 - Ent: 59 (SERVER) - NT1 - Tarefa Aperiódica (3) Termina de ser Atendida por Server Task (59)  
Tempo: 001.412,86 - Ent: 3 (VBR) - NT1 - Monta Pedido de Conexão de 64 bytes de NT 1 para NT 3. Tamanho da Mensagem 6144 bytes  
Tempo: 001.412,86 - Ent: 3 (VBR) - NT1 - Inicia Transmissão de Pedido de Conexão pelo BS  
Tempo: 001.415,00 - Ent: 75 (SERVER) - NT7 - Tarefa é Preemptada faltando 5 us. Prioridade 200  
Tempo: 001.415,42 - Ent: 3 (VBR) - NC - Pedido de Conexão entre NT 1 e NT 3 chega ao NC com Prioridade 0  
Tempo: 001.417,62 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.417,62 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 1417.62 e 21 comunicações  
Tempo: 001.417,62 - Ent: 75 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.417,62 - Ent: 75 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 39.8800000000001  
Tempo: 001.419,42 - Ent: 3 (VBR) - NC - Determina Existência de 1 Canais Físicos Disponíveis entre NT 1 e NT 3  
Tempo: 001.419,42 - Ent: 3 (VBR) - NC - Aloca 1 Canais Físicos Disponíveis. Tempo de conexão: 4. Deadline: 1411.87409678156  
Tempo: 001.419,42 - Ent: 3 (VBR) - NC - Inicia Transmissão de Mensagem de 6144 bytes pelo Crossbar. Tempo de Transmissão: 61.44  
Tempo: 001.420,00 - Ent: 54 (HARD) - NT1 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 001.422,62 - Ent: 75 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 001.422,62 - Ent: 75 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (75)  
Tempo: 001.422,62 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 20 comunicações  
Tempo: 001.425,00 - Ent: 58 (SERVER) - NT3 - Tarefa é Preemptada faltando 8.19917930081988 us. Prioridade 200  
Tempo: 001.435,00 - Ent: 78 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 21 comunicações  
Tempo: 001.435,00 - Ent: 78 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 55  
Tempo: 001.435,00 - Ent: 78 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.435,00 - Ent: 78 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 45.1199999999999  
Tempo: 001.437,50 - Ent: 80 (HARD) - NT8 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 001.437,50 - Ent: 80 (HARD) - NT8 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 001.445,00 - Ent: 58 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.450,00 - Ent: 70 (HARD) - NT5 - Tarefa Termina Execução  
Tempo: 001.450,00 - Ent: 70 (HARD) - NT5 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.453,20 - Ent: 58 (SERVER) - NT3 - Tarefa Termina Execução  
Tempo: 001.453,20 - Ent: 58 (SERVER) - NT3 - Tarefa Aperiódica (56) Termina de ser Atendida por Server Task (58)  
Tempo: 001.453,20 - Ent: 56 (VBR) - NT3 - Monta Pedido de Conexão de 64 bytes de NT 3 para NT 2. Tamanho da Mensagem 64 bytes  
Tempo: 001.453,20 - Ent: 56 (VBR) - NT3 - Inicia Transmissão de Pedido de Conexão pelo BS  
Tempo: 001.455,00 - Ent: 78 (SERVER) - NT7 - Tarefa é Preemptada faltando 5 us. Prioridade 200  
Tempo: 001.455,00 - Ent: 49 (HARD) - NT7 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 001.455,76 - Ent: 56 (VBR) - NC - Pedido de Conexão entre NT 3 e NT 2 chega ao NC com Prioridade 0  
Tempo: 001.459,76 - Ent: 56 (VBR) - NC - Determina Existência de 1 Canais Físicos Disponíveis entre NT 3 e NT 2  
Tempo: 001.459,76 - Ent: 56 (VBR) - NC - Aloca 1 Canais Físicos Disponíveis. Tempo de conexão: 4. Deadline: 1440.21735807014  
Tempo: 001.459,76 - Ent: 56 (VBR) - NC - Inicia Transmissão de Mensagem de 64 bytes pelo Crossbar. Tempo de Transmissão: .64  
Tempo: 001.460,40 - Ent: 56 (VBR) - NT2 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.460,40 - Ent: 56 (VBR) - NT2 - Tarefa Chega com Prioridade 1460.39917930082 e 1 comunicações  
Tempo: 001.460,40 - Ent: 56 (VBR) - NT2 - Monta Pedido de Desconexão  
Tempo: 001.460,40 - Ent: 56 (VBR) - NT2 - Inicia Transmissão de Pedido de Desconexão pelo BS  
Tempo: 001.460,40 - Ent: 81 (SERVER) - NT2 - Tarefa Chega com Prioridade 200 e 1 comunicações

Tempo: 001.460,40 - Ent: 81 (SERVER) - NT2 - Inicializa Servidor Aperiódico (56). Capacidade 2.26195507935324

Tempo: 001.460,40 - Ent: 81 (SERVER) - NT2 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.462,66 - Ent: 81 (SERVER) - NT2 - Tarefa é Preemptada faltando 208.267720034329 us. Prioridade 200

Tempo: 001.462,96 - Ent: 56 (VBR) - NC - Pedido de Desconexão entre NT 2 e NT 3 chega ao NC com Prioridade 0

Tempo: 001.465,00 - Ent: 65 (HARD) - NT3 - Tarefa Inicia Execução com Prioridade 500

Tempo: 001.466,96 - Ent: 56 (VBR) - NC - Libera 1 Canais Físicos do Crossbar entre NT 2 e NT 3

Tempo: 001.475,00 - Ent: 49 (HARD) - NT7 - Tarefa deixa o Sistema Após Perder Deadline por 100 us

Tempo: 001.475,00 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12

Tempo: 001.480,12 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar

Tempo: 001.480,12 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 1480.12 e 22 comunicações

Tempo: 001.480,12 - Ent: 78 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.480,12 - Ent: 78 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 17.3800000000001

Tempo: 001.480,86 - Ent: 3 (VBR) - NT3 - Mensagem Termina Transferência pelo Crossbar

Tempo: 001.480,86 - Ent: 3 (VBR) - NT3 - Tarefa Chega com Prioridade 1480.86312426732 e 4 comunicações

Tempo: 001.480,86 - Ent: 3 (VBR) - NT3 - Monta Pedido de Desconexão

Tempo: 001.480,86 - Ent: 3 (VBR) - NT3 - Inicia Transmissão de Pedido de Desconexão pelo BS

Tempo: 001.480,86 - Ent: 82 (SERVER) - NT3 - Tarefa Chega com Prioridade 200 e 4 comunicações

Tempo: 001.480,86 - Ent: 82 (SERVER) - NT3 - Inicializa Servidor Aperiódico (3). Capacidade 51.8008206991801

Tempo: 001.480,86 - Ent: 65 (HARD) - NT3 - Tarefa é Preemptada faltando 184.136875732683 us. Prioridade 500

Tempo: 001.483,42 - Ent: 3 (VBR) - NC - Pedido de Desconexão entre NT 3 e NT 1 chega ao NC com Prioridade 0

Tempo: 001.485,00 - Ent: 82 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.485,00 - Ent: 82 (SERVER) - NT3 - Tempo de Resposta dessa Tarefa: 72.1368757326827

Tempo: 001.485,12 - Ent: 78 (SERVER) - NT7 - Tarefa Termina Execução

Tempo: 001.485,12 - Ent: 78 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (78)

Tempo: 001.485,12 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 21 comunicações

Tempo: 001.487,42 - Ent: 3 (VBR) - NC - Libera 1 Canais Físicos do Crossbar entre NT 3 e NT 1

Tempo: 001.495,00 - Ent: 83 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 22 comunicações

Tempo: 001.495,00 - Ent: 83 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 30

Tempo: 001.495,00 - Ent: 83 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.495,00 - Ent: 83 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 47.6199999999999

Tempo: 001.500,00 - Ent: 85 (HARD) - NT1 - Tarefa Chega com Prioridade 500 e 0 comunicações

Tempo: 001.500,00 - Ent: 54 (HARD) - NT1 - Tarefa é Preemptada faltando 90.2837138867339 us. Prioridade 500

Tempo: 001.505,00 - Ent: 82 (SERVER) - NT3 - Tarefa é Preemptada faltando 163.541500943249 us. Prioridade 200

Tempo: 001.512,50 - Ent: 74 (HARD) - NT6 - Tarefa Termina Execução

Tempo: 001.512,50 - Ent: 74 (HARD) - NT6 - Tarefa Deixa o Sistema Normalmente. 0 comunicações

Tempo: 001.520,00 - Ent: 85 (HARD) - NT1 - Tarefa Inicia Execução com Prioridade 500

Tempo: 001.520,00 - Ent: 83 (SERVER) - NT7 - Tarefa Termina Execução

Tempo: 001.520,00 - Ent: 83 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (83)

Tempo: 001.520,00 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 22 comunicações

Tempo: 001.520,00 - Ent: 77 (HARD) - NT7 - Tarefa Inicia Execução com Prioridade 500

Tempo: 001.525,00 - Ent: 82 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.525,00 - Ent: 82 (SERVER) - NT3 - Tempo de Resposta dessa Tarefa: 112.136875732683

Tempo: 001.537,50 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12

Tempo: 001.540,00 - Ent: 85 (HARD) - NT1 - Tarefa é Preemptada faltando 180 us. Prioridade 500

Tempo: 001.542,62 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar

Tempo: 001.542,62 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 1542.62 e 23 comunicações

Tempo: 001.542,62 - Ent: 86 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 23 comunicações

Tempo: 001.542,62 - Ent: 86 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 5

Tempo: 001.542,62 - Ent: 77 (HARD) - NT7 - Tarefa é Preemptada faltando 177.38 us. Prioridade 500

Tempo: 001.545,00 - Ent: 82 (SERVER) - NT3 - Tarefa é Preemptada faltando 143.541500943249 us. Prioridade 200

Tempo: 001.555,00 - Ent: 86 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.555,00 - Ent: 86 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 2.5

Tempo: 001.560,00 - Ent: 54 (HARD) - NT1 - Tarefa Inicia Execução com Prioridade 500

Tempo: 001.562,50 - Ent: 81 (SERVER) - NT2 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.562,50 - Ent: 88 (HARD) - NT2 - Tarefa Chega com Prioridade 500 e 0 comunicações

Tempo: 001.562,50 - Ent: 81 (SERVER) - NT2 - Tarefa é Preemptada faltando 208.267720034329 us. Prioridade 200

Tempo: 001.565,00 - Ent: 82 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200

Tempo: 001.565,00 - Ent: 82 (SERVER) - NT3 - Tempo de Resposta dessa Tarefa: 152.136875732683

Tempo: 001.575,00 - Ent: 86 (SERVER) - NT7 - Tarefa é Preemptada faltando 5 us. Prioridade 200

Tempo: 001.580,00 - Ent: 54 (HARD) - NT1 - Tarefa deixa o Sistema Após Perder Deadline por 80 us  
Tempo: 001.582,50 - Ent: 81 (SERVER) - NT2 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.585,00 - Ent: 82 (SERVER) - NT3 - Tarefa é Preemptada faltando 123.541500943249 us. Prioridade 200  
Tempo: 001.600,00 - Ent: 85 (HARD) - NT1 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 001.600,00 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.602,50 - Ent: 81 (SERVER) - NT2 - Tarefa é Preemptada faltando 188.267720034329 us. Prioridade 200  
Tempo: 001.605,00 - Ent: 82 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.605,00 - Ent: 82 (SERVER) - NT3 - Tempo de Resposta dessa Tarefa: 192.136875732683  
Tempo: 001.605,12 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.605,12 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 1605.12 e 24 comunicações  
Tempo: 001.615,00 - Ent: 86 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.615,00 - Ent: 86 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 2.5  
Tempo: 001.620,00 - Ent: 86 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 001.620,00 - Ent: 86 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (86)  
Tempo: 001.620,00 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 23 comunicações  
Tempo: 001.622,50 - Ent: 81 (SERVER) - NT2 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.625,00 - Ent: 82 (SERVER) - NT3 - Tarefa é Preemptada faltando 103.541500943249 us. Prioridade 200  
Tempo: 001.625,00 - Ent: 90 (HARD) - NT3 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 001.625,00 - Ent: 91 (SERVER) - NT3 - Tarefa Chega com Prioridade 200 e 4 comunicações  
Tempo: 001.625,00 - Ent: 91 (SERVER) - NT3 - Inicializa Servidor Aperiódico (3). Capacidade 60  
Tempo: 001.625,00 - Ent: 91 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.625,00 - Ent: 91 (SERVER) - NT3 - Tempo de Resposta dessa Tarefa: 212.136875732683  
Tempo: 001.635,00 - Ent: 92 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 24 comunicações  
Tempo: 001.635,00 - Ent: 92 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 75  
Tempo: 001.635,00 - Ent: 92 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.635,00 - Ent: 92 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 42.5  
Tempo: 001.637,50 - Ent: 80 (HARD) - NT8 - Tarefa Termina Execução  
Tempo: 001.637,50 - Ent: 80 (HARD) - NT8 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.642,50 - Ent: 81 (SERVER) - NT2 - Tarefa é Preemptada faltando 168.267720034329 us. Prioridade 200  
Tempo: 001.645,00 - Ent: 91 (SERVER) - NT3 - Tarefa é Preemptada faltando 163.541500943249 us. Prioridade 200  
Tempo: 001.645,00 - Ent: 65 (HARD) - NT3 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 001.660,00 - Ent: 92 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 001.660,00 - Ent: 92 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (92)  
Tempo: 001.660,00 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 24 comunicações  
Tempo: 001.660,00 - Ent: 77 (HARD) - NT7 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 001.662,50 - Ent: 81 (SERVER) - NT2 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.662,50 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.665,00 - Ent: 65 (HARD) - NT3 - Tarefa deixa o Sistema Após Perder Deadline por 40 us  
Tempo: 001.667,62 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.667,62 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 1667.62 e 25 comunicações  
Tempo: 001.667,62 - Ent: 93 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 25 comunicações  
Tempo: 001.667,62 - Ent: 93 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 50  
Tempo: 001.667,62 - Ent: 77 (HARD) - NT7 - Tarefa é Preemptada faltando 169.76 us. Prioridade 500  
Tempo: 001.675,00 - Ent: 93 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.675,00 - Ent: 93 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 22.5  
Tempo: 001.682,50 - Ent: 81 (SERVER) - NT2 - Tarefa é Preemptada faltando 148.267720034329 us. Prioridade 200  
Tempo: 001.685,00 - Ent: 82 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.685,00 - Ent: 82 (SERVER) - NT3 - Tempo de Resposta dessa Tarefa: 272.136875732683  
Tempo: 001.687,50 - Ent: 95 (HARD) - NT4 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 001.687,50 - Ent: 95 (HARD) - NT4 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 001.695,00 - Ent: 93 (SERVER) - NT7 - Tarefa é Preemptada faltando 5 us. Prioridade 200  
Tempo: 001.702,50 - Ent: 81 (SERVER) - NT2 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.705,00 - Ent: 82 (SERVER) - NT3 - Tarefa é Preemptada faltando 83.541500943248 us. Prioridade 200  
Tempo: 001.715,00 - Ent: 93 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.715,00 - Ent: 93 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 22.5  
Tempo: 001.720,00 - Ent: 93 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 001.720,00 - Ent: 93 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (93)  
Tempo: 001.720,00 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 25 comunicações  
Tempo: 001.722,50 - Ent: 81 (SERVER) - NT2 - Tarefa é Preemptada faltando 128.267720034329 us. Prioridade 200

Tempo: 001.725,00 - Ent: 91 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.725,00 - Ent: 91 (SERVER) - NT3 - Tempo de Resposta dessa Tarefa: 312.136875732683  
Tempo: 001.725,00 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.730,12 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.730,12 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 1730.12 e 26 comunicações  
Tempo: 001.730,12 - Ent: 96 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 26 comunicações  
Tempo: 001.730,12 - Ent: 96 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 25  
Tempo: 001.730,12 - Ent: 96 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.730,12 - Ent: 96 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 47.3800000000001  
Tempo: 001.745,00 - Ent: 91 (SERVER) - NT3 - Tarefa é Preemptada faltando 143.541500943249 us. Prioridade 200  
Tempo: 001.750,00 - Ent: 98 (HARD) - NT5 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 001.750,00 - Ent: 98 (HARD) - NT5 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 001.755,12 - Ent: 96 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 001.755,12 - Ent: 96 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (96)  
Tempo: 001.755,12 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 26 comunicações  
Tempo: 001.755,12 - Ent: 77 (HARD) - NT7 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 001.780,00 - Ent: 85 (HARD) - NT1 - Tarefa Termina Execução  
Tempo: 001.780,00 - Ent: 85 (HARD) - NT1 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.782,50 - Ent: 81 (SERVER) - NT2 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.787,50 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.792,62 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.792,62 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 1792.62 e 27 comunicações  
Tempo: 001.802,50 - Ent: 81 (SERVER) - NT2 - Tarefa é Preemptada faltando 108.267720034329 us. Prioridade 200  
Tempo: 001.805,00 - Ent: 82 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.805,00 - Ent: 82 (SERVER) - NT3 - Tempo de Resposta dessa Tarefa: 392.136875732683  
Tempo: 001.812,50 - Ent: 100 (HARD) - NT6 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 001.812,50 - Ent: 100 (HARD) - NT6 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 001.815,00 - Ent: 101 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 27 comunicações  
Tempo: 001.815,00 - Ent: 101 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 80  
Tempo: 001.815,00 - Ent: 77 (HARD) - NT7 - Tarefa é Preemptada faltando 109.88 us. Prioridade 500  
Tempo: 001.822,50 - Ent: 81 (SERVER) - NT2 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.825,00 - Ent: 82 (SERVER) - NT3 - Tarefa é Preemptada faltando 63.5415009432488 us. Prioridade 200  
Tempo: 001.835,00 - Ent: 101 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.835,00 - Ent: 101 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 42.3800000000001  
Tempo: 001.842,50 - Ent: 81 (SERVER) - NT2 - Tarefa é Preemptada faltando 88.2677200343292 us. Prioridade 200  
Tempo: 001.845,00 - Ent: 91 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.845,00 - Ent: 91 (SERVER) - NT3 - Tempo de Resposta dessa Tarefa: 432.136875732683  
Tempo: 001.850,00 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.855,00 - Ent: 101 (SERVER) - NT7 - Tarefa é Preemptada faltando 5 us. Prioridade 200  
Tempo: 001.855,12 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.855,12 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 1855.12 e 28 comunicações  
Tempo: 001.855,12 - Ent: 101 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.855,12 - Ent: 101 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 42.3800000000001  
Tempo: 001.860,12 - Ent: 101 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 001.860,12 - Ent: 101 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (101)  
Tempo: 001.860,12 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 27 comunicações  
Tempo: 001.862,50 - Ent: 81 (SERVER) - NT2 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.865,00 - Ent: 91 (SERVER) - NT3 - Tarefa é Preemptada faltando 123.541500943249 us. Prioridade 200  
Tempo: 001.875,00 - Ent: 102 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 28 comunicações  
Tempo: 001.875,00 - Ent: 102 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 55  
Tempo: 001.875,00 - Ent: 102 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.875,00 - Ent: 102 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 42.61999999999999  
Tempo: 001.875,00 - Ent: 104 (HARD) - NT7 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 001.875,00 - Ent: 102 (SERVER) - NT7 - Tarefa é Preemptada faltando 25 us. Prioridade 200  
Tempo: 001.875,00 - Ent: 77 (HARD) - NT7 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 001.882,50 - Ent: 81 (SERVER) - NT2 - Tarefa é Preemptada faltando 68.2677200343292 us. Prioridade 200  
Tempo: 001.885,00 - Ent: 82 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.885,00 - Ent: 82 (SERVER) - NT3 - Tempo de Resposta dessa Tarefa: 472.136875732683  
Tempo: 001.887,50 - Ent: 95 (HARD) - NT4 - Tarefa Termina Execução

Tempo: 001.887,50 - Ent: 95 (HARD) - NT4 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.895,00 - Ent: 77 (HARD) - NT7 - Tarefa deixa o Sistema Após Perder Deadline por 20 us  
Tempo: 001.902,50 - Ent: 81 (SERVER) - NT2 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.905,00 - Ent: 82 (SERVER) - NT3 - Tarefa é Preemptada faltando 43.5415009432488 us. Prioridade 200  
Tempo: 001.912,50 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.915,00 - Ent: 102 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.915,00 - Ent: 102 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 22.5  
Tempo: 001.917,62 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.917,62 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 1917.62 e 29 comunicações  
Tempo: 001.917,62 - Ent: 102 (SERVER) - NT7 - Tarefa é Preemptada faltando 22.3800000000001 us. Prioridade 200  
Tempo: 001.922,50 - Ent: 81 (SERVER) - NT2 - Tarefa é Preemptada faltando 48.2677200343292 us. Prioridade 200  
Tempo: 001.925,00 - Ent: 91 (SERVER) - NT3 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.925,00 - Ent: 91 (SERVER) - NT3 - Tempo de Resposta dessa Tarefa: 512.136875732683  
Tempo: 001.935,00 - Ent: 102 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.935,00 - Ent: 102 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 42.5  
Tempo: 001.937,50 - Ent: 106 (HARD) - NT8 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 001.937,50 - Ent: 106 (HARD) - NT8 - Tarefa Inicia Execução com Prioridade 500  
Tempo: 001.945,00 - Ent: 91 (SERVER) - NT3 - Tarefa é Preemptada faltando 103.541500943249 us. Prioridade 200  
Tempo: 001.950,00 - Ent: 98 (HARD) - NT5 - Tarefa Termina Execução  
Tempo: 001.950,00 - Ent: 98 (HARD) - NT5 - Tarefa Deixa o Sistema Normalmente. 0 comunicações  
Tempo: 001.955,00 - Ent: 102 (SERVER) - NT7 - Tarefa é Preemptada faltando 2.38000000000011 us. Prioridade 200  
Tempo: 001.975,00 - Ent: 102 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.975,00 - Ent: 102 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 22.5  
Tempo: 001.975,00 - Ent: 6 (CBR) - NC - Inicia Transmissão de Mensagem de 512 bytes pelo Crossbar. Tempo de Transmissão: 5.12  
Tempo: 001.977,38 - Ent: 102 (SERVER) - NT7 - Tarefa Termina Execução  
Tempo: 001.977,38 - Ent: 102 (SERVER) - NT7 - Tarefa Aperiódica (6) Termina de ser Atendida por Server Task (102)  
Tempo: 001.977,38 - Ent: 6 (CBR) - NT7 - Tarefa Deixa o Sistema Normalmente. 28 comunicações  
Tempo: 001.980,12 - Ent: 6 (CBR) - NT7 - Mensagem Termina Transferência pelo Crossbar  
Tempo: 001.980,12 - Ent: 6 (CBR) - NT7 - Tarefa Chega com Prioridade 1980.12 e 30 comunicações  
Tempo: 001.980,12 - Ent: 107 (SERVER) - NT7 - Tarefa Chega com Prioridade 200 e 29 comunicações  
Tempo: 001.980,12 - Ent: 107 (SERVER) - NT7 - Inicializa Servidor Aperiódico (6). Capacidade 30  
Tempo: 001.980,12 - Ent: 107 (SERVER) - NT7 - Tarefa Inicia Execução com Prioridade 200  
Tempo: 001.980,12 - Ent: 107 (SERVER) - NT7 - Variação da Taxa de Execução dessa Amostra: 57.3800000000001  
Tempo: 002.000,00 - Ent: 109 (HARD) - NT1 - Tarefa Chega com Prioridade 500 e 0 comunicações  
Tempo: 002.000,00 - Ent: 109 (HARD) - NT1 - Tarefa Inicia Execução com Prioridade 500

**ANEXO 13** – Resultados da Regressão Linear dos Experimentos de 5 Fatores (2<sup>5</sup>)

Tempo de Resposta VBR

<b>Fator</b>	<b>Coefficiente Estimado</b>	<b>Graus de Liberdade</b>	<b>Erro Padrão</b>	<b>I.C. 95% Lim. Inf.</b>	<b>I.C. 95% Lim. Sup.</b>
Independ.	5502,06	1	357,33	4723,50	6280,63
A	287,33	1	357,33	-491,24	1065,89
B	-1606,55	1	357,33	-2385,12	-827,99
C	-698,73	1	357,33	-1477,29	79,84
D	-411,07	1	357,33	-1189,64	367,49
E	-2235,47	1	357,33	-3014,03	-1456,90
AC	274,42	1	357,33	-504,15	1052,99
AD	-723,74	1	357,33	-1502,31	54,82
AE	-97,01	1	357,33	-875,57	681,56
BC	650,96	1	357,33	-127,61	1429,52
BD	842,13	1	357,33	63,56	1620,69
BE	854,99	1	357,33	76,46	1633,56
CD	-382,92	1	357,33	-1161,48	395,65
CE	559,83	1	357,33	-218,74	1338,39
DE	65,41	1	357,33	-713,16	843,97
ACD	-653,82	1	357,33	-1432,39	124,74
ACE	-148,88	1	357,33	-927,44	629,69
ADE	-216,13	1	357,33	-994,70	562,43
CDE	290,55	1	357,33	-488,02	1069,11
ACDE	785,92	1	357,33	7,36	1564,49

<b>Num.</b>	<b>Resultado Atual</b>	<b>Valor Predito</b>	<b>Residuo</b>	<b>T Outlier</b>
1	14967,00	13724,83	1242,17	1,004
2	11992,00	11782,63	209,37	0,162
3	4573,40	5815,57	-1242,17	-1,004
4	3664,00	3873,37	-209,37	-0,162
5	5975,70	7526,65	-1550,95	-1,287
6	15056,00	13036,62	2019,38	1,771
7	3772,17	2221,22	1550,95	1,287
8	5711,80	7731,18	-2019,38	-1,771
9	10770,00	10570,27	199,73	0,155
10	13087,00	12356,62	730,38	0,573
11	5829,80	6029,53	-199,73	-0,155
12	7085,50	7815,88	-730,38	-0,573
13	7692,50	7437,21	255,29	0,198
14	2052,40	5157,76	-3105,36	-3,483
15	5245,00	5500,29	-255,29	-0,198
16	6326,20	3220,84	3105,36	3,483
17	3107,50	4766,69	-1659,19	-1,392
18	5909,50	7040,19	-1130,69	-0,907
19	1936,60	277,41	1659,19	1,392
20	3681,60	2550,91	1130,69	0,907
21	3385,70	3384,83	0,87	0,001
22	6501,15	5632,11	869,04	0,686
23	1498,50	1499,37	-0,87	-0,001
24	2877,60	3746,64	-869,04	-0,686
25	5168,40	4719,78	448,62	0,349
26	2665,00	2705,38	-40,38	-0,031
27	3150,40	3599,02	-448,62	-0,349
28	1625,00	1584,62	40,38	0,031
29	3069,50	2440,04	629,46	0,492
30	2338,50	1456,22	882,28	0,697
31	3293,64	3923,10	-629,46	-0,492
32	2057,00	2939,28	-882,28	-0,697

## ANEXO 14 – Resultados da Regressão Linear dos Experimentos de 5 Fatores (2<sup>5</sup>)

### Varição da Taxa de Execução CBR

Fator	Coefficiente Estimado	Graus de Liberdade	Erro Padrão	I.C. 95% Lim. Inf.	I.C. 95% Lim. Sup.
Independ.	3,95	1	0,060	3,82	4,07
A	0,24	1	0,060	0,11	0,36
C	-0,47	1	0,060	-0,59	-0,34
D	0,059	1	0,060	-0,065	0,18
AC	-0,29	1	0,060	-0,42	-0,17
AD	0,16	1	0,060	0,034	0,28
CD	0,23	1	0,060	0,10	0,35

<b>Num.</b>	<b>Resultado Atual</b>	<b>Valor Predito</b>	<b>Residuo</b>	<b>T Outlier</b>
1	3,81	4,21	-0,40	-1,344
2	4,84	4,96	-0,11	-0,370
3	4,16	4,21	-0,049	-0,158
4	5,20	4,96	0,24	0,797
5	3,44	3,41	0,031	0,102
6	2,68	2,98	-0,30	-1,001
7	3,64	3,41	0,22	0,736
8	2,85	2,98	-0,13	-0,432
9	3,49	3,56	-0,074	-0,241
10	4,93	4,94	-0,013	-0,043
11	3,59	3,56	0,030	0,098
12	5,03	4,94	0,090	0,292
13	3,55	3,66	-0,12	-0,380
14	3,93	3,87	0,058	0,187
15	3,59	3,66	-0,075	-0,244
16	5,02	3,87	1,15	5,759
17	4,11	4,21	-0,093	-0,303
18	4,90	4,96	-0,057	-0,185
19	4,43	4,21	0,22	0,711
20	5,21	4,96	0,25	0,834
21	3,14	3,41	-0,28	-0,911
22	2,72	2,98	-0,26	-0,863
23	3,76	3,41	0,35	1,151
24	3,36	2,98	0,37	1,246
25	3,60	3,56	0,037	0,119
26	4,59	4,94	-0,35	-1,164
27	3,89	3,56	0,33	1,105
28	4,89	4,94	-0,052	-0,169
29	3,57	3,66	-0,097	-0,316
30	3,40	3,87	-0,47	-1,613
31	3,63	3,66	-0,036	-0,117
32	3,46	3,87	-0,41	-1,384

## ANEXO 15 – Resultados da Regressão Linear dos Experimentos de 5 Fatores (2<sup>5</sup>)

Tempo de Conexão

Fator	Coefficiente Estimado	Graus de Liberdade	Erro Padrão	I.C. 95% Lim. Inf.	I.C. 95% Lim. Sup.
Inpedend.	58,83	1	2,06	54,58	63,08
A	10,35	1	2,06	6,10	14,60
B	-13,95	1	2,06	-18,20	-9,70
D	-2,68	1	2,06	-6,93	1,57
E	-17,16	1	2,06	-21,40	-12,91
BD	5,75	1	2,06	1,50	10,00
BE	-7,43	1	2,06	-11,68	-3,18

<b>Num.</b>	<b>Resultado Atual</b>	<b>Valor Predito</b>	<b>Resíduo</b>	<b>T Outlier</b>
1	78,35	80,60	-2,24	-0,213
2	97,66	101,29	-3,63	-0,345
3	59,79	56,05	3,74	0,356
4	74,53	76,74	-2,22	-0,211
5	70,71	80,60	-9,89	-0,957
6	113,52	101,29	12,23	1,195
7	44,81	56,05	-11,25	-1,094
8	62,61	76,74	-14,13	-1,396
9	72,25	63,73	8,52	0,820
10	107,03	84,42	22,61	2,389
11	61,06	62,18	-1,13	-0,107
12	90,46	82,87	7,58	0,728
13	57,38	63,73	-6,35	-0,607
14	63,17	84,42	-21,25	-2,215
15	73,05	62,18	10,87	1,056
16	89,39	82,87	6,52	0,624
17	47,45	61,14	-13,69	-1,349
18	96,94	81,83	15,11	1,501
19	15,97	6,89	9,08	0,876
20	32,27	27,58	4,69	0,447
21	51,45	61,14	-9,69	-0,937
22	93,63	81,83	11,80	1,151
23	15,80	6,89	8,91	0,859
24	28,76	27,58	1,18	0,112
25	52,59	44,27	8,32	0,801
26	64,67	64,96	-0,29	-0,028
27	17,31	13,02	4,29	0,409
28	21,28	33,71	-12,43	-1,216
29	42,65	44,27	-1,62	-0,154
30	55,02	64,96	-9,94	-0,962
31	15,14	13,02	2,12	0,202
32	15,88	33,71	-17,83	-1,805