UNIVERSIDADE FEDERAL DE SANTA CATARINA

DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO

Gerência das Reservas Maleáveis de Recursos na Grade

Rodrigo Micheli

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau Bacharel em Ciências da Computação

Florianópolis – SC

2007/2

Rodrigo Micheli

Gerência das Reservas Maleáveis de Recursos na Grade

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau Bacharel em Ciências da Computação

Orientador: Professor Carlos Becker Westphall

Banca Examinadora

_____

Mestre Leornado Kuranth

_____

Professora Doutora Carla Merkle Westphall

Resumo

Este trabalho apresenta um estudo de gerência de reservas dos recursos na Grade. Foi abordado neste projeto o conceito de reservas maleáveis que poderão ser empacotadas de forma imediata, antecipada e com relaxamento. As reservas maleáveis são muito utilizadas para reservar a largura de banda da rede, pois com estas reservas o aproveitamento da rede se torna maior e a taxa de aceitação dos pedidos também, porque o gerenciador consegue ajustar mais facilmente os pedidos, verificando a utilização momentânea do recurso através das reservas existentes fita. As reservas de recursos são importantes para obter QoS em grades e com as reservas maleáveis é possível obter uma melhor qualidade na transmissão dos dados pela rede. O modelo proposto neste trabalho, para fazer a representação das reservas maleáveis e dos recursos, é baseado no empacotamento em fita. Os usuários, o gerenciador de recursos e o gerenciador de reservas são entidades separadas e que se relacionam. Foram implementados algoritmos, que são utilizados pelo gerenciador, para fazer o escalonamento das reservas maleáveis na fita.

Sumário

Lista de Figuras

Lista de abreviaturas e siglas

QoS	Qualidade de serviço,	p. 9
RM	Reserva Maleável,	p. 10
RA	Reserva Antecipada,	p. 10
RI	Reserva Imediata,	p. 10
FIFO	First in, First out,	p. 15
GARA	Globus Architecture for Reservation and Allocation,	p. 22
MSS	MetaScheduling Service,	p. 25
RMS	Resource Management Systems,	p. 25
GARG	Gerenciador Adaptativo Gerenciador dos Recursos da Grade,	p. 26
EST	Tempo mais cedo possível para começar a operação,	p. 31
LET	Tempo limite para a operação acabar,	p. 31
VS	Variable Slots,	p. 36
APN	Armazena no Próximo Nível,	p. 38
APNP	Armazena no primeiro nível possível,	p. 39
AMN	Armazena no Melhor Nível,	p. 41
AUN	Armazena no Último Nível,	p. 42

1 Introdução

Para resolver problemas computacionais complexos existe a necessidade de um alto poder computacional, processamento e armazenamento dos dados. (ASSUNÇÃO, 2003) afirma que estes problemas necessitam de um grande número de recursos que não podem ser mantidos em um mesmo domínio por questões econômicas e técnicas. Estes problemas são freqüentemente encontrados em trabalhos nas áreas científicas e comercias em aplicações de grande escala.

A solução para estes problemas poderia envolver a utilização de um hardware com alto desempenho, disponibilizados por cluster de computadores ou supercomputadores, visto que há anos atrás eram as únicas arquiteturas disponíveis para a execução destas aplicações. Elas têm um elevado custo de aquisição e manutenção e nem sempre possuem os recursos suficientes para as resoluções dos problemas. Esperava-se que todas as necessidades computacionais fossem supridas por clusters e supercomputadores reunidos em um único domínio administrativo, mas conforme (ASSUNÇÃO, 2003), esta abordagem se tornou impraticável e este cenário tem mudado com a grande disponibilidade de recursos computacionais e de rede a um custo muito menor, além da constante evolução das tecnologias de software.

A solução para estes problemas tem sido a utilização de recursos distribuídos com intuito de agregá-los, formando um ambiente computacional distribuído em grande escala. Com este poder computacional podemos encontrar a solução para os problemas mais complexos. Esta agregação e compartilhamento de recursos conectados em rede, formando um sistema distribuído em larga escala é chamada de computação em grade. O objetivo desta tecnologia é processar e manipular grande quantidade de dados em grande escala, utilizando recursos da computação distribuída. A grade funciona como um supercomputador, pois obtém recursos de vários computadores para resolver um determinado problema que levaria muito tempo para ser resolvido por um único computador.

A grade possui uma plataforma de software, que geralmente é chamada de middleware. Esta plataforma fornece as funcionalidades básicas de interligação entre os computadores, identificação dos recursos disponibilizados, das aplicações dos usuários da grade, funções para localizar

os recursos desejados, e funções de segurança, como criptografia. “Aplicação de Grade” é o nome dado a uma aplicação construída sobre um middleware da grade e que pode utilizar os recursos e as funcionalidades fornecidas pela Grade (KUNRATH; WESTPHALL; KOCH, 2008).

As estruturas e arquiteturas da grade são tecnologias que estão em desenvolvimento. Existem várias propostas para a melhoria da grade, sendo estas propostas as melhorias das funcionalidades já existentes ou a inclusão de novas funcionalidades. Espera-se, desta forma, popularizar a utilização da grade, ou seja, tornar o seu uso simples e disponível no mundo todo para qualquer indivíduo. Um dos desafios, para estas melhorias, seria o fornecimento dos mais diversos recursos com qualidade de Serviços (QoS). Por isso, neste trabalho foi proposto um gerenciador de reservas maleáveis dos recursos para garantir QOS na grade ao usuário.

A alta qualidade de serviço (QoS) é algo desejável na computação em grade. Esta qualidade pode ser garantida através de uma configuração adequada, reservas e alocação dos recursos correspondentes a grade. Este trabalho está relacionado com o gerenciamento das reservas maleáveis de recursos. A reserva maleável pode ser feita de forma imediata ou de forma antecipada. (SIDDIQUI; VILLAZÓN; FAHRINGER, 2006) explica que o problema se torna mais difícil e mais desafiador com o rápido crescimento de recursos e aplicações na grade, pois as aplicações têm que competir por recursos, enquanto lidam com a complexidade do middleware. Por isso é interessante utilizar reservas antecipadas (Advance Reservation), porque elas possuem um horário que indica o inicio e o fim da aplicação.

Quanto mais é utilizada a grade em várias instituições, também é maior o desafio de conseguir uma melhor gerência na alocação de reservas para utilização de determinados recursos existentes no ambiente, pois através de uma boa gerência destas reservas, se obtém uma elevada qualidade de serviço (QoS) na Grade.

Este gerenciador pode gerenciar uma ou mais reservas de vários recursos, controlados por um ou mais gerenciadores de recursos. Porém não será a função deste gerenciador alocar o recurso para o usuário, esta função é feita pelos gerenciadores de recursos. Portanto, o gerenciador de reservas não precisa estar localizado internamente ao gerenciador de recursos, pode estar localizado inclusive em outro nó da grade.

Existem alguns algoritmos que realizam a gerência da alocação das reservas. Neste trabalho serão implementados alguns algoritmos que farão o agendamento de reservas maleáveis, para comparar a performance e a eficiência destes algoritmos em determinados casos. A idéia é apresentar como as reservas maleáveis se comportam nos casos de utilização do recurso, apresenta a taxa de perda nos pedidos de reserva para comparar com o desempenho das reservas fixas.

Os algoritmos implementados neste trabalho têm a função de trabalhar com reservas maleáveis (Malleable Reservation) de forma antecipada (Advance Reservation) imediata (Immediata Reservation) e com relaxamento num ambiente de grade computacional, fornecendo assim uma boa qualidade de serviço ao usuário que deseja utilizá-lo.

1.1 Justificativa

Para que a grade computacional seja utilizada com mais freqüência comercialmente é necessário que o serviço fornecido seja de qualidade. Desta forma, a sua quantidade de usuários aumentará significativamente.

Segundo (FOSTER et al., 1999a) "a correta execução das redes, com ascensão no desempenho e orientadas a aplicativos, é uma prestação de serviço com elevada qualidade ponto a ponto". Esta qualidade de serviço (QoS) ponto a ponto pode ser conseguida de várias maneiras como: uma boa configuração, reservas e alocação dos recursos correspondentes. Por exemplo, a análise simultânea de dados interativos pode exigir o acesso a um sistema de armazenamento guardando uma cópia dos dados, um supercomputador para fazer a análise dos elementos da rede na hora da transferência dos dados e um dispositivo para visualizar esta interação. Alguns dos critérios para avaliação da qualidade de serviço são a viabilidade, disponibilidade do recurso, custos e desempenhos.

Algumas aplicações podem exigir mecanismos de reservas antecipadas que proporcionam uma expectativa sobre a disponibilidade dos recursos, verificando se os recursos podem ser reservados quando exigido, igual ao comportamento de uma companhia aérea que sabe se e possível obter um assento (FOSTER et al., 1999a).

Na grade os recursos podem ser de vários tipos como: computadores, memória, disco, redes, processadores e outros mais, sendo esses recursos distribuídos em domínios administrativos diferentes, assim como os mecanismos e a política de controle. Por isso é indispensável ter um sistema que faça o gerenciamento da alocação destes recursos. As reservas maleáveis são muito utilizadas para reservar a largura de banda da rede, por exemplo, quando um usuário necessitar do backup dos dados de um servidor. Esta transferência é feita pela rede entre dois nós da grade, por isso é importante reservar a largura de banda da rede entre estes dois nós. Para isso, basta o usuário enviar um pedido (com a quantidade de bytes que serão transmitidos e os nós de origem e destino) para o gerenciador agendar a reserva.

1.2 Objetivos

1.2.1 Objetivos Gerais

A proposta deste trabalho de conclusão de curso é contribuir para a melhoria da qualidade de serviço (QoS) na grade computacional, utilizando um sistema que gerencia a alocação de reservas maleáveis. Para isso serão utilizados alguns algoritmos que já existem, para comparar a eficiência e performance destes algoritmos, verificando como os algoritmos se comportam nos seguintes casos: utilização dos recursos e taxa de pedidos aceitos.

Outro objetivo é mostrar a utilidade que um gerenciador de reservas maleáveis dos recursos da grade pode trazer. O intuito do gerenciador é fornecer um serviço com qualidade para os usuários finais e conseguir grande utilização dos recursos da grade, verificando se este gerenciamento dos recursos atende às necessidades procuradas por muita aplicações e instituições.

1.2.2 Objetivos específicos

Para realizar os objetivos gerais definidos para o trabalho, os seguintes objetivos específicos são listados:

- Levantamento de requisitos necessários para implementação de um sistema que faz o gerenciamento das reservas maleáveis dos recursos na grade;
- Verificar o desempenho dos algoritmos na taxa de perda de reservas e a utilização do recurso, fazendo a comparação entre estes;
- Pesquisar algoritmos utilizados para a manipulação das reservas maleáveis de recursos e implementa-los para conseguir a comparação desejada;
- Estudar os tipos de reservas existentes, tal como reservas imediatas, reservas antecipadas e reservas antecipadas com relaxamento, verificando as vantagens e desvantagens apresentadas por cada uma;
- Verificação da eficiência do modelo proposto, apresentando as vantagens que foram encontradas e o que pode ser melhorado neste trabalho, para desta forma mostrar inovações e sugerir trabalhos futuros nesta área.

1.3 Metodologia

O capítulo 2 apresenta a fundamentação teórica que explica o funcionamento das reservas de recursos e os tipos de reservas , como as reservas imediatas, reservas antecipadas e reservas maleáveis. O capítulo 3 apresenta os trabalhos relacionados que mostra os trabalhos relevantes para a gerencia das reservas de recursos na grade e para a gerencia das reservas maleáveis na grade que é o tema específico deste trabalho. O capítulo 4 apresenta a proposta que explica arquitetura do gerenciador e os tipos de reservas suportadas pelo mesmo. Também é apresentado o suporte a outros tipos de reservas que não são suportadas por este gerenciador, mas que são relevantes para a gerência de reservas. O capítulo 5 apresenta Implementação e Resultados Obtidos que explica os algoritmos implementados neste trabalho para gerenciar as reservas maleáveis. Também são apresentados os resultados obtidos com estes algoritmos. O capítulo 6 apresenta a conclusão onde são feitas considerações sobre a importância do tema e os resultados obtidos. Também são sugeridos os trabalho futuros a serem desenvolvidos.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo descreve os conceitos sobre os tipos de reservas de recursos em grades computacionais.

2.1 Reservas De Recursos

Entende-se por "reserva de um recurso" o período que o recurso se torna exclusivamente dedicado a um usuário da grade ou a uma aplicação da mesma. Porém, antes do uso exclusivo de um recurso, existe a fase de negociação, onde o usuário e o provedor do recurso entram em um acordo sobre o recurso utilizado, como o tempo de uso e as políticas ou regras que devem ser respeitadas (KUNRATH; WESTPHALL; KOCH, 2008).

A reserva do recurso tem como objetivo deixar o usuário com acesso exclusivo ao mesmo. Isso é importante, pois o usuário utilizará toda a capacidade do recurso armazenado. O recurso reservado fica garantido para aquele usuário, independente do quão disputado esteja este recurso. Portanto, o acesso ao recurso não pode ser parado no meio da execução por outras requisições.

A grade computacional está se tornando a principal tecnologia para o compartilhamento de recursos distribuídos em aplicações de larga escala. O gerenciamento dos recursos da grade é muito difícil e complexo, pois os recursos da grade têm as seguintes características: estão distribuídos geograficamente; são heterogêneos; e estão sob a administração de organizações diferentes que possuem suas próprias políticas de acesso e controle sobre os recursos (MINGBIAO et al., 2007).

Praticamente existem dois tipos de reservas, as reservas imediatas e as reservas antecipadas. As reservas imediatas são feitas na hora que o recurso será utilizado. As reservas antecipadas são feitas com antecedência ao tempo de utilização do recurso. Duas ou mais reservas não ocorrem ao mesmo tempo, ou seja, um recurso não é reservado para usuários diferentes ao mesmo tempo (KUNRATH; WESTPHALL; KOCH, 2008). Também existem as reservas maleáveis

(RM), ou seja, reservas que não tem parâmetros fixos. Podem ser utilizadas para reservar a largura de banda da rede, pois estas reservas dão uma melhor performance à rede. A reserva maleável pode ser feita de forma imediata ou antecipada.

Reservas de recursos são eficientes, para quem compete por recursos distribuídos que estão localizados em diferentes domínios na grade. É um tema importante que tem investigações em aberto. Os recursos da grade não estão submetidos a um controle centralizado, mas são controlados pelos interesses de um domínio administrativo local e compartilhados por usuários concorrentes. Porém, os conflitos entre os usuários e os fornecedores são inevitáveis, entretanto, estes conflitos de interesses (de ambos os lados) precisam ser reconciliados. A reserva de recursos na grade computacional pode ser modelada como dinâmica e distribuída (GRIMSHAW; WULF, 1997). É, portanto, essencial organizar algum tipo de acordo entre o requerente e o prestador de serviços na utilização dos recursos (CZAJKOWSKI et al., 2006).

A tarefa de reservar um recurso envolve a seleção deste. Isto é, descobrindo um nó correspondente para cada atividade, ou seja, encontrando os recursos que podem estar em nós diferentes, mas que devem ser reservados no mesmo intervalo de tempo. Para conseguir um serviço de qualidade na execução da aplicação (SIDDIQUI; VILLAZÓN; FAHRINGER, 2006).

É conhecido que os componentes concorrentes precisam ser reservados, porém é essencial a reserva imediata que muitas vezes é negligenciada pelos projetistas. Nas Reservas imediatas, os recursos devem ser alocados com o menor tempo de atraso, depois da solicitação de reserva. Esta capacidade é fundamental para a flexibilidade da grade, pois os usuários necessitam ter acesso aos recursos em um momento conveniente. Porém os usuários são incapazes de prever quando vai começar e vai terminar um trabalho, portanto falhas podem ocorrer ao longo do percurso e por isso as reservas imediatas são importantes para recuperar as falhas. Definir o modelo a ser utilizado, é a primeira questão a ser resolvida, para que o projeto fique apropriado (GRIMSHAW; WULF, 1997).

2.1.1 Reservas Imediatas (RI)

As reservas imediatas só são armazenadas se o recurso estiver disponível no momento, pois o recurso pode estar desocupado ou ocupado na hora do pedido. Se o recurso estiver disponível será alocado imediatamente, caso contrário o usuário terá que aguardar até a liberação do mesmo. No pedido de uma reserva imediata não existe um horário explicitando o começo e o fim da operação, por isso elas necessitam de uma alocação do recurso o mais rápido possível.

Conforme (KUNRATH; WESTPHALL; KOCH, 2008) quando várias aplicações da grade

estão em execução em um mesmo processador, uma fila é utilizada para organizá-las. As regras da fila são bem simples, pois uma aplicação está em execução por vez e depois de um certo tempo, esta aplicação é interrompida e recolocada na fila, e dá lugar para a próxima da fila. Neste exemplo a fila é do tipo FIFO (first in, first out), ou seja, é uma fila que segue a ordem de chegada das aplicações.

Como as reservas imediatas normalmente possuem prioridade maior que as outras aplicações de grade, estas reservas imediatas são colocadas à frente de outras aplicações, porém atrás de outras reservas já feitas. É importante salientar que existe um revezamento de execução entre as aplicações, mas que este revezamento de execução não existe entre as reservas, pois a reserva só libera o recurso quando acabar sua execução e as outras reservas só poderão utilizá-lo quando aquela reserva liberar o recurso. O tempo de execução das reservas é determinado e finito, sendo negociado com o provedor do recurso.

Conforme (TACHIBANA; KASAHARA, 2006) "o ponto forte das reservas imediatas é a simplicidade na implementação", mas neste tipo de reserva pode ocorrer uma grande busca por um determinado recurso, causando um congestionamento para o seu uso e isto acaba acarretando na baixa qualidade de serviço. Por isso surgiu a proposta de utilizar reservas antecipadas, que possibilita reservar um recurso muito antes do início da sua utilização, melhorando assim a qualidade do serviço prestado.

2.1.2 Reservas Antecipadas (RA)

As reservas antecipadas são importantes para a grade computacional, quando uma aplicação ou um usuário for utilizar um determinado recurso. O recurso pode não estar disponível no momento desejado, desta forma a aplicação ou usuário ficará impedido de utilizá-lo, por isso é importante que a reserva de um recurso seja feita de forma antecipada. As reservas antecipadas garantem que uma aplicação ou um usuário terá acesso ao recurso no momento desejado. Dessa maneira a grade fornece um serviço com qualidade, pois quem usufruir o recurso fará no momento ambicionado, sem esperar para que o mesmo seja liberado.

Reservas antecipadas são especialmente úteis para computação em grade, mas também servem para uma variedade de aplicações que exijam uma rede com qualidade de serviço, como as redes de distribuição de conteúdo, ou cliente móvel que precisam de reservas antecipadas para apoiar o fluxo das transferências de vídeos (BURCHARD; HEISS; ROSE, 2003). as reservas antecipadas são essenciais para executar a maioria dos mecanismos de qualidade de serviços.

Antes de se discutir a respeito da sua performance, falaremos brevemente sobre o ambiente

das reservas antecipadas. (BURCHARD; HEISS; ROSE, 2003) afirma que as reservas antecipadas diferem das reservas imediatas pelo horário do pedido que é submetido à gestão da rede, assim dissociando totalmente a apresentação do pedido a partir da utilização dos recursos.

As reservas antecipadas devem permitir ao sistema da rede um controle de admissão viável, por isso estas reservas possuem variáveis para indicar o horário que o recurso começará a ser utilizado (Tstart), o tempo de duração da reserva sobre o recurso e também o horário que a reserva sobre o recurso terminará (Tstop). A requisição do pedido é enviada para o gerenciador em uma determinada hora (Tresv), porém este horário não deve ser maior que o Tstart. Se isto ocorrer o sistema deve avisar ao usuário, que não foi possível agendar a reserva.

Segundo (BURCHARD; HEISS; ROSE, 2003) isto é necessário para obter informações sobre estado de utilização do recurso no decorrer do tempo e assim realizar o controle viável de admissão, já que a idéia, de utilizar reservas antecipadas, é prever a disponibilidade dos recursos na Grade.

A reserva antecipada é dividida em duas fases: a fase intermediária que é o tempo entre a realização do pedido e o começo da utilização do recurso e a fase que o recurso é utilizado, sendo delimitada pelo Tstart e pelo Tstop.O que não pode ocorrer é o tstart, da reserva, começar antes que o tpedido (horário que o pedido foi enviado ao gerenciador). Na figura 2.1 observamos estas duas fases com os seguintes atributos:

- **Tpedido:** tempo que o pedido é feito para o gerenciador.
- **Tstart:** Tstart é o tempo que o recurso começa a ser utilizado.
- **Tstop:** é neste momento que termina a utilização do recurso.
- **Largura De banda**: Este atributo mostra a capacidade da rede utilizada. Respeitando sempre o limite máximo de utilização que existe em cada recurso, para que o pedido seja reservado.

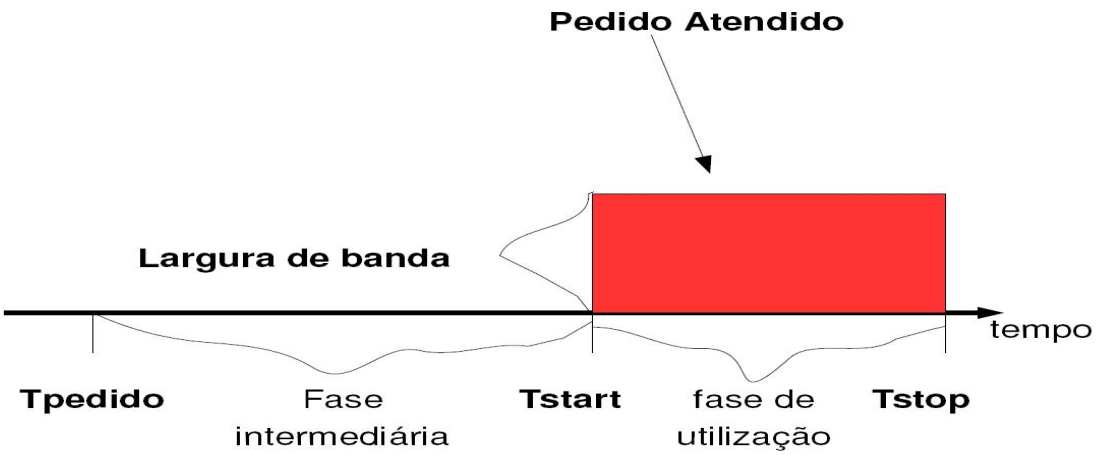

Figura 2.1: As duas fases e os atributos que existem na reserva antecipada.

Uma desvantagem das chamadas "reservas fixas" (que não são reservas maleáveis) é a possibilidade de existir intervalos de tempo, entre duas reservas, sem utilização do recurso. Isto implica em uma utilização mais baixa do recurso, sendo estabelecido picos de utilização do recurso. Esses picos são mostrados na figura 2.2 onde a utilização do recurso não acontece de forma homogênea, pois no decorrer do tempo acontecem os picos de utilização do recurso.

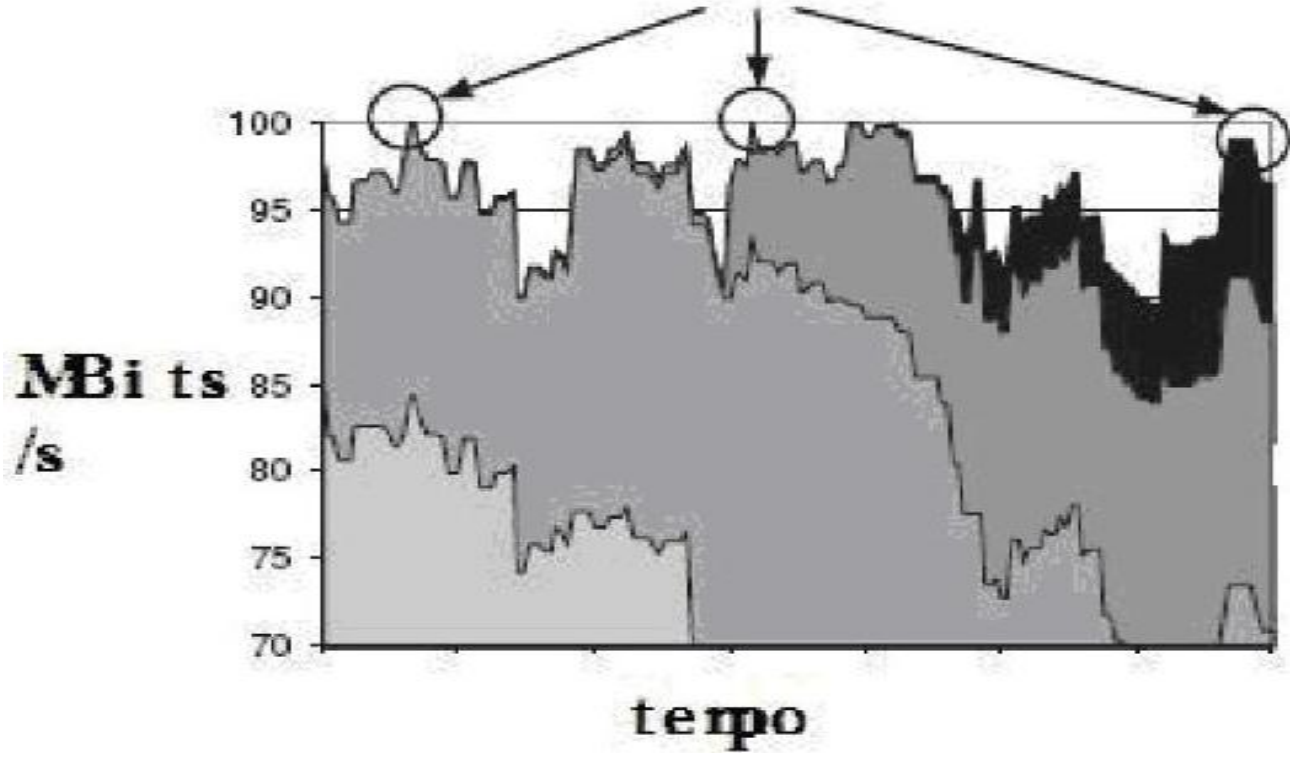

Figura 2.2: Picos ocorridos nas reservas fixas (BURCHARD; HEISS; ROSE, 2003)

existem momentos, que a utilização do recurso cai bastante em relação aos picos apresentado no gráfico, deixando a utilização do recurso desproporcional (transferências de bits) no decorrer do tempo. Isto acontece porque as reservas são fixas, ou seja, o tempo e a utilização do recurso são estipulados pelo usuário que deseja utilizar a rede para transmitir uma determinada quantidade de dados. Na figura 2.3 mostra um pedido sendo rejeitado, porque não é possível reservá-lo. Como a reserva não é maleável, este pedido é rejeitado, causando os picos que foram mostrado na figura 2.2, pois entre dois picos fica muito difícil armazenar uma reserva fixa. A figura 2.3 apresenta o motivo pelo qual o pedido é rejeitado.

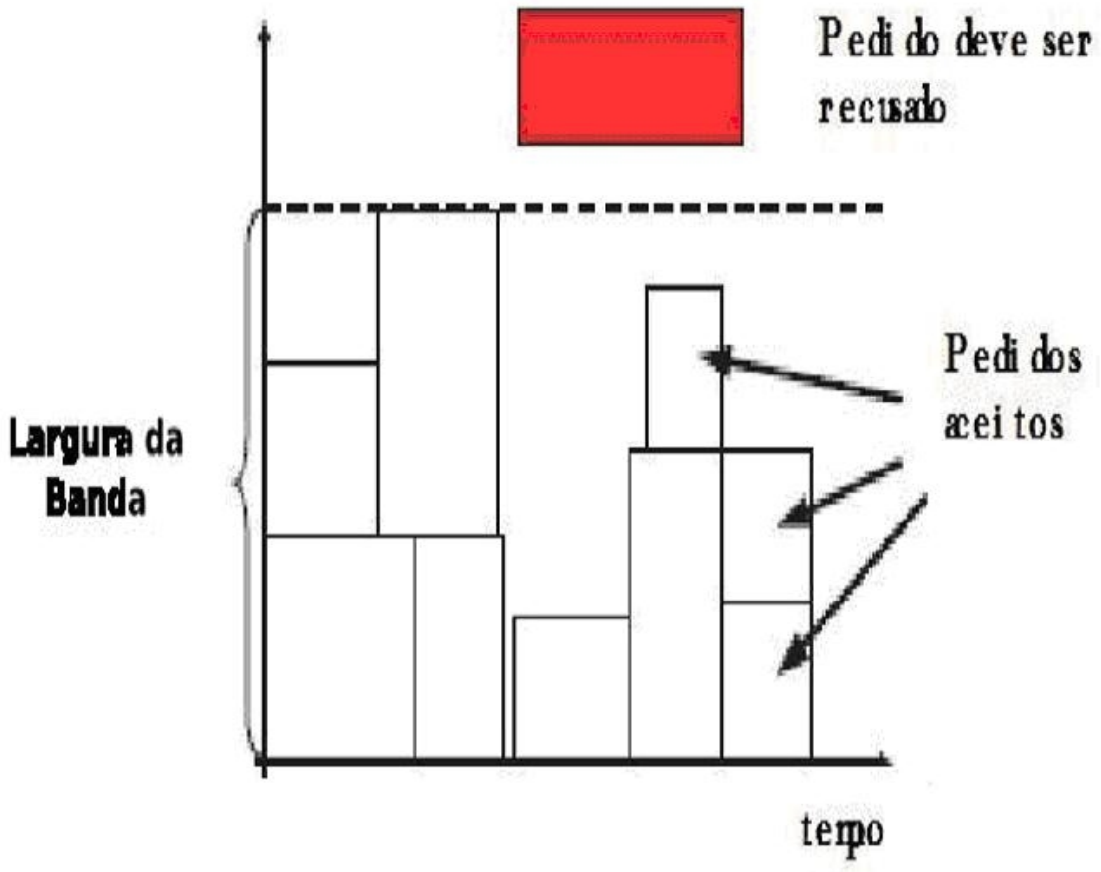

Figura 2.3: Pedido sendo recusado

A figura 2.3 mostra a importância da reserva maleável que será detalhado na seção 2.4.3, pois não foi possível reservar o pedido enviado porque a quantidade de recurso a ser utilizado e o tempo de reserva do recurso deste pedido não se encaixa com a disponibilidade da rede.

2.1.3 Reservas Maleáveis (RM)

Este tipo de reserva é usado para um grupo especial, onde nem a quantidade de recurso armazenada e nem a duração são valores absolutos (BARZ et al., 2005).Os parâmetros, tempo e duração, são escolhidos pelo algoritmo que faz a gerência das reservas dos recursos, por isso a reserva é chamada de maleável. As reservas maleáveis são utilizadas, por exemplo, em backups, onde há uma transmissão em larga escala de dados para que sejam armazenados em um servidor.

Alguns dos casos que a reserva maleável pode ser utilizada são: a transmissão de uma determinada quantidade de dados pela rede ou a utilização de um certo número de ciclos do processador, levando em conta que existem vários processadores na grade. Portanto, a reserva é maleável porque o tempo de utilização do recurso e a capacidade utilizada são adaptados pelo gerenciador que faz o agendamento da reserva. Segundo (BARZ et al., 2005) em redes de computadores o serviço de reserva pode transformar o pedido para uma reserva maleável. Isto é, o horário de inicio, o horário de fim e a largura de banda podem ser escolhidos de acordo com a disponibilidade dos recursos da rede.

Na figura 2.4 existem duas formas de reservar um pedido, para transmitir uma certa quantidade de dados. Nesta figura podemos observar como as reservas foram adaptadas, conforme a disponibilidade da largura de banda da rede e finalmente mostrando onde as reservas foram armazenadas.

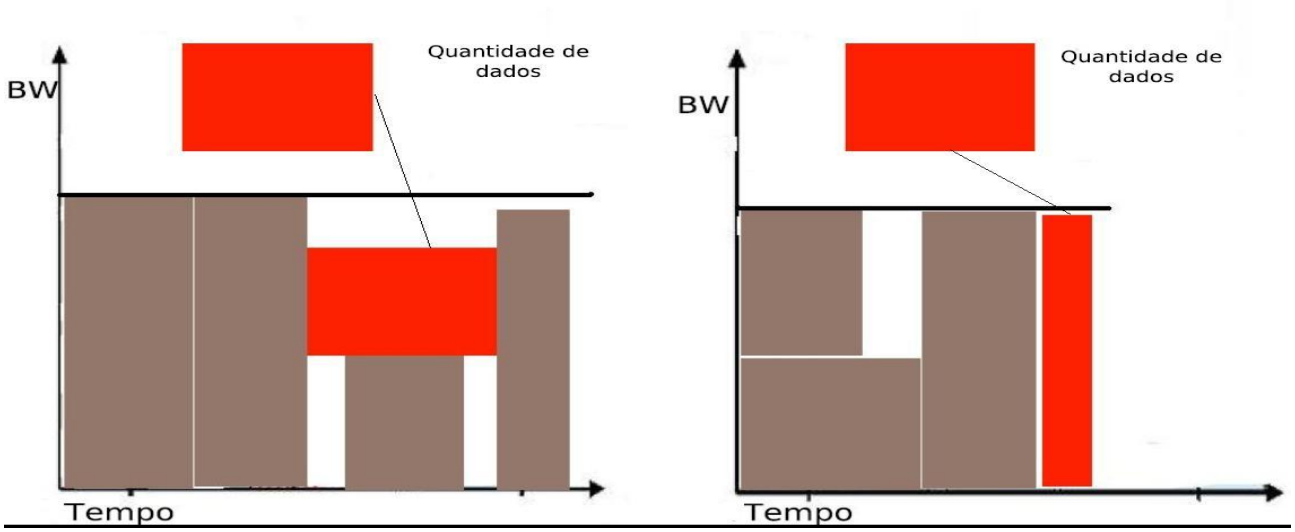

Figura 2.4: Como uma reserva maleável pode ser armazenada

Na figura 2.5 podemos observar que a duração e a quantidade de recurso reservado são decididas pelo gerenciador que faz o armazenamento das reservas, conforme a disponibilidade dos recursos ou a lógica do algoritmo utilizado. Neste caso existem duas ou mais alternativas

de reservas para um certo volume de transferência.

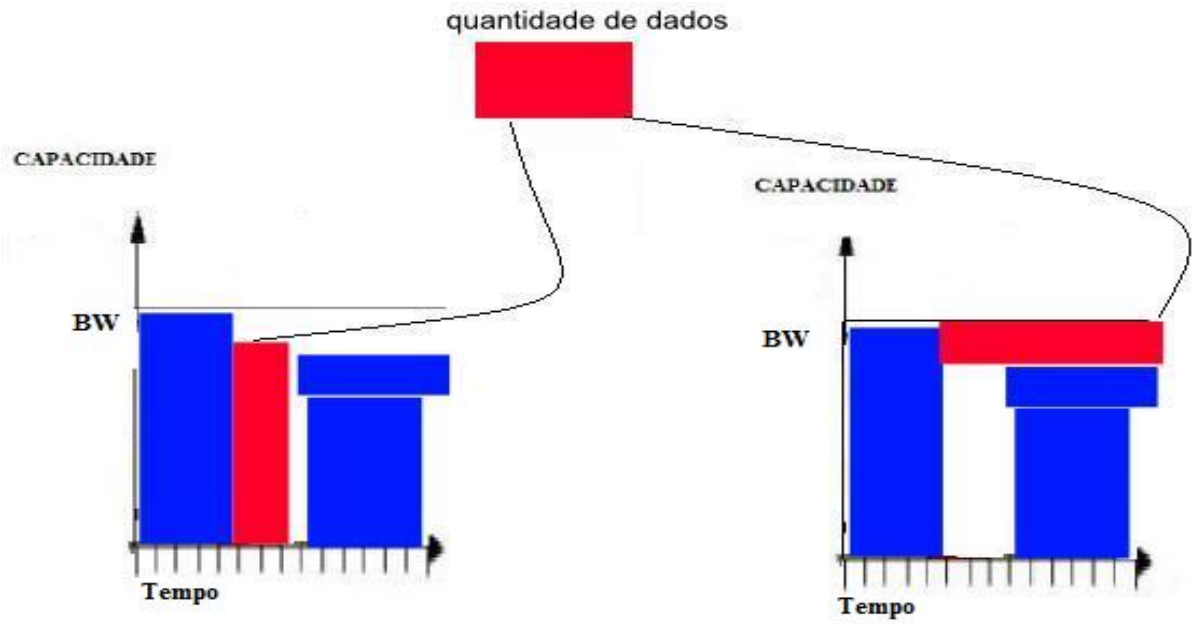

Figura 2.5: Duas opções para reservar um pedido

Outro ponto importante, mostrado na figura 2,5, é que a duração e a capacidade do recurso reservado são inversamente proporcionais, ou seja, se o tempo de utilização do recurso aumenta a capacidade do recurso reservado diminui e vice-versa. A reserva maleável segue a seguinte fórmula:
Z = X * Y. Z pode ser a quantidade de dados a ser transferido pela rede ou a quantidade de ciclos de um processador em uma grade com vários processadores, sendo X a duração e Y a capacidade do recurso a ser reservado. O gerenciador das reservas maleáveis deve levar em consideração a fórmula mostrada acima, para conseguir uma melhor utilização do recurso que será utilizado. As reservas, que são maleáveis, podem ser feitas de forma imediata ou de forma antecipada.

Levando em consideração a figura 2.3 onde não foi possível reservar o pedido porque a largura de banda e o tempo de utilização, estipulados pelo cliente, não estavam adequados para a disponibilidade da rede naquele momento. A figura 2.6 mostra aquele mesmo pedido sendo armazenado, pois a reserva é maleável. Portanto, em reservas maleáveis, quem deseja transferir dados pela rede deve indicar quantos bytes serão transmitidos, para o algoritmo reservar o pedido, através da combinação entre o tempo de utilização e a capacidade do recurso arma-

zenado. Na figura dizer 2.6 é apresentada uma alternativa para reservar o pedido, ajustando o tempo de duração e a largura de banda, conforme a disponibilidade da rede no momento, aumentando desta forma a taxa de aceitação dos pedidos. É importante lembrar que existem outras alternativas para reservar este mesmo pedido.

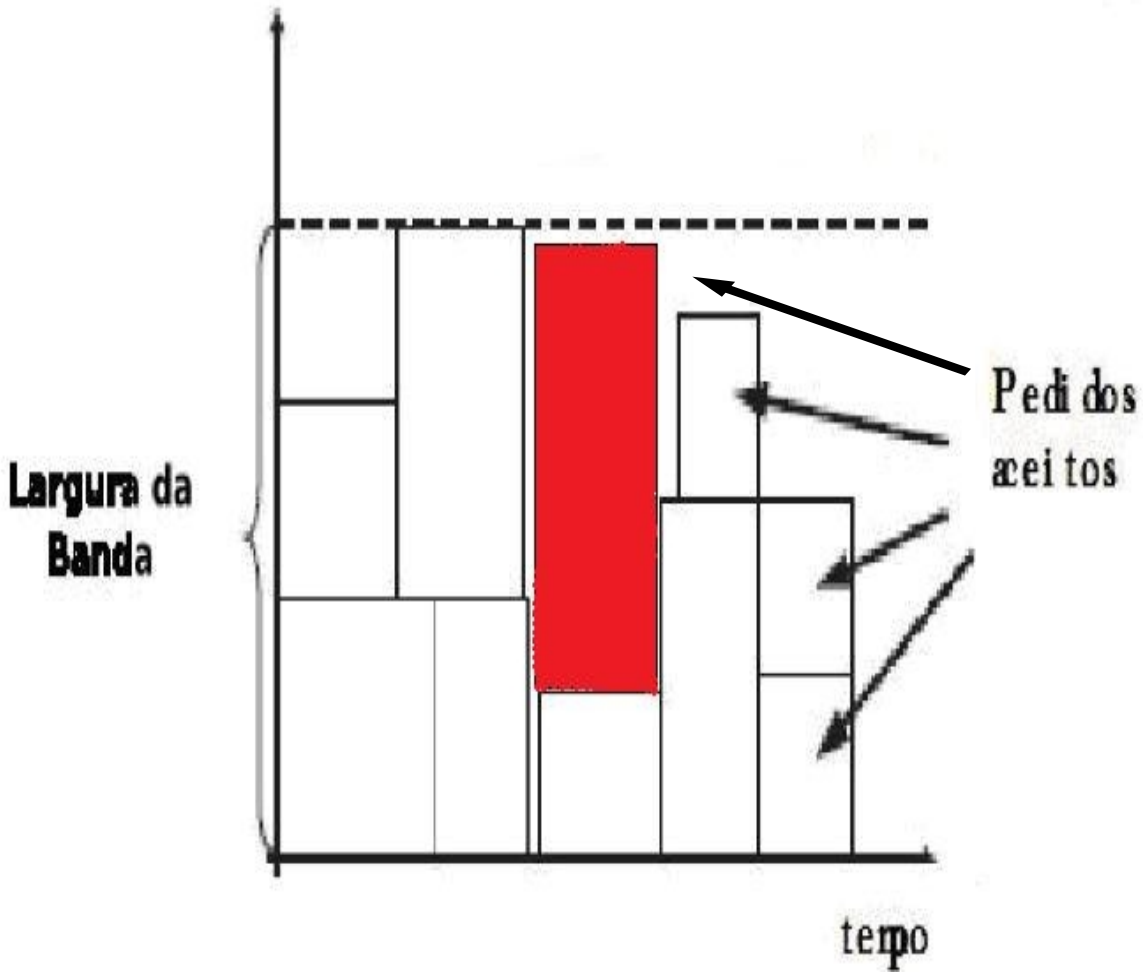

Figura 2.6: O pedido foi adaptado à disponibilidade da rede

A figura 2.6 mostra o pedido do usuário sendo atendido, pois ele, o usuário, enviou a quantidade de bytes que deseja transmitir e o gerenciador adaptou o pedido conforme a disponibilidade da rede. Por isso, o seu tempo de utilização foi encurtado e a largura de banda da reserva expandida. Conseguindo desta forma a transferência de todos os bytes desejados. Isto mostra a importância das reservas maleáveis para reservar a largura de banda da rede em grade computacional.

3 *Trabalhos Relacionados*

Neste capítulo, serão discutidos trabalhos relacionados ao tema de alocação de recursos em grades computacionais, visando a qualidade de serviço, com a utilização de AR e MR. Serão apresentados os pontos fortes e as carências de cada trabalho. Levando em conta as suas soluções aplicadas na grade computacional.

3.1 Gara

Este trabalho apresenta a sigla GARA que significa *Globus Architecture for Reservation and Allocation* (FOSTER et al., 1999a), ou seja, é uma arquitetura com a função de reserva e alocar recursos em computadores paralelo no Globus. O Globus é um middleware de grade muito utilizado. Sua arquitetura constitui de três componentes principais: um serviço local, gerenciamento local dos recursos, e vários tipos de agentes, para co-alocação, que implementam estratégias para descobrir e alocar os recursos pedidos, satisfazendo a qualidade de serviço requerida.

No GARA os recursos são apresentados como objetos genéricos. Neste caso, processadores, largura de banda, discos e todos os demais tipos de recursos são apresentados pelo mesmo tipo de objeto. Este tipo de representação permitiu que as interfaces fossem desenvolvidas de forma homogêneas para publicação e acesso aos recursos, porém tornou se complicado a utilização dos recursos, pois na hora do uso as diferenças entre os recursos vêm à tona. O GARA divide a criação do objeto genérico em duas fases: reserva e alocação. Na fase de reserva, é criada a reserva por uma operação genérica “Reservation Create”, que interage com o gerenciador local do recurso para garantir a quantidade e qualidade requerida no pedido e verificar também se o recurso estará disponível no horário desejado, caso isto não seja possível a operação “Reservation Create” falha. Na fase de alocação o objeto é criado e o recurso é alocado para o usuário ou aplicação que fez o pedido.

Para reservas, este modelo homogêneo se tornou bastante útil, porque facilitou que as re-

servas fossem feitas de forma mais simples. O GARA trata AR (Advance Reservation) e IR (Immediate Reservation). Porém, todas as reservas são feitas através de timeslot tables ou simplesmente slot tables, sendo trabalhados pelos gerenciadores chamados de timeslot managers. Os Slot tables são estruturas bidimensionais que levam em conta o período de tempo da reserva e a porcentagem do recurso que será utilizada. Eles são bons apenas para alguns tipos de recursos, onde se pode reservar parte da capacidade para um usuário, parte para outro e assim por diante. Para os recursos que não seguem esta idéia, como os processadores que são reservas como um todo e não apenas uma capacidade, o GARA não é uma boa solução (KUNRATH; WESTPHALL; KOCH, 2008).

O GARA não trata reservas maleáveis MR (Malleable Reservation) que são muito importante para reservar a largura de banda da rede. Isto torna a utilização da rede menor, pois as reservas maleáveis conseguem um melhor ajuste das reservas em determinadas situações da rede.

3.2 Gerenciadores

(SIDDIQUI; VILLAZÓN; FAHRINGER, 2006) apresenta um trabalho gerencia as reservas de recurso no grid, utilizando reservas antecipadas. O protocolo oferece a negociação entre os clientes e o sistema de reservas, tendo como objetivo alcançar uma grande utilização do recurso. Este protocolo é dividido em três camadas: alocação, co-alocação e coordenação.

A primeira camada, a de alocação, é dirigida pelos alocadores que negociam a alocação do recurso com nó individual do grid. O principal objetivo desta camada é otimizar a utilização do recurso, através das reservas de capacidade dos recursos. Esta camada é implementada através de um algoritmo de empacotamento em forma de fita (retângulos) adaptado para reservas, com largura finita (capacidade do recurso) e altura infinita (tempo). O algoritmo utilizado para fazer o empacotamento em fitas, adaptado para reservas, é o VSHSH, sendo esta forma utilizada uma boa solução.

A segunda camada, a de co-alocação, é moderada pelo co-alocador e esta camada leva em conta as preferências dos clientes para fazer a alocação do pedido, enquanto se preocupa em otimizar a utilização global do recurso. Portanto, a segunda camada melhora a qualidade das ofertas geradas num sentido amplo. Esta camada é construída sobre a anterior e gerencia uma ou mais aplicações. Como os recursos são compartilhados, a mesma opção de reserva pode ser oferecida a múltiplas aplicações. Para que isto não ocorra é necessário a coordenação entre os co-alocadores, sendo a terceira camada a responsável por esta função.

A terceira camada, a de coordenação, pode ser ativada pelos clientes ou pelo próprio sistema, quando o recurso aderir ou sair do grid. A terceira camada trabalha também com as "reservas em abertas", que são alocações flexíveis, e que podem ser mudadas nos nó antes das aplicações começarem a executar. Isto visa um maior aproveitamento na utilização dos recursos no momento. Esta camada busca resolver os problemas gerados pela competição de usuários ou de aplicações pelos mesmos recursos, gerando soluções não conflitantes, através de uma reorganização das reservas na agenda. Isto mostra a importância das reservas estarem em "abertas", pois estas reservas são flexíveis e podem ser reorganizadas.

Os únicos recursos gerenciados por este trabalho são os processadores, sendo possível a reserva de forma avançada e imediata.

3.3 Reservas para Transferências de dados

Neste trabalho é apresentado o uso das reservas maleáveis (BURCHARD; HEISS; ROSE, 2003),para adquirir uma melhor performance da rede. Nos testes apresentados, com as reservas fixas, a taxa de utilização da rede alternava entre os picos máximo. Portanto, com as reservas maleáveis, a utilização da rede se mostrou homogênea e com uma taxa de aceitação dos pedidos maior, para transmitir dados pela rede.

Em reservas maleáveis é desejado encontrar a melhor relação entre o tempo de utilização e a capacidade do recurso armazenado. Neste trabalho, foram apresentadas quatro estratégias para conseguir uma relação entre a capacidade do recurso armazenado e o tempo de utilização, baseado na quantidade de bytes que serão transmitidos. O pedido de uma reversa maleável foi definido, neste trabalho, com os seguintes atributos:

- **U**: nó de origem dos dados;
- **V**: nó de destino dos dados;
- **Tmin**: tempo mais cedo que a transmissão pode começar;
- **Tmax**: o tempo mais tarde que transmissão pode acabar;
- **Dmim**: duração mínima que a transmissão pode ter;
- **Dmax**: duração máxima pode ocorrer na transmissão;
- **C**: este atributo representa a quantidade de bytes que devem ser transmitidos

Este trabalho utiliza o *bandwidth broker* que implementa o controle de admissão dos pedidos. A tarefa básica, do bandwidth broker, é checar se existe recurso disponível, para satisfazer um pedido e comunicar-se com os componentes da rede.

O resultado, entre os testes obtidos com as reservas maleáveis e com as reservas fixas, mostrou que as reservas maleáveis conseguem um aproveitamento melhor da performance da rede e que a taxa de pedidos aceitos, feitos pelos usuários, é maior.

3.4 Alocação dos Recursos no Viola

(BARZ et al., 2006) Apresenta um trabalho que faz a alocação dos recursos no projeto VIOLA. A arquitetura do UNICORE é formada por três camadas. A primeira camada, Java-Client, é a interface do grid com o usuário, a segunda camada é quem fornece e gerencia o acesso dos trabalhos do usuário ao grid e finalmente a terceira camada é quem executa estes trabalhos. Uma trabalho(ou aplicação) consiste de vários sub-trabalhos.

Existe um MetaScheduling Service (MSS) que negocia os recursos com o local Resource Management Systems (RMS- Sistema que gerencia os recursos locais). Esta negociação ocorre em quatro fases principais:
1.Examina o local, através do RMS, para encontrar o slot livre no período desejado.
2.Determinando o slot livre.
3.Se existe um slot com tempo livre, o pedido deve ser reservado neste slot com o nome do utilizador. Caso contrário, a consulta deve ser reiniciada com um tempo depois do período pesquisado anteriormente.
4.Verificar se a reserva foi feita no horário correto do slot em todos os sistemas. Se isto ocorreu o pedido é reservado, senão reiniciar a consulta com um período de tempo depois do último pesquisado.

Caso nenhum tempo livre em algum slot seja identificado, através do RMS, um erro é enviado ao usuário. Este projeto trabalha com reservas antecipada e também com reservas maleáveis para alcançar um serviço com qualidade (QoS) e um maior desempenho da rede.

4 Proposta

A forma de fornecer qualidade de serviço em grade computacional é diferente do tradicional QoS em redes. A grade executa sobre das redes. A fim de satisfazer a QoS ponto a ponto, garantindo o alto desempenho das aplicações, a grade deve ter a implementação da qualidade de serviço além da rede. Mas por causa da heterogeneidade, complexidade e autonomia da propagação dinâmica dos recursos da grade, qualquer recurso pode aleatoriamente participar da Grade ou suspender os serviços nela. Todas estas questões tornam a implementação de QoS fim a fim na grade uma dificuldade (WANDAN et al., 2005).

Neste projeto a proposta é fazer um gerenciador de reservas maleáveis, que faz a alocação dos recursos da grade, para fornecer um serviço com qualidade para o usuário. A alocação do recurso pode ser através de reservas imediatas e reservas antecipadas. A opção de escolher um trabalho nesta área justifica-se por que:

- É um tema que não foi muito explorado ainda;
- Pela importância de fornecer para o usuário qualidade de serviço (QoS);
- Por que as reservas maleáveis dão uma boa utilização ao recurso;

Neste trabalho é proposto o GARG (Gerenciador Adaptativo Gerenciador dos Recursos da Grade) que tem como objetivo fornecer qualidade de serviço através da alocação de recursos. Como citado anteriormente, esta alocação pode ser imediata ou antecipada, isto depende da necessidade do usuário, mas a preferência é dada para as reservas antecipadas, pois é mais fácil prever o funcionamento da rede e a utilização dos recursos. A reserva imediata do recurso será efetuada se o recurso estiver disponível no momento que o pedido foi enviado.

4.1 Arquitetura do GARG

A arquitetura do GARG (Gerenciador Adaptativo dos Recursos da Grade) está definida na figura 4.1:

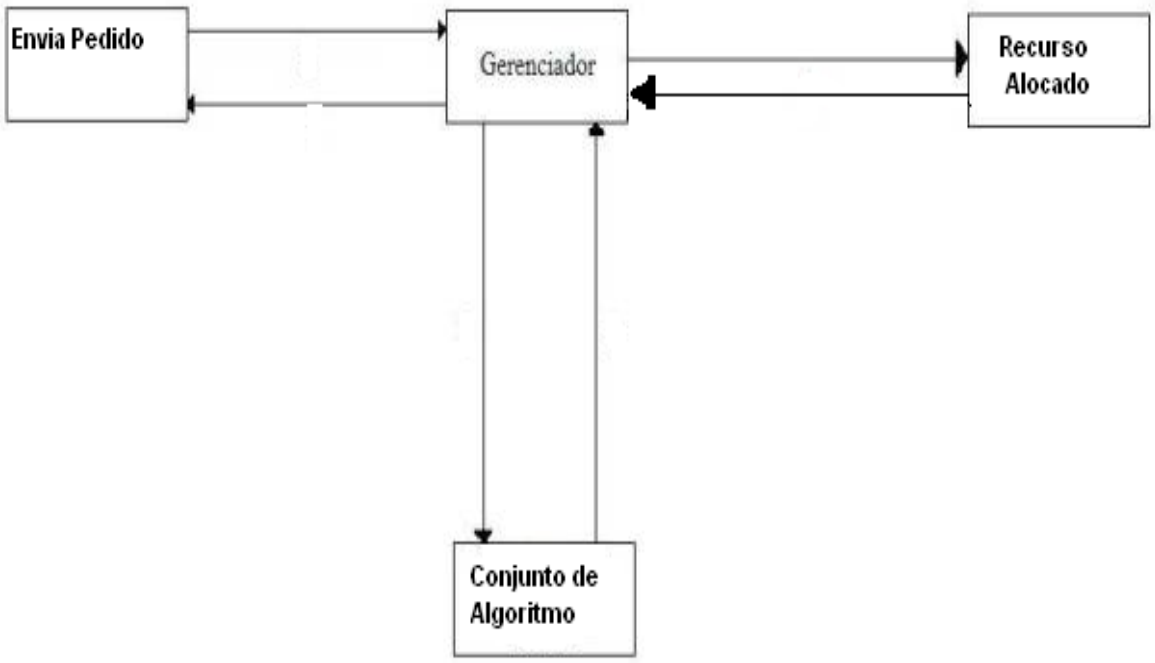

Figura 4.1: Arquitetura do GARG

Os passos descritos a seguir mostram como funciona o GARG:

- O usuário envia um pedido com a quantidade de dados, que ele deseja transmitir pela rede, por exemplo;
- O gerenciador utiliza um algoritmo adequado, para reservar o recurso;
- O gerenciador verifica como está a utilização do recurso, no horário solicitado pelo usuário.
- Uma mensagem é enviada ao usuário avisando se foi possível armazenar o recurso.

4.2 Agendamento das reservas em fita

Foi observado no trabalho (NTENE; VUUREN, 2007) o conceito de empacotamento em fita para ser utilizado neste projeto.

As reservas são agendadas ou empacotadas através de dois elementos geométricos: uma fita e alguns retângulos. A fita é um objeto bidimensional que possui comprimento e largura. A

largura é dada por um valor finito e conhecido, já o comprimento é um valor muito grande com fim desconhecido, ou seja, temos um início definido na fita e um final desconhecido.

O retângulo também é um objeto bidimensional que possui largura e comprimento, sendo estes finitos e conhecidos, porém cada retângulo possui a sua própria dimensão, ou seja, varia de um retângulo para outro. Os retângulos podem ter diversos formatos como: a forma de um quadrado, ter a largura maior que o comprimento e o comprimento maior que a largura.

O problema, entre a fita e os retângulos, deve seguir os seguintes itens:

- Os retângulos devem ter a largura menor ou igual a da fita, por isso todos os retângulos devem estar contidos na fita.
- Nenhum retângulo pode ocupar a mesma área pertencente a um outro retângulo, ou seja, não pode existir interseção entre as áreas dos de retângulos distintos.

Na figura 4.2 existem três retângulos empacotados em uma fita. Para armazenar outro retângulo na fita, este deve ter largura menor ou igual a largura da fita. Caso contrário o empacotamento não é possível.

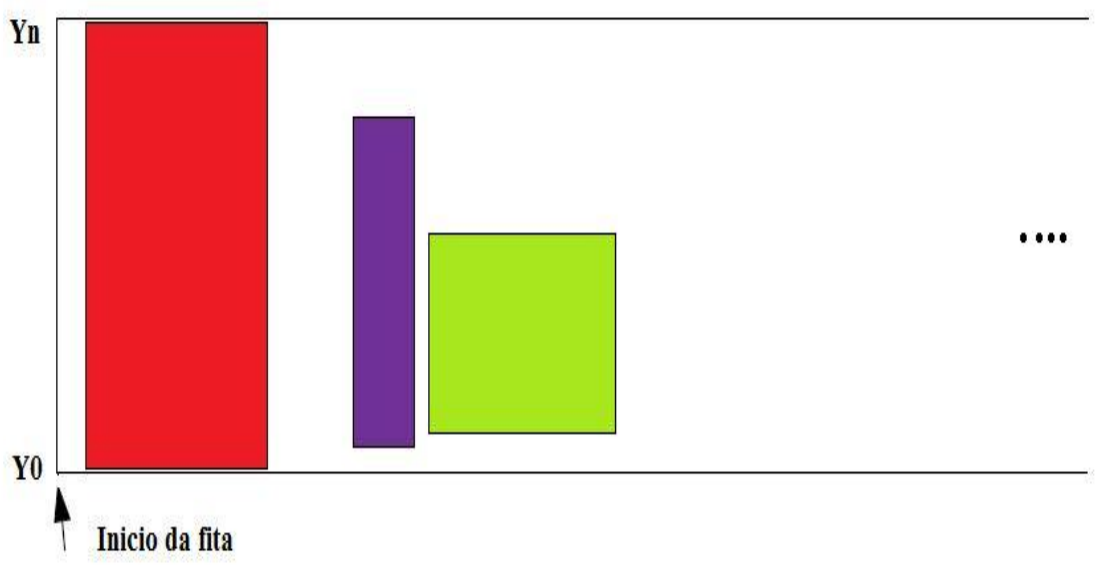

Figura 4.2: Agendamento em fita

Existem vários objetivos para realizar-se o empacotamento em fita, por exemplo, uma boa utilização do recurso, uma pequena taxa de rejeição dos pedidos de armazenamento, velocidade e agendamento das reservas e outros mais. Existem vários algoritmos, para agendar os retângulos na fita, buscando a maior utilização do recurso.

Pela grande complexidade, vários algoritmos com diferentes heurísticas foram criados para gerar boas soluções de utilização da área da fita, porém com baixa complexidade algorítmica (NTENE; VUUREN, 2007). Algumas heurísticas organizam os retângulos por ordem de comprimento, outras por ordem de largura e outras armazenando próximos os retângulos similares. Um tipo de heurística interessante é dado pelo estabelecimento de cortes virtuais na fita, desta forma, separando a fita em partes (também chamadas de gavetas) e tentando organizar os retângulos de diferentes maneiras em cada gaveta. A forma em que os cortes são feitos, bem como as formas de organização podem ser bem variadas. Também é possível estabelecer cortes verticais e também cortes horizontais, ou até ambos na mesma heurística. Portanto com estas técnicas várias soluções podem ser estabelecidas para o problema (boa utilização da fita). Como citado anteriormente, não existe uma melhor solução estabelecida, porque diferentes soluções podem ser consideradas como melhores dependendo de onde o empacotamento da fita é usado (KUNRATH; WESTPHALL; KOCH, 2008).

Existem também duas formas distintas de armazenamento dos retângulos na fita, que são o armazenamento on-line e o armazenamento off-line. Para cada forma de armazenamento, existem soluções distintas de armazenamento.

- *Agendamento on-line* – O armazenamento de retângulos na fita são on-line porque uma vez que a sua posição seja definida na fita, esta definição não pode ser refeita, ou seja, a posição do retângulo não pode ser trocada posteriormente. Portanto, a fita não pode ser reorganizada para buscar uma melhor solução. Os retângulos são armazenados conforme o tempo que vão chegando na fita, sendo conhecido somente os espaços vagos existentes na fita. Assim, não se tem conhecimento das dimensões dos outros retângulos que virão a ser armazenados. Por isso, a fita não possui a maior utilização possível no armazenamento on-line, pois o retângulo é armazenado na fita sem sofrer alteração em sua posição posteriormente. As soluções existentes para este tipo de armazenamento são menos complexas, pois os algoritmos devem achar uma solução para um retângulo de cada vez e não para todos simultaneamente. No armazenamento on-line não existe uma solução ideal, pois é um problema np-completo.

- *Agendamento off-line* – Este armazenamento dos retângulos é como se fosse o oposto do anterior. Os retângulos podem ser reorganizados na fita o quanto se desejar, ou seja, o retângulo pode ser re-alocado em outra posição mais tarde, conforme os outros retângulos vão sendo conhecidos. Existe a situação em que todos os retângulos são conhecidos antes de começar o empacotamento. Desta forma, este problema gera soluções mais complexas porque são considerados todos os retângulos simultaneamente, porém é possível obter

uma melhor utilização da área na fita, pois existem mais possibilidades. Inclusive, a solução ideal da fita, que é muito complexa, só é possível para o armazenamento off-line.

Neste projeto será utilizado somente o armazenamento on-line, pois os pedidos são enviados um de cada vez, ou seja, somente uma reserva é armazenada por instante.

4.3 Gerência de Reservas

A etapa de negociação é muito importante para o gerenciador de reservas ter um bom desempenho. Já que neste momento o gerenciador dá a resposta, confirmando ou não o agendamento do pedido, ao usuário. O gerenciador deve ser capaz de suportar reserva de vários tipos de recursos, assim como suportar reservas imediatas, reservas antecipadas, reservas antecipadas com relaxamento e reservas maleáveis.

O gerenciador de recursos da grade deve levar em conta os mais diversos tipos de reservas que existem, como reserva de um único recurso, reserva de vários recursos idênticos, e reservas de capacidade de um recurso. Como o objetivo deste trabalho é usar reservas maleáveis, serão usadas reservas maleáveis de recursos idênticos e reserva da capacidade de recursos. Como no suporte de reserva para único recurso não é interessante usar reservas maleáveis, só foi detalhado brevemente como seria o suporte AR.

Existe também a co-alocação de recursos, ou seja, o uso simultâneo de vários recursos diferentes que podem ou não estar no mesmo domínio. Isto não é um suporte prestado diretamente pelo gerenciador, mas sim um serviço prestado pela grade. Para que este serviço seja prestado é necessário que várias reservas antecipadas sejam feitas dos determinados recursos. Todos os recursos devem ser reservados num mesmo intervalo de tempo, para que o pedido possa ser efetuado. Não será tratado neste trabalho a co-alocação de recursos, apenas será explicado brevemente como funcionaria um gerenciador com este suporte, utilizando reservas antecipadas.

4.3.1 Gerência de Reservas Antecipadas Para Um único Recurso

Este secção apresenta um detalhamento sobre a reserva antecipada em um único recurso. Já que as reservas maleáveis não são trabalhadas, pois neste caso não temos a possibilidade de reservar por capacidade, ou seja, o recurso só pode estar totalmente disponível ou inteiramente reservado. Segundo (KUNRATH; WESTPHALL; KOCH, 2008) este tipo de AR é o mais simples que existe, pois no período que um usuário reservou o recurso, somente ele pode utilizá-lo, impossibilitando o acesso de outros usuários e outras aplicações a este recurso

Conforme (KUNRATH; WESTPHALL; KOCH, 2008), um recurso com suporte a este tipo de AR pode se encontrar, em um determinado instante de tempo, reservado ou disponível. É impossível que este recurso esteja parcialmente utilizado ou parcialmente disponível, por isso é inviável o uso de reservas maleáveis.

Os algoritmos utilizados para gerenciar este tipo de suporte a AR são unidimensionais, cuja dimensão é o tempo. Portanto a reserva de um único recurso é determinada com o seguinte modelo: um tempo de início para utilização do recurso e um tempo para o fim da utilização do recurso. Duração é horário de início menos o horário final da utilização, portanto se existe uma reserva de um recurso no período de 12:00 horas a 15:00 horas, a duração desta reserva é de três horas e durante este período o recurso fica para uso restrito ao usuário, sem competição e sem interrupção entre as 12:00 horas e as 15:00 horas. Os algoritmos utilizados para fazer o gerenciamento destas reservas são mais simples, pois basta verificar se o recurso está disponível ou não no período desejado pelo usuário.

A figura 4.3mostra um exemplo de gerenciamento das reservas de um único recurso, levando em conta que este recurso pode ser um processador, disco rígido, memória e qualquer outro recurso. Existe um novo pedido para reservar um recurso com duração de 3 horas, com EST (tempo mais cedo possível para começar a operação) de 5 horas e com LET (tempo limite para a operação acabar) de 15 horas. Existem três possibilidades, a alternativa X, a alternativa Y e a alternativa Z, para que esta reserva seja agendada. Portanto, o algoritmo utilizado visa agendar a reserva num lugar onde o espaço entre as reservas fique o menor possível. Por isso, neste caso a alternativa Y é a escolhida para agendar o pedido de reserva.

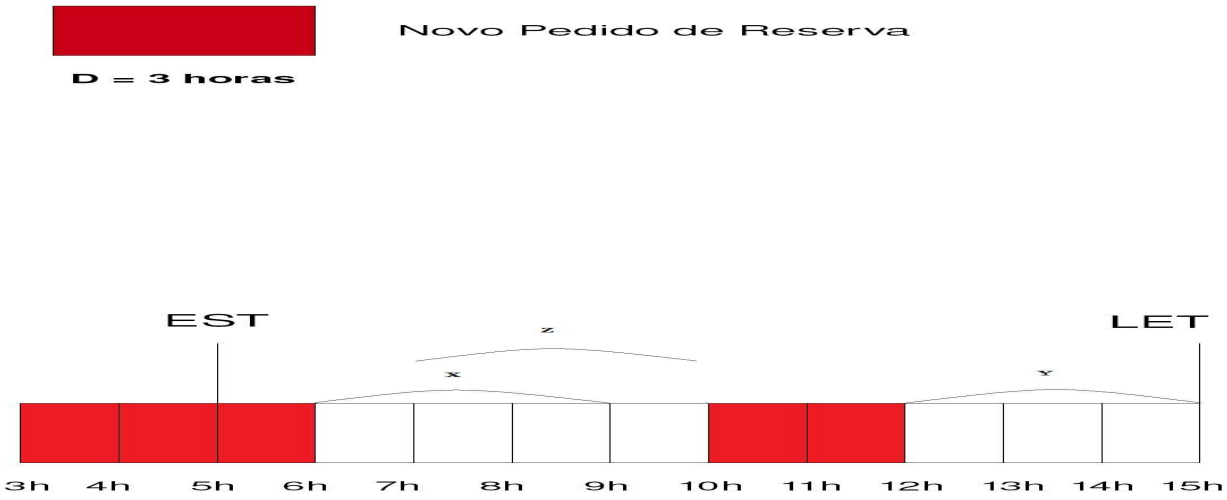

Figura 4.3: Exemplo de uma AR de único recurso

Os períodos vagos, que existem entre as reservas, podem ser armazenados em uma lista ou em uma árvore binária. Porém a forma mais adotada é armazenar os períodos vagos em árvores por ser mais eficiente. Foi dada uma breve explicação de como usar AR para um único recurso, sem entrar em detalhes dos algoritmos.

4.3.2 Co-Alocação de recursos

Aqui será tratado o conceito de co-alocação dos recursos, mostrando alguns exemplos onde isto é necessário.

Co-alocação de recurso é o uso sincronizado de diferentes recursos que podem pertencer a domínios administrativos distintos, tendo cada domínio as suas políticas, de acesso aos recursos, que devem ser respeitadas. A co-alocação dos recursos tem a intenção de possibilitar que as aplicações possam desfrutar inteiramente da grade para realizarem suas operações.

Uma aplicação que processa grande quantidade de imagens coletadas por câmeras digitais, executa operações complexas de reconhecimento de padrões e armazena as imagens para uma possível utilização, é um exemplo de uma aplicação na grade, onde a co-alocação de recursos seria utilizado. Para que esta aplicação tenha uma boa execução são necessários: utilizar os recursos que geram as imagens (câmeras), recursos que processam funções (processadores) e recursos que armazenam as imagens (discos rígidos). Se a massa de dados das imagens for muito grande, deve também levar em conta a transmissão dos mesmos entre os nós da grade (recurso da rede), na transmissão, entre os nós da grade. Por isso, é interessante utilizar reservas maleáveis para conseguir um melhor desempenho da rede (KUNRATH; WESTPHALL; KOCH, 2008).

A utilização destes recursos de forma sincronizada é difícil de obter. Segundo (FOSTER et al., 1999b), a forma mais fácil e viável para conseguir a co-alocação dos recursos é através de AR. Porém, não se usa apenas uma reserva porque os recursos são diferentes e pertencem a domínio administrativo distinto. Portanto, são feitas várias reservas antecipadas, uma de cada recurso diferente, para conseguir, através de um algoritmo, a utilização dos recursos no período desejado de forma sincronizada. Foi resumido como funcionaria a co-alocação de recursos na grade, porém neste trabalho não será tratado a co-alocação de recursos.

4.3.3 Gerência das Reservas Maleáveis para Recursos Idênticos

É importante que o gerenciador de reservar de recursos suporte reservas maleáveis de recursos idênticos. Este tipo de reserva é diferente da co-alocação, pois os recursos, além de

serem idênticos, pertencem a um único domínio. Na co-alocação, os recursos podem estar em domínios diferentes e serem distintos. É interessante que o middleware consiga, através de um gerenciador, agendar os recursos idênticos em conjuntos.

O gerenciador não trabalha de forma unidimensional, como no caso de um único recurso, pois agora existem duas dimensões, o tempo de utilização e a quantidade de recursos que serão utilizados para fazer as reservas.

Por exemplo, um nó da grade funcionando com um servidor para fazer armazenamento dos dados e para alcançá-lo existem quatro portas de redes distintas. É possível reservar uma porta de rede, duas portas, três portas ou quatro portas, por um certo período de tempo. Existem duas dimensões, o tempo e a largura de rede que leva os dados até o servidor. Neste exemplo foi utilizado como recurso, a largura de banda. A figura 4.4 apresenta um exemplo de reservas maleáveis para recursos idênticos, onde existem quatro portas de rede em um servidor de armazenamento. A figura 4.4 mostra reservas de várias portas de redes no mesmo instante.

Uma característica importante, da reserva maleável de vários recursos, é a possibilidade de existir várias reservas simultaneamente. No exemplo apresentado há quatro placas de rede que podem ser reservadas por usuários distintos.

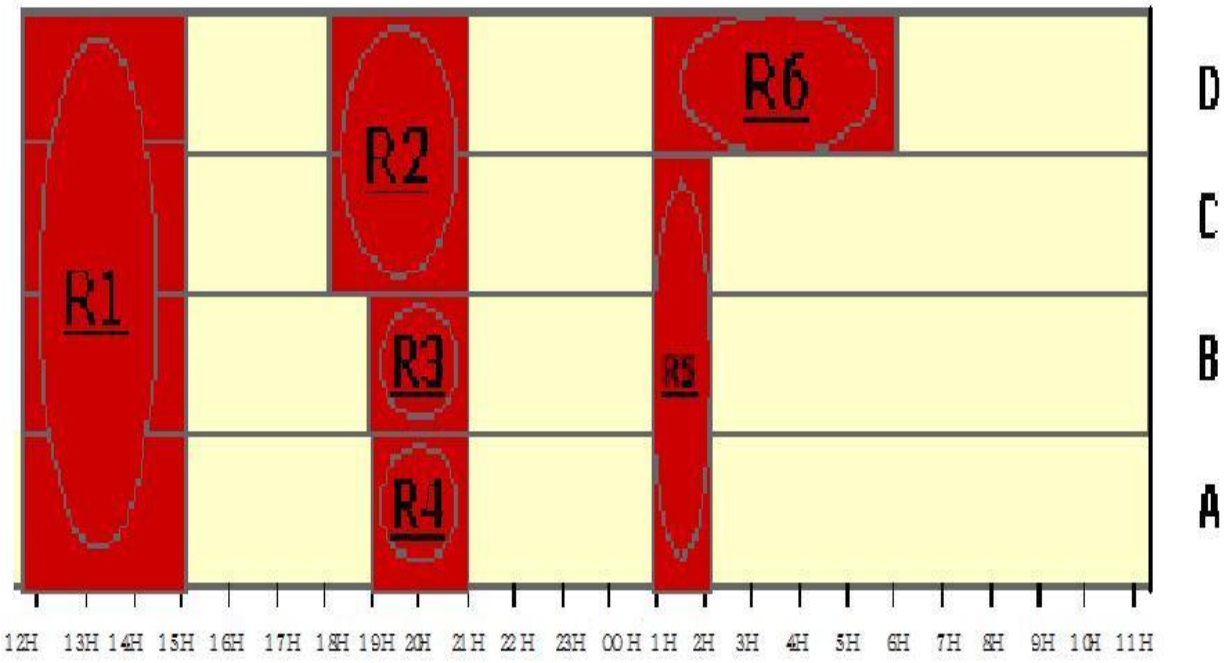

Figura 4.4: Reserva de Recursos Idênticos:

Isso fica evidente na figura 4.4 onde as reservas R2, R3 e R4 estão alocadas em um mesmo

período (19:00h as 21:00h) de tempo, porém cada reserva utiliza uma das quatro portas da rede, ou seja, não existe compartilhamento de recurso (porta de redes) entre estas reservas. Vale ressaltar que os recursos não podem estar parcialmente reservados, ou seja, os recursos estão disponíveis ou reservados. Neste exemplo foi utilizada reserva maleável sobre os recursos idênticos. No pedido deve ter um atributo sobre a quantidade, em bytes, de dados que serão transmitidos. É maleável a escolha de reservar uma ou mais porta de redes, conforme a disponibilidade destas portas no período que o usuário deseja transmitir os dados. O algoritmo gerencia as reservas maleáveis de recursos idênticos da seguinte maneira:

- Considerando o exemplo das placas de redes, onde um pedido leva como atributo a quantidade de bytes que serão transmitidos, o gerenciador verifica a disponibilidade dos recursos no período desejado pelo usuário.
- O algoritmo verifica, através da disponibilidade dos recursos idênticos, qual seria a melhor solução: reservar um, dois, três ou todos os recursos, para ter uma boa adaptação a este pedido.
- A soA solução mais apropriada é aquela que deixa o menor espaço de tempo, entre uma reserva e a outra. Caso não exista recurso reservado ainda, o algoritmo busca reservar o maior número de recursos possível. Se não for possível reservar todos os recursos, o algoritmo vai diminuindo o número de recursos um a um até que a reserva seja possível, porém se o número de recursos reservados diminuir, o tempo de duração da reserva vai aumentar.
- Caso não consiga reservar o pedido, uma resposta negativa é enviada ao usuário.

Segundo (KUNRATH; WESTPHALL; KOCH, 2008), a forma mais comum de representar reservas de vários recursos idênticos é através de algoritmos para os problemas de empacotamento em fita. Estes algoritmos podem ser modelados de forma a representar os recursos e reservas, e assim, provêem uma boa solução para o suporte a este tipo de reservas antecipadas. Na próxima secção será apresentado o conceito de empacotamento em fita e a sua forma de gerenciamento.

4.3.4 Gerência das Reservas Maleáveis de Capacidade de Recurso

Neste tipo de reserva, os recursos podem ser reservados parcialmente para um usuário, ou seja, um recurso pode estar reservado para vários usuários em um mesmo instante. Por exemplo,

um recurso pode estar com oitenta por cento reservado para um usuário e o restante para outro usuário. Portanto, o recurso contém três estados distintos: estar inteiramente reservado, estar totalmente disponível ou estar parcialmente disponível. Neste trabalho foi utilizado como recurso a largura de banda da rede, pois o foco deste projeto é trabalhar com as reservas maleáveis. Existem algoritmos, que permitem ao dono do recurso disponibilizar apenas uma parcela de um determinado recurso como, por exemplo, oitenta por cento da capacidade total do recurso.

Como o objetivo deste trabalho é gerenciar as reservas maleáveis, não nos detemos no armazenamento das reservas. Neste trabalho serão montadas as reservas, através dos algoritmos implementados que servem para definir o tempo e a capacidade de recurso armazenado. Para fazer o armazenamento das reservas foi utilizado um algoritmo desenvolvido pela (KUNRATH; WESTPHALL; KOCH, 2008) em forma de variable slots que será detalhado na próxima secção. Foram implementados alguns algoritmos para encontrar o tempo e a capacidade do recurso utilizado que melhor se adaptem a disponibilidade da rede. Conforme, o horário que o usuário deseja utilizar este recurso, a reserva é criada com tempo e capacidade estipulados pelo gerenciador de reservas maleáveis.

Na figura 4.5 vemos como ficam as reservas por capacidade de um recurso (largura de banda da rede). Existe mais de uma reserva feita ao mesmo instante. Isso mostra que é possível reservar apenas uma parcela dos recursos para um usuário.

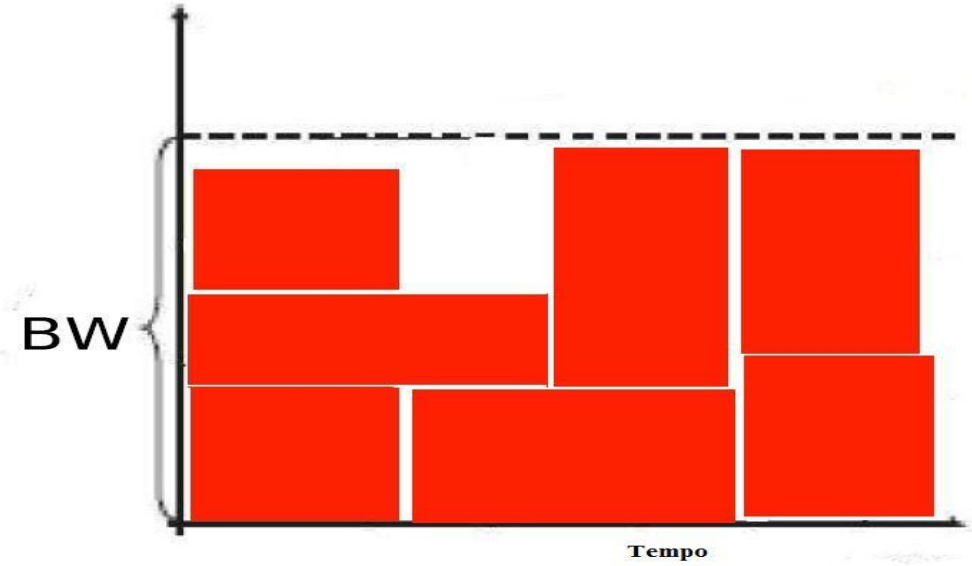

Figura 4.5: Reserva de Capacidade do Recurso

Os algoritmos que fazem o gerenciamento das reservas de capacidade de recursos maleáveis são mais complexos que as outras formas de reservas. Foram implementados alguns algoritmos

para gerenciar as reservas maleáveis, estes algoritmos serão detalhados no capitulo no capitulo 5. A seguir será detalhado o algoritmo utilizado para fazer o empacotamento das reservas. Este algoritmo foi desenvolvido pela Universidade Federal de Santa Catarina (UFSC) (KUNRATH; WESTPHALL; KOCH, 2008) e é chamado de Variable Slots.

4.3.5 Armazenamento em Variable Slots

É um método de armazenamento das reservas que são retângulos na fita. Este método de empacotamento foi desenvolvido, pelo Laboratório de redes e Gerência (LRG) da Universidade Federal de Santa Catarina (UFSC), com o objetivo de conseguir uma boa utilização na área da fita, para as reservas de capacidades de recursos e este algoritmo será utilizado neste trabalho que utiliza reservas maleáveis, pois após encontrar a dimensão da reserva é necessário armazená-la.

VS são formados por reservas inteiras, reservas parciais ou por parte de reservas parciais. As reservas são juntadas em um mesmo VS quando estão num mesmo intervalo de tempo, ou seja, VS são estruturas determinadas por intervalos de tempo e compostas por reservas ou pedaços de reservas. Na figura 4.6 está apresentada uma fita com duas reservas R1 e R2. Na etapa 2 R2 fica dividida em duas partes porque isso não altera a reserva, pois a quantidade de recurso armazenado é igual na etapa 1 nos mesmos intervalos de tempo. Na etapa três são utilizados VS. Na figura 4.6 há três VS: o primeiro é o intervalo no qual somente R1utliza o recurso; o segundo é o intervalo onde ocorre a intersecção entre R1 e R2; e o terceiro é o intervalo no qual só existe R2.

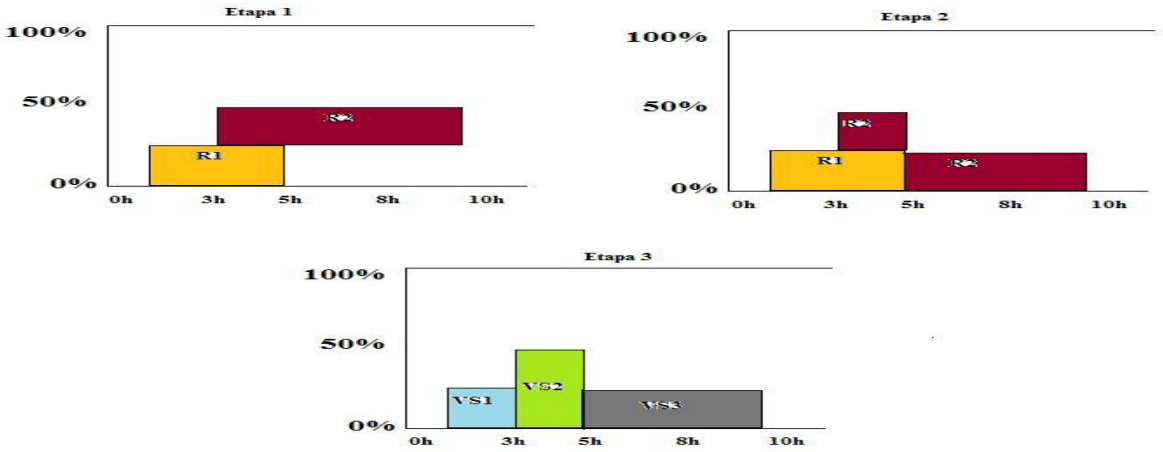

Figura 4.6: Armazenamento em Variable Slots

VS são úteis quando se deseja representar a capacidade de recursos utilizado num determi-

nado intervalo de tempo. Através do VS é mais fácil verificar quais são os períodos que tem capacidade de recursos suficiente para uma reserva.

Uma característica do VS é a seguinte: um VS pode possuir várias reservas ou partes de reservas. Uma reserva pode ser representada por diversos VS, mas uma reserva não pode possuir apenas uma parte de um VS, ou seja, no intervalo de tempo de um VS terá sempre as mesmas reservas ou pedaços de reservas. Isto pode ser visto na figura 4.6 onde o VS3 pega uma parte da reserva. Fica visível no VS2, que um único VS é formado por uma parte da reserva R1 e uma parte da reserva R2. Mostrando que existe a possibilidade de um VS ser formado por várias partes de reservas distintas.

VS são úteis para este trabalho, pois serão procurados os espaços vazios na fita para armazenar as reservas maleáveis. E com VS é mais fácil de encontrar a capacidade do recurso disponível num determinado intervalo de tempo.

5 *Implementação e Resultados Obtidos*

Aqui serão apresentados os resultados obtidos e também serão explicados os algoritmos utilizados para fazer o gerenciamento das reservas maleáveis neste trabalho. Para os algoritmos que tiveram resultados mais significativos serão apresentados os resultados obtidos.

5.1 Implementação

Neste capitulo serão explicados os algoritmos implementados neste trabalho. No gerenciamento das reservas parciais e das reservas de recursos idênticos. Nestes casos as reservas maleáveis são úteis e por isso foram utilizadas. Todos o algoritmos deste trabalho obedecem aos seguintes passos:
1.Todos os algoritmos, apresentados neste trabalho, buscam obter o máximo possível do recurso disponível em um determinado nível. Utilizando a seguinte fórmula: Z = X * Y. Onde o Z é a quantidade de dados a serem transmitido, o X é o tempo de utilização do recurso e o Y é a capacidade de recurso armazenado pela reserva. Lembrando que buscamos sempre X e Y como números inteiros, sendo o X e o Y econtrados pelos algoritmos utilizados.
2. Caso o algoritmo não pegue a capacidade máxima de recurso de um determinado nível, o valor de Y vai sendo reduzido e o de X aumentado. Até que a fórmula acima seja satisfeita.
3. O pedido só pode ser armazenado se a sua operação não ultrapassar o Tstop (tempo limite para realizar a operação).

Vale lembrar que as reservas são transformadas em VS (Variable Slots) e cada VS é um nível da fita.

5.1.1 Armazena no próximo nível (APN)

No algoritmo APN (armazena no próximo nível), os retângulos, que são reservas, vão sendo armazenados na fita conforme a hora de chegada, o recurso é armazenado no último nível da fita se obedecer a seguinte fórmula.F(Y) >= UN(Y) + P(Y). Onde F(Y) é a capacidade máxima

que a fita suporta, UN(Y) é a capacidade de recurso utilizado pelo último nível da fita e P(Y) é a quantidade de recurso desejada para este pedido. Caso contrário o retângulo é empacotado em um próximo nível. Cada nível da fita é definido pelo tempo de utilização de um único VS. A figura 5.1 mostra como o algoritmo APN se comporta em dois casos distintos com a chegada de um novo pedido para ser agendado na fita.

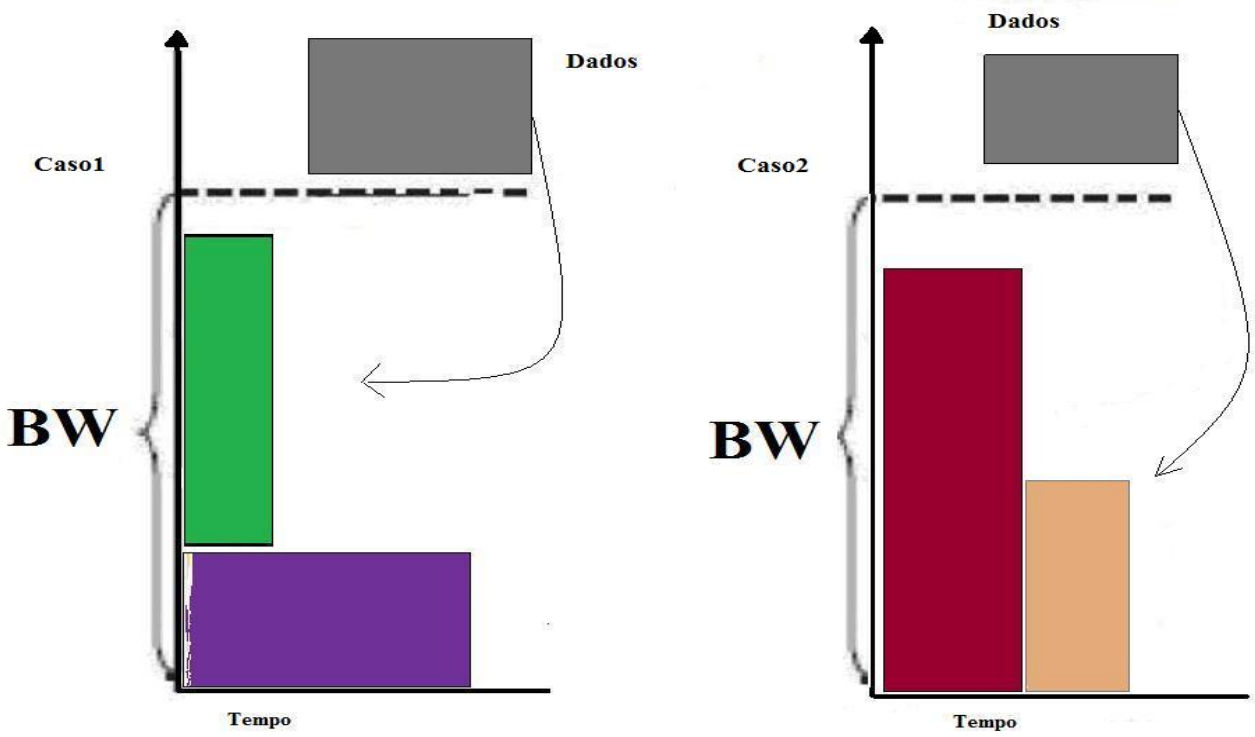

Figura 5.1: Agendamento feito através do APN

Na figura 5.1 aparecem dois casos distintos para utilização do algoritmo APN (Armazena no próximo nível). No caso 1 foi possível adaptar a reserva maleável no último nível existente na fita, mas no caso 2 não foi possível fazer a adaptação da reserva maleável e por isso foi criado um outro nível onde a reserva foi adicionada. O algoritmo APN não tem um resultado expressivo porque sempre busca o último nível ou cria um novo nível para empacotar as reservas. Por isso, muitas reservas são rejeitadas, pois não é possível armazená-las no intervalo de tempo desejado pelo usuário. Então ocorre a rejeição do pedido, acarretando na baixa utilização da fita.

5.1.2 Armazena no primeiro nível possível (APNP)

No algoritmo APNP (Armazena no primeiro nível possível) os retângulos também são armazenados conforme a hora de chegada. Este algoritmo percorre os níveis da fita, até encontrar um nível que seja possível armazenar a reserva maleável de um determinado pedido, ou seja,

a reserva é empacotada no primeiro nível possível. O pedido é enviado para o gerenciador de reservas maleáveis que utiliza o algoritmo APNP. Para que este pedido seja armazenado deve respeitar as seguintes condições:

- A capacidade de recurso ocupado por um nível da fita mais o recurso utilizado pela reserva não pode ultrapassar a capacidade (altura) da fita.
- O intervalo de tempo da reserva maleável que será armazenada tem que ser menor ou igual o intervalo do nível da fita. Sabemos que cada nível é formando por um VS.

Porém se não for possível armazenar o pedido em algum nível, um outro é criado para armazenar este pedido. Isto se este novo nível estiver dentro do tempo estimado pelo cliente. Caso contrário o pedido de reserva do recurso é rejeitado. A figura 5.2 mostra como funciona o APNP.

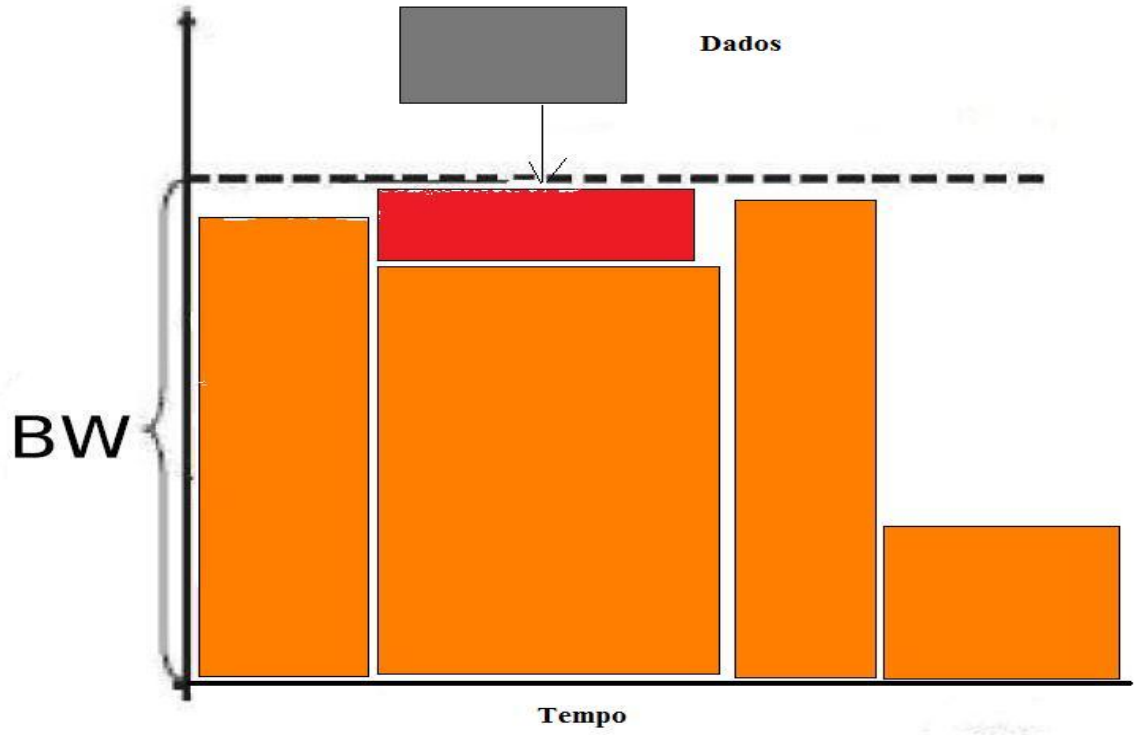

Figura 5.2: Agendamento feito através do algoritmo APNP

Na figura 5.2 o pedido foi armazenado no segundo nível, porém a reserva também podia ser armazenada no quarto nível, mas para seguir o algoritmo APNP o pedido foi armazenado no segundo nível. Vale ressaltar que a reserva armazenada e o segundo VS se transformam em um ou mais VS. Cada nível da fita é formado por um VS (Variable Slots).

5.1.3 Armazena no Melhor nível (AMN)

No algoritmo AMN (Armazena no Melhor Nível) os retângulos (reservas) também são armazenados pela ordem de chegada. Este algoritmo percorre todos os níveis existentes na fita, até encontrar a melhor opção possível para poder empacotar o pedido do usuário. Este melhor nível possível é encontrado da seguinte forma:

- O algoritmo percorre todos os níveis da fita, verificando em quais níveis é possível armazenar a reserva maleável, guardando o nível da melhor posição em uma variável, para depois armazenar a reserva neste nível.
- É feita uma subtração do tempo de utilização da reserva pelo tempo de utilização do nível. O nível que tiver o menor tempo nesta subtração será o escolhido, todos os níveis são percorridos.

A figura 5.3 mostra uma reserva maleável sendo armazenada através do algoritmo (AMN).

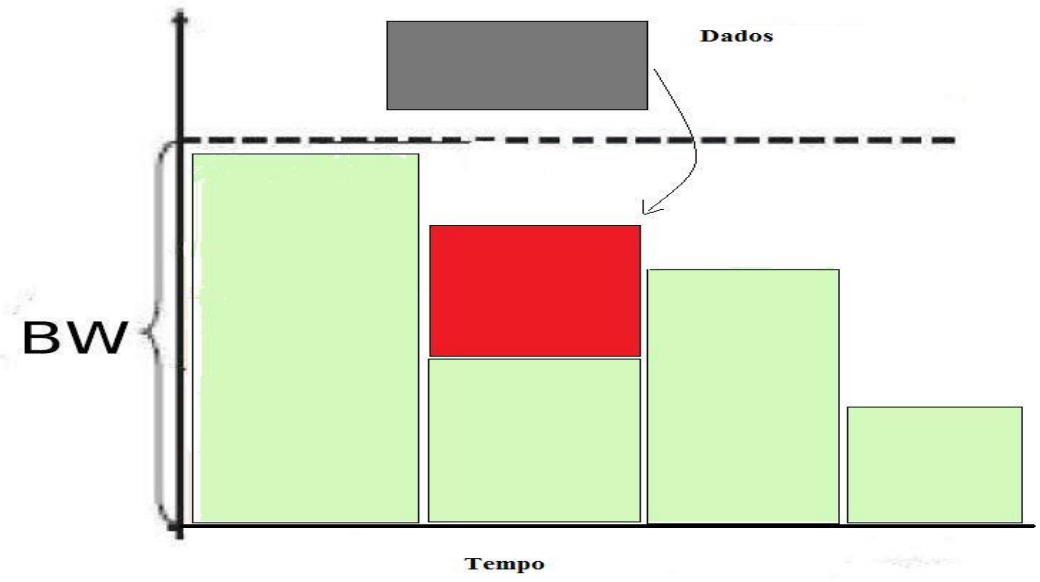

Figura 5.3: Agendamento feito através do algoritmo AMN

Na figura 5.3 existem três níveis onde o pedido poderia ser armazenado, porém o segundo nível foi o melhor local para armazenar esta reserva maleável, pois o intervalo de tempo do segundo nível e o da reserva se equivalem. Todos os níveis foram percorridos pelo algoritmo, mas não tiveram um agendamento melhor que o segundo nível. Como este algoritmo percorre todos os níveis existentes da fita, a sua complexidade algorítima é de O(n). Por isso, o AMN

contém um tempo de resposta maior para armazenar os pedidos que os outros algoritmos citados anteriormente.

5.1.4 Armazena no Último nível (AUN)

No algoritmo AUN (Armazena no último nível) os retângulos são armazenados no último nível da fita. Caso não seja possível armazenar no último VS, um outro nível é criado para armazenar o pedido. Este algoritmo tenta preencher o último nível da fita, buscando uma grande utilização.

O AUN pega se possível a capacidade de recurso disponível no último nível da fita, porém o tempo desta reserva pode ou não ultrapassar o fim do nível. A figura 5.4 mostra dois casos de armazenamento das reservas maleáveis com este algoritmo.

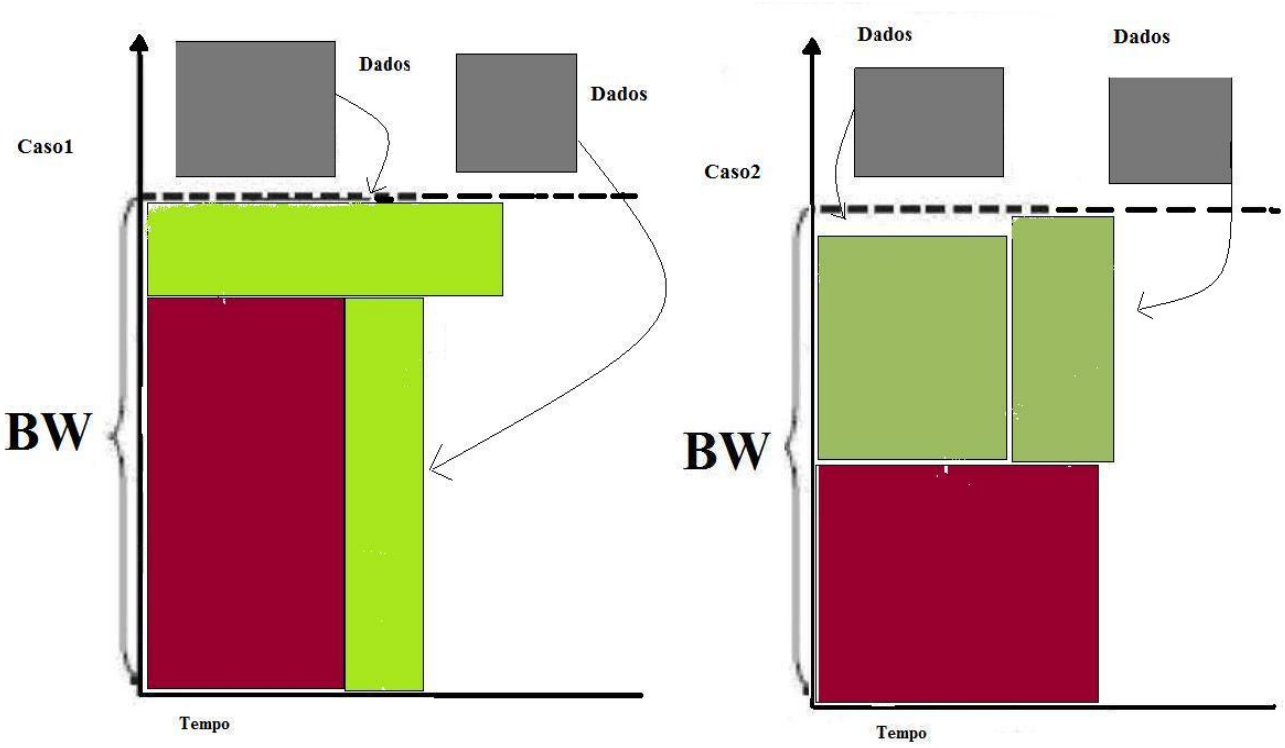

Figura 5.4: Agendamento das reservas feito através do AUN

A figura 5.4 mostra dois casos distintos para utilização do algoritmo AUN. No caso 1, o primeiro pedido utilizou todo o recurso disponível no último nível. Este pedido foi reservado com intervalo de tempo maior que o intervalo do último nível e por isso ultrapassou o fim deste nível. O segundo pedido foi armazenado entre o fim do nível anterior e o final da reserva. Portanto no caso 1 foram formados dois níveis (VS). No caso 2, o primeiro pedido é armazenado entre o intervalo do último nível, porém não foi possível pegar todo o recurso disponível naquele

nível. O outro pedido enviado foi armazenado após o tempo do primeiro pedido armazenado e um pouco a frente do último nível da fita. Portanto neste caso também ficaram dois níveis, mas o último nível ficou com um intervalo muito pequeno, pois está entre o fim da reserva que já estava armazenada e o fim do último pedido armazenado.

Portanto, as reservas são armazenadas com início igual ao do último nível, porém o final da reserva pode ultrapassar ou não o fim deste nível. O algoritmo AUN consegue boa utilização na fita, porém possui uma taxa de rejeição de pedidos alta, pois o algoritmo sempre armazena as reservas no último nível e muitas vezes não condiz com o tempo limitado pelo cliente para acabar a aplicação. Esse algoritmo foi idealizado neste trabalho para obter um bom aproveitamento da fita, ou seja, conseguir uma grande utilização do recurso.

5.1.5 Algoritmo Junção

Este algoritmo é a junção do AUN (Armazena no Último Nível) com o AMN (Armazena no Melhor Nível). Este algoritmo funciona da seguinte maneira:

- Este algoritmo começa usando a estratégia do AMN, buscando a melhor posição para armazenar o pedido enviado pelo usuário.
- Porém se o AJ (algoritmo junção) não encontrar a melhor posição através do ANM, não cria um nível novo, como acontece no AMN, e começa a utilizar as estratégias de armazenamento do AUN.

Este algoritmo conseguiu a melhor utilização da fita entre os implementados, pois uniu as melhores estratégias dos algoritmos AMN e AUN. Porém o AJ é o algoritmo que tem a maior complexidade algorítmica e, portanto, não é o mais indicado quando se deseja um rápido tempo de reposta para reservar um pedido e não se deseja boa utilização do recurso. Neste trabalho surgiu a idéia de fazer a junção deste dois algoritmos (AMN e AUN) e este algoritmo junção se mostrou mais eficaz, para reservas parciais de recursos e para reservas de recursos idênticos na utilização da fita.

5.2 Resultados Obtidos

Foi comparado o desempenho da reserva fixa e da reserva maleável para os casos de aceitação dos pedidos e utilização da fita.

Para fazer transferências dos dados pela rede com reservas maleáveis, o usuário deve enviar os seguintes atributos no pedido:

- O nó de origem;
- O nó de destino;
- A quantidade de bytes que serão transmitidos;
- Um atributo indicando qual o horário mais cedo possível para transferência dos dados começar;
- E um atributo indicando o horário que a transferência deve acabar.

O gerenciador armazena as reservas em um dos níveis da fita que está armazenada em uma lista. Cada posição desta lista contém um VS que tem a seguinte dimensão: o tempo de utilização é o seu comprimento e o recurso utilizado é a sua largura. A figura 5.5 mostra um gráfico com o desempenho dos algoritmos implementados neste trabalho.

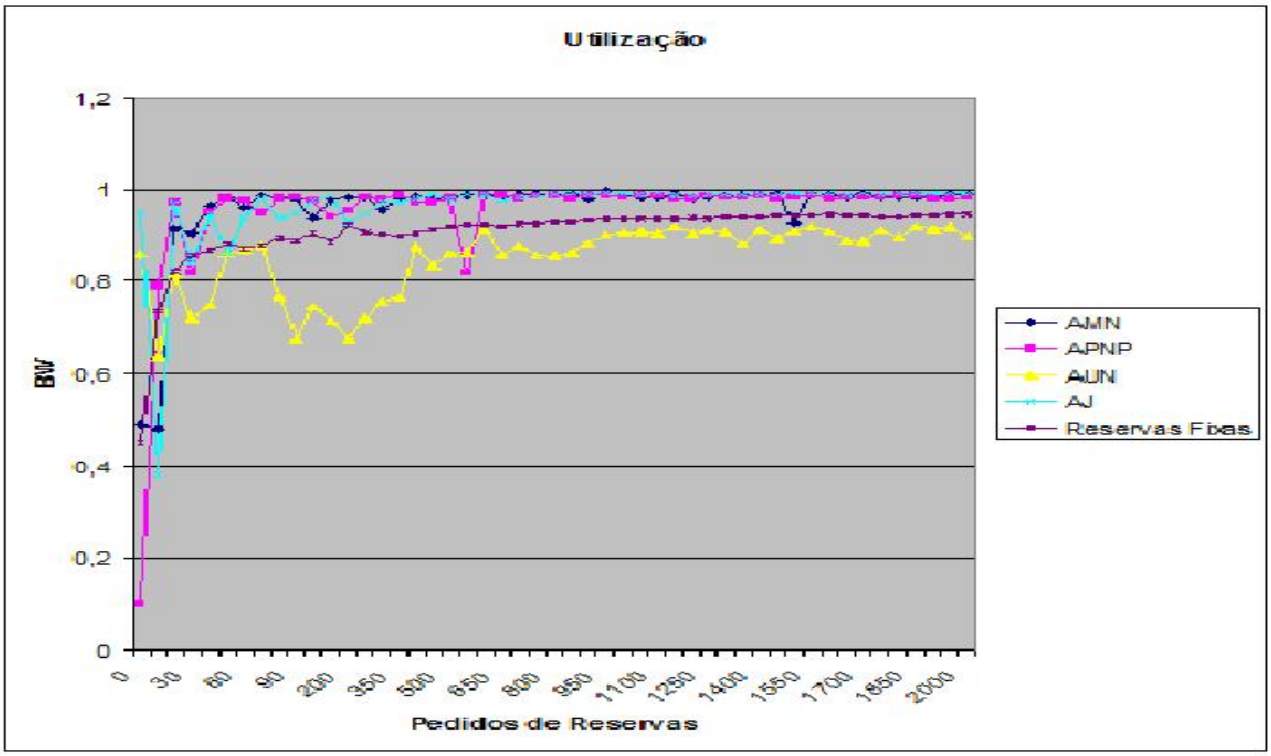

Figura 5.5: Resultado dos algoritmos na utilização da fita para reservas parciais

A figura 5.5 apresenta uma comparação entre os algoritmos (para reservas maleáveis) e para as reservas fixas na utilização da fita. Mostrando que as reservas maleáveis conseguem

uma maior utilização da fita e que o algoritmo AJ (algoritmo junção) tem o melhor desempenho e com utilização mais constante. Todos as reservas foram feitas de formas antecipadas (RA). Sendo este gráfico é para reservas parciais de recursos.

A taxa de aceitação dos pedidos é maior para as reservas maleáveis, pois é possível ajustar a reserva conforme a utilização da fita. Porém o tempo de reposta avisando se o pedido foi aceito é mais lento, porque o gerenciador tem que calcular as dimensões da reserva, enquanto nas reservas fixas o gerenciador só armazena as reservas, pois o usuário já manda o pedido com as dimensões das reservas definidas.

O AJ (Algoritmo Junção) também foi utilizado para fazer o gerenciamento das reservas maleáveis para recursos idênticos, pois este algoritmo foi o que obteve o melhor resultado para a utilização da fita. A figura 5.6 mostra o desempenho do AJ para as reservas maleáveis de recursos idênticos. Foram simulados cincos recursos idênticos que podem ser cincos placas de redes ou cinco processadores.

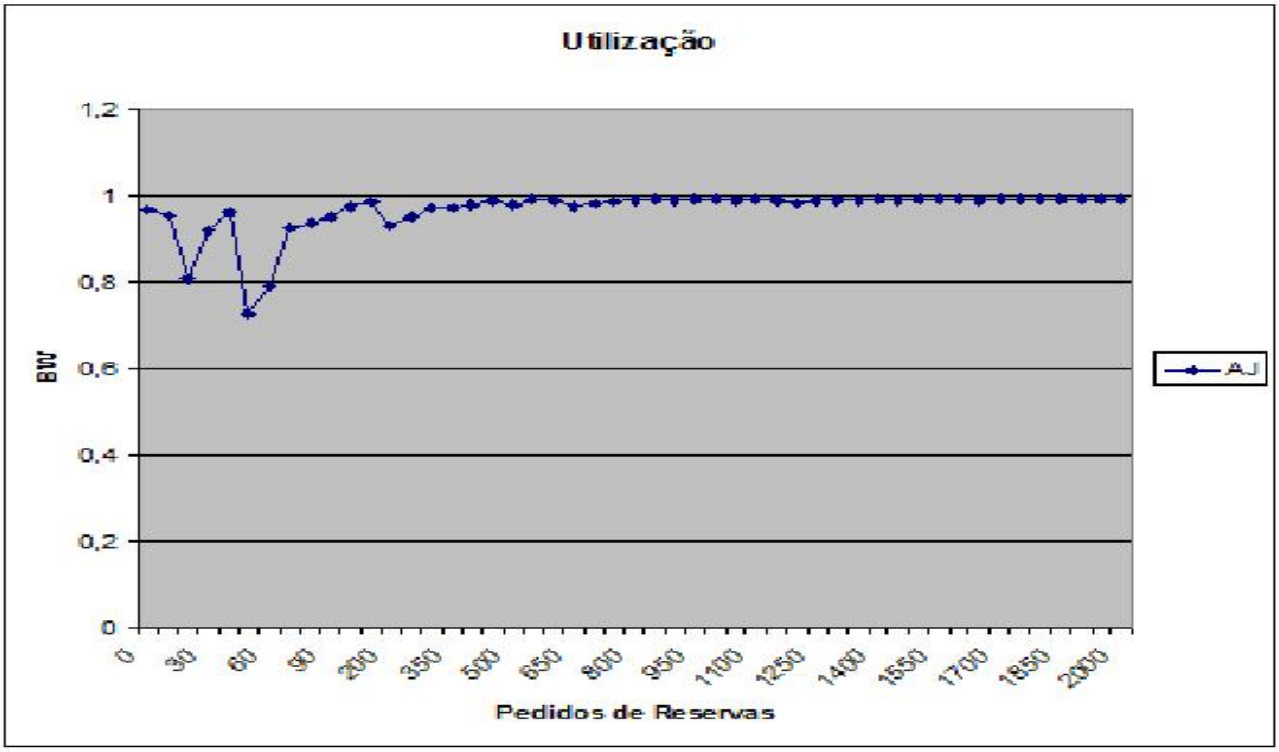

Figura 5.6: Resultado do algoritmo AJ na utilização da fita para reservas maleáveis de recursos idênticos

A taxa de aceitação dos pedidos para reservas maleáveis de recursos idênticos é mais baixa, pois é mais difícil fazer a adaptação das reservas na fita, pois cada um dos recursos deve estar totalmente armazenado ou totalmente disponível. Por isso, a taxa de aceitação das reservas maleáveis para reservas parciais é maior que a taxa de aceitação para os recursos idênticos,

mas para utilização da fita, o resultado é muito semelhante. É fornecida a qualidade de serviço pelo gerenciador, pois a utilização do recurso e taxa de aceitação dos pedidos são altas. Além de facilitar a utilização do usuário, pois ele não precisar se preocupar com as dimensões das reservas.

6 *Conclusão*

As atividades de pesquisa realizadas neste período foram sobre reservas e alocação de recursos na Grade, sendo marcadas por muita leitura de trabalhos científicos e interação com o orientador e com outras pessoas que trabalham no LRG (Laboratório de Redes e Gerência).

O objetivo foi à modelagem e criação de um gerenciador de reservas maleáveis de recursos na Grade. Esse gerenciamento foi feito através da verificação da disponibilidade dos recursos. A grande dificuldade foi encontrar um algoritmo eficiente para fazer o armazenamento das reservas de recurso na fita, pois é um problema NP - completo.

Das experiências obtidas, com a pesquisa sobre Grade, foi possível verificar a importância da área, que está se projetando como um padrão, para computação distribuída, cada vez mais aceito e em expansão. A contribuição com os estudos e modelos dos gerenciadores de reservas de recursos é algo muito gratificante. Pois é muito importante que a Grade tenha políticas de reserva para os recursos. Pois assim fornecerá uma boa qualidade de serviço a quem deseja utiliza-lo. Uma vez que os resultados de pesquisa adquiridos permitem uma melhoria significativa na área, acelerando a adesão deste padrão em ascensão.

Para trabalho futuros, é possível citar algumas carências que podem ser supridas. Uma delas é o suporte a co-reservas, ou seja, reserva de vários recursos heterogêneos de domínios distintos ou organizações virtuais, pois este suporte é importante para uma aplicação que utilize uma grande quantidade de recursos. O gerenciador também deve lidar com as reservas fixas e maláveis ao mesmo tempo, pois existem alguns recursos que não podem ser armazenados pelas reservas maleáveis. Seria interessante o empacotamento de forma off-line, pois existem casos que a aplicação gera grande fluxos de tarefas e sabem-se antecipadamente os recursos que seriam utilizados. Portanto, os algoritmos de empacotamento off-line teriam uma melhor desempenho.

Referências Bibliográficas

ASSUNÇÃO, M. *Implementação e Análise de uma Arquitetura de Grids de Agentes para a Gerência de Redes e Sistemas: UFSC, 2003. 188p*. Tese (Doutorado) — Dissertação Mestrado, 2003.

BARZ, C. et al. Argon-allocation and reservation in gridenabled optical networks. *BMBF-VIOLA Project, Tech. Rep. B*, v. 2, 2005.

BARZ, C. et al. Co-allocation of compute and network resources in the VIOLA-testbed. *Terena Networking Conference, Catania, May*, 2006.

BURCHARD, L.; HEISS, H.; ROSE, C. D. Performance issues of bandwidth reservations for grid computing. *Computer Architecture and High Performance Computing, 2003. Proceedings. 15th Symposium on*, p. 82–90, 2003.

CZAJKOWSKI, K. et al. Usage Scenarios for a Grid Resource Allocation Agreement Protocol. *draft-ggf-graap-usagescenarios-01 T*, 2006.

FOSTER, I. et al. A distributed resource management architecture that supportsadvance reservations and co-allocation. *Quality of Service, 1999. IWQoS'99. 1999 Seventh International Workshop on*, p. 27–36, 1999.

FOSTER, I. et al. A distributed resource management architecture that supports advance reservations and co-allocation. In: *Proceedings of the 7th IEEE International Workshop on Quality of Service (IWQoS'99)*. London, England: IEEE, 1999. ISBN 9780780356719.

GRIMSHAW, A.; WULF, W. The Legion vision of a worldwide virtual computer. *Communications of the ACM*, ACM Press New York, NY, USA, v. 40, n. 1, p. 39–45, 1997.

KUNRATH, L.; WESTPHALL, C. B.; KOCH, F. L. Towards advance reservation in large-scale Grids. IEEE Computer Society Press, 2008. To appear in *The Proceedings of The Third International Conference on Systems (ICONS 2008)*. IEEE Computer Society Press.

MINGBIAO, L. et al. Optimization of Grid Resource Allocation Combining Fuzzy Theory with Generalized Assignment Problem. *Grid and Cooperative Computing, 2007. GCC 2007. Sixth International Conference on*, p. 142–146, 2007.

NTENE, N.; VUUREN, J. van. A survey and comparison of level heuristics for the 2D oriented strip packing problem. *Discrete Optimization, Submitted*, 2007.

SIDDIQUI, M.; VILLAZÓN, A.; FAHRINGER, T. Grid capacity planning with negotiation-based advance reservation for optimized QoS. In: *SC'2006 Conference CD*. Tampa, FL, USA: IEEE/ACM SIGARCH, 2006.

TACHIBANA, T.; KASAHARA, S. Burst-cluster transmission: service differentiation mechanism for immediate reservation in optical burst switching networks. *IEEE Communications Magazine*, v. 44, n. 5, p. 46–55, 2006.

WANDAN, Z. et al. G-RSVPM: A Grid Resource Reservation Model. *Semantics, Knowledge and Grid, 2005. SKG'05. First International Conference on*, p. 79–79, 2005.