

UNIVERSIDADE FEDERAL DE SANTA CATARINA – UFSC

**PROPOSTA DE UM PROTÓTIPO PARA A CONFIGURAÇÃO E
EXECUÇÃO DE APLICAÇÕES NO AMBIENTE DE GRADE
COMPUTACIONAL GLOBUS 3.0**

PAULI FIGUEREDO GOMES

Orientador: Carlos Becker Westphall

Florianópolis, 2008

PAULI FIGUEREDO GOMES

**PROPOSTA DE UM PROTÓTIPO PARA A CONFIGURAÇÃO E
EXECUÇÃO DE APLICAÇÕES NO AMBIENTE DE GRADE
COMPUTACIONAL GLOBUS 3.0**

Projeto de pesquisa para o desenvolvimento do Trabalho de conclusão de curso apresentado como parte dos requisitos para a obtenção de título de Bacharel em Sistemas de Informação à Universidade Federal de Santa Catarina – UFSC, no curso de Sistemas de Informação.

Orientador: Prof.Dr. Carlos Becker Westphall

Florianópolis, 2008

Autoria:Pauli Figueredo Gomes

Titulo: Proposta de um protótipo para a configuração e execução de aplicações no ambiente de Grade computacional Globus 3.0.

Os componentes da banca de avaliação, abaixo listados, consideram este trabalho aprovado.				
	Nome	Titulação	Assinatura	Instituição
1	Carlos Becker Westphall	Doutor		UFSC
2	Carla Merkle westphall	Doutora		UFSC
3	Fernando Augusto da Silva Cruz	Doutor		UFSC

Data da aprovação: ____ de _____ de _____

“Dedico este trabalho a meus pais.”

AGRADECIMENTOS

*Agradeço acima de tudo aos meus pais
Evaldo e Maria, que sempre me apoiaram e
mostraram a importância do estudo e nunca se
importaram com as despesas, que foram
muitas, para que eu pudesse concluir este
curso.*

RESUMO

A computação em Grade é um das tecnologias que vem ganhando cada vez mais espaço, e recebendo cada vez mais atenção não somente por parte dos pesquisadores mas de áreas comerciais, indicando o surgimento de um novo paradigma.

Porém a utilização deste paradigma ainda não é totalmente utilizado por empresas e usuários em geral e apesar dos investimentos, o acesso a recursos através de ambientes de grades ainda é bastante complexo necessitando de grande especialização por parte dos usuários que pretendem utilizar esta tecnologia. Assim, constata-se a existência de uma poderosa tecnologia mas que ainda é pouco utilizado por usuários não especialistas devido a seu alto grau de complexidade.

Por esta razão torna-se considerável o estudo de meios para que os usuários sem grande especialização possam fazer o uso mais transparente possível de um ambiente de grande computacional complexo como por exemplo o Globus 3.

Assim este trabalho tem como objetivo propor uma solução para que um usuário iniciante nesta plataforma possa realizar de maneira mais transparente possível a execução de uma aplicação no ambiente de Computação em Grade Globus 3.

Palavras-chave: Computação em Grades, Grid Computing, Globus 3.

ABSTRACT

This work, we propose a small prototype for configuring and run na application in a Grid Computing environment Globus. The mean goal is to facility the configuring and execution applications for a beginners users in Globus.

SUMÁRIO

1	Introdução	11
1.1	<i>Visão Geral</i>	11
1.2	<i>Objetivos Gerais</i>	12
1.3	<i>Objetivos Específicos</i>	12
1.4	<i>Proposta e Motivação</i>	13
1.5	<i>Organização do Trabalho</i>	14
2	Computação em Grade	15
2.1	<i>Conceituação</i>	15
2.2	<i>Arquitetura</i>	16
2.2.1	<i>Ambiente</i>	17
2.2.2	<i>Conectividade</i>	17
2.2.3	<i>Recursos</i>	17
2.2.4	<i>Coletivo</i>	18
2.2.5	<i>Aplicação</i>	19
2.3	<i>Áreas de Aplicação</i>	19
2.3.1	<i>Supercomputação Distribuída</i>	19
2.3.2	<i>Computação de alta vazão</i>	19
2.3.3	<i>Computação sob demanda</i>	20
2.3.4	<i>Computação de processamento intensivo de dados</i>	20
2.3.5	<i>Computação Colaborativa</i>	20
2.4	<i>Comparação com outras tecnologias</i>	21
2.4.1	<i>Peer-to-Peer</i>	21
2.4.2	<i>Cluster</i>	22
2.5	<i>Padronização</i>	23
2.5.1	<i>Open Grid Service Architecture (OGSA)</i>	23
2.5.2	<i>Open Grid Service Infrastructure (OGSI)</i>	23
2.5.3	<i>Serviços de Grade (Grid Services)</i>	23
2.5.4	<i>Web Service Resource Framework (WRSF)</i>	24

2.6	<i>Plataformas</i>	25
2.6.1	My Grid e OurGrid.....	25
2.6.2	Condor.....	25
2.7	<i>Globus</i>	26
2.7.1	Visão Geral.....	26
2.7.2	Arquitetura do sistema.....	26
2.7.2.1	Gerenciamento de recursos.....	26
2.7.2.2	Serviço de Informação.....	27
2.7.2.3	Serviço de Segurança.....	27
2.7.2.4	Serviços de Transferência de arquivos.....	28
2.8	<i>Conquistas da plataforma e do projeto Globus</i>	28
2.9	<i>Experiência com o Globus 3</i>	28
3	Modelo Conceitual	30
3.1	<i>Ferramentas</i>	32
4	Execução do protótipo	33
4.1	<i>Introdução</i>	33
4.2	<i>Contexto</i>	33
4.3	<i>Execução</i>	34
4.4	<i>Configuração</i>	35
4.5	<i>Cadastro da aplicação</i>	38
4.6	<i>Executar Aplicação</i>	40
5	Resultado, dificuldades e trabalhos futuros.	41
6	Considerações Finais	43

INTRODUÇÃO

Este trabalho apresenta o desenvolvimento de um protótipo de software que tem como objetivo facilitar a execução de aplicações em um ambiente de Grade Computacional Globus 3. Para isso, mostramos o conceito de grade computacional, de forma mais geral apresentamos outros ambientes que existem, e mais detalhado falamos sobre o Globus 3 e os problemas de complexidade que este ambiente possui para um usuário iniciante.

A seguir é apresentado o modelo conceitual que é a base para a implementação do protótipo. Nele são definidas todas as funcionalidades bem como a sua arquitetura. Seguindo é mostramos todo o funcionamento do protótipo através da apresentação das telas que são mostradas para o usuário em cada etapa. Por último temos as considerações finais onde são apontados os problemas e possíveis melhorias, bem como os trabalhos futuros.

1 INTRODUÇÃO

1.1 Visão Geral

Por volta dos anos 90, cientistas da comunidade de Processamento de alto desempenho criaram o conceito de computação em Grade. A proposta inicial era disponibilizar uma infraestrutura de hardware e de software que possibilitasse o acesso a grande capacidade computacional distribuídos geograficamente.

Embora num primeiro momento a computação em grade se pareça com outras técnicas de computação de alto desempenho tais como Clusters, ela apresenta algumas diferenças essenciais, como por exemplo, grande distribuição geográfica e grande variedade de recursos envolvidos, tratando-se de um novo paradigma.

A utilização deste paradigma ainda não é totalmente utilizado por empresas e usuários em geral e apesar dos investimentos, o acesso a recursos através de ambientes de grades ainda é bastante complexo necessitando de grande especialização por parte dos usuários que pretendem utilizar esta tecnologia. Assim, constata-se a existência de uma poderosa tecnologia mas que ainda é pouco utilizado por usuários não especialistas devido a seu alto grau de complexidade.

Por esta razão torna-se considerável o estudo de meios para que os usuários sem grande especialização possam fazer o uso mais transparente possível de um ambiente de grande computacional.

1.2 Objetivos Gerais

Este trabalho tem como objetivo geral fazer um estudo para as possibilidades de uso de computação em grade por usuários comuns, ou seja sem grande especialização nesta tecnologia, usando o Globus 3 como plataforma de computação em grade.

Alem de estudar suas características e complexidades no contexto de um usuário não especializado e descrever os possíveis problemas e dificuldades para a utilização da mesma.

A idéia básica é realizar um estudo de caso em que exemplifique o uso desta plataforma por um usuário sem grandes especialização. Verificar suas dificuldades, analisar a complexidade da plataforma e propor uma solução para os possíveis problemas encontrados.

1.3 Objetivos Específicos

Com base nos objetivos gerais descritos acima, os seguintes objetivos específicos são propostos:

- Levantamento teórico e aprendizado de conceitos relacionados a ambientes de Grade Computacional e tecnologias envolvidas;
- Estudo de ferramentas de programação e APIs específicos para esta plataforma, como por exemplo a CoK;
- Adquirir conhecimento teóricos e práticos da plataforma Globus 3.0;
- Implementar uma solução de software, ou seja uma aplicação, que torne o mais transparente possível a execução de uma aplicação qualquer no ambiente de Grade estudado, com o objetivo de facilitar o uso deste ambiente para usuários iniciantes no uso desta plataforma.

Para a realização de tais objetivos um levantamento bibliográfico foi efetuado, além da realização de várias tarefas práticas, como instalação, tutoriais, configuração, testes, programação de exemplos, etc. Tudo para se verificar o funcionamento do Globus 3.

1.4 Proposta e Motivação

Ao contrário das Grades *plug-and-play*, o Globus 3 requer grande esforço para ser instalado, e igualmente um grande esforço para ser configurado, tornando tudo bem complexo. Diversas configurações como acesso de usuários, grid services disponíveis, serviços a serem utilizados, escalonadores de aplicações e escalonadores de recursos. Além de uma verdadeira maratona para criação e autenticação de uma série de certificados de segurança para cada usuário e estação de trabalho da Grade.

A instalação do sistema, além de bastante complexa como já foi dito, também é bastante longa. A fase de geração, assinatura e repasse de certificados de autorização é bastante confusa para não chamá-la de obscura. Em uma rápida busca na Internet podemos encontrar dezenas de relatos de administradores, estudantes e pesquisadores frustrados por não terem conseguido a conclusão da instalação correta do Globus após diversas tentativas, desistindo até mesmo de utilizar a plataforma.

É justificável que a proposta do Globus é complexa demais para uma solução *plug-and-play*, uma vez que considera ambientes altamente heterogêneos com múltiplos domínios administrativos, etc. Mas o Globus realmente carece de meios que facilitem a sua implantação, como por exemplo uma padrão de instalação pré-configurado para um único domínio administrativo.

Assim ao nos depararmos com estas dificuldades iniciais ao lidar com o Globus 3 ficou evidente a necessidade de uma solução para o problema de configuração e o envio de uma aplicação neste ambiente em um domínio único, como por exemplo, um laboratório ou departamento.

Por esse motivo este trabalho tem por objetivo dar uma pequena contribuição para facilitar os passos necessários para a configuração e submissão de uma aplicação na Grade. É exatamente este o foco principal que propõe este trabalho. Para isso é proposto a criação de um protótipo funcional que tem por objetivo tornar estes passos o mais transparente possível para o usuário.

1.5 Organização do Trabalho

O capítulo 2 é apresentado a parte conceitual de computação em Grade que é a principal tecnologia na qual este trabalho foi elaborado. Juntamente neste capítulo é apresentada uma descrição do Globus 3

No capítulo 3 são apresentadas a parte conceitual do protótipo e as ferramentas adicionais que foram utilizadas para a sua elaboração

O capítulo 4 apresenta todo o funcionamento do protótipo através da explicação de cada tela e a funcionalidade da mesma.

No capítulo 5 são apresentados os resultados obtidos e os possíveis trabalho futuros e por último no capítulo 6 as conclusões do trabalho são apresentadas.

2 COMPUTAÇÃO EM GRADE

2.1 Conceituação

A computação em Grade é uma nova abordagem para os antigos sistemas distribuídos com a diferença de possuir compartilhamento de recursos em larga escala. Com a constante melhoria do desempenho que redes de computadores vem experimentando, naturalmente surgiu a idéia de se utilizar computadores independentes conectados em rede como plataforma para execução de aplicações paralelas (Foster e Kesselman, 1999). Esta abordagem permite executar aplicações complexas com grande demanda computacional, alocando diversos recursos espalhados na internet, a um custo relativamente baixo.

É comum encontrarmos na literatura uma analogia da computação em grade com a distribuição de energia elétrica. Assim da mesma forma que um usuário da rede elétrica possui acesso a energia elétrica de acordo com a sua demanda, com a computação em Grade pretende-se oferecer poder computacional na medida em que o usuário necessitar. Já o usuário não tem que se preocupar de onde e aonde estes recursos se encontram e sim apenas fazer o uso que lhe é necessário.

O termo Grade, “Grid” em inglês, foi criado por volta dos anos 90 para denotar a então proposta de infraestrutura de computação distribuída para ciência e engenharia avançada (Foster e Kesselman, 2004).

Uma Grade computacional pode ser classificada também como uma estrutura distribuída, heterogênea e complexa. Suas principais características são:

- Múltiplos Domínios Administrativos e autônomos: Todos os recursos de uma Grade estão distribuídos em vários domínios e não pertencem a uma mesma organização. Por essa razão a autonomia dos proprietários de cada domínio precisa ser mantida.
- Heterogeneidade: Grande quantidade de recursos que são compostos de vários tipos de tecnologia.

- Escalabilidade: Pode crescer de uma pequena quantidade de recursos para uma grande quantidade indefinidamente.
- Dinamicidade e Adaptabilidade: Capacidade de que se houver falha em um recurso o comportamento deverá ser adaptado dinamicamente e fazer o uso dos recursos de maneira efetiva.

Na figura 2.1 podemos ver esquematicamente um exemplo de uma Grade computacional.

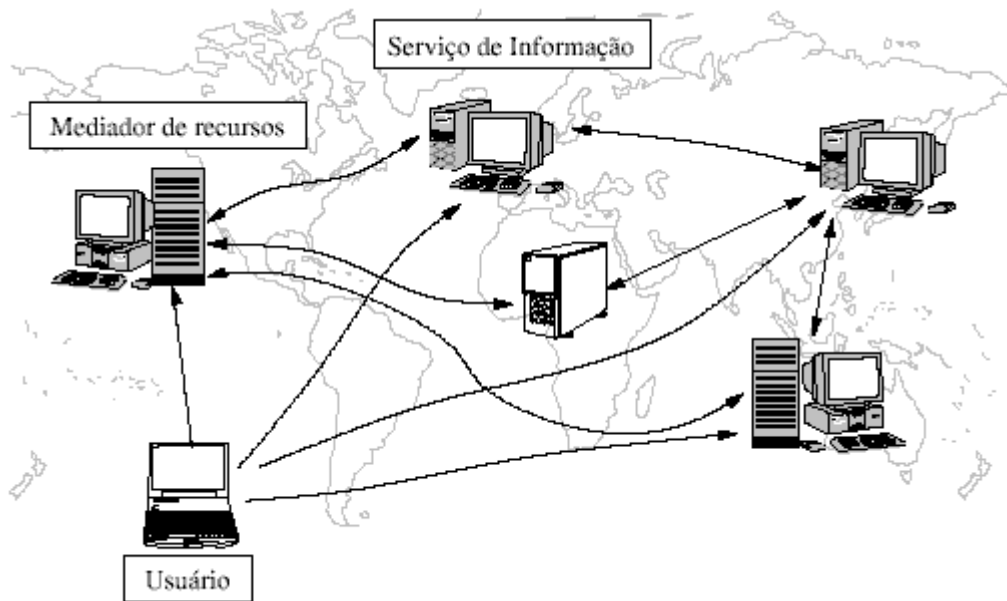


Figura 2.1: Exemplo de uma Grade Computacional (Foster et al., 2001), adaptado

2.2 Arquitetura

A arquitetura de Grade é desenvolvida com base na grande abrangência e grande número de parceiros e recursos envolvidos. Por isso é necessário que essa arquitetura tenha como base protocolos e padrões que definam seus mecanismos básicos. Assim uma arquitetura aberta baseada em padrões facilita a escalabilidade, interoperabilidade, portabilidade e compartilhamento de código (Foster e Kesselman, 2004).

Isto é fundamental pois irá permitir o compartilhamento de recursos entre as partes, além da adição de novos participantes.

Em (Foster et al., 2001) é proposto uma arquitetura que mostra os principais componentes como podemos ver na figura 2.2.

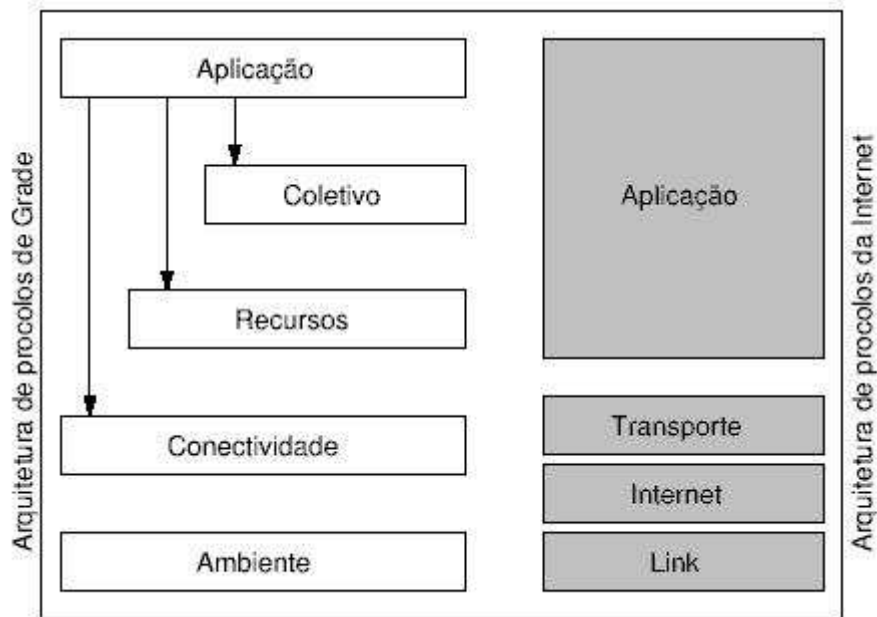


Figura 2.2: Arquitetura de Grade Computacional (Foster et al., 2001), adaptado

A seguir segue uma breve descrição de cada componente desta arquitetura juntamente com suas principais funções.

2.2.1 Ambiente

Camada responsável por fornecer os recursos compartilhados, como por exemplo: armazenamento, catálogos e sensores. Alguns destes componentes são considerados básicos para o funcionamento desta camada. Um exemplo disto é o mecanismo de pesquisa que permite descobrir informações sobre a estrutura e estado dos recursos. Responsável também pelo gerenciamento para fins de controle e garantia da qualidade de serviço.

2.2.2 Conectividade

Nesta camada são definidos os protocolos centrais de comunicação e autenticação requeridos por transações exclusivas de uma rede em Grade. Eles permitem a troca de dados entre os recursos e a camada ambiente.

2.2.3 Recursos

Nesta camada são definidos alguns protocolos, (APIs) Application Programming Interfaces e (SDKs) Software Development Kit's para uma série de utilizações; como por

exemplo: iniciação, monitoração, controle, geração de relatórios, dentre outras tarefas para o compartilhamento de recursos individuais. Estas duas camadas (Recursos e Conectividade) juntas formam o gargalo da arquitetura, por esta razão os protocolos a serem utilizados devem ser limitados a um conjunto pequeno.

2.2.4 Coletivo

Esta camada também fornece uma série de protocolos, APIs e SDKs que possibilitam e efetivam a interação de recursos. Seus componentes fornecem muitos recursos sendo a maioria destes compartilhados.

Os serviços desta camada são:

- Serviços de diretórios: permite a descoberta e a verificação da existência de recursos.
- Serviços de co-alocação: escalonamento e mediação (brokering): permite a alocação de um ou mais recursos para propósitos específicos e escalonamento.
- Serviços de monitoramento de diagnósticos: dão suporte a monitoração de falhas, ataques e outros recursos.
- Serviços de aplicação de dados: dão suporte ao gerenciamento de recursos de armazenamento e aumenta a velocidade de acesso aos dados.
- Serviços de programação para a grade: são modelos de programação para serem usados em ambientes de grades.
- Serviços de descoberta de softwares: permite a descoberta e seleção da aplicação e a plataforma que a mesma será executada.
- Serviço de autorização: define políticas de acesso a recursos e força a definição de políticas de acesso.
- Serviço de compatibilização e pagamento: coletam informações de usos de recursos para propósitos de cobrança.
- Serviço de colaboração: fornecem a troca coordenada de informação.

2.2.5 Aplicação

Seu desenvolvimento é realizado a partir de chamadas de serviços fornecidos pelas camadas anteriores.

Cabe aqui lembrar que embora esta arquitetura represente as necessidades requeridas por definição para um ambiente de Grade computacional ainda não existe um padrão estabelecido de arquitetura, sendo a proposta descrita acima uma das mais aceitas pela comunidade em geral e uma das fortes candidatas a se tornar padrão de fato.

2.3 Áreas de Aplicação

São muitas as áreas de aplicação que podem fazer o uso de uma infraestrutura de computação em Grade, as de maiores relevâncias podem ser agrupadas em 5 principais categorias como definidas em (Foster e Kesselman, 1999) e abaixo relacionadas.

2.3.1 Supercomputação Distribuída

Faz o uso de Grades para agregar uma quantidade grande de poder computacional para que seja possível resolver problemas onde os outros sistemas não conseguem disponibilizar seus recursos. Estes recursos podem ser uma grande quantidade de computadores espalhados pela internet ou todos os computadores de uma determinada empresa.

Neste tipo de aplicação encontra bastante uso em áreas que necessitam fazer simulações de problemas complexos, como na pesquisa científica ou em problemas de física, biologia, metrologia, campo militar, economia, finanças e computação gráfica.

2.3.2 Computação de alta vazão

Já neste tipo de aplicação as Grades são utilizadas para escalonar uma grande quantidade de recursos independentes, com o objetivo principal a utilização dos recursos ociosos, geralmente estes recursos são estações de trabalho (desktops). Buscando reunir uma certa quantidade de recursos disponíveis para a solução de um determinado problema.

Um projeto bastante conhecido nesta classe de aplicação, é o projeto “SETI@Home”, para a suposta busca de vida extraterrestre, onde grandes quantidades de dados são analisadas em computadores comuns espalhadas na internet quando estes computadores estão ligados porém fora de uso.

2.3.3 Computação sob demanda

Geralmente neste tipo de aplicação utiliza-se a computação em Grade para fazer o uso, por um período determinado de recursos que não podem ser mantidos localmente por alguma motivo, como por exemplo, questões financeiras. Estes recursos são bastante diversificados, podemos citar por exemplo: aplicações, repositório de dados, sensores, etc. Também neste tipo de uso, as decisões de utilização do recurso nas aplicações costumam ser orientadas por questões financeiras ao invés de questões de desempenho como até então vinham sendo utilizadas.

2.3.4 Computação de processamento intensivo de dados

Neste tipo de aplicação que como o nome diz faz uso intensivo de dados, o objetivo principal é sintetizar novas informações a partir de dados que são mantidos em repositórios, bibliotecas digitais e bancos de dados geograficamente distribuídos. Neste tipo de aplicação se faz uso intensivo de recurso computacional e da comunicação de dados.

2.3.5 Computação Colaborativa

Já neste tipo de aplicação a atenção é voltada para o objetivo de melhorar a interação homem-a-homem, onde é definido um espaço virtual estruturado. Muitas destas aplicações buscam melhorar o uso compartilhado de recursos computacionais, tais como arquivos de dados e simulações. Como exemplo podemos citar: ambientes de realidade virtual, entretenimento e educação.

2.4 Comparação com outras tecnologias

Podemos perceber que a computação em grade não é a única plataforma para execução de aplicações paralelas. Por essa razão nesta seção será feito um breve comparativo de Grades computacionais com duas outras tecnologias: Peer-to-Peer (P2P) e Clusters.

2.4.1 Peer-to-Peer

O Peer-to-Peer, conhecido por P2P, é uma tecnologia que provocou uma mudança considerável em relação aos paradigmas até então existentes. Neste tipo de paradigma não há dependência de uma organização central ou hierárquica, além de dispor aos seus integrantes as mesmas capacidades e responsabilidades.

Podemos definir P2P como um tipo de sistemas e aplicações que utilizam recursos distribuídos para executar tarefas de forma descentralizada.

A inexistência de um servidor central significa que é possível cooperar para a formação de uma rede P2P sem qualquer investimento adicional em hardware de alto desempenho para coordená-la (Rocha et al., 2004).

Uma outra vantagem é o de agregar e utilizar a capacidade de processamento e armazenamento que fica ociosos em computadores que não estão sendo utilizados no momento por seus usuários. O modelo P2P apresenta a vantagem da escalabilidade para fazer uso de crescimento no número de usuários e equipamentos conectados, aumentando a capacidade de processamento.

Em geral, é aceito pela comunidade científica que sistemas P2P devam suportar os seguintes requisitos (Rocha et al., 2004):

- Capacidade de lidar com diferentes taxas de transmissão entre os usuários;
- Autonomia parcial ou total dos nós;
- Garantia parcial ou total dos nós em relação a um servidor central;
- Capacidade dos nós se comunicarem diretamente uns com os outros.

2.4.2 Cluster

A computação em Cluster surgiu no início da década de 90, quando foi constatado que interligando-se dois ou mais computadores, poderia-se obter uma grande quantidade de poder computacional, ao invés de se preocupar em construir processadores cada vez mais potentes (Baker e Buyya, 1988).

Um cluster é um tipo de sistema de processamento paralelo, que consiste de uma coleção de computadores interconectados trabalhando juntos como se fosse um único recurso integrado. O que tem contribuído para que os clusters sejam difundidos como uma plataforma prática para processamento paralelo é a padronização de ferramentas e utilitários usados por aplicações paralelas, como por exemplo a biblioteca Message Passing Interface (MPI) (Brucke al., 1995). Na figura 2.3 temos um exemplo de Cluster.

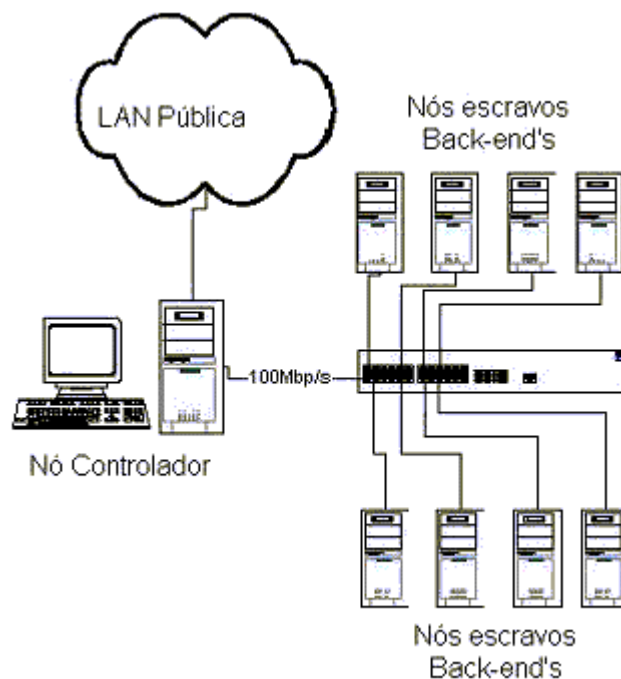


Figura 2.3: Exemplo de cluster

O principais tipos de Clusters são:

- Alta Disponibilidade (High Availability): são construídos para prover uma disponibilidade de serviços e recursos de forma ininterruptas através do uso da redundância implícitas ao sistema. Assim se um nó do cluster vir a falhar as aplicações ou serviços continuarão disponíveis no restante do cluster.
- Balanceamento de carga (Load Balancing): distribui o tráfego nos nodos que executam os mesmas aplicações ou serviços entre os computadores do cluster, desta forma se um nó falhar as requisições são redistribuídas entre os nós disponíveis.

- Combinação High Availability & Load Balancing: Faz a combinação das duas técnicas anteriores aumentando a escalabilidade e a disponibilidade dos serviços ou recursos.
- Processamento Distribuído ou Processamento Paralelo: neste modelo aumenta a disponibilidade e a performance para aplicações, é utilizado para grandes tarefas computacionais, tais como aplicações científicas e grandes análises financeiras.

2.5 Padronização

O resultado do esforço para o desenvolvimento de padrões para a tecnologia de Grades foi a criação do Open Grid Service Architecture (OGSA). Este padrão foi desenvolvido juntamente com a especificação chamada de Open Grid Service Infrastructure (OGSI).

Segue-se uma breve descrição das principais padronizações atuais:

2.5.1 Open Grid Service Architecture (OGSA).

É um modelo cuja a abordagem é baseada em serviços. Esta abordagem vem no sentido de que as Grades poderiam proporcionar serviços com fins científicos e comerciais para qualquer usuário. Estes serviços podem ser um recurso computacional qualquer, armazenamento de dados, programas ou banco de dados.

2.5.2 Open Grid Service Infrastructure (OGSI).

Esta especificação define um conjunto de convenções e padrões para permitir a criação de serviços Web, os Web Services. Nela são definidos abordagens para a criação, declaração e manipulação de conjunto destes serviços.

2.5.3 Serviços de Grade (Grid Services)

São semelhantes a abordagem dos WebServices com algumas características específicas para o ambiente de Grades. A padronização da interface e do método de acesso aos Grid Services permite a criação de diversas implementações com uma interface comum.

2.5.4 Web Service Resource Framework (WRSF)

Foi desenvolvido com o principal objetivo de definir convenções para manipular estados para que aplicações possam buscar, analisar e interagir como os recursos de uma modo padronizado.

A maioria dos esforços para a implementações do OSGA/OGSI é através da plataforma de programação Java, utilizando tecnologias como J2SE, J2EE/Servlets ou J2EE/Jboss. Na figura 2.4 temos representado a estrutura de camadas da arquitetura OSGA/OGSI.

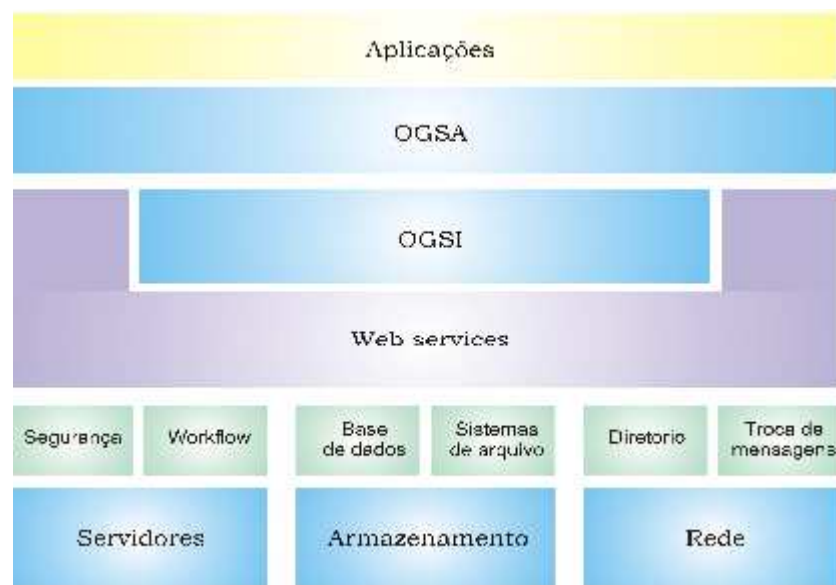


Figura 2.4: Estrutura de camadas da arquitetura OSGA/OGSI.

Juntamente com estes projetos, outros vem trabalhando em conjunto no mesmo sentido, tal como o Grid Interoperability Project (GRIP).

Todos estes padrões tem investido esforços no desenvolvimento de serviços centrais, tais como:

- Gerenciamento e automação de sistemas;
- Segurança;
- Gerenciamento de carga e desempenho;
- Gerenciamento de serviços;
- Serviços de Cluster;

- Gerenciamento de recursos físicos.

Uma das implementações da OGSA/OGSI, trata-se do Globus 3.0, que utiliza contêineres J2EE/Servlets em um servidor Tomcat como base, e o OGSA.NET, que utiliza a plataforma .NET como base do desenvolvimento. Entretanto, a OGSI já está prestes a ser substituída pela especificação Web Services Resource Framework (WSRF), representada na figura 2.5, que separa as funcionalidades em diversas especificações. Além de já contar com os avanços na especificação de Web Services .A versão 4 do Globus já é baseada em WSRF.

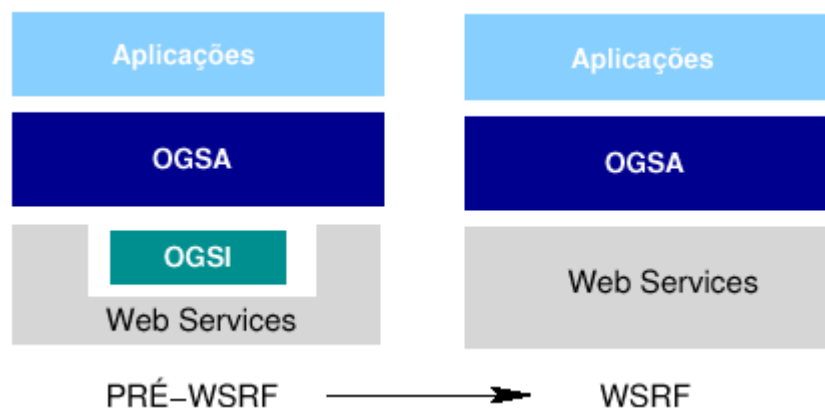


Figura 2.5: Evolução para a especificação WSRF

2.6 Plataformas

Atualmente existem diversas plataformas e projetos em desenvolvimento para a tecnologias de Grades computacionais. Estes projetos vão desde iniciativas puramente acadêmicas e ou mantidos por comunidades abertas até projetos comerciais desenvolvidos por empresas. Dentre todas as existentes algumas merecem destaque como os projetos My Grid e OurGrid, Condor e Globus.

2.6.1 My Grid e OurGrid

É um projeto desenvolvido pela universidade UFCG e é um dos pioneiros de Grades no Brasil. Sua proposta principal é o desenvolvimento de um sistema simples e completo.

2.6.2 Condor

Seu objetivo principal é fornecer grande quantidade de poder computacional a médio e longo prazos utilizando recursos ociosos na rede. No caso seu foco está na alta vazão e não no

alto desempenho computacional. Pode ser usado também no gerenciamento de nós de clusters dedicados.

2.7 Globus

2.7.1 Visão Geral

O Globus é um conjunto de serviços de código e arquiteturas abertas, e uma biblioteca de funções que dão suporte a Grades computacionais e a suas aplicações (Foster et al., 2001).

Seu nome real é Globus Toolkit e foi desenvolvido por um consórcio de várias instituições chamado de Globus Alliance.

Trata-se de uma implementação completa da especificação OGSI totalmente modular e independente como uma caixa de ferramentas (toolkit).

Fornecer uma grande variedade de recursos, serviços, programas e utilitários. É dividido em cinco principais componentes que são: segurança, gerenciamento de recursos, gerenciamento de dados, comunicação, detecção de falhas e portabilidade.

O Globus é implementado de tal forma que seus componentes estão organizados em serviços independentes e podem ser instalados conforme a sua necessidade.

2.7.2 Arquitetura do sistema

2.7.2.1 Gerenciamento de recursos

É definido em 3 camadas e mais uma linguagem para a definição de requisições, a RSL (Resource Specification Language). A camada de fora a qual o usuário estabelece contato é composta por negociadores de recursos (Resource Brokers), que permitem aos usuário realizarem solicitações de alto nível.

A camada intermediária é formada por co-alocadores que possuem a tarefa de quebrar as requisições em sub-requisições, de forma que cada uma possa ser tratada individualmente por um GRAM, permitindo ao usuário abstrair os recursos alocados como uma entidade única.

Já a camada mais interna é chamada de GRAM (Globus Resource Allocation Manager) e tem por finalidade fazer a ponte entre o Globus e o gerenciador de recursos local. As principais responsabilidades do GRAM são:

- Receber requisições RSL conforme a disponibilidade;
- Monitoramento e gerenciamento de processos criados na Grade;
- Atualizar o Serviço de informação (MDS) com as informações dos processos criados e a disponibilidade dos recursos gerenciados.

2.7.2.2 Serviço de Informação

O MDS (Monitoring and Discovery Service) é o serviço responsável por prover informações sobre o estado da grade, criando uma coleção de entidades e protocolos que recebem as informações de diversos outros serviços. As principais entidades que compõem esse serviço são os Provedores de Informação e os Diretórios de Agregados.

Os Provedores de Informação (Information Providers) são responsáveis pela coleta e o monitoramento do estado dos recursos.

Os Diretórios agregados (Aggregate Directories) coletam a informação de diversos provedores de informação, organiza esta informação e responde as consultas sobre as informações e disponibilidade de recursos. Algumas destas informações coletadas e disponibilizadas são: computadores, redes, dispositivos de armazenamento e hardware não tradicionais como equipamentos científicos.

2.7.2.3 Serviço de Segurança

É chamado de GSI (Globus Security Infrastructure) e responsável pela garantia de autenticação e comunicação seguras dentro da Grade. Assim quando o usuário faz a sua autenticação para realizar a sua requisição na Grade, o GSI cria uma representação do usuário conhecida como User Proxy que a partir deste momento fará todas as autenticações posteriores nos diferentes recursos da Grade computacional.

2.7.2.4 Serviços de Transferência de arquivos

É conhecido como GASS (Global Access to Secondary Storage) e possui alto desempenho de transferência de arquivos.

Já o GridFTP é uma extensão do FTP adequada para as necessidades da computação em Grades e que também foi criado para resolver alguns problemas de interoperabilidade que haviam surgido em razão do GASS por este não ser suportado por algumas outras arquiteturas. E embora o GASS seja bem superior ao GridFTP, o segundo devido as suas facilidades o tornaram padrão para o serviço de transferência de arquivos no Globus.

2.8 Conquistas da plataforma e do projeto Globus

Sempre esteve à frente no paradigma de computação em grade, inovando em recursos e estabelecendo padrões. Sendo o principal referencial de produto na área. Algumas de suas conquistas de maior destaques são:

- Implementação mais abrangente em termos de serviços, escalonadores e protocolos;
- O mais pesquisado, estando atualmente na sua 4 versão;
- Implementação de uma Grade para ambientes realmente heterogêneos, sendo permitida a entrada na Grade de até mesmo estações de trabalhos Windows, para a execução de aplicações em Java. Assim aplicações Java podem ser submetidos para qualquer nó de uma Grade Globus composta de estações Linux e Windows.

2.9 Experiência com o Globus 3

O Globus 3 por ser uma solução de serviços independentes, adaptáveis e configuráveis possui uma considerável complexidade e instabilidade. Requer um entendimento aprofundado por parte dos administradores sobre o funcionamento destes serviços e protocolos utilizados, além de exigir bastante conhecimento prévio em sistemas Linux para facilitar sua instalação e manuseio. Somam-se a isso a carente documentação que a plataforma possui.

Por estas razões o Globus 3 é bastante desanimador para quem gostaria de começar a estudá-lo mas não tem uma larga experiência nos itens mencionados acima. É bastante

comum encontramos em listas de discussões na internet vários relatos de pessoas que após dezenas de tentativas desistiram definitivamente de instalar a plataforma. Sendo que a instalação é apenas uma das etapas para sua utilização.

Sempre que conseguimos instalar o Globus com sucesso em determinada distribuição Linux, vem imediatamente a vontade de verificar se esta instalação está funcionando corretamente e se somos capazes de submeter uma aplicação simples apenas para verificar o seu funcionamento. É exatamente aí que entra a proposta deste trabalho, através de um protótipo simples que tem por objetivo configurar e submeter uma aplicação na grade em determinado contexto para verificar o seu funcionamento.

3 MODELO CONCEITUAL

Neste capítulo será apresentado o modelo conceitual que foi seguido para o desenvolvimento da aplicação.

A primeira coisa que foi definida antes da realização da aplicação foi a definição de todas as suas funcionalidades. Estas funcionalidades encontram-se representadas no diagramas de casos de uso da figura 3.1.



Figura 3.1: Diagrama de caso de uso

Após a definição destas funcionalidades, partiu-se para a elaboração da arquitetura da aplicação. Esta arquitetura foi dividida em duas partes maiores, que são: a parte independente da plataforma Globus e a parte dependente do Globus.

Dentro da parte independente da plataforma Globus temos a seguinte classe: InterfaceCliente.

Já na parte dependente temos: CadastrarAplicacao, SubmetAplicação, TransferênciaArquivos e AutenticarGrade.

Podemos ver na figura 3.2 a representação desta arquitetura.

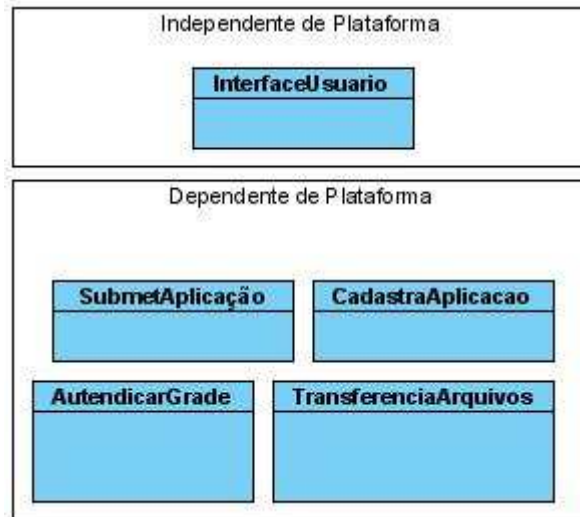


Figura 3.2: Arquitetura da aplicação

Classes que fazem uso da API Globus: AutenticadordaGrade, SubmeteAplicação, TransferênciaArquivos.

Classes independentes do Globus: InterfaceCliente

Na figura 3.3 podemos ver o diagrama de classes do projeto juntamente com o relacionamento destas classes. Das classes apresentadas no diagrama algumas são adaptações de classes que já existem na API do Globus, outras fazem uso apenas de algumas funcionalidade destas classes e as demais são independentes desta API.

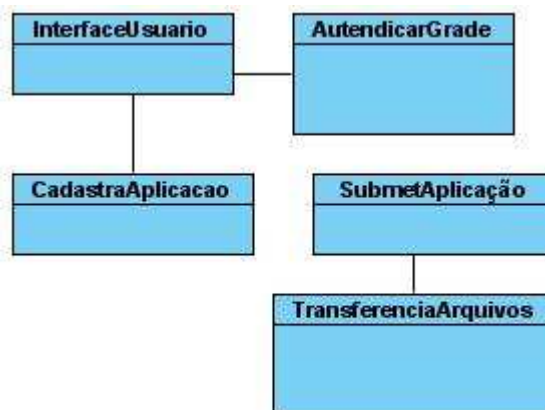


Figura 3.3: Diagrama de Classes

Cabe lembrar aqui que por razões de simplicidade não esta sendo considerado o controle de concorrência que é de responsabilidade do escalonador, realizado na plataforma Globus por um serviço chamado de Master Fork Managed Job Factory Service. Esta função poderá ser considerada num trabalho futuro.

3.1 Ferramentas

Para a implementação do protótipo foram utilizadas APIs do Globus na linguagem Java. Como linguagem de desenvolvimento foi utilizado o Java Standard Edition (J2SE), na versão 1.4, juntamente com o ambiente de desenvolvimento NetBeans 3.6.

Nas estação de trabalho para o desenvolvimento foram utilizados SuSe 9.2 e Mandriva 10.1 como sistema operacional.

4 EXECUÇÃO DO PROTÓTIPO

4.1 Introdução

Neste capítulo propõe-se a apresentação do funcionamento do protótipo, ou seja a sua execução. Nele será apresentado todo o seu funcionamento. Para isto é apresentado cada tela da interface gráfica do protótipo que aparece para o usuário juntamente com a sua finalidade e função, e as principais ações que o usuário pode realizar. Podemos ver na figura 4.1 o fluxo de execução do protótipo.

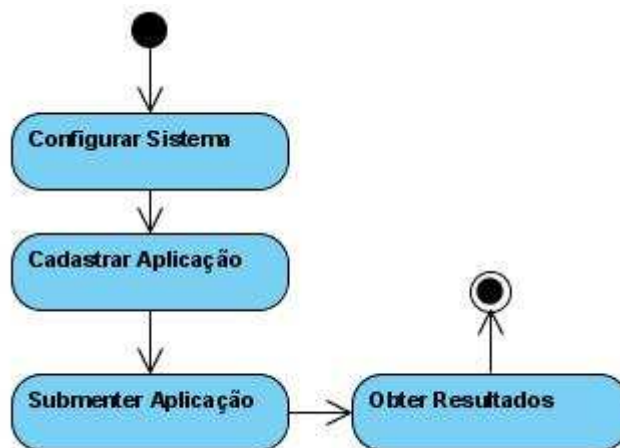


Figura 4.1: Fluxo de execução do protótipo

4.2 Contexto

Consideramos que o usuário tenha bons conhecimentos de programação, e seja um administrador avançado do sistemas Windows mas não tem familiaridade com comandos em modo texto no Linux (Shell).

Possui conhecimentos iniciais nos aspectos teóricos referentes ao Globus 3, tais como sua arquitetura, principais serviços, conceitos de arquivos RSL etc, todo este conhecimento teórico pode ser obtido facilmente lendo a documentação oficial do Globus 3.

Vamos partir da idéia que o Globus acabou de ser instalado por este usuário em uma máquina utilizando-se da documentação oficial do mesmo.

Não ocorreram problemas durante a instalação e esta foi finalizada com sucesso. Imediatamente após a instalação o usuário deseja submeter uma aplicação qualquer neste ambiente para verificar se tudo esta funcionando corretamente.

Também consideramos que toda a infraestrutura esta no mesmo domínio e tem um único servidor com os seguintes serviços: Index Server, Resource Finder Service, Trust Information Server Provider e Certificate Authority (CA). Como podemos ver na figura 4.2.

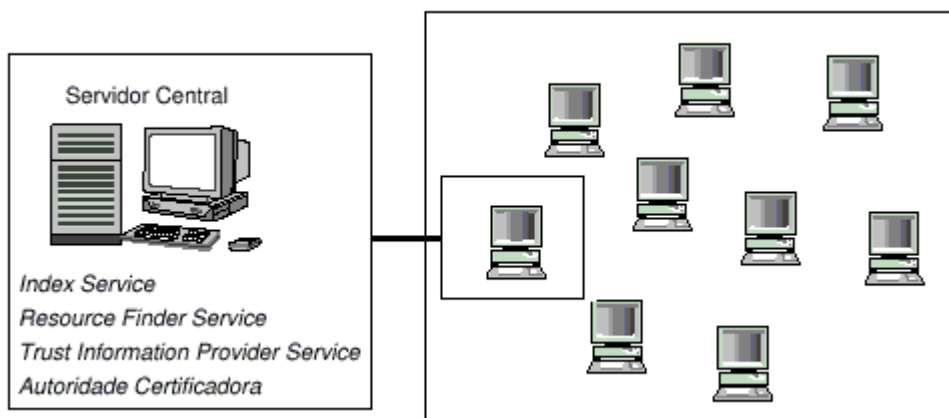


Figura 4.2: Cenário de execução

4.3 Execução

De posse do arquivo cliente_teste.jar em seu computador, para a execução do protótipo basta apenas executá-lo no computador em questão. Isto poderá ser feito basicamente de duas maneiras; com um clique duplo no arquivo ou através de linha de comando. Imediatamente após a execução do arquivo a seguinte janela de interface gráfica será apresentada ao usuário. Como vemos na figura 4.3.

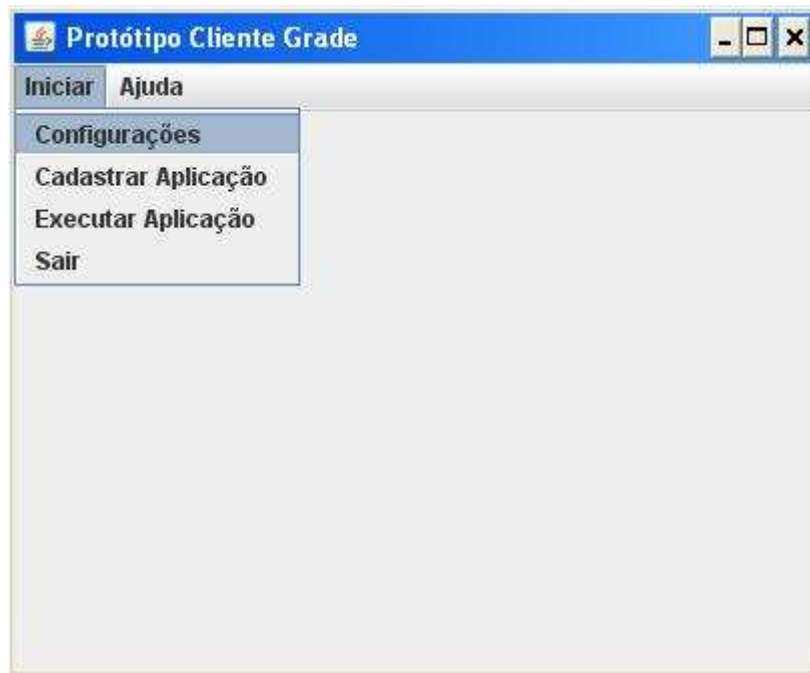


Figura 4.3: Janela principal

Para que o usuário consiga submeter uma aplicação a grade, ele tem que realizar antes a configuração do ambiente. Esta etapa é bastante deficiente em termos de documentação e costuma tomar muito tempo dos administradores que tem um primeiro contato com o Globus. Esta etapa não precisa seguir a ordem imposta pelo protótipo, uma vez que um especialista no Globus poderá realizá-la de várias maneiras, mas aqui se optou por esta ordem específica por parecer a mais lógica e fácil de entender.

O protótipo encontra-se dividido em duas etapas maiores: Configuração, Cadastrar Aplicação e Execução.

4.4 Configuração

Aqui é preciso definir alguns parâmetros para a instalação do Globus, estes parâmetros são: URL do servidor de serviços, nomes de serviços, e-mail do CA e diretórios com a localização dos serviços e certificados de segurança.

Todas estas configurações que serão utilizadas a seguir, são as configurações padrão do Globus 3. Realizadas através da interface gráfica do protótipo e em uma seqüência pré-estabelecida. Estas configurações poderiam ser feitas, por um administrador especialista no Globus através de linhas de comando no Linux (Shell) não necessariamente nesta seqüência.

De todos os recursos disponíveis em uma Grade de um determinado domínio temos que ter um que funcione como servidor central. Aqui temos que informar o endereço do computador do domínio que será o servidor e será responsável por vários serviços dentro da Grade, tais como: Index Server, Resource Finder Service, Trust Information Server Provider e Certificate Authority (CA).

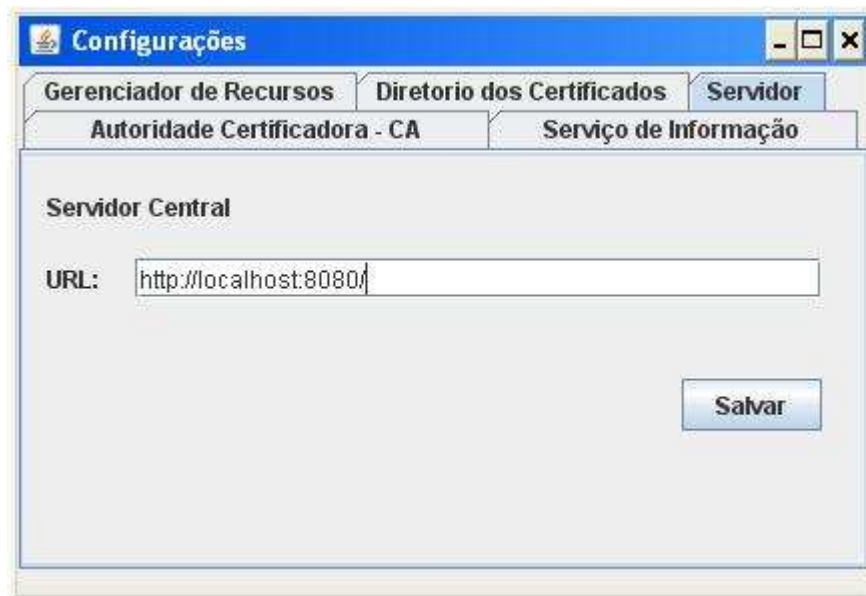


Figura 4.4: Definição do Servidor

Aqui na figura 4.5 devemos que indicar a localização do Serviço de Informação que é responsável por prover mecanismos para a descoberta e monitoração de serviços na Grade e dos recursos utilizados na infra-estrutura.



Figura 4.5: Definição do Serviço de Informação

Já na tela apresentada na figura 4.6 devemos informar o diretório onde se encontra o Resource Manager Service: responsável pela gerenciamento de alocação de recursos na infraestrutura da Grade.

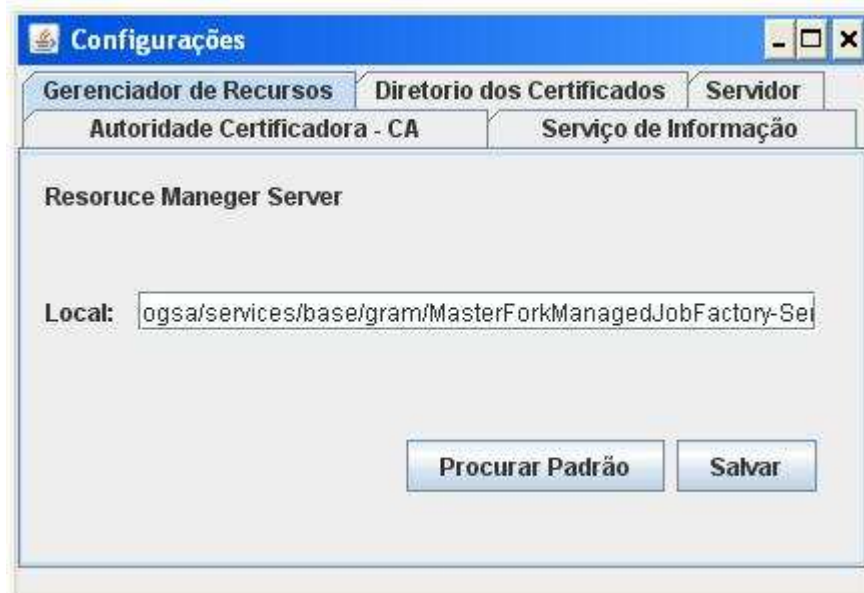


Figura 4.6: Resource Manager Server

Em autoridade Certificadora– CA (Figura 4.7) devemos informar o email da autoridade certificadora para o qual o usuário envia um email solicitando um Proxy na Grade. Este Proxy será a sua representação na Grade e esta relacionado com os mecanismo de autenticação e segurança da plataforma. Este email é recebido pela autoridade certificadora e assinado. Depois é devolvido para o usuário.



Figura 4.7: Email da Autoridade Certificadora

Requisição do certificado de segurança

Aqui o usuário informa o diretório onde se encontram os Certificados de segurança que foram assinados e devolvidos pela autoridade certificadora.

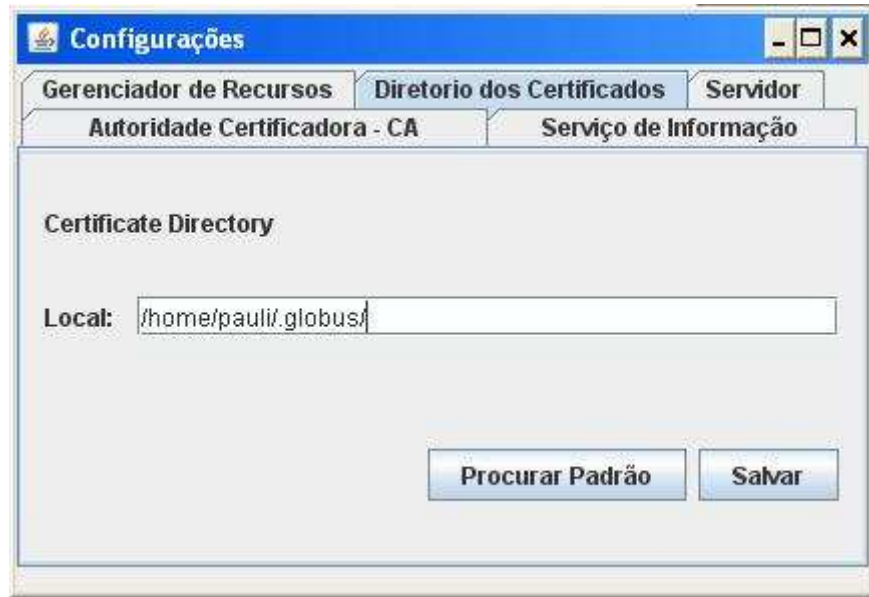


Figura 4.8: Diretório dos Certificados de Segurança

Com esta etapa encerra-se a parte de configuração. Assim através de passos bem definidos o usuário realizou a configuração da instalação do Globus em sua máquina.

4.5 Cadastro da aplicação

O primeiro passo para a submissão de aplicações na Grade é o cadastro da aplicação, onde são fornecidos todos os parâmetros para a execução da aplicação na Grade, tais como: Sistema Operacional, Arquitetura, Memória Mínima, Espaço Mínimo no HD, etc.

Podemos ver a tela de Cadastro da Aplicação representado na figura 4.9.

Cadastrar Aplicação

CadastroAplicação: _____

Executável:

Arquivos:

Comando de Execução:

Paramétros da Aplicação: _____

Sistema Operacional:

Arquitetura:

Processador:

HD:

Figura 4.9: Cadastrar Aplicação

No campo Executável devemos informar o arquivo que deverá ser executado na Grade. Neste exemplo trata-se de uma classe Java (um arquivo .Class) contendo um programa de cálculo matemático.

No item Comando de execução é definida a linha de comando que devera ser executada nos nós remotos da Grade. Já em Arquivos, deverão ser indicados os arquivos (APIs, arquivos de configuração, DLLs, etc) que devem ser transferidos para os nós remotos juntamente com a aplicação para sua correta execução no ambiente de Grade.

Para se testar a execução de aplicação na grade foi utilizado como teste uma aplicação que estava disponível em Sotomayor, B. (2004). trata-se de uma classe Java compilada, um arquivo .class, que pode ser executado em qualquer máquina Java. Essa aplicação necessita de alguns arquivos adicionais (APIs) que deverão ser enviados juntamente com a aplicação para a sua execução no nó remoto. Eles estão no arquivos CalculatorTestAPI.jar que deverá ser informado a sua localização no campo Arquivos.

4.6 Executar Aplicação

O último passo é mostrado na figura 4.10, onde temos a janela Executar Aplicação.

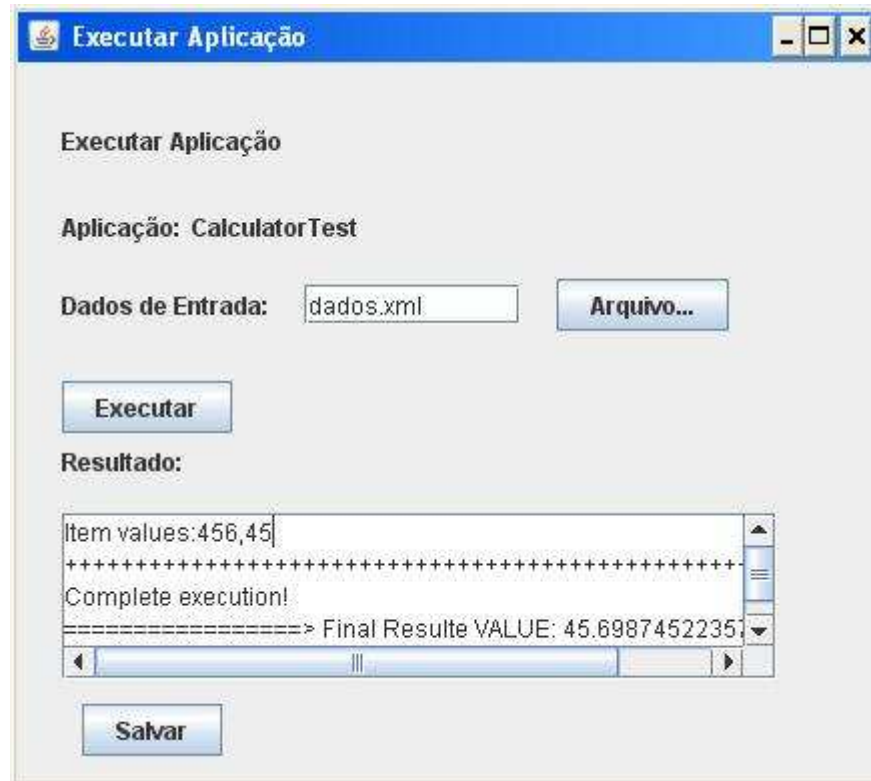


Figura 4.10: Janela Executar Aplicação

No campo Dados de entrada devemos informar a localização de um arquivo no formato xml com os dados de entrada da aplicação.

O próximo passo é clicar no botão Executar e esperar o resultado do processamento ser apresentado no campo Resultado, podendo se salvo através do botão Savar Resultado.

5 RESULTADO, DIFICULDADES E TRABALHOS FUTUROS.

Como já foi mencionado vários problemas e dificuldades foram encontradas durante a fase de instalação e configuração do Globus e da utilização do mesmo. O Globus apresenta uma série de problemas de instabilidade e uma enorme quantidade de bibliotecas e APIs que devem estar previamente instalados no computador, sendo que estas APIs dependem muito da distribuição Linux que se está utilizando. Sendo que o Globus parece ter total falta de compatibilidade com algumas distribuições Linux como por exemplo o Ubuntu e Mandraque, este fato foi descoberto a duras penas depois de se perder bastante tempo tentando instalá-lo nestas plataformas.

Muitos destes problemas não possuem nenhuma documentação e foram resolvidos com a ajuda de pessoas através de listas de discussões e tutoriais espalhados na internet, e de colegas que possuíam mais experiência na plataforma.

Já a implementação, embora tenha bastante APIs disponíveis na linguagem Java, existe pouca documentação e exemplos de como usá-las, sendo que muitas funcionalidades foram descobertas através de várias tentativas e de muita persistência.

O resultado final se mostrou satisfatório para os objetivos propostos que são ainda bastante limitantes se comparado com as diversas possibilidades do ambiente. Mas permite ao usuário uma alternativa para o acesso ao ambiente além de um ponto de partida para o aprendizado de todas as demais funcionalidades do Globus.

Uma série de outras funcionalidades podem se agregadas ao protótipo e que podem gerar melhorias e trabalhos futuros. Como exemplo podemos citar a busca por recursos, monitoração de recursos na grande, transferência de arquivos através do GridFTP, etc.

6 CONSIDERAÇÕES FINAIS

Este trabalho propôs uma pequena solução em forma de aplicação executável que tem por finalidade tornar o mais transparente possível a configuração básica e a execução de uma aplicação em um plataforma de computação em Grade a fim de testar o funcionamento desta plataforma após a sua instalação.

Tudo é feito através de uma interface gráfica que torna mais fácil para administradores que estão começando a estudar a plataforma e que também não possuem conhecimentos avançados do sistema operacional Linux.

Apesar das dificuldades para se dominar de maneira satisfatória a plataforma Globus, enfrentando diversos problemas de instabilidade, falta de documentação, comportamentos diferentes em versões diferentes do mesmo sistema operacional, etc, o protótipo funciona no contexto descrito, apesar de possuir uma série de instabilidades.

Sua usabilidade é extremamente simples e o usuário só tem que lidar com uma interface gráfica, seguir os passos apresentados ao invés de ter que realizar todos estes passos em enormes linhas de comando Linux sem qualquer seqüência previa, tarefa não muito apreciada para quem esta usando a plataforma pela primeira vez.

REFERÊNCIAS

Baker, M. e Buyya, R. (1988). **Cluster computing: The commodity supercomputing**. *Journal of Software - Practice & Experience*. v. 1, n. 1.

Berstis, V. e Ferreira, L. (2002). **Fundamentals of Grid Computing** - IBM RedPaper.

Brooke, J. e Garwood, K. (2003). **Interoperability of grid resource descriptions**. Em **GGF9 Semantic Grid Workshop**, Chicago, USA.

CoG (2005). **Commodity Grid Kits**. <http://www.cogkit.org/>, último acesso em : Maio de 2007.

Condor (2005). **The Condor Project: High Throughput Computing**. <http://www.cs.wisc.edu/condor/>, último acesso em: Abril de 2007.

DANTAS, Mario. **Computação Distribuída de alto desempenho: Redes, clusters e grids computacionais**. Axcel Books, 2005.

Foster, I. e Kesselman, C. (2001). **Globus: A metacomputing infrastructure toolkit**. *The International Journal of Supercomputer Applications and High Performance Computing*. v. 11, n. 2, p. 115–128.

Foster, I. e Kesselman, C., editors (1999). **The grid: blueprint for a new computing infrastructure**. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Foster, I. e Kesselman, C., editors (2004). **The grid 2: blueprint for a new computing infrastructure**. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Globus (2005). **Globus Alliance**. <http://www.globus.org>, último acesso em: Março de 2008.

Grid, A. (2005). **Access Grid Project: a global community**. <http://www.accessgrid.org/>, último acesso em : Novembro de 2007.

Rocha, J., Domingues, M., Callado, A., Souto, E., Silvestre, G., Kamienski, C., e Sadok, D.

(2004). **Peer-to-Peer: Computação colaborativa na internet**. Em 22 o Simpósio Brasileiro de Redes de Computadores.

SETI@home (2005). **The Search for Extraterrestrial Intelligence**.
<http://setiathome.ssl.berkeley.edu/>, último acesso em : Abril de 2005.

Sotomayor, B. (2004). **The Globus Toolkit 3 Programmer's Tutorial**.
<http://gdp.globus.org/gt3-tutorial/>, último acesso em : Abril de 2008.

[17] TANENBAUM, Andrew S. **Distributed System: Principles and Paradigms**. 1. st. edition. Prentice Hall, 2002.