Moacir Alves de Campos Junior

**Arquitetura de computação em grade aplicada a
saúde: Um estudo de caso em bioinformática para
oncologia**

Dissertação apresentada à Escola Politécnica da
Universidade de São Paulo para a obtenção do
Título de Mestre em Engenharia Elétrica.

São Paulo
2008

Moacir Alves de Campos Junior

**Arquitetura de computação em grade aplicada a
saúde: Um estudo de caso em bioinformática para
oncologia**

Dissertação apresentada à Escola Politécnica da
Universidade de São Paulo para a obtenção do
Título de Mestre em Engenharia Elétrica.

Área de Concentração: Sistemas Eletrônicos

Orientador: Prof. Dr. Marcelo Knörich Zuffo

São Paulo

2008

*Dedico esse trabalho a meus pais e avós e a todos aqueles
que me ajudaram a chegar onde cheguei, pois sem vocês
nada seria possível.*

Agradecimentos

Agradeço aos meus pais, Moacir e Diva, a minha irmã Tatiana, a minha namorada Thays e a minha sogra Alzira pelo carinho, dedicação, amor e incentivo.

Agradeço aos meus amigos pessoais Luiz F. Carvalho, Fernando O. Moraiz, Adão Cunha, Shirley Aparecida pelo incentivo.

Agradeço a coordenação da Escola Politécnica da USP e do Laboratório de Sistemas Integráveis pelo apoio oferecido.

Agradeço ao Prof. Dr. Marcelo Knörich Zuffo, pelas orientações, amizade e oportunidade de trabalharmos em conjunto.

Agradeço ao Prof. Dr. Sergio Takeo Kofuji e Profa. Dra. Liria M. Sato pelos ensinamentos disseminados que foram fundamentais para a realização deste trabalho.

Agradeço o Dr. André Nebel, Prof. Dr. Vicente Odone Filho, Profa. Dra. Chong Ae Kin e a toda a equipe do ICR-FMUSP, pelo apoio e colaboração.

Agradeço a Consultora Anne Picorone pela amizade e pelas dicas sobre o mercado da Tecnologia da Informação.

Agradeço a administração da Escola Politécnica e do Laboratório de Sistemas Integráveis por trabalharem para proporcionar as melhores condições possíveis para a execução deste e de muitos outros trabalhos.

Agradeço o apoio da Rede Nacional de Ensino e Pesquisa pela disponibilização da sua infra-estrutura para a realização de validações experimentais.

Agradeço aos amigos de trabalho: Francisco Fernandes, Higor Alves, Thiago Petersen, Vitor Pavani, Marcelo Arbore, Marcia Kondo, Maryana Alegro, Emerson Moretto, Henrique Calazans, Ilana Souza, Marcio Almeida, Felipe, Claudinei Sanches, Eduardo, Rodrigo Alves, Rodrigo Vaez, Vinicius, Maria Francesca Neglia, Ana Alcantara, Fabio Kung, André Rosa, Nadia Gobato, Walesca Parros, Sandra Segalo, Daniel Savoy, Jannaina, Marcio Hatano, Celina Kukue, Edvaldo, Danilo, Airtton Guelfi, Natanael Santos, Cícero da Conceição e Silvio Soriano, pelas colaborações e incentivos para realização deste

trabalho.

Ofereço especiais agradecimentos ao amigo Msc. Adilson Hira, gerente do núcleo de Telessaúde Laboratório de Sistemas Integráveis por ter acreditado na minha capacidade, pelas dicas, sugestões, ensinamentos e pela amizade construída durante os anos de trabalho conjunto.

Financiamento

O Trabalho aqui relatado foi possível graças ao apoio, patrocínio e financiamento das pesquisas e projetos do LSI-EPUSP pela FINEP - Financiadora de Estudos e Pesquisas do MCT, Ministério de Ciências e Tecnologias, Projetos ONCONET Fase II (Processo LSI-TEC/ONCONET2 N° 01.04.1010.00) e Teleonco Fase II (Processo LSI-EPUSP/ TELEONCO2 N°01.06.0145.00).

Resumo

Atualmente o câncer é a terceira causa de óbitos em ambos os sexos e em todas as faixas de idade. O Ministério da Saúde relata que a alta complexidade da atenção oncológica demanda alto custo financeiro e alta tecnologia. Cada vez mais, as atividades de pesquisa, prevenção, diagnóstico e tratamento do câncer são realizadas sobre um domínio multidisciplinar envolvendo medicina, química, genética, bioinformática e engenharia. Estas atividades são responsáveis por gerar e manipular grandes quantidade de dados que são geograficamente distribuídos entre as unidades de tratamento deste país. Estes dados necessitam ser integrados e processados, a sua utilização é empregada para melhorar a gestão e viabilizar colaborações de pesquisa entre as instituições do setor. O processamento destas informações demandaria grande capacidade computacional. Particularmente, verificamos que a bioinformática manipula uma grande quantidade de dados, e tem cada vez mais contribuído para a pesquisa e no diagnóstico de várias doenças, principalmente do câncer. A proposta desta dissertação consiste na pesquisa e desenvolvimento de uma infra-estrutura computacional baseada na tecnologia de grades computacionais para atender as demandas de alto processamento computacional, relacionadas com as informações do setor da saúde, em específico para bioinformática em oncologia. A aplicação da tecnologia de grade computacional no cenário da saúde brasileira é desafiadora do ponto de vista da pesquisa e desenvolvimento, porque esta tecnologia aborda a complexidade de integração e processamento de dados utilizando compartilhamento colaborativo de recursos computacionais sobre um ambiente multi-institucional geograficamente distribuído. A validação deste trabalho consistiu na aplicação de alguns dos procedimentos que envolvem a bioinformática em oncologia, em especial a identificação de similaridade genética, através de uma implementação experimental da arquitetura proposta. Os resultados obtidos a partir dos testes de validação nos permitiram avaliar os atendimentos dos requisitos do ambiente relacionados com as necessidades do setor de saúde em oncologia, assim oferecendo indicações e direções técnicas para o emprego da tecnologia de grades computacionais no apoio de diversas atividades relacionadas ao setor da saúde.

Abstract

Cancer is currently the third cause of death in both sexes and among all age ranges. The Brazilian Ministry of Health reports that the high complexity of oncology care demands high financial costs and high technology. The activities of cancer research, prevention, diagnosis and treatment are increasingly becoming a multidisciplinary environment, including medicine, chemistry, genetic, bioinformatics and engineering. Those activities are responsible for the generation and management of huge quantities of data, which are geographically distributed among the treatment units on this country. This data needs to be integrated and processed, and its use is for the improvement of the cancer management and to enable research exchanges among institutions in the field. The processing of this information requires a large computational capacity. Particularly, we verify that bioinformatics handles a huge amount of data and has increasingly contributed for the research and diagnosis of several diseases, especially cancer. The proposal of this thesis is the research and development of a computational infrastructure based on grid computational technology to serve the requirements of high computational processing, related to the data from health care systems, in particular to bioinformatics in oncology. The use of grid computing technology in the brazilian health scenario is challenging in terms of research and development, since this technology faces the complexity of data integration and processing using collaborative sharing of computational resources onto a multi-institutional geographically distributed environment. The validation of this work was in the application of some of the procedures that involve bioinformatics in oncology, especially the identification of genetic similarity, using an experimental implementation of the proposed architecture. The results obtained from the validation tests enabled us to evaluate the meeting of environment requirements, related with the needs of health sector in oncology, thus offering indications and technical directions to the use of grid computing technology in support of several activities related with the health sector.

Lista de Figuras

1.1	Crescimento do poder computacional agregado dos supercomputadores classificados na lista TOP500 de 2003 a 2006 (TOP500, 2007).	2
2.1	Telemedicina, Telessaúde, interseção entre áreas de conhecimento. Adaptado de (BASIC, 2001).	12
2.2	Método de preparação de um <i>microarray</i> (SEBASTIANI et al., 2003; CRISTO, 2003).	18
2.3	Imagem de uma lâmina de <i>microarray</i>	19
2.4	Etapas de processamento de um experimento de <i>microarray</i>	20
3.1	Taxonomia Sistemas de Grade (KRAUTER; BUYYA; MAHESWARAN, 2002).	27
3.2	Exemplo de ambiente de grade computacional trocando diferentes tipos de mensagens dentro de uma OV (Organização Virtual).	30
3.3	Descoberta de recursos em ambiente de grade computacional (DUAN; TALLEY; SEETHA, 2007).	32
3.4	Visão esquemática de uma arquitetura composta por escalonadores locais e um meta-escalonador (FERREIRA et al., 2003).	34
3.5	Acesso a dados no ambiente de grade computacional por meio de serviços Web.	35
3.6	Componentes do Globus Toolkit 4.0.X (GLOBUS..., 2007).	39
3.7	Visão das camadas da arquitetura GSI.	40
3.8	Arquitetura MDS apresentada em forma de ampulheta (SCHOPF et al., 2005).	43
3.9	Arquitetura básica de um projeto BOINC (RATTEI et al., 2006).	45
3.10	Arquitetura OurGrid (ANDRADE et al., 2005).	47

3.11	Arquitetura base de um aglomerado de computadores baseado no PBS. Adaptado de (HENDERSON, 1995)	51
3.12	Condor e Condor-G dispostos sobre um ambiente de grade computacional baseado no Globus Toolkit (THAIN; TANNENBAUM; LIVNY, 2005). . .	52
3.13	Arquitetura do meta-escalador GridWay (GLOBUSALIANCE, 2007). . .	53
4.1	Arquitetura OncoGrid expressada sobre diagrama de camadas funcionais. .	60
4.2	Componentes fundamentais do OncoGrid	63
4.3	Arquitetura do ambiente de processamento do OncoGrid representada no modelo de camadas funcionais.	67
4.4	Submissão de uma tarefa independente no OncoGrid.	69
4.5	Submissão de um lote de tarefa no OncoGrid.	70
4.6	Submissão de uma tarefa paralela no OncoGrid.	72
4.7	Distância euclidiana de um gene expressados em dois experimentos de <i>mi-</i> <i>croarray</i>	75
4.8	Modo de operação do módulo ISGG recebendo os dados de saída do módulo ISGL.	77
4.9	Exemplo de distribuição da aplicação no ambiente OncoGrid	79
5.1	Implementação inicial da arquitetura OncoGrid apresentada sob o modelo de camadas funcionais.	84
5.2	Processo de autenticação estabelecido no OncoGrid (ALVES et al., 2008). .	86
5.3	Representação esquemática do sistema de informação do OncoGrid. . . .	89
5.4	Página do WebMDS para a visualização das informações do ambiente de processamento OncoGrid.	90
5.5	Fluxograma do módulo de Identificação de Semelhanças Genéticas Local - ISGL.	98
5.6	Fluxograma do módulo de Identificação de Semelhanças Genéticas Global - ISGG.	99
6.1	Gráfico representando o tempo total de processamento das cinquenta e duas tarefas ISGL nos testes realizados com 1, 3, 5, e 7 processadores.	104

6.2	Gráfico representando o tempo médio por tarefa de utilização de CPU e de comunicações de rede nos testes com 1, 3, 5 e 7 CPUs.	105
6.3	Gráfico representando o crescimento de desempenho obtido na execução da bateria de testes um contra todos.	106
6.4	Gráfico representando as médias das distâncias euclidianas de alguns dos cruzamentos testados.	111
6.5	Gráfico representando as médias das distância euclidianas obtidas dos experimentos classificados em primeiro e quinto lugar.	112

Lista de Tabelas

1.1	Distribuição dos supercomputadores mais poderosos entres as regiões geográficas do mundo em 2006 (TOP500, 2007).	2
2.1	Densidade populacional e despesas com saúde por habitante.	13
3.1	Comparação dos itens de flexibilidade das ferramentas para implementação de grades computacionais	48
3.2	Comparação dos itens referentes à modularização das ferramentas para implementação de grades computacionais	48
3.3	Comparação dos itens referentes à escalabilidade das ferramentas para implementação de grades computacionais	48
4.1	Cálculo das distâncias euclidianas entre a expressão genética original $\mathbf{o}$ e as comparadas $\mathbf{c}_1$ e $\mathbf{c}_2$	76
4.2	Matriz resposta.	77
4.3	Matriz classificação.	78
4.4	Matriz pontuação.	78
5.1	Componentes físicos aplicados na implementação piloto do OncoGrid . . .	85
5.2	Componentes físicos aplicados no ambiente de processamento implementado.	88
5.3	Configurações aplicadas no GridWay para o estabelecimento da integração do meta-escalonador com as ferramentas de informações, gerência de execução e transporte de dados do OncoGrid.	91
5.4	Informações sobre os recursos de processamento identificadas pelo GridWay.	92
5.5	Parâmetros de escalonamento para configuração das políticas relacionadas com as tarefas.	93
5.6	Parâmetros de escalonamento para configuração das políticas relacionadas com os recursos.	93

5.7	Informações sobre os processos computacionais (tarefas) em execução no meta-escalador GridWay.	94
6.1	Classificação geral do teste comparando a expressão lc8n006rex contra todas as outras.	107
6.2	Processo de criação das tarefas para execução do teste de todos contra todos.	108
6.3	Dados referente a execução do teste todos contra todos (1 recurso de processamento, 2 escalonador de tarefa, 3 estação de usuário).	109
6.4	Resultados dos teste de avaliação todos contra todos, apresentando os dez primeiros classificados.	110
B.1	Parâmetros para configuração de ambiente para submissão de tarefas no GridWay.	120

Lista de Acrônimos

AC Autoridade Certificadora

ANL *Argonne National Laboratory*

BoT *Bag of Tasks*

CFM Conselho Federal de Medicina

CLARA Cooperação Latino Americana de Redes Avançadas

CPU *Central Processing Unit*

DARPA *Defense Advanced Research Projects Agency*

DNA Ácido Desoxirribonucleico

DRS *Data Replication Service*

FINEP Financiadora de Estudos e Projetos

FLOPS *Floating point Operations Per Second*

FTP *File Transfer Protocol*

GRAM *Grid Resource Allocation Manager*

GSi *Grid Security Infrastructure*

GT *Globus Toolkit*

HT *Hyperthreading*

HTML *HyperText Markup Language*

HTTPS *HyperText Transfer Protocol Secure*

I-Way *Information Wide Area Year*

INCA Instituto Nacional do Câncer

LNH Linfoma Não-Hodking

MDS *Monitoring and Discovery System*

MIAME *Minimum Information About a Microarray Experiment*

MPI *Message Passing Interface*

OGF *Open Grid Forum*

OGSA *Open Grid Services Architecture*

OGSA-DAI *OGSA-Data Access and Integration*

OV *Organização Virtual*

PBS *Portable Batch System*

PCR *Polymerase Chain Reaction*

PTT *Pontos de Troca de Tráfego*

QoS *Quality of Service*

RFT *Reliable File Transfer*

RLS *Replica Location Service*

RNA *Ácido Ribonucleico*

RNP *Rede Nacional de Ensino e Pesquisa*

SAML *Security Assertion Markup Language*

SGBD *Sistema Gerenciador de Base de Dados*

SOAP *Simple Object Access Protocol*

TB *TeraByte*

TI *Tecnologia da Informação*

TIS *Tecnologia da Informação em Saúde*

TLS *Transport Layer Security*

WS-GRAM *Web Services-GRAM*

WSDL *Web Service Definition Language*

WSRF *Web Services Resource Framework*

XML *eXtensible Markup Language*

Sumário

1	Introdução	1
1.1	Motivação e Relevância	3
1.1.1	Tecnologia da Informação no Setor de Saúde	4
1.1.2	Bioinformática Aplicada à Prática Clínica	5
1.1.3	Grades Computacionais Aplicada ao Setor de Saúde e a Bioinformática	6
1.2	Objetivos	7
1.2.1	Objetivos Específicos	7
1.3	Metodologia	8
1.4	Trabalhos Correlatos	8
1.5	Organização da Dissertação	10
2	Tópicos em Telessaúde e Bioinformática	11
2.1	Telessaúde e Telemedicina	11
2.1.1	Realidade dos sistemas de Telessaúde no Brasil	12
2.2	Bioinformática	14
2.2.1	Desafios da Bioinformática	15
2.2.2	Conceituação	15
2.2.3	Tecnologia de <i>Microarrays</i>	17
2.2.4	Obtenção dos Dados do <i>Microarrays</i>	19
2.3	Resumo do Capítulo	21
3	Estado da Arte em Grades Computacionais	23

3.1	Definindo Sistemas de Grade Computacional sobre uma Abordagem Histórica	23
3.2	Taxonomia dos Sistemas de Grades Computacionais	26
3.3	Componentização das Grades Computacionais	28
3.3.1	Interface do Usuário	29
3.3.2	Segurança	29
3.3.3	Agentes de Busca de Informações	31
3.3.4	Escalonadores	32
3.3.5	Gerenciamento de Dados	34
3.3.6	Gerenciamento de Tarefas e de Recursos	36
3.4	OGSA - Arquitetura Aberta de Serviços de Grade	36
3.5	Ferramentas para Implementação de Grades Computacionais	37
3.5.1	Globus Toolkit 4.0.X	38
3.5.2	BOINC - <i>Berkeley Open Infrastructure for Network Computing</i> . .	44
3.5.3	OurGrid	46
3.5.4	Comparativo entre as Ferramentas Abordadas	47
3.6	Ferramentas para Distribuição de Processamento em Grade Computacional	49
3.6.1	Portable Batch System - PBS	49
3.6.2	Condor e Condor-G	50
3.6.3	GridWay	52
3.7	Projetos Similares em Grade para Saúde	54
3.7.1	Genegrid: Grade Computacional Aplicada a Bioinformática	55
3.7.2	caBIG: Grade de Informações Biomédicas em Câncer	55
3.8	Resumo do Capítulo	55
4	Concepção do Ambiente de Processamento para Grade Computacional em Oncologia e Bioinformática	57
4.1	OncoGrid - Grade Computacional em Oncologia	57

4.1.1	Requisitos do OncoGrid	58
4.1.2	Arquitetura OncoGrid	59
4.2	Proposta do Ambiente de Processamento Distribuído em Grade Aplicado ao OncoGrid	63
4.2.1	Requisitos do Ambiente de Processamento do OncoGrid	64
4.2.2	Arquitetura do Ambiente de Processamento do OncoGrid	64
4.2.3	Métodos de Distribuição de Tarefas	68
4.3	Concepção de Aplicação de Bioinformática para Validação do Ambiente . .	73
4.3.1	Conjunto de Dados Utilizado para Validação	74
4.3.2	Módulo Identificador de Semelhança Genética Local	74
4.3.3	Módulo Identificador de Semelhança Genética Global	76
4.3.4	Modelo de Distribuição do Algoritmo	79
4.4	Resumo do Capítulo	80
5	Avaliação Experimental	82
5.1	Implementação do Ambiente OncoGrid	82
5.1.1	Autenticação no OncoGrid	85
5.2	Arquitetura de Processamento do OncoGrid	87
5.2.1	Recursos Físicos	87
5.2.2	Sistema de Informações	88
5.2.3	Gerenciamento de Execução de Tarefas	90
5.2.4	Movimentação de Dados	91
5.2.5	Sistema de Meta-Escalonamento	91
5.2.6	Tolerância a Falhas no Processamento de Tarefas	95
5.3	Desenvolvimento da Aplicação de Bioinformática para Validação do Ambiente	96
5.3.1	Implementação do Módulo Identificador de Semelhanças Genéticas Local	96

5.3.2	Implementação do Módulo Identificador de Semelhanças Genéticas	
	Global	98
5.4	Resumo do Capítulo	100
6	Análises de Resultados	102
6.1	Resultados do Teste Um Contra Todos	102
6.2	Resultados do Teste Todos Contra Todos	106
6.3	Análises e Discussões dos Resultados	110
6.4	Resumo do Capítulo	114
7	Conclusões, Trabalhos Futuros e Considerações Finais	115
7.1	Conclusões	115
7.2	Trabalhos Futuros	116
7.3	Contribuições Oferecidas	118
	Apêndice A – Artigos Publicados	119
	Apêndice B – Parâmetros de Configuração de Ambiente de Execução no GridWay	120
	Referências Bibliográficas	121

1 *Introdução*

A partir da 1980, ocorreu um aumento significativo de pesquisas multidisciplinares e de trabalhos com abrangência multi-institucional. Dentre as áreas que se destacaram podemos citar: geofísica, engenharia e bioinformática. Muitas das atividades atribuídas a estas pesquisas envolviam mais de um centro de pesquisa ou local de atuação, assim caracterizando a distribuição geográfica destas atividades.

É considerável a quantidade de dados gerados em tais atividades de pesquisa, que, em muitos casos, ultrapassa a oferta de recursos computacionais convencionais. Neste caso podemos citar os trabalhos para mapeamento do genoma humano, que demandou aproximadamente 10.000 horas de CPU (*Central Processing Unit*) (BADER, 2004) e os sistemas de simulação para previsão atmosférica (FREITAS et al., 2007; LONGO et al., 2007).

Para suprir as necessidades computacionais requisitadas por aplicações complexas, a tecnologia da informação vem, ao longo do tempo, propondo alternativas tecnológicas baseadas em protocolos e ferramentas para trabalhos colaborativos que apresentavam características de larga distribuição de dados e grande demanda de processamento computacional (aglomerados de computadores, computadores vetoriais, grades computacionais) e padrões para interoperabilidade de informação e comunicação no nível de aplicação (SOA, XML, Web Services, sistemas de segurança interoperáveis) (BERMAN; HEY; FOX, 2003).

Podemos constatar a crescente demanda por plataformas computacionais de alto desempenho observando o aumento da capacidade de processamento dos supercomputadores mais poderosos do mundo. Os dados sobre estes equipamentos são publicados semestralmente pelo portal *Top 500 Supercomputing*. A figura 1.1 apresenta um gráfico gerado com os dados das listas publicadas no período de junho de 2003 a novembro de 2006. A curva representa a capacidade computacional agregada dos supercomputadores classificados.

Figura 1.1: Crescimento do poder computacional agregado dos supercomputadores classificados na lista TOP500 de 2003 a 2006 (TOP500, 2007).

Para complementar as informações expostas na curva apresentada na figura 1.1, a tabela 1.1 indica a distribuição das plataformas de processamento de alto desempenho entre as regiões geográficas no mundo, segundo a lista publicada no mês novembro de 2006.

Região Geográfica	Super Computadores	Concentração de processamento	Capacidade em Gflops	Quantidade de Processadores
América do Norte	317	65,44%	2.308.307	697.546
Ásia Oriental	56	11,29%	398.116	100.663
Europa Ocidental	38	9,31%	328.566	95.864
Norte da Europa	37	6,32%	223.025	57.746
Sul da Europa	15	3,71%	130.772	27.824
Centro Sul da Ásia	10	0,97%	34.162	10.908
Ásia Ocidental	8	0,78%	27.477	9.128
Austrália e Nova Zelândia	5	0,69%	24.426	5.888
Sudeste da Ásia	5	0,52%	18.449	4.722
América do Sul	4	0,39%	13.668	3.780
Europa Oriental	2	0,28%	9.705	2.460
Sul da África	2	0,16%	5.696	3.072
América Central	1	0,14%	5.090	1.360
Total	500	100%	3.527.459	1.020.961

Tabela 1.1: Distribuição dos supercomputadores mais poderosos entre as regiões geográficas do mundo em 2006 (TOP500, 2007).

É possível verificar que a grande concentração do poder de processamento se encontra na América do Norte, devido à forte economia e à grande concentração de pólos tecnológicos. A América do Sul assume o décimo lugar entre as treze regiões geográficas que compõem a tabela.

O Brasil possui todos os supercomputadores da América do Sul citados na lista de

novembro de 2006, sendo que três deles são de propriedade da Petrobrás - Petróleo Brasileiro S.A., dedicados a executar aplicações geofísicas para identificação de petróleo no subsolo, sendo um deles da Universidade de São Paulo, aplicado a diversos segmentos da pesquisa científica (TOP500, 2007).

Extrapolando as observações realizadas sobre os dados apresentados, podemos constatar que existe uma forte demanda mundial por plataformas computacionais de alto desempenho.

Da mesma forma que os demais áreas, onde a pesquisa é uma atividade fundamental, o setor de saúde apresenta um amplo conjunto de necessidades relacionadas à Tecnologia da Informação (TI), sendo responsável por uma grande produção de dados (registros médicos textuais, informações genéticas, imagens radiológicas, vídeos, etc.), estando dispersos entre as unidades de saúde. A extensão territorial do Brasil implica numa dispersa distribuição de dados deste setor.

Os dados gerados pela saúde são utilizadas para diferentes fins, como: atenção clínica, pesquisa e gestão do setor, e formam uma rica base de dados. Atualmente é discutido na comunidade científica a gestão integralizada das informações sobre a atenção e pesquisa da saúde (CUNHA, 2002; BILYKH, 2003; ALEGRO et al., 2006; BARROS, 2006; HIRA, 2005).

1.1 Motivação e Relevância

Atualmente o câncer é a terceira causa de óbitos em ambos os sexos e em todas as faixas de idade. Estima-se a ocorrência de 232 mil novos casos em homens e 235 mil em mulheres em 2008 (NORONHA et al., 2008). Na população infanto-juvenil, o câncer é a terceira causa de morte por doença entre 1 e 14 anos e a quarta considerando todas as outras causas (RODRIGUES; CAMARGO, 2003; HIRA, 2005).

A taxa de incidência do câncer infantil tem crescido em torno de 1% ao ano. Este crescimento tem sido inversamente proporcional ao crescimento da taxa de mortalidade e estima-se que a taxa de cura global esteja em torno de 85%. É esperado que, em 2010, um em cada 250 adultos seja um sobrevivente de câncer na infância (RODRIGUES; CAMARGO, 2003).

O Ministério da Saúde estabelece que a atenção a doenças de alta complexidade, que é o caso da Oncologia, é “ O Conjunto de procedimentos que, no contexto do SUS,

envolve alta tecnologia e alto custo, objetivando propiciar à população acesso a serviços qualificados, integrando-os demais níveis de atenção à Saúde (atenção básica e de média complexidade) ”(SAUDE, 2008).

Os procedimentos envolvidos na pesquisa, diagnóstico, prevenção e tratamento do câncer são atividades multidisciplinares, envolvendo conhecimentos de diversas áreas como medicina, engenharia, química, genética, bioinformática, telessaúde, dentre outras.

A intersecção entre as áreas da engenharia de computação, oncologia, bioinformática e a telessaúde é abordada no estudo deste trabalho. Por este fato, consideramos que a utilização de redes de computadores e a disponibilização de sistemas computacionais adequados podem colaborar significativamente para a gestão, processamento e análise das informações da saúde que envolvem a bioinformática aplicada na pesquisa e atenção à oncologia. Devido às características referentes à distribuição geográfica do país, o setor de saúde nacional apresenta larga distribuição de dados. Acreditamos que para a gestão das informações deste setor, se faz necessária a aplicação de uma infra-estrutura computacional adequada para suportar procedimentos de larga escala computacional.

1.1.1 Tecnologia da Informação no Setor de Saúde

O setor de tecnologia da informação (TI) possui um segmento denominado Tecnologia da Informação em Saúde-TIS, que se dedica a atender de forma customizada as necessidades de gestão da informação do setor de saúde.

Dentre os desafios das TIS que envolvem procedimentos computacionais de larga escala computacional, destacamos três grupos, classificados segundo as suas aplicabilidades:

- Interoperabilidade de informações: Estes desafios consistem na consolidação de dados de unidades de saúde geograficamente dispersas, possibilitando eficiência no manuseio de informações através da coleta constante de dados, redução no trâmite de papéis, eficiência na troca de informações e na realização de estatísticas para avaliação de indicadores de tratamento e da gestão do setor;
- Colaboração entre equipes a distância: Estes desafios consistem na realização de conferências de trabalhos coletivos, colaboração em diagnósticos, tratamentos a distância, treinamento de equipes, assim reduzindo os impactos causados pela distância, beneficiando regiões remotas ou menos favorecidas de recursos financeiros;
- Processamento de alto desempenho: Estes desafios consistem na realização de proce-

dimentos computacionais em extensos setores de dados ou tarefas de processamento computacionalmente complexo, como: operações estatísticas, mineração de dados, procedimentos complexos de integração de informações distribuídas, busca e identificação de padrões em base de informação multimídia (texto, imagens e vídeos), identificação de proteomas, seqüenciamentos genéticos, formação e manipulação de depósitos de dados.

A literatura científica apresenta uma tendência em promover alternativas destinadas ao auxílio da informatização e da proliferação de recursos tecnológicos no setor de saúde. Os objetivos destas iniciativas convergem em pontos comuns que são: consolidação de informações (ALEGRO et al., 2006; CAMPOS et al., 2006), aprimoramento dos métodos estatísticos (ALEGRO et al., 2006), promover a colaboração entre equipes (ROSA et al., 2006; SOUZA et al., 2006), promover métodos para o processamento de extensas massas de dados (BADER, 2004).

1.1.2 Bioinformática Aplicada à Prática Clínica

A bioinformática é considerada uma disciplina de pesquisa e desenvolvimento na área da tecnologia da informação em saúde (TIS). Esta disciplina combina conhecimentos de química, física, biologia, ciência da computação, informática e matemática/estatística para processar dados biológicos ou biomédicos (MUNIZ, 2003; GIBAS; JAMBECK, 2003).

Basicamente a bioinformática atua de duas formas, durante o manejo e apresentação das extensas massas de dados geradas em procedimentos da biologia molecular, e na análise destas informações por meios de algoritmos e procedimentos computacionais (MUNIZ, 2003; GIBAS; JAMBECK, 2003).

Além de outros benefícios, a bioinformática possibilita que os avanços da biologia molecular sejam utilizados na prática clínica. Uma das formas de conseguir esta integração é avaliando a expressão quantitativa dos genes expressos no paciente e comparar estes dados com indícios já conhecidos, como por exemplo, a comparação de casos clínicos que apresentem o mesmo diagnóstico e tenham forte semelhança genética.

Um dos métodos para adquirir a expressão genética de um determinado indivíduo é a tecnologia de *microarray*. Esta tecnologia nos permite adquirir a expressão de milhares de genes de um indivíduo simultaneamente. As informações genéticas obtidas a partir deste método permite realizar diferentes tipos de avaliações, como, a identificação dos genes que são expressos em um grupo de pacientes com o mesmo diagnóstico, buscando descobrir

quais são os genes que interferem na ocorrência desta determinada doença.

As informações dos *microarrays* permitem identificar a pré disposição de um indivíduo de apresentar uma doença como câncer antes mesmo da sua incidência, assim possibilitando assumir atitudes preventivas. De outra forma, torna-se possível identificar os genes de um indivíduo que apresente uma determinada doença de natureza genética, com base nestas informação, personalizar o seu tratamento. Também podemos utilizar informação de tratamentos já realizados com pacientes de perfil genético semelhante buscando melhores práticas aplicadas (YANG; SUN, 2007).

1.1.3 Grades Computacionais Aplicada ao Setor de Saúde e a Bioinformática

As atividades de seqüenciamento do genoma e as tecnologias pós-genômicas como as dos *microarrays* estão gerando uma quantidade considerável de dados que necessitam ser gerenciados e processados, assim colocando a bioinformática à frente das disciplinas que requisitam alto poder computacional em ambiente de larga escala altamente colaborativo (JITHESH et al., 2005).

Aplicar esta tecnologia ao cenário da saúde brasileira é desafiador do ponto de vista da pesquisa e desenvolvimento. Esta afirmação se justifica pelo fato das grades computacionais estabelecerem interfaces comuns para compartilhamento de serviços e recursos físicos e serem escaláveis em todos os níveis de sua estrutura, podendo ser estendida conforme as suas necessidades.

Dentro de um ambiente de grade computacional podem ser disponibilizados dinamicamente dados, recursos físicos, serviços e aplicativos, que podem ser acessados por um usuário de qualquer instituição que seja participante do ambiente.

O compartilhamento de dados em grade possibilita a consulta das informações de forma integrada, ampliando as bases de informação para pesquisa e para gestão do setor. Este fato pode elevar as chances de realizar novas descobertas e melhorar a qualidade da gestão do setor de saúde por meio de dados atualizados e abrangentes. O compartilhamento de recursos estende a capacidade de processamento de uma única instituições, possibilitando que processadores e equipamentos dos participantes da grade sejam utilizados de forma colaborativa para solucionar problemas computacionalmente complexos.

Os estudos e desenvolvimentos relacionados com a bioinformática que serão apresentados neste trabalho se encontram no domínio da intersecção entre a bioinformática a

telessaúde e a tecnologia de grades computacionais.

1.2 Objetivos

O objetivo deste trabalho é a pesquisa, desenvolvimento e avaliação de um ambiente de grade computacional concebendo uma infra-estrutura que agregue as capacidades computacionais de máquinas dispostas em redes de computadores mundialmente conectadas, buscando alcançar alto desempenho para suporte a aplicações do setor de saúde que apresentem alta complexidade computacional.

1.2.1 Objetivos Específicos

Neste trabalho consideramos a formação da base de conhecimento necessária para propor uma alternativa computacional que forneça alto grau de interoperabilidade e desempenho computacional, entre os diferentes sistemas de saúde em nível de infra-estrutura, operando em abrangência nacional, desta forma, otimizando os investimentos já concretizados e beneficiando as regiões menos favorecidas de infra-estrutura tecnológica, como é o caso de unidades de saúde nas regiões norte e nordeste do nosso país.

Os objetivos específicos deste trabalho se relacionam ao desenvolvimento e às avaliações realizadas neste trabalho:

- A consolidação dos conhecimentos necessária para desenvolvimento e implantações de ambientes de grades computacionais;
- A elaboração de um modelo para especificação e a especificação dos componentes para a implementação do ambiente de grade computacional;
- A implementação de um ambiente que permita realizar avaliações experimentais;
- O desenvolvimento de uma aplicação de bioinformática que utilize a tecnologia de *microarrays* para a identificação de similaridades genéticas entre grupos de indivíduos com o mesmo diagnóstico;
- A validação experimental do ambiente utilizando a aplicação desenvolvida.

1.3 Metodologia

A metodologia utilizada na elaboração deste trabalho consiste em:

- Levantamento bibliográfico do estado da arte das tecnologias de grade computacional e Bioinformática envolvidas neste trabalho;
- Proposta da arquitetura do ambiente de processamento em grade computacional baseado em modelo de camadas funcionais;
- Uso de protocolos e padrões abertos de informação visando o compartilhamento de dados e a interoperabilidade de aplicações;
- Utilização de *software* livre visando a redução do custo de implantação do sistema tornando-o viável economicamente;
- Consideração dos fatores relacionados como a escalabilidade da arquitetura possibilitando a sua expansão conforme se faça necessário, e das alternativas para proporcionar tolerância a falhar no ambiente, visando a alta disponibilidade da arquitetura.

1.4 Trabalhos Correlatos

Nesta seção são apresentados os trabalhos desenvolvidos pelo núcleo de Telessaúde do Laboratório de Sistemas Integráveis da Escola Politécnica da USP, dos quais participei de forma direta ou indireta, consistindo na motivação para a realização das pesquisas apresentadas nesta dissertação.

O projeto ONCONET é uma iniciativa apoiada pelo Governo Federal, por meio da FINEP (Financiadora de Estudos e Projetos do Governo Federal), que visa a aplicação da tecnologia da informação a favor do suporte a saúde. A execução deste projeto consistiu em diversas etapas de trabalho e subprojetos. A seguir são indicados os trabalhos que possuem alguma relação com esta dissertação:

- Portal de Serviços ONCONET: O trabalho desenvolvido por Hira (2005) apresenta uma proposta de um sistema de informações em saúde para gestão da atenção em oncologia pediátrica. Este projeto, utiliza tecnologias baseadas em software livre, é disponibilizado por meio de um portal de serviços na Web. O grande objetivo desta iniciativa é promover a homogeneização do tratamento de oncologia pediátrica

no território brasileiro, elevando os índices de cura das regiões menos favorecidas do Brasil (norte e nordeste). Este trabalho contou com a colaboração de toda a equipe do núcleo de Telessaúde do LSI;

- Rede Física ONCONET: A implantação da rede física ONCONET consistiu na interligação física entre as unidades hospitalares e centros de saúde com foco em oncologia de diversos estados brasileiros à Rede Nacional de Ensino e Pesquisa (RNP). Esta interligação possibilitou a qualidade de comunicação necessária para o estabelecimento da utilização dos serviços do Portal ONCONET e da colaboração entre equipes médicas por meio de vídeo conferência e de outros meios interativos. Até o momento foram interligadas dez instituições médicas e uma instituição de pesquisa com foco em ensino a distância;
- Cluster ONCONET Fase I: Foi desenvolvido pela equipe de infra-estrutura do núcleo de Telessaúde do LSI uma arquitetura de aglomerado de computadores para missão crítica, otimizado para execução de serviços disponibilizados pelo portal ONCONET. Este ambiente proporciona alta disponibilidade baseada na replicação de servidores, balanceamento de carga e escalabilidade, estando em serviço há mais de quatro anos.
- Cluster ONCONET Fase II: Foi desenvolvido um ambiente similar ao Cluster ONCONET Fase I, em parceria com o INCA (Instituto Nacional do Câncer), Intel do Brasil e Accept. Este ambiente foi otimizado para atender os serviços demandados na consolidação nacional dos registros de câncer.
- Integrador de Registro Hospitalar de Câncer: O integrador de registro hospitalar de câncer RHCNET foi uma aplicação desenvolvida dentro do projeto ONCONET em conjunto com o Instituto Nacional do Câncer (INCA) com a finalidade de integrar todos os registros de câncer do Brasil. Este é o principal sistema que é executado pelo Cluster ONCONET Fase II (ALEGRO et al., 2006).
- OncoGrid: O OncoGrid nasceu como um projeto de pesquisa dentro do núcleo de saúde digital, tendo como objetivo explorar tecnologicamente a aplicabilidade das arquiteturas de grades computacionais, a fim de empregar esta tecnologia em prol dos serviços de saúde nacional. Hoje é prevista a integração do Cluster Onconet Fase I com os Hospitais interligados a Rede ONCONET, a fim de disponibilizar serviços de integração e processamento de dados de forma transparente ao usuário. O desenvolvimento desta dissertação é especialmente motivado pelas pesquisas

realizadas no projeto OncoGrid (CAMPOS et al., 2006).

1.5 Organização da Dissertação

Esta dissertação está dividida em sete Capítulos, organizados como descrito abaixo:

O Capítulo 1 apresenta a introdução, motivação, objetivos e os trabalhos correlatos, fornecendo a visão geral e os motivos que levaram ao desenvolvimento das pesquisas envolvidas neste trabalho;

O Capítulo 2 apresenta os estados da arte sobre os assuntos relacionados com a disciplina de Bioinformática.

O Capítulo 3 apresenta os estados da arte sobre grades computacionais e possíveis ferramentas para distribuição de processamento neste tipo de infra-estrutura.

O Capítulo 4 apresenta a proposta do projeto OncoGrid, que é um modelo de uma infra-estrutura de grade computacional para suporte à oncologia. Neste trabalho abordamos como foco central os sistemas de processamento acoplados ao ambiente, a sua modularização e métodos de distribuição de processos. Também apresentamos a proposta da aplicação utilizada para validação do ambiente.

O Capítulo 5 apresenta o desenvolvimento desta pesquisa, na qual a implementação da arquitetura se dá na forma de um projeto piloto, permitindo que partir deste piloto, realizar a validação experimental da aplicação proposta e da própria arquitetura.

O Capítulo 6 apresenta os resultados obtidos a partir dos testes para validações e as análises e discussões pertinentes referentes aos resultados obtidos.

O Capítulo 7 apresenta as conclusões e os trabalhos futuros.

2 Tópicos em Telessaúde e Bioinformática

Neste capítulo será abordado o conjunto de conceitos que estão integrados na área da saúde que são relevantes para o desenvolvimento deste trabalho.

Na seção 2.1, abordamos os conceitos sobre telessaúde e telemedicina. A seção 2.2 abordamos alguns tópicos sobre a bioinformática.

2.1 Telessaúde e Telemedicina

O Conselho Federal de Medicina (CFM), na resolução número 1643 artigo primeiro, define a telemedicina como sendo: “o exercício da Medicina através da utilização de metodologias interativas de comunicação audiovisual e de dados, com o objetivo de assistência, educação e pesquisa em Saúde ” (CFM, 2007).

De acordo com a ATA (*American Telemedicine Association*), “a telemedicina consiste no uso de informação médica veiculada de um local para outro, por meio de comunicação eletrônica, visando à saúde e educação dos pacientes e do profissional médico, para assim melhorar a assistência a saúde” (ATA, 2005).

Bacic (2001) apresenta a definição de telemedicina como sendo: “o intercâmbio de informações para o suporte a prática médica à distância, por meio eletrônico, quando há necessidade por uma das partes”.

A Telemedicina, então, está sempre vinculada à idéia do uso de informação de forma eletrônica, objetivando serviços na saúde remotamente (HIRA, 2005).

O termo Telessaúde (*Telehealth*) é entendido com um escopo mais abrangente do atendimento à saúde (ATA, 2005; HIRA, 2005). Este termo abrange todos os serviços remotos dedicados à atenção à saúde.

Apresentamos na figura 2.1 o domínio de atuação de cada uma das áreas abordadas nesta seção. A telessaúde é a interseção entre a medicina, a tecnologia da informação e telecomunicação. A telemedicina é vista como um subconjunto da telessaúde.

Figura 2.1: Telemedicina, Telessaúde, interseção entre áreas de conhecimento. Adaptado de (BACIC, 2001).

2.1.1 Realidade dos sistemas de Telessaúde no Brasil

Quando abordamos a aplicação da Telessaúde em âmbito nacional, se faz necessário considerar fatores relacionados à geografia, distribuição de renda, oferta ou qualidade de serviço de saúde.

O Brasil possui extensões continentais. A distribuição dos serviços de saúde, assim como a qualidade dos serviços oferecidos é divergente entre as regiões. Este fato é atribuído à diferentes fatores, como: falta de profissionais especialistas nas localidades mais remotas do país; deficiência no aprimoramento dos profissionais de saúde de regiões remotas; divergências nos tratamentos devido a falta de protocolos de conduta médica; regiões menos favorecidas financeiramente tendenciam atendimentos de pior qualidade.

A tabela 2.1 apresenta as divergências territoriais, populacionais e os gastos realizados pela saúde com a população brasileira, segundo informações obtidas a partir do censo realizado em 2007 e de informações publicadas pelo Ministério da Saúde em 2004, refletindo a posição mais próxima da atual (SAUDE, 2008).

Região	População	Área em Km ²	habitantes/Km ²	R\$ / habitante
Norte	14.623.316	3.851.560	3,8	308,36
Nordeste	51.534.406	1.556.001	33,12	251,95
Sudeste	77.873.120	927.286	83,98	360,59
Sul	26.733.595	575.316	46,47	306,69
Centro-Oeste	13.222.854	1.604.852	8,24	325,62
Total	183.987.291	8.515.015	21,61	315,69

Tabela 2.1: Densidade populacional e despesas com saúde por habitante.

A partir da análise da tabela, podemos observar a baixa densidade populacional nas regiões Norte e Centro-Oeste, onde a distância geográfica dos centros de tratamento é um fator que prejudica o acesso a saúde. A região Nordeste apresenta o menor gasto por habitante. Nesta região, são encontradas famílias em situação de extrema pobreza e deficiência no saneamento básico e abastecimento de água, como é o caso do sertão nordestino.

Hira (2005), em sua dissertação de mestrado, sugere a discussão sobre alguns pontos, nos quais o emprego da telessaúde pode ser uma alternativa tecnológica que possibilite a elevação da qualidade dos serviços de saúde em regiões menos favorecidas.

- Divergências na qualidade do atendimento entre regiões do país;
- Deficiências de recursos humanos em regiões remotas;
- Migração de pacientes para tratamento em centros de excelência;
- Deficiência de ferramentas e métodos informatizados para auxílio à saúde;
- Ausência de métodos informatizados nos locais remotos;
- Necessidade de integração de diferentes tecnologias;
- Revisão de expansão segundo evolução tecnológica.

Observando os modelos de telessaúde aplicados aos países desenvolvidos, constatamos que eles são concebidos com base em tecnologias de alto custo, como comunicação dedicada, sistemas e soluções proprietárias. Devido a este fato, devemos realizar uma reflexão sobre o custo/benefício das soluções. Devido à indisponibilidade financeira no setor de

saúde brasileiro, devemos considerar a aplicação de técnicas que otimizem a utilização dos recursos já existentes, excluindo ou reduzindo a necessidade de grandes investimentos para sua concepção.

O emprego de redes de computadores mundiais ou continentais como a Internet, Internet2, RNP, Rede Clara, que constituem uma estrutura física já disponível para a sociedade, em conjunto com a aplicação de padrões abertos e tecnologias livres para o estabelecimento de sistemas de telessaúde, formam um modelo funcional e economicamente viável indicado para uso em países em desenvolvimento.

A seção seguinte descreve conceitos e aplicabilidades relacionados com bioinformática que são fundamentais para o trabalho em questão. A importância de integrar esses dois temas é atribuída ao fato de tornar os benefícios da bioinformática mais próximos da prática clínica por intermédio de serviços de telessaúde. Adicionalmente, devemos considerar que as implementações realizadas nesta dissertação contemplam a movimentação de dados médicos por meio das infra-estruturas de telecomunicações e o processamento dos mesmos, atividades que enquadram no domínio da telessaúde.

2.2 Bioinformática

A bioinformática é uma área multidisciplinar da ciência que utiliza recursos tecnológicos (computacionais, físicos, matemáticos, estatísticos) para organizar, analisar e disponibilizar informações referentes à biologia molecular, com a finalidade de promover informações para suportar novas descobertas sobre organismos vivos. Esta disciplina atua basicamente de duas formas, no desenvolvimento e implementação de ferramentas para o acesso, uso, manejo e visualização das informações e no desenvolvimento de novos algoritmos ou processos para análise de dados gênicos e relacionados (MUNIZ, 2003).

Existe a tendência do uso de informações relativas a biologia molecular, genética e informações cromossômicas para possibilitar a melhoria da qualidade de diagnósticos e o aprimoramento dos protocolos de tratamento em diversas patologias. Particularmente, diversos estudos em oncologia seguem esta tendência. Desta forma, pesquisas no campo da bioinformática têm se tornado bastante relevantes para proporcionar melhorias nos tratamentos do câncer.

Nas seções seguintes vamos abordar conceitos pertinentes à genética e alguns métodos empregados para estudos e trabalho com genes utilizados no campo da bioinformática.

2.2.1 Desafios da Bioinformática

As pesquisas da biologia na era da genética buscam uma compreensão de como os organismos vivos são formados, a partir da codificação genética. Descobrir a codificação genética dos organismos vivos, identificar genes desconhecidos pela análise de seqüências genética e modelar como uma corrente de aminoácidos se organiza em uma proteína funcional são tarefas que apresentam custo computacional alto, e necessitam na maioria dos casos, de integração multidisciplinar.

Hoje estão disponíveis ao público diversas bases genéticas. O GenBank é uma das mais conhecidas, sendo uma base de dados pública que armazena e disponibiliza a maioria das seqüências DNA, RNA, hemoglobinas e de proteínas conhecidas até o momento. A quantidade de informações genéticas inseridas nestas bases de dados cresce de forma exponencial. A tarefa de gerenciar esses dados requer alto nível de padronização e alto desempenho na realização de consultas (GIBAS; JAMBECK, 2003).

É conhecido que cada gene que compõe o genoma não é uma entidade independente. Os genes são integrados para formar os caminhos bioquímicos que, por sua vez, alimentam outros caminhos. As ligações bioquímicas são influenciadas por diferentes elementos, como: ambiente externo, interação patogênicas, etc. Consolidar informações cruzadas da bioquímica, genética e dos fatores externos, formando modelos concisos do funcionamento de organismos, e conseguir as predições sobre a evolução dos organismos são tarefas que requerem grande integração de conhecimentos de diferentes áreas (GIBAS; JAMBECK, 2003). Um exemplo desta atividade é o estudo correlacionando da expressão genética de um paciente com o seu histórico clínico, considerando fatores como o diferencial de tempo da evolução do caso clínico, metástases, hábitos. Estudos com estas características são aplicados para descobrir quais genes interferem em uma determinada doença, um forte indício de acerto é quando um determinado grupo de genes é expressado em um grupo de indivíduos com o mesmo diagnóstico de câncer e não é expressado em um grupo saudável.

Sinteticamente, a bioinformática é uma disciplina que reúne conhecimentos de outras áreas como matemática, estatística, biologia, bioquímica, física e engenharia. É nítido que o termo colaboração seja aplicado quando nos referimos a esta disciplina.

2.2.2 Conceituação

Abordamos nesta seção as definições básicas sobre os principais focos de estudo das disciplinas de biologia molecular, e por consequência da bioinformática, que estão relaci-

onados com este trabalho: DNA, RNA e os Genes.

A célula é a menor unidade com vida. O ciclo de vida deste organismo envolve uma extensa quantidade de moléculas variando de água, pequenas compostos orgânicos (por exemplo, ácidos graxos e açúcares) e macromoléculas (DNA, proteínas e polissacarídeos). As macromoléculas controlam a maioria das atividades celulares. Além de outras funcionalidades, o ácido desoxirribonucleico (DNA) armazena informações sobre as macromoléculas, permitindo a sua replicação (reprodução celular) de acordo com as especificações iniciais da célula (LING; LEE, 2004).

O DNA é composto por cadeias químicas construídas com blocos chamados nucleotídeos. Os nucleotídeos são formados por um grupo fosfato, uma molécula de desoxirribose e uma das quatro diferentes bases azotadas (nitrogenadas), conhecidas pelas letras iniciais Guanina, Citosina, Adenina, Timina. A codificação contida no DNA é dada pela organização das seqüências das bases azotadas (LING; LEE, 2004).

O RNA (ácido ribonucléico) é localizado no núcleo das células e no citoplasma. Uma de suas funções é participar da síntese celular durante o processo de multiplicação celular. Existem basicamente duas categorias de RNA's: mRNA (RNA mensageiro), que participa no processo de decodificação dos genes e RNA funcional, que é o responsável pela síntese protéica, que traduz a informação do mRNA em proteína (LING; LEE, 2004).

Os genes são as unidades da seqüência do DNA que controlam os caracteres hereditários identificáveis de um organismo. Um gene pode ser definido como um segmento de DNA que especifica um RNA funcional. O conjunto total dos genes de uma célula de um indivíduo é conhecido como genoma.

O genoma define a construção genética de uma célula e, por conseqüência, do organismo. Hoje, são conhecidas seqüências completas do genoma de diversas espécies, incluindo várias bactérias, mamíferos e humanos. (LING; LEE, 2004).

Um dos propósitos da biologia molecular e da bioinformática aplicada à prática clínica é a identificação genes causadores de diferentes fenômenos biológicos, como a evolução de espécies, anomalias genéticas e patologias (CRISTO, 2003).

Atualmente a sociedade dispõe de recursos tecnológicos da bioinformática e da genética que possibilitam o estudo de grande parte dos genes de uma pessoa. Este processo consiste na avaliação quantitativa da extração da expressão genética de um conjunto pré-determinado de genes.

Existem diferentes técnicas para a extração da expressão gênica. As mais conhecidas

e utilizadas são a transcrição reversa da reação polimérica em cadeia (RT-PCR - *Reverse Transcription Polymerase Chain Reaction*) e os *microarrays*.

A técnica RT-PCR é um processo baseado na adição de enzimas (*Taq polimerase*) a uma amostra de fragmentos de cDNA (ácido desoxirribonucleico complementar). A amostra é submetida a variações de temperatura em estágios pré-definidos; em cada ciclo de resfriamento, o DNA se desestrutura quebrando-se em duas fitas e a enzima as complementa formando uma cópia da original. A cada repetição deste processo, a cadeia gênica é replicada. Para o estudo de cada gene, é necessário um reagente específico. Desta forma, a extração da expressão gênica é pontual, ou seja, a cada experimento teremos a expressão de um único gene. Esta técnica pode se tornar imprecisa por variações no procedimento experimental (variação de tempo e temperatura). Por este motivo sempre são utilizadas amostras de controle.

A tecnologia de *microarray* permite a extração da expressão gênica de milhares de genes em paralelo. Por ser o processo que originou os dados utilizados neste trabalho, dedicamos a seção 2.2.3 para abordar especificamente este tema.

2.2.3 Tecnologia de *Microarrays*

Existem diferentes tipos de *microarrays*. Nós vamos abordar a tecnologia de *cDNA-microarray* pelo método de hibridização. Os dados experimentais utilizados nos testes de validação realizados neste trabalho foram originados a partir de lâminas deste tipo.

No processo de extração da expressão gênica por *microarray*, é utilizada uma lâmina que possui a geometria de uma matriz. Esta lâmina pode ser comprada ou construída com clones de DNA de um banco de clones. Os *spots* se referem a cada posição desta matriz onde é fixada uma amostra de cDNA (*probe*) de um gene conhecido. Isso não indica que em todas as posições da lâmina haverá uma amostras de um gene. Normalmente alguns *spots* são utilizados para controle e outras a amostra de combinações de genes e de outras estruturas como proteínas.

Para ativar a lâmina, é necessária a extração do mRNA do indivíduo através de uma amostra de tecido por biopsia. Quando a quantidade de mRNA extraído não é suficiente para a realização do experimento, ela é amplificada por técnicas similares às utilizadas no RT-PCR (CRISTO, 2003; LING; LEE, 2004).

O primeiro passo do processo consiste na adição de mRNA do ser examinado nos *spots* da lâmina feita por um robô que consegue a precisão micrométrica desejada. São

dispostas na lâmina duas amostras de mRNA que foram extraídas dos tecidos a serem analisados. Normalmente são utilizadas uma amostra de tecido saudável e uma de tecido doente. Essas amostras são contaminadas com a adição de dois reagentes fluorescentes Cy3 (verde) e Cy5 (vermelho). Com a junção do cDNA e do mRNA depositados na lâmina ocorre o pareamento das moléculas. Quanto maior o número de pareamentos em um determinado *spot*, maior é a fluorescência emitida, por consequência, o gene de referência do *spot* é mais expressivo no organismo estudo.

A figura 2.2 apresenta esquematicamente o procedimento de hibridização de uma lâmina, até a obtenção da imagem com os tons de fluorescência.

Figura 2.2: Método de preparação de um *microarray* (SEBASTIANI et al., 2003; CRISTO, 2003).

As indicações de intensidade de fluorescência são cruzadas sobre dois canais de *laser*. Um dos canais apresenta a reação de fluorescência com Cy3 e o outro do Cy5. Um exemplo de imagem resultante deste processo é apresentado na figura 2.3. As amplitudes das ondas utilizadas para obtenção destas imagens são 532nm para o canal 1 e 635nm para o canal 2 indicando o Cy3 e o Cy5 respectivamente.

Após a obtenção da imagem da lâmina, o próximo passo é a submissão da imagem para processamento computacional. Os processos são apresentados em síntese na seção 2.2.4

Figura 2.3: Imagem de uma lâmina de *microarray*.

2.2.4 Obtenção dos Dados do *Microarrays*

Após a realização do experimento é obtida a imagem digital da lâmina por meio de um *scanner* digital de alta definição.

A imagem resultante passa por uma série de processos de análise computacional para corrigir erros e imperfeições na imagem, normalização, identificação da localização dos *spots*, extração de ruídos, entre outros (PASANEN et al., 2003).

Figura 2.4: Etapas de processamento de um experimento de *microarray*

A figura 2.4 apresenta as principais etapas do processamento da imagem da lâmina, até o ponto da conversão das informações da imagem em dados numéricos.

Os dados numéricos resultantes da análise da imagem formam um arquivo em formato texto com os dados brutos conhecido com *Grid File*. O Grid File é posteriormente convertido em um arquivo tabulado com informações sobre o experimento e os níveis de expressão em ambos os dois canais.

Existem diversos aplicativos que realizam a conversão das informação gráficas em dados numéricos, como o ScanAlyze e o NETAFFX por exemplo.

Particularmente o ScanAlyze realiza a leitura da imagem identificando os valores expressos em cada *spot* por meio de processamento de imagens. As informações adquiridas a partir deste processo são inseridas em um arquivo tabulado em formato DAT, que pode ser utilizado em planilhas eletrônicas de cálculos ou por outros aplicativos que consigam compreender este formato. Dentro do arquivo DAT encontramos o resultado da expressão de cada um dos *spots* da lâmina analisada e adicionalmente são incluídos diversos dados sobre a análise da imagem, por exemplo, o endereço do *spot* na matriz, informações sobre os cálculos de bordas, mediana e a quantidade de pixels contidos nos *spots* (EISEN, 1999).

O NETAFFX é uma ferramenta Web disponibilizada pela empresa AFFYMETRIX. Este software possibilita a leitura de lâminas de experimentos, inserção de anotações sobre os genes, marcadores e nucleotídeos. Este software disponibiliza a saída de dados, em formato tabulado contendo as informações que o usuário requisitou ou diretamente para um arquivo de planilha eletrônica.

Após a extração dos dados da imagem, os próximos passos são relacionados com a avaliação das informações em busca de respostas biológicas. Podemos citar os seguintes exemplos deste tipo de estudo: a identificação dos genes de referência em um conjunto de expressões genéticas, os estudos pontuais sobre a ação da expressão de algum gene em relação a doença abordada e a identificação do perfil genético de uma determinada doença, entre outros.

Atualmente são dedicados grandes esforços para a padronização dos dados de experimentos de *microarrays*. Um dos resultados destes trabalhos é o protocolo MIAME (*Minimum Information About a Microarray Experiment*), que indica quais informações deverão ser apresentadas nas experimentações envolvendo *microarrays* (BRAZMA et al., 2001).

2.3 Resumo do Capítulo

A telemedicina consiste no suporte à prática médica a distância, por meio eletrônico, quando há necessidade por uma das partes. A telessaúde é um termo mais abrangente, que se relaciona a todos os serviços remotos dedicados à atenção à saúde incluindo a telemedicina.

O Brasil, é um país de extensões continentais, e em suas regiões existem diferenças na arrecadação de renda, que se refletem nos investimentos realizados na saúde, repercutindo na qualidade da oferta dos serviços de saúde. Tais fatores nos levam a pensar sobre como as aplicações de telessaúde podem contribuir para a equalização da qualidade de serviços para atenção ao câncer.

A bioinformática, por sua vez, é a área que explora os recursos tecnológicos para organizar, analisar e disponibilizar as informações referentes à biologia molecular, que estuda os eventos celulares.

Dentro do conjunto das aplicações da bioinformática, o estudo sistemático da expressão genética de um indivíduo é bastante relevante para as pesquisas em oncologia, porque

o câncer é uma doença genética. A tecnologia de *microarray* permite o estudo da expressão genética de um indivíduo em larga escala, ou seja, possibilita a análise de milhares de genes de um indivíduo em um único experimento. As informações obtidas a partir deste tipo de experimento podem ser estudadas de forma individual ou em grupo. Parte dos estudos que correlacionam extensos grupos de expressões genéticas necessitam de plataformas computacionais de alto desempenho para tornar possível o seu processamento e a integração dos seus dados.

3 Estado da Arte em Grades Computacionais

Neste capítulo serão apresentados os conceitos necessários para a compreensão deste trabalho que estão relacionados com as tecnologias de grades computacionais. Serão abordadas nas seções seguintes as definições, taxonomias, padrões e ferramentas que são atualmente empregadas para estabelecer ambientes computacionais baseados nesta tecnologia.

3.1 Definindo Sistemas de Grade Computacional sobre uma Abordagem Histórica

Durante a década de 80, os aplicativos para computação paralela eram focados em prover mecanismos para comunicação entre processadores. As aplicações de computação distribuída destinadas a alto desempenho proporcionaram um extenso potencial para realizar compartilhamento e distribuição de memória (BERMAN; HEY; FOX, 2003).

As primeiras manifestações de sistemas de computação em grade consistiam em aplicações que utilizavam paradigmas da computação paralela e distribuída aplicada a alta desempenho para distribuir segmentos de processamento entre computadores distribuídos geograficamente (BERMAN; HEY; FOX, 2003; DEFANTI et al., 1996).

Ainda nos anos 80, a comunidade científica se deparou com o desafio de trabalhar em pesquisas multidisciplinares, nas quais eram envolvidas grandes quantidades de informações a serem transportadas, processadas, disponibilizadas e apresentadas (BERMAN; HEY; FOX, 2003). A chave deste desafio consistia em promover ferramentas para infraestrutura computacional de larga escala, a fim de possibilitar novas descobertas científicas (BERMAN; HEY; FOX, 2003).

Os estudos para solucionar os desafios do gerenciamento da informação em atividades

de pesquisas multidisciplinares proporcionaram o estabelecimento de um conjunto de soluções, modelos e métodos para trabalhos colaborativos a distância, impactando na forma como a ciência conduzia seus processos de descobertas até o momento. Este fato chegou a gerar novas disciplinas, tal como a bioinformática.

No início da década de 1990, incentivados pelo uso de redes de computadores mundiais, computação de larga escala e o crescimento das massas de informações geradas, surgiu uma série de projetos de pesquisa que, de alguma forma, faziam uso de computação distribuída. Para suprir as necessidades a comunidade de desenvolvedores na área da supercomputação direcionou seus estudos para compor novas ferramentas para computação de larga escala. Tais ferramentas eram destinadas a agregar computadores fornecendo uma imagem única do ambiente ao usuário (ABBAS, 2004).

Em 1995, durante a conferência de supercomputação promovida pelo IEEE e pela ACM, foram utilizadas onze redes de computadores de alta velocidade para conectar equipamentos por dezessete localidades distribuídas geograficamente, para demonstrar a criação de um supercomputador geograficamente distribuído. Este projeto recebeu o nome de I-Way (*Information Wide Area Year*) e foi conduzido por Ian Foster, sendo a primeira implementação do modelo atual de grades computacionais (ABBAS, 2004; DEFANTI et al., 1996).

O sucesso do I-WAY incentivou de forma direta o desenvolvimento de mais de setenta aplicações, e muitos dos novos conceitos de grade computacional surgiram e foram testados dentro deste ambiente (ABBAS, 2004). A repercussão positiva deste projeto fez com que diversas instituições governamentais norte-americanas incentivassem a continuidade desta iniciativa. Em outubro do ano de 1996 a DARPA (*Defense Advanced Research Projects Agency*) fomentou um projeto para pesquisa e desenvolvimento de ferramentas para computação distribuída de larga escala. Tal projeto foi conduzido por Ian Foster (ANL) e Carl Kesselman (University of Southern California) e denominado GLOBUS (DEFANTI et al., 1996; BERMAN; HEY; FOX, 2003).

No ano de 1998, durante a SC98 (Conferência de Supercomputação), foi estabelecido um fórum com o objetivo de discutir as definições dos padrões e as melhores práticas no uso de grades computacionais. Em paralelo surgiram outras iniciativas similares. No ano de 2000 o *European Grid Forum* (eGrid) e o *AsiaPacific Grid Forum* se uniram formando a *Global Grid Forum*. Em meados do ano de 2006 ocorreu a fusão do *Global Grid Forum* e o *Enterprise Grid Alliance* (EGA) formando o *Open Grid Forum* (OGF) (OGF, 2007).

Hoje o OGF é composto por uma comunidade que conduz trabalhos para a padroni-

zação da computação em grade, integrando profissionais da pesquisa, indústria, iniciativa privada e governamental, envolvendo mais de quatrocentas organizações distribuídas em mais de 50 países (OGF, 2007).

É possível observar que a computação em grade sofreu evoluções que repercutiram em sua definição. Em 1998 Foster e Kesselman publicaram no livro “The Grid: Blueprint for a New Computing Infrastructure” uma definição primária para os ambientes de grade computacional (FOSTER, 2002):

“Uma grade computacional é uma infra-estrutura de *hardware* e *software* capaz de prover dependabilidade, consistência e pervasibilidade de baixo custo para computação de alto desempenho” .

A definição apresentada em 1998 ficou obsoleta rapidamente. Em 2000 a definição foi refinada incorporando os participantes envolvidos na disponibilização e utilização de recursos. Assim é assumido até os dias atuais (FOSTER; KESSELMAN; TUECKE, 2001; FOSTER, 2002):

“Uma grade computacional é concernida do compartilhamento coordenado de recursos dedicada à solução de problemas de forma dinâmica por organizações virtuais multi institucionais.”

É importante citar que existem divergências entre as definições sobre computação em grade. As variantes das definições estão relacionadas normalmente com a aplicação do ambiente. Dentre as estudadas e apresentadas, uma em especial apresenta melhor adequação ao nosso estudo, citada em Addas (ABBAS, 2004):

“Uma grade computacional permite que organizações virtuais compartilhem recursos distribuídos geograficamente em busca de um propósito comum, assumindo a ausência de uma localização centralizada, controle centralizado, onisciência e a existência de um relacionamento de confiança.”

Podemos completar esta última definição com a afirmação citada por Foster (FOSTER, 2002).

“A Finalidade de um sistema de grade é promover acesso direto a computadores, programas, dados e outros recursos, quando necessários, aplicados à solução de problemas complexos de forma colaborativa, sendo indicado para o uso no campo da indústria, ciência e engenharia.”

3.2 Taxonomia dos Sistemas de Grades Computacionais

É comum encontrar na literatura, duas diferentes classificações dos sistemas de grades computacionais. No meio comercial, a classificação é feita considerando a distribuição geográfica: departamentais (*departmental grids*), empresariais (*enterprise grids*), extra-empresariais (*extraprise grids*), globais (*global grids*). De outra forma, no campo da pesquisa científica, são classificadas segundo as suas aplicabilidades, sendo identificadas três categorias neste contexto: grades computacionais (*computational grids*), grades de dados (*data grids*) e grades de serviços (*services grids*). Neste trabalho, vamos assumir a segunda forma de classificação (ABBAS, 2004; BERMAN; HEY; FOX, 2003).

- A categoria de grades computacionais é composta por sistemas destinados a agregar recursos de processamento de máquinas distribuídas geograficamente. A sua função é executar tarefas de alta complexidade computacional de forma distribuída sobre um conjunto de máquinas distantes geograficamente. Dentro deste nicho temos dois modelos de agregação de recursos: o primeiro se dedica a executar tarefas de forma paralela em múltiplas máquinas; o segundo se dedica a distribuir tarefas em múltiplas máquinas que não necessitam de relação entre os dados durante o seu processamento (*bag-of-task applications*) ;
- A categoria de grades de dados é composta por sistemas destinados a promover uma infra-estrutura para sintetizar novas informações a partir de repositórios de dados, bibliotecas digitais ou *data warehouse*, sendo estes dados distribuídos geograficamente. Tipicamente, as aplicações para grades de dados têm o propósito de mineração de dados, correlacionamento de informações entre fontes de dados de diferente formato e desempenho computacional trabalhando em ambiente de larga distribuição de dados;
- A categoria de grades de serviços é composta por sistemas destinados a estabelecer serviços que não podem ser disponibilizados por uma simples máquina. Esta categoria é formada principalmente por aplicações destinadas a colaboração, integração de demanda (*on demand*) e multimídia. Quando se tratando de grades para colaboração, elas são utilizadas para conectar usuários e aplicações em um grupo de trabalho virtual. As grades por demanda agregam recursos de diferentes espécies para promover um novo serviço como por exemplo correlacionar o registro base do paciente armazenado em um determinado hospital às imagens de exames médicos

que estão em outros hospitais. As grades para conteúdo multimídia promovem uma infra-estrutura para aplicações multimídia que requerem necessidades especiais, tais como: comunicação em tempo real, QoS (*Quality of Service*) entre domínios distintos e alcance a serviços externos.

A figura 3.1 apresenta a organização taxonômica dos sistemas de grade computacional.

Figura 3.1: Taxonomia Sistemas de Grade (KRAUTER; BUYYA; MAHESWARAN, 2002).

Na maioria dos casos, os ambientes de grade computacional possuem características em comum, como ambiente de segurança e serviços para descoberta e localização de recursos. Por este motivo é comum encontrar implementações de ambientes de grade que envolvem mais de um dos modelos apresentados, ou até todos eles. Tais ambientes são utilizados para solucionar problemas que apresentem alta complexidade computacional que envolva extensas massas de dados distribuídas e requeiram alta demanda de colaboração e processamento.

3.3 Componentização das Grades Computacionais

Para concepção de ambientes de grades computacionais é indispensável conhecer os seus componentes e como serão compartilhados os seus recursos. Foster (2002) indica três características fundamentais de um ambiente de grade, possibilitando assim diferenciá-lo de um sistema distribuído convencional:

1. Coordenação de recursos não subordinada a um controle centralizado: o ambiente deve ser apto a integrar e coordenar recursos de diferentes espécies, alocados sobre domínios administrativos distintos, por exemplo, computadores pessoais, centrais de processamento e dados distribuídos sobre organizações físicas distintas;
2. Assumir o uso de padrões, protocolos e interfaces de domínio público para propósitos gerais: o ambiente deve prover a interface necessária para proporcionar interoperabilidade entre as aplicações fundamentais tais como as ferramentas de segurança e descoberta de recursos e submissão de tarefas. Deve também assumir padrões e protocolos abertos, assim proporcionando alto grau de interoperabilidade no ambiente;
3. Promover qualidade de serviço não trivial: devido à complexidade da distribuição do ambiente torna-se necessário promover qualidade de serviço em diferentes níveis da arquitetura, abrangendo, gerenciamento, tolerância a falhas, estabilidade e velocidade de acesso, segurança e alocação/desalocação de recursos.

Ao longo do ciclo evolutivo das grades computacionais, diversas arquiteturas foram concebidas sem seguir nenhuma padronização comum, como é caso do SETI@home que a princípio possuía arquitetura proprietária. Posteriormente o projeto SETI@home concebeu uma arquitetura para processamento computacional de larga escala baseada no aproveitamento de recursos públicos (usuários de Internet colaboradores). A falta de padronização em todos os sentidos (comunicação, dados, plataformas, protocolos) gerou um impacto negativo na interoperabilização dos sistemas (MINOLI, 2004)

Com o aperfeiçoamento dos protocolos empregados na Internet e da própria tecnologia de grades computacionais, foi concebida a componentização lógica mínima necessária para o estabelecimento de ambientes de grade, sendo formada por: interface do usuário, segurança, agente de busca, escalonador de tarefas, gerenciador de dados e gerenciador de tarefas e de recursos. Respectivamente as funcionalidades destes componentes são apresentados nas seções 3.3.1, 3.3.2, 3.3.3, 3.3.4, 3.3.5, 3.3.6.

3.3.1 Interface do Usuário

Um ambiente de grade disponibiliza diferentes interfaces para interação com um usuário. As mais utilizadas são interpretador de comando (shell, sh), aplicação cliente com funcionalidades pré definidas e portais de serviços.

Em muitos casos, a utilização do ambiente distribuído é completamente transparente ao usuário tornando-se análogo à infra estrutura de malha elétrica, onde, o usuário utiliza os recursos da grade computacional sem tomar conhecimento da complexidade empregada na composição do ambiente (MUNGIOLI, 2005).

A submissão de tarefas através do interpretador de comando é o mecanismo mais primitivo de interação com o ambiente, porém ainda muito utilizada principalmente por usuários mais experientes (RIBEIRO; OLIVEIRA; B., 2003).

As aplicações cliente dedicadas à interface com o ambiente de grade são instaladas na estação do usuário e possuem a interface de comunicação com a grade. Normalmente são utilizadas por operadores não experientes, executando tarefas determinadas. Tais aplicações se encarregam de conectar-se ao ambiente e com comandos simples dos dispositivos de interface do usuário requisitam o recurso ou a execução de tarefa ao ambiente. Um exemplo de utilização deste tipo de interface é a submissão de dados por meio de arquivos em lote ou XML.

Os portais de serviços são as interfaces de maior utilização na atualidade, devido à expansão massiva da Internet em todos os níveis da sociedade. Um determinado portal de grade pode disponibilizar uma gama de serviços para uma comunidade, necessitando apenas que o usuário tenha cumprido os requisitos de acesso e um navegador Web. Os administradores podem disponibilizar e excluir serviços do portal dinamicamente, não interferindo na máquina do usuário. (MINOLI, 2004; JITHESH et al., 2006).

3.3.2 Segurança

Para esclarecer o problema de segurança em grade computacional, é conveniente que seja apresentado um cenário baseado em um caso genérico. A figura 3.2 apresenta um ambiente de computação largamente distribuído e os envolvidos estão dispersos geograficamente em instituições distintas.

Figura 3.2: Exemplo de ambiente de grade computacional trocando diferentes tipos de mensagens dentro de uma OV (Organização Virtual).

Dentro deste cenário, podemos observar a existência de sete sítios (A - G), cada um dispondo de recursos e necessidades diferentes. As setas de interligação indicam os processos de comunicação entre os envolvidos.

Os processos de comunicação entre as organizações participantes (sítios) são realizados como segue: o sítio A solicita um procedimento de análise de dados para o sítio C; C adquire informações sobre recursos no sítio D; D solicita tarefas de processamento para os sítios G e E; G e E consultam parâmetros e dados no sítio F.

As operações apresentadas neste cenário constituem um exemplo clássico do funcionamento de uma grade computacional. Dentro de cenários com características semelhantes ao apresentado, devemos considerar os seguintes pontos chave descritos por Foster (1998):

1. O grupo de usuários das organizações multi-institucionais é extenso e dinâmico. Os participantes estão em diferentes instituições constituintes da organização virtual, sendo que tais instituições podem alterar, incluir ou excluir os integrantes conforme

as suas necessidades;

2. Os recursos são vastos e dinâmicos. As instituições podem disponibilizar ou não recursos, por tempo determinado ou constantemente;
3. Os processos computacionais apresentam características distribuídas. Durante a execução podem solicitar recursos e dados de diferentes sítios dinamicamente;
4. Os processos podem utilizar formas diversificadas para se comunicar, tais como *unicast* e *multicast*. Podem criar e destruir conexões dinamicamente em tempo de execução;
5. Os recursos podem ser acessados por diferentes formas de autenticação, autorização ou política, fato devido a divergências do ambiente de segurança empregado nas instituições participantes;
6. Um participante da OV deve ser associado aos diferentes sítios, identificando por meio de autenticação para controle de acesso e contabilidade de uso. O Gerenciamento dos usuários na OV deve ser dinâmico: por exemplo, o usuário em um determinado sítio pode ser um cliente regular e em outros pode ser convidado temporário.

Assim como a Internet, a proteção da infra-estrutura de grade visa prevenir: ataques na configuração de recursos dispersos na grade; roubo de recursos da grade; congestionamento malicioso de nós; e inclusão de dados espúrios que impossibilitam legitimidade da informação. Desta forma, o ambiente deve contemplar as funcionalidades de autenticação, controle de acesso, integridade, confidencialidade e irretratabilidade.

3.3.3 Agentes de Busca de Informações

O Agente de Busca de Informações ou *Broker* é o componente responsável por manter as informações sobre os recursos e serviços que fazem parte do ambiente e informar aos serviços e usuários a disponibilidade e a localização destes. A localização dos componentes do ambiente é realizada através de descrições publicadas e serviços de nome e de diretório (DNS, LDAP) (DUAN; TALLEY; SEETHA, 2007). Também compete ao *Broker* a atividade de controlar a informação sobre o estado dos recursos (disponível, trabalhando, ocupado, indisponível, falho) (MINOLI, 2004).

Figura 3.3: Descoberta de recursos em ambiente de grade computacional (DUAN; TALLEY; SEETHA, 2007).

A figura 3.3 apresenta como agente de busca, realiza o processo de localização das informações sobre os recursos para execução das tarefas desejadas. Segundo o esquema apresentado, quando existe a necessidade de uma execução, é lançada uma requisição de execução ao agente de busca que, por sua vez, consulta o provedor de serviços e os registros de serviços, recebendo como resposta a informação sobre a disponibilidade dos recursos no ambiente. O registrador de serviços é alimentado com a descrição dos serviços pelo provedor de serviços.

3.3.4 Escalonadores

Os escalonadores são componentes responsáveis por coordenar a execução de processos computacionais. São extensamente aplicados em sistemas operacionais multiprogramados para escalonamento de processos e na execução de aplicações paralelas e distribuídas para distribuição de tarefas em ambientes de computadores em redes (SILBERSCHATZ; GAGNE; GALVIN, 2004).

Uma aplicação distribuída necessita de um escalonador quando há necessidade de um controle externo para sua execução, por motivos de interdependências de execução ou quando o ambiente compartilha os recursos disponíveis (MINOLI, 2004).

Um exemplo simplificado das atividades de um escalonador aplicado em sistemas distribuídos é a submissão de trabalhos computacionais para um conjunto de unidades de processamento previamente selecionadas. Este método é normalmente utilizado em aglomerados de computadores (*clusters*) onde as unidades de processamento na maioria dos casos são homogêneas.

Quando em ambiente de grade computacional, os escalonadores necessitam de maior grau de dinamismo, identificando ou sendo informados sobre as características do ambiente que se encontra sob o seu domínio, conhecendo os recursos de processamento aptos a executar determinada tarefa. Neste caso os recursos dispostos no ambiente são heterogêneos em diversos sentidos, como arquitetura dos equipamentos, quantidade de processadores, sistema operacional e tipo dos equipamentos (computador pessoal, aglomerados de computadores).

Os escalonadores desenvolvidos ou adaptados para grades computacionais são conhecidos como meta-escalonadores (*metascheduler*), contendo maior quantidade de funcionalidades quando comparados com os escalonadores aplicados a aglomerados de computadores.

Os meta-escalonadores contemplam ferramentas para designar prioridade de execução, re-submissão automática de tarefas e escalonamento multinível. Para a execução destas tarefas, os componentes recebem informações sobre capacidade de processamento disponível entre os recursos e informações do estado dos recursos (disponível ou ocupado).

Os escalonadores para grade computacional são compreendidos como sendo o escalonador global dos escalonadores locais e de equipamentos que compõem a infra-estrutura. A figura 3.4 apresenta a visão esquemática de uma arquitetura na qual é aplicado meta-escalonamento.

Por interagir com componentes e submeter tarefas na grade computacional, um meta-escalonador necessita ter compatibilidade com as ferramentas de segurança, sistema de informações e ambiente de execução da grade (FERREIRA et al., 2003; MINOLI, 2004).

Figura 3.4: Visão esquemática de uma arquitetura composta por escalonadores locais e um meta-escalador (FERREIRA et al., 2003).

3.3.5 Gerenciamento de Dados

O gerenciamento de dados em um ambiente de grade computacional é um problema abordado de diversas formas. Para alguns tipos de aplicações, esta tarefa ainda é vista como um desafio.

O modelo de gerenciamento dos conjuntos de dados é dependente da taxonomia ou finalidade do ambiente (veja seção 3.2), por exemplo: um ambiente destinado à computação de alto desempenho sofre com a aquisição de dados para processamento, necessitando de uma organização arquitetural otimizada para transporte de dados; um ambiente destinado a integrar dados distribuídos geograficamente em sistemas que não requerem respostas em tempo real faz com que o serviço de gerência de dados e os canais de comunicação que o compõem passem a ser menos estressados, do que no caso anterior (CAMPOS et al., 2006).

Dentro de uma grade computacional podemos encontrar diferentes tipos e formatos de dados, que devem ser interoperados de forma comum no ambiente, sendo: arquivos convencionais, bases de dados relacionais, bases de dados XML, objetos de dados, dados virtualizados, setores de dados e arquivos de metadados (BERMAN; HEY; FOX, 2003).

Quando tratamos da gerência e movimentação de arquivos ou diretórios, o transporte destas informações fica a cargo dos protocolos convencionalmente utilizados na Internet que foram adaptados para o uso em grades computacionais, como é o caso do GridFTP.

Nos demais casos, como base de dados relacional, base de dados XML, objeto de dados, dados virtualizados e setores de dados, torna-se indispensável uma camada de *software* adicional (*middleware*) que possibilite a manipulação de tais informações com a transparência necessária.

A função básica desta camada de *software* é compreender o provedor de informações (i.e. SGBD) e transformar estas informações em um padrão comum à grade computacional, tal como documento XML. No caso de uma nova inserção de informação, o recíproco deve ser atendido (ANTONIOLETTI et al., 2004; CAMPOS et al., 2006).

A figura 3.5 apresenta um modelo de acesso a bases de dados relacionais em ambiente de grade computacional. É possível verificar o acesso a dados por meio da utilização de serviços Web.

Figura 3.5: Acesso a dados no ambiente de grade computacional por meio de serviços Web.

3.3.6 Gerenciamento de Tarefas e de Recursos

Normalmente, o componente responsável por gerenciar os recursos e as tarefas na grade é conhecido como GRAM Gerenciador de Alocação de Recursos na Grade (*Grid Resource Allocation Manager*). A principal funcionalidade do GRAM é lançar tarefa a um recurso específico, checar o *status* e recuperar o resultado quando a tarefa é finalizada. (MINOLI, 2004)

Tipicamente os componentes de gerenciamento se mantêm informados sobre os recursos disponíveis por meio dos agentes de informações do ambiente e sobre quais usuários podem utilizar os recursos. Estas informações são utilizadas para decidir quais recursos irão executar quais tarefas de quais usuários.

Outra funcionalidade deste componente é a determinação da capacidade ou poder computacional dos recursos, conhecendo o nível de utilização dos recursos no instante desejado. Esta função é importante para prevenir congestionamentos, excesso de agendamentos e de utilização dos recursos. (MINOLI, 2004)

Pelo fato do GRAM participar no controle da submissão das tarefas aos recursos, ele detém a responsabilidade de contabilizar a utilização dos recursos por usuário (FELLER; FOSTER; MARTIN, 2007).

3.4 OGSA - Arquitetura Aberta de Serviços de Grade

Dentro dos trabalhos realizados pelos grupos do OGF, o mais influente na comunidade é a especificação da Arquitetura Aberta de Serviços de Grade (OGSA-Open Grid Service Architecture). A estrutura da arquitetura OGSA é dividida em quatro camadas, contemplando (FOSTER et al., 2006):

1. Camada de aplicação: é a camada responsável por suportar as aplicações utilizadas pelos usuários, sendo vista como uma interface da grade com o mundo externo, de forma que o usuário somente tem acesso a esta camada;
2. Camada de serviços de grade OGSA: esta camada é responsável por gerir os serviços que mantêm o ambiente funcional, sendo que os componentes de software desta camada são sempre baseados em serviços Web e contemplam tarefas como descoberta, gerenciamento de estado do ambiente, agrupamento de serviços, fábrica de serviços, notificações;

3. Camada de serviços Web: é a camada responsável pelo estabelecimento dos padrões utilizados para os serviços Web que compõem o ambiente; utiliza a padronização aplicada aos serviços Web para definir interfaces comuns para comunicação baseadas em WSDL (*Web Service Definition Language*) e XML;
4. Camada de recursos físicos e lógicos: esta camada tem por finalidade prover para o ambiente os recursos para a grade computacional. Os recursos físicos são os equipamentos em geral como sistemas de armazenamento de dados, redes de computadores, servidores, aglomerados de servidores, entre outros. Os recursos lógicos são recursos que oferecem funcionalidades para os recursos físicos, como os sistemas de virtualização de ambiente. Um recurso lógico pode ser apresentado como um serviço disposto na camada de serviços Web (MINOLI, 2004; FOSTER et al., 2006).

Os objetivos centrais da especificação são:

- Gerenciar de forma distribuída os recursos em plataformas de *hardware* e *software* heterogêneas;
- Prover serviços com níveis de segurança elevados;
- Estabelecer o gerenciamento autônomo de plataformas computacionais de larga escala;
- Estabelecer interfaces de comunicação entre aplicações sobre protocolos abertos incentivando assim a interoperabilidade entre serviços no ambiente (MINOLI, 2004; FOSTER et al., 2006).

Muitas implementações de aplicações para grade computacional utilizam a especificação do OGSA, com o finalidade de proporcionar a interoperabilidade com outras ferramentas já existentes (MINOLI, 2004).

3.5 Ferramentas para Implementação de Grades Computacionais

Atualmente podemos encontrar diversas ferramentas para a implementação de ambientes de grades computacionais. Nesta seção, vamos apresentar as que consideramos relevantes no escopo da pesquisa em questão.

3.5.1 Globus Toolkit 4.0.X

O desenvolvimento do *Globus Toolkit* (GT) foi iniciado no final dos anos 90 e atualmente esta ferramenta é distribuída na versão 4.2. Os componentes do GT formam um conjunto de ferramentas fundamentais para o estabelecimento de ambientes de grades computacionais, sendo destinadas a promover segurança, acesso a recursos, gerenciamento de recursos, gerenciamento de dados e serviços de informações do ambiente. A implementação dos componentes desta ferramenta é baseada na especificação do OGSA (MINOLI, 2004; MUNGIOLI, 2005; FOSTER, 2005).

O Globus, a partir da sua versão 3, utiliza *Web Services* para definir as interfaces e estruturas dos seus componentes. Este fato permite adotar mecanismos baseados em XML para descrever, descobrir e invocar serviços na rede, suportando padronizações, como, WSRF (*Web Services Resource Framework*), WS-*Notification* e WS-*Security* (OASIS, 2007; GLOBUS... , 2007).

O núcleo funcional do Globus é formado por componentes de *software* que realizam tarefas de gerenciamento de execução (GRAM); movimentação, acesso e gerenciamento de replicação de dados (GridFTP, RFT, RLS, DRS, OGSA-DAI); monitoramento e descoberta de informações (WebMDS); gerenciamento de credenciais (Myproxy, Delegation, SimpleCA); armazenamento e disponibilização de serviços sobre ambiente de execução comum (Contêiners de serviços Web); e disponibilização de bibliotecas de *software* para desenvolvimento (Java, Python, C) (FOSTER, 2005).

Em resumo, o Globus promove um conjunto de serviços de infra-estrutura que implementam as interfaces necessárias para gerenciar recursos computacionais (armazenamento, CPUs, serviços, recursos de rede e equipamentos remotos) (FOSTER, 2005; GLOBUS... , 2007). A figura 3.6 apresenta os componentes do GT e as previsões dos novos componentes.

Figura 3.6: Componentes do Globus Toolkit 4.0.X (GLOBUS..., 2007).

Serviços de Segurança

A segurança é um fator de fundamental importância para o correto funcionamento do ambiente. A arquitetura de segurança empregada no Globus é baseada no GSI (*Globus Security Infrastructure*), que é uma especificação integrante do OGSA, composta por camadas, onde cada uma delas implementa um nível de segurança (FOSTER et al., 1998; CHAKRABARTI, 2007).

A figura 3.7 apresenta a disposição das camadas de segurança aplicada ao Globus.

O sistema de segurança aplicado ao Globus é baseado em infra-estrutura de chaves públicas (ICP), certificados X.509 e TSL/SSL (*Transport Layer Secure / Secure Sockets Layer*) para proteção dos canais de comunicação e utiliza os padrões de segurança já utilizados nos serviços Web (CHAKRABARTI, 2007).

Figura 3.7: Visão das camadas da arquitetura GSI.

O GSI utiliza padrões já empregados na Internet, portanto o Globus emprega ferramentas que estendem a funcionalidades destes padrões para realizar tarefas mais aprimoradas no trabalho em ambiente de grade computacional, sendo o Myproxy, Delegation e SimpleCA.

A SimpleCA é uma ferramenta para criar e gerenciar certificados baseados em uma autoridade certificadora (AC) própria, formando uma infra-estrutura de chaves públicas (ICP) para gerenciamento de certificados de servidores e usuários da grade. Esta ferramenta é utilizada em ambientes que não requeiram uma autoridade certificadora externa. No caso da saúde, o Conselho Federal de Medicina (CFM) exige o uso de um certificado ICP-Brasil para o trânsito de informações digitais da saúde (Resolução CFM N° 1.821, de 11 de Julho de 2007).

Para possibilitar que um usuário ou serviço utilize um ambiente baseado no Globus, eles devem realizar uma autenticação. A partir do certificado padrão do usuário é gerado um certificado secundário com tempo de vida curto, conhecido como certificado Proxy. O usuário acessa os recursos do ambiente utilizando somente este certificado.

A ferramenta MyProxy é um servidor que armazena os certificados Proxy's válidos e os disponibiliza quando necessário, para o próprio usuário ou para algum serviço que a requisite em nome do utilizador. Uma vez que um certificado é armazenado neste servidor ele pode ser recuperado a partir de qualquer lugar do ambiente.

A ferramenta Delegation é utilizada para possibilitar que um determinado serviço possa realizar um ou mais procedimentos no ambiente em nome do usuário ou de outro serviços.

Gerente de Alocação de Recursos da Grade

O sistema para gerenciamento e alocação de recursos do Globus é o GRAM (*Grid Resource Allocation and Management*). Esta ferramenta provê uma interface baseada em serviço Web para inicialização, gerenciamento de execução de processos em estações remotas pertencentes ao ambiente. As interfaces do GRAM permitem que o usuário possa expressar as suas necessidades, como quantidade e tipo de recursos desejados para executar a tarefa (FOSTER, 2005)

O GRAM pode ser utilizado de diferentes maneiras. Por exemplo, podemos submeter uma grande quantidade de tarefas individuais para processamento em recursos computacionais da grade. Podemos utilizar esta ferramenta para submeter tarefas baseadas em bibliotecas de comunicação para programação paralela a aglomerados de computadores paralelos (FELLER; FOSTER; MARTIN, 2007; FOSTER, 2005).

Apesar destas funcionalidades o GRAM não é um meta-escalonador para grade computacional devido ao fato de prover somente as interfaces para submissão de tarefas. Todos os meta-escalonadores que são utilizados em ambientes Globus utilizam o GRAM como a interface para submissão das tarefas.

Gerenciamento de Dados

O Globus oferece quatro ferramentas básicas para gerência de dados (GridFTP, RFT, RLS, OGSA-DAI), que atuam de diferentes formas na movimentação e serviços de dados no ambiente.

A primeira delas é o GridFTP. Esta ferramenta é uma extensão do protocolo FTP convencional, com ferramentas e bibliotecas adaptadas para promover a integração com a segurança do ambiente, confiabilidade e desempenho. (FOSTER, 2005; BRESNAHAN et al., 2007)

O RFT (Transferidor de Arquivo Confiável) é um componente utilizado em conjunto com o GridFTP para prover confiabilidade para transferência.

O RLS (Serviço de Localização de Réplica) é um serviço que permite a localização

dos setores de dados replicados, promovendo disponibilidade nas transações de recepção de arquivos de dados no ambiente. Este componente trabalha como um suporte para o RTF.

O DRS (Serviço de Replicação de Dados) combina as funcionalidades do GridFTP e do RLS para promover o gerenciamento de replicações de dados. Este serviço utiliza o GridFTP para replicar os setores de dados e o RLS para localizar as réplicas.

O OGSA-DAI (*OGSA-Acesso e Integração de Dados*) permite o acesso à base de dados relacional, XML e a arquivos indexados, por meio de serviços Web. A consulta em formato SQL é enviada para a base de dados por meio de um documento XML, previamente convertido pelo OGSA-DAI que cria este documento XML para passagem da informação pelo ambiente. Ao chegar à base de dados destino a mensagem novamente é traduzida para o formato SQL e então é processada. A solicitação é respondida para o solicitante pelo processo reverso ao da consulta.

O OGSA-DAI nos permite configurar o acesso a dados de diferentes formas: utilizando uma única base de dados, abrangendo um conjunto de bases de dados que possuem a mesma estrutura de forma federativa ou homogenizando o acesso a base de dados com estruturas diferentes. Esta funcionalidade pode ser explorada para diversas finalidades como criação de recursos de metadados, aproveitamento do legado de dadas instituições integradas ao ambiente de grade e homogeneização de setores de dados distintos, entre outros (ANTONIOLETTI et al., 2004).

Serviços de Informação do Ambiente

Os serviços de monitoramento e descoberta de recursos e serviços são vitais para o funcionamento de um ambiente de grade computacional. As tarefas de monitoramento consistem em detectar e diagnosticar problemas em diferentes níveis da topologia. As tarefas de descoberta consistem em identificar os recursos e serviços registrados no ambiente com as suas propriedades e estado (FOSTER, 2005).

O *Globus* disponibiliza a ferramenta MDS (*Monitoring and Discovery System*) para execução das tarefas de monitoramento e descoberta de serviços. A sua arquitetura é exposta em forma de uma ampulheta, como apresentado na figura 3.8.

Figura 3.8: Arquitetura MDS apresentada em forma de ampulheta (SCHOPF et al., 2005).

Na parte inferior da ampulheta, diversas fontes transmitem informações para o MDS, que são traduzidas para mensagens padronizadas no formato XML e transmitidas através de protocolos WSRF. Na parte superior da ampulheta, as aplicações e ferramentas podem acessar as informações sobre os recursos de forma padronizada a partir de consultas a serviços Web (SCHOPF et al., 2005).

O MDS permite que outras ferramentas de monitoramento possam ser acopladas ao ambiente, assim expandindo as suas capacidades.

O Ganglia é uma ferramenta que possibilita monitorar o estado dos recursos, indicando a disponibilidade de memória, processador, disco, versão do sistema operacional, arquitetura do processador, entre outras. As informações disponibilizadas pelo Ganglia ao MDS são utilizadas pelos meta-escalonadores para realizar seus serviços com maior eficácia.

Ambiente de Execução Comum

O Ambiente para execução comum disponibilizada pelo GT é composto por um conjunto de bibliotecas, contêiner para disponibilização de serviços em diferentes linguagens (C, Python e Java) que, por seguirem padronização WSRF, conseguem alto grau de interoperabilidade.

Os contêineres podem ser utilizados para manter serviços de infra-estrutura adicionais e serviços de aplicação.

Considerações Sobre o Globus Toolkit

Somente as ferramentas e serviços do GT não atendem todos os requisitos de grande parte das aplicações distribuídas. Este fato é atribuído ao escopo do projeto *Globus*, que consiste em estabelecer o núcleo funcional de uma grade computacional, possibilitando que o utilizador personalize o seu ambiente conforme as suas necessidades.

Assim, existem diversas ferramentas que possuem interoperabilidade com o GT, desenvolvidas em outros projetos ou em parceria com o projeto *Globus*, indicadas ou incluídas na distribuição do GT, complementando de forma significativa as necessidades dos usuários, oferecendo colaboração, meta-escalonamento, empacotamento e distribuição de *software*, bibliotecas para programação de serviços e aplicações paralelas e distribuídas (GLOBUS..., 2007).

3.5.2 BOINC - *Berkeley Open Infrastructure for Network Computing*

O BOINC é um conjunto de módulos de aplicação de código aberto e distribuição livre para construir plataforma de computação distribuída que explora a utilização de recursos públicos (equipamentos de usuários conectados à Internet) para realizar tarefas de processamento em larga escala de dados que não possuem interdependências na sua execução. Este modelo de computação distribuída também é conhecido como computação oportunista (RATTEI et al., 2006; GOLDCHLEGER, 2004).

A arquitetura do BOINC é dividida em três partes: BOINC servidor, núcleo BOINC do cliente e aplicação BOINC.

O BOINC servidor é composto por um núcleo que contém o escalonador de tarefas, servidor de dados, serviços Web e serviços para controle das aplicações do projeto e do próprio BOINC. Neste lado da arquitetura, contamos com alguns componentes já utilizados comumente em aplicações desenvolvidas para o ambiente de Internet, como servidor de Web, para disponibilização das aplicações e transporte de dados para os usuário e bancos de dados relacionais (MySQL, Postgresql) para armazenamento dos resultados dos processamentos realizados pelos cliente (RATTEI et al., 2006).

A parte do sistema executada no cliente é conhecida como núcleo BOINC cliente,

sendo responsável por receber as subtarefas submetidas pelo escalonador, recepcionar os dados a serem processados, submeter os dados recebidos para o processamento na aplicação e devolver ao servidor BOINC os resultados obtidos no processamento da sub tarefa. A aplicação BOINC é o módulo de aplicação. Cada projeto deve desenvolver o seu próprio. Neste módulo estão contidas as operações que a estação do cliente deve realizar sobre os dados recebidos do núcleo BOINC cliente (RATTEI et al., 2006). O BOINC ainda disponibiliza API (Interface de Programação de Aplicação) para padronização do desenvolvimento, reduzindo a dificuldade de implementação.

A figura 3.9 apresenta a arquitetura estrutural de um projeto baseado na ferramenta BOINC.

Figura 3.9: Arquitetura básica de um projeto BOINC (RATTEI et al., 2006).

Alguns autores não consideram esta infra-estrutura como sendo um ambiente de computação em grade porque não contemplam os três aspectos fundamentais de uma grade computacional (seção 3.3), como controle descentralizado. No entanto, os resultados em função do custo e do aproveitamento de recursos ociosos são muitos consideráveis. Para demonstrar este fato, podemos citar o projeto SIMAP, baseado nesta plataforma, que em julho de 2006 conseguiu agregar o poder computacional efetivo de 3.5 Tflops contando

com a contribuição de mais de dez mil usuários que disponibilizaram em torno de vinte mil estações (RATTEI et al., 2006).

O BOINC surgiu da evolução do SETI@home. A princípio, este projeto possuía uma arquitetura própria com o método de distribuição similar ao BOINC, mas não permitia que a aplicação fosse alterada. Após o surgimento do BOINC o projeto SETI@home foi migrado para esta plataforma.

3.5.3 OurGrid

O OurGrid é uma ferramenta para implementação de grade computacional dedicada à distribuição de tarefas entre instituições distantes geograficamente utilizando uma rede P2P (*Peer To Peer*).

O OurGrid é formado por um conjunto de três componentes principais: MyGrid, OurGrid *Communit* e o SWAN. O MyGrid é o componente responsável por disparar as tarefas para as estações disponíveis em ambiente de rede local. O OurGrid *Communit* é o componente responsável pela interligação das instituições participantes por meio de uma rede P2P, desta forma agrupando os ambientes de processamento MyGrid. O SWAN é o componente responsável por garantir o acesso de forma segura aos recursos do ambiente.

A figura 3.10 apresenta a arquitetura do OurGrid e de seus componentes.

O OurGrid é uma ferramenta que complementa as funcionalidades da sua versão antecessora, o MyGrid que por sua vez é dedicado a distribuir as tarefas em ambiente de rede local.

As duas ferramentas são frutos de iniciativas do Laboratório de Sistemas Distribuídos de Universidade Federal de Campina Grande (LSD) em parceria com a Hewlett-Packard do Brasil (BRASILEIRO; CIRNE, 2005b, 2005a).

Figura 3.10: Arquitetura OurGrid (ANDRADE et al., 2005).

3.5.4 Comparativo entre as Ferramentas Abordadas

Cada uma das ferramentas apresentadas possui características muito particulares. Para realizar a avaliação devemos considerar a motivação central para o desenvolvimento deste trabalho, que consiste em empregar as tecnologias de grade no setor de saúde nacional, voltando para atenção à oncologia e bioinformática, abordando as seguintes características: flexibilidade, modularidade e escalabilidade.

Os pontos chave de cada característica são:

1. Flexibilidade: código fonte aberto, binários gratuitos, implementação do OGSA;
2. Modularização: sistema de segurança padronizado, quantidade de camadas funcionais básicas, indexação de serviços, acoplamentos de sistemas terceiros;
3. Escalabilidade: forma de acesso a dados; suporte a múltiplos serviços, múltiplos métodos de escalonamento, processamento oportunista, acoplamento de nós de processamento, acoplamento de novas aplicações em tempo de execução.

A interoperabilidade também é um fator fundamental para este tipo de ambiente computacional, porém o fator principal que promove este benefício é a padronização. Desta

forma, consideramos o fator promotor da interoperabilidade é a utilização de padrões como o OGSA na implementação da ferramenta.

As tabelas 3.1, 3.2, 3.3, apresentam as avaliações comparativas realizadas.

Flexibilidade	Globus	BOINC	OurGrid
Código fonte aberto	Sim	Sim	Sim
Binários gratuitos	Sim	Sim	Sim
Implementação do OGSA	Sim	Não	Não

Tabela 3.1: Comparação dos itens de flexibilidade das ferramentas para implementação de grades computacionais

Modularização	Globus	BOINC	OurGrid
Sistema Segurança Padronizado	Sim	Não	Sim
Camadas funcionais básicas	5	3	3
Indexação de Serviços	Sim	Não	Não
Acoplamentos de sistemas terceiros	Sim	Não	Não

Tabela 3.2: Comparação dos itens referentes à modularização das ferramentas para implementação de grades computacionais

Escalabilidade	Globus	BOINC	OurGrid
Forma de acesso a dados	múltiplos	único por aplicação	único por aplicação
Suporte a múltiplos serviços	Sim	Sim	Sim
Sistemas à meta-escalonamento	Sim	Não	Não
Processamento Oportunista	Não	Sim	Não
Acoplamento de estações de processamento dinamicamente	Sim	Sim	Sim
Acoplamento de novas aplicações em tempo de execução	Sim	Não	Não

Tabela 3.3: Comparação dos itens referentes à escalabilidade das ferramentas para implementação de grades computacionais

A partir da análise realizada podemos verificar que o modelo do BOINC é indicado para a solução de problemas computacionalmente extensos, porém é direcionado a projetos, cada projeto tem a sua própria aplicação e grupo de colaboradores. Com o *OurGrid* conseguimos utilizar a infra-estrutura para solucionar diferentes projetos mas o suporte

apresentado é apenas para a disponibilização de processamento utilizando padronização particular. Assim, somente pode ter interoperabilidade com outras infra-estruturas baseadas em *OurGrid*. O Globus, por sua vez, possibilita alto nível de padronização, distribuição de tarefas sobre diferentes plataformas de processamento, emprega padronização nos sistemas de segurança, permitindo o emprego de serviços para diferentes finalidades. Sendo apresentado como a opção mais adequada para a utilização neste trabalho.

3.6 Ferramentas para Distribuição de Processamento em Grade Computacional

Nesta seção vamos apresentar as ferramentas que realizam a tarefa de gerenciar a distribuição de tarefas computacionais sobre um grupo de recursos e que possam ser utilizados em conjunto ou integradas aos ambientes de grade computacionais baseados na ferramenta Globus.

3.6.1 Portable Batch System - PBS

O *Portable Batch System* (PBS) é um sistema para gerenciamento de lotes de tarefas para aglomerados de processadores (nós) distribuídos em ambiente de rede (CLUSTER-RESOURCE, 2007). A primeira versão deste sistema foi desenvolvida entre 1993 e 1997 pela NASA (*National Aeronautics and Space Administration*) para substituir os sistemas de distribuição de tarefas em lote até então utilizados pela agência. Posteriormente, a empresa *Veridian System* assumiu a distribuição comercial do produto.

O PBS é indicado para a distribuição de tarefas que não possuem interdependências (BoTs-Bag of Task), e a sua função principal é conhecer os recursos disponíveis, distribuir os trabalhos para as filas de execução, gerenciar as filas de execução e monitorar o estado do processamento da tarefa em ambiente de aglomerados de computadores (HENDERSON, 1995; CLUSTER-RESOURCE, 2007).

Esta ferramenta tem um arquitetura simplificada, porém funcional, formada basicamente por quatro componentes:

- Servidor de Tarefas: A principal função do servidor é fornecer os serviços básicos para o aglomerado, tais como recebimento/criação de uma tarefa, modificação e envio da tarefa para fila de execução;

- **Monitor de Recurso:** A principal função deste componente é verificar as informações relacionadas ao estado dos nós de processamento. Cada unidade de processamento executa uma instância do Monitor de Recursos;
- **Servidor de Execução de Tarefas:** Este componente é responsável pela execução das tarefas nos nós de processamento. Esta ferramenta também é conhecida como MOM (Machine Oriented Miniserver). Em cada equipamento do ambiente é executado uma instância deste servidor;
- **Escalonador:** Este componente obtém informações dos Servidores de Execução de Tarefas (MOMs) e dos Monitores de Recursos para conhecer o estado do ambiente no instante desejado. Através dos dados coletados, o escalonador aplica as melhores políticas locais para submeter as tarefas da fila de espera para execução nos nós de processamento. (HENDERSON, 1995).

Em um ambiente de aglomerado de computadores baseado no PBS (*PBS-cluster*) existem dois tipos de máquinas, uma máquina de gerenciamento conhecida como diretor ou *master*, que executa as funções do Servidor de Tarefas e do Escalonador; e o conjunto de tamanho N variável de nós para processamento que executa as funções de Servidor de Execução de Tarefas e Monitor de Recurso (HENDERSON, 1995).

A figura 3.11 apresenta a organização da arquitetura de um aglomerado de computadores baseado em PBS.

O sistema PBS possui diversas distribuições, sendo OpenPBS, PBSpro e TORQUE-PBS. Os aglomerados concebidos com estas ferramentas podem ser utilizados como recursos de processamento de ambientes de grade computacional baseado no Globus trabalhando em conjunto com o WS-GAM e MDS do Globus.

3.6.2 Condor e Condor-G

O Condor é um gerenciador de lotes de tarefas para ambientes de rede locais, que provê mecanismos para gerenciamento de filas de execução, políticas de escalonamento, gerenciamento de prioridades de execução e classificação de recursos computacionais. O processo de execução do Condor consiste em recepcionar as tarefas dos clientes, colocá-las em uma fila de execução, executar a tarefa e por fim devolver a resposta ao cliente. O Condor também possibilita a execução das tarefas de forma oportunista, utilizando as estações ociosas de um ambiente de rede local (THAIN; TANNENBAUM; LIVNY, 2005).

Figura 3.11: Arquitetura base de um aglomerado de computadores baseado no PBS. Adaptado de (HENDERSON, 1995)

O Condor-G é um componente adicional de gerenciamento da execução de tarefas sobre aglomerados Condor, PBS e outros gerenciadores de recursos computacionais, portátil com os padrões de segurança, com os serviços de indexação de recursos possibilitando, assim o acesso a equipamentos em ambientes multi-institucionais (FREY et al., 2002).

O Condor é considerado o gerenciador de tarefas em lote para os recursos sobre o ambiente institucional e o Condor-G o gerenciador de execução multi-institucional que faz a interface com as ferramentas de grade computacional do Globus. A figura 3.12 apresenta de forma esquemática a disposição do Condor e do Condor-G sobre um ambiente de grade.

A forma de execução oportunista empregada no Condor utiliza as máquinas que estão sem utilização no momento, como por exemplo, sem atividade nas interfaces de entrada (teclado, mouse) e utilização de CPU próxima a zero. Caso a estação apresente algum tipo de utilização o Condor migra o processo em execução para outra CPU. Alguns processos não podem ser preemptados, assim, o Condor reserva as estações que possibilitarão manter a execução do processo do início ao fim (THAIN; TANNENBAUM; LIVNY, 2005).

Figura 3.12: Condor e Condor-G dispostos sobre um ambiente de grade computacional baseado no Globus Toolkit (THAIN; TANNENBAUM; LIVNY, 2005).

3.6.3 GridWay

O *GridWay* é uma ferramenta que foi desenvolvida especificamente para trabalhar com ambientes Globus, provendo interfaces amigáveis entre o usuário e a grade computacional nas atividades de gerenciamento da execução de tarefas, e fornece mecanismos para execução eficaz de tarefas possibilitando a troca adaptativa dinâmica das condições de execução (HUDO; MONTEIRO; LLORENTE, 2005).

Para o usuário, o *GridWay* apresenta dois componentes principais: interface de usuário por linha de comando, que provê comandos para simples submissão, parada, reinício e cancelamento da execuções de tarefas. O agente pessoal de recursos é o responsável por descobrir os recursos disponíveis, submeter e escalonar as tarefas do usuário e também monitorar o desempenho da tarefa (HERRERA et al., 2004).

A arquitetura do *GridWay* é apresentada na figura 3.13 e os tópicos a seguir descrevem as funcionalidades de cada um dos componente apresentados na figura.

- Interface do usuário: responsável por submeter os comandos inseridos pelo usuário. Este módulo inclui o DRMAA (*Distributed Resource Management Application API*) que é a implementação de um padrão OGF para suporte ao desenvolvimento de aplicações distribuídas baseadas em C e Java;
- Mediador de Acesso ao Dispositivo: Conhecido como MAD (*Middleware Access Driver*), é o componente responsável por integrar o ambiente externo à grade com o interno. O MAD adquire as informações do ambiente (recursos disponíveis, gerente

Figura 3.13: Arquitetura do meta-escalonador GridWay (GLOBUSALIANCE, 2007).

de recursos) e as disponibiliza para o usuário final. Todos os componentes do núcleo do GridWay o acessam para realizar as suas tarefas;

- Núcleo GridWay: é uma conjunto de componentes que provê o completo gerenciamento da execução e agenciamento de recursos, oferecendo escalonamento avançado e capacidades para recuperação de falhas de execução. O gerenciador de disparos realiza todos os estágios de submissão da tarefa e monitora se a execução está acontecendo de maneira correta. O gerenciador de informações é o responsável por descobrir e monitorar as informações sobre as estações e execuções. O gerenciador de execução é o responsável pela submissão e monitoramento da execução da tarefa. O gerenciador de transferência é responsável pelas operações que envolvem o acesso e a preparação dos dados para processamento, incluindo o envio de códigos binários e a coleta dos dados não mais utilizados nos recursos de processamento;
- Escalonador: responsável por decidir o método de distribuição das tarefas sobre os recursos disponíveis no momento. No GridWay o escalonador trabalha de forma adaptativa, avaliando as melhores condições de escalonamento no momento da execução. Esta operação é realizada com base nas informações do ambiente, no histórico de execuções anteriormente realizadas e nas prioridades configuradas para os

recursos e usuário. O escalonador do GridWay permite a submissão de tarefas em ambiente composto com recursos heterogêneos, por exemplo, aglomerados baseados no PBS, estações convencionais, estações multiprocessadas e clusters paralelos.

- Interface do Gerenciador de Informações: realiza a interface com o núcleo do GridWay e com os sistemas de informação da infra-estrutura da Grade (MDS2, MDS4);
- Interface do Gerenciador de Execução: realiza a interface com os serviços de gerenciamento de tarefas e recursos da grade (pré WS-GRAM, WS-GRAM);
- Interface de Gerenciador de Transferência: realiza a interface com os serviços de transporte de dados da grade (GridFTP, RFT).

Por ser um aplicativo mais recente que os demais apresentados, o GridWay possibilita a distribuição de tarefas baseadas em duas plataformas de desenvolvimento, sendo C/C++ e Java. Os demais apresentados conseguem somente distribuir tarefas baseadas em C e Fortran (GLOBUSALIANCE, 2007).

Esta ferramenta possibilita que o usuário, por meio de um arquivo de descrição de tarefas, possa discriminar as características do ambiente onde seu(s) processo(s) será(ão) executado(s). Entre as vinte e cinco opções disponíveis para configurar o ambiente de execução, podemos citar algumas como: indicar a máquina ou a localidade que deve ser executada (i.e. REQUIREMENTS = HOSTNAME = “maquina01.lsi.usp.br ” ou REQUIREMENTS = HOSTNAME = “*.lsi.usp.br ”), indicar a arquitetura do processador (i.e. REQUIREMENTS = ARCH = “i686”), indicar o gerenciador de tarefas local (i.e. fork, pbs, condor, etc.), o sistema e a versão do sistema operacional (i.e. Linux, Windows, etc.). Na apêndice B apresentamos todos os parâmetros possíveis para configuração do ambiente de execução.

3.7 Projetos Similares em Grade para Saúde

Nesta seção vamos apresentar alguns trabalhos que exploram a tecnologia de grade computacionais para integração de dados e processamento distribuído em benefício da área da saúde e bioinformática.

3.7.1 Genegrid: Grade Computacional Aplicada a Bioinformática

O projeto GeneGrid propõe e implementa uma arquitetura grade computacional baseada na especificação OGSA. As principais finalidades deste ambiente são proporcionar a integração de recursos de dados e processadores distribuídos geograficamente e agregar serviços e aplicações dedicados à bioinformática formando um laboratório virtual. O acesso ao ambiente é realizado por uma interface amigável baseada em portais Web para manipulação e processamento de dados biológicos. Este sistema promove a colaboração entre equipes de pesquisa de genética e bioinformática. A arquitetura empregada neste ambiente engloba aspectos de grades de serviços de dados e computacional (JITHESH et al., 2005, 2006).

3.7.2 caBIG: Grade de Informações Biomédicas em Câncer

O caBIG é uma iniciativa apoiada pelo governo federal Norte-Americano e desenvolvida pelo Instituto Nacional do Câncer dos EUA. Este projeto visa criar uma infraestrutura computacional para promover a colaboração entre pesquisadores e instituições de tratamento do câncer, possibilitando o acesso aos dados distribuídos geograficamente nos hospitais e centros de pesquisa.

A arquitetura do caBIG utiliza o Globus como sua ferramenta principal e utiliza o OGSA-DAI e GRAM para oferecer serviços como gerenciamento de experimentos clínicos, ferramentas patológicas, dicionários de vocabulários de dados comuns e bancos de tecidos (FENSTERMACHER et al., 2005).

Para atender as funcionalidades relacionadas com segurança do ambiente o projeto caBIG desenvolveu duas ferramentas próprias: a *Grid Authentication and Authorization with Reliably Distributed Services* (GAARDS) que implementa a especificação GGF-GSI assim possibilitando a integração com o Globus; e a *Grid Trust Service* (GTS) que possibilita a utilização de diversas autoridades certificadoras, que foi incorporada ao ambiente devido à grande quantidade de instituições participantes (SALTZ et al., 2006).

3.8 Resumo do Capítulo

Este capítulo abordou os componentes e ferramentas que atualmente são utilizados para o estabelecimento de ambientes de larga escala computacional baseados na tecnologia

de grades computacionais.

Os ambientes de grade computacional são subdivididos taxonomicamente entre grade computacional, grade de dados e grade de serviços. Um ambiente de grade computacional pode possuir o enquadramento de uma ou mais de uma destas subdivisões.

As funcionalidades básicas que um ambiente de grade deve apresentar são segurança, gerenciamento de informações, meta-escalonamento, gerenciamento de dados e gerenciamento de tarefas e recursos.

O OGF é o fórum mais influente sobre o assunto de grades computacionais. Este fórum especifica os padrões que mais são aceitos pela comunidade. O principal trabalho deste grupo é a especificação da Arquitetura Aberta de Serviços de Grade (OGSA) que aborda como se estabelecem os níveis de serviço de um ambiente de grade computacional.

Foram avaliadas ferramentas para implementação de ambientes de Grade computacional, como o BOINC, o OurGrid e o Globus Toolkit. Dentre estas, a que apresentou melhores características para cumprimento dos objetivos foi o Globus, porque possui melhores características relacionadas à flexibilidade e atende as necessidades para a implementação deste trabalho. Esta ferramenta implementa a especificação OGSA e por este motivo proporciona alto nível de interoperabilidade com ferramentas e com outros ambientes que utilizam este padrão.

Também foram avaliadas ferramentas para escalonar tarefas em ambientes distribuídos globalmente baseados no Globus, como o *Portable Batch System* (PBS), o Condor e o GridWay. O PBS e o Condor são sistemas que foram adaptados para o trabalho de meta-escalonamento em ambientes globais pois são implementações antigas e de eficácia comprovada. O GridWay é uma ferramenta que foi desenvolvida especificamente para trabalhos com grades computacionais baseadas na ferramenta Globus, sendo uma opção a ser avaliada para a aplicação no cenário do setor de saúde nacional, em especial para atender as demandas computacionais requisitadas pelas pesquisas em bioinformática.

Atualmente podemos encontrar projetos e implantações no campo da pesquisa que abordam assuntos similares aos tratados nesta dissertação, como o GeneGrid, que estabelece uma infra-estrutura de grade para ser utilizada como um laboratório de pesquisas em bioinformática, e o caBIG, que estabelece um arcabouço de ferramentas para diversas áreas de pesquisa e atenção primária à oncologia.

4 Concepção do Ambiente de Processamento para Grade Computacional em Oncologia e Bioinformática

Este capítulo apresenta a proposta do ambiente de grade computacional OncoGrid e se estrutura em três seções principais: a seção 4.1 apresenta a arquitetura do projeto OncoGrid; a seção 4.2 apresenta a proposta do ambiente de processamento de alto desempenho em grade computacional para ser acoplado ao OncoGrid; por fim, a seção 4.3 apresenta a aplicação desenvolvida para as avaliações da utilização de procedimento de bioinformática em oncologia no ambiente de grade computacional implementado.

4.1 OncoGrid - Grade Computacional em Oncologia

A iniciativa OncoGrid foi originada com o propósito de investigar as possibilidades do emprego da tecnologia de grade no setor de saúde, em especial em Oncologia. Os trabalhos realizados no desenvolvimento deste projeto consistiram na pesquisa das tecnologias envolvidas e na investigação das aplicabilidades no campo da saúde em oncologia. O desenvolvimento consistiu no estabelecimento de um projeto piloto que agregava a componentização fundamental de um ambiente de grade computacional.

Partindo dos estudos realizados no desenvolvimento do projeto piloto OncoGrid, constatamos dois grandes desafios da tecnologia da informação em saúde (TIS), onde esta abordagem tecnológica poderia expressar forte colaboração: na gestão distribuída de informações médicas e no processamento de larga escala que é demandado da por diversas áreas da saúde, como é o caso da bioinformática. Após a concepção da componentização básica do OncoGrid, foram dedicados esforços e pesquisas nessas duas linhas.

A primeira aplicabilidade prática desenvolvida e testada no OncoGrid foi a integra-

ção de registros hospitalares de câncer geograficamente distribuídos, por meio da infraestrutura de grade computacional (ALVES et al., 2008).

A proposta e as implementações contidas nesta dissertação tratam da segunda linha de pesquisa abordada no projeto OncoGrid, que se refere ao estabelecimento da plataforma de processamento distribuído para suporte a aplicações computacionalmente complexas no setor de Oncologia.

Para compreensão integral da proposta desta dissertação, dedicamos esta seção para descrever a arquitetura do projeto OncoGrid. Para isso detalhamos os requisitos elencados, camadas funcionais e a componentização fundamental.

4.1.1 Requisitos do OncoGrid

Os requisitos elencados para o OncoGrid são direcionados a estabelecer as seguintes funcionalidades ao ambiente: interoperabilidade, segurança, escalabilidade e interação amigável. Os demais componentes que possam ser executados sobre esta plataforma devem seguir os requisitos aqui elencados, podendo complementá-los com requisitos adicionais.

1. Redes de Comunicação: as intercomunicações do ambiente devem ser realizadas por redes de comunicação mundiais ou nacionais, sendo elas comerciais ou científicas. A rede que suportar a comunicação de uma instituição participante deve possuir PTT (Pontos de Troca de Tráfego) em comum com as dos demais.
2. Utilizar padronização estabelecida pela OGF: o ambiente deve ser concebido com componentes que implementem os padrões especificados pela OGF, assim preservando a interoperabilidade com componentes e ambientes já estabelecidos que utilizam este padrão;
3. Interoperabilidade com sistemas de segurança: a infra-estrutura de segurança do sistema deve ser compatível com os padrões e políticas de segurança já existentes nas instituições participantes;
4. Acesso aos dados de forma otimizada: o acesso aos dados do ambiente deve ser realizado com transparência e utilizar padronizações pré-definidas (XML, SQL, entre outros);
5. Suportar componentes físicos heterogêneos: o ambiente deve ser executado sobre a

componentização física altamente heterogênea e fracamente acoplada, suportando diferentes tipos de arquiteturas de processadores e sistemas operacionais;

6. Utilizar softwares livres: o ambiente deve ser concebido a partir de softwares livres mantendo a independência de tecnologia, possibilitando alterações e correção e também otimizando o custo de implantação;
7. Suportar alta escalabilidade: o ambiente deve suportar a inclusão e exclusão de participantes, componentes e recursos; mesmo sob essas condições ele deve conseguir reorganizar a sua topologia;
8. Interface Usuário: o ambiente deve proporcionar diferentes interfaces para interação dos usuários, atendo as necessidades de usuários mais experientes como linha de comando (bash, sh, command), e de usuários principiantes ou operadores, como aplicações cliente e interfaces Web.

4.1.2 Arquitetura OncoGrid

Após avaliarmos as funcionalidades necessárias para contemplar os requisitos apresentados, optamos por especificar a arquitetura do OncoGrid sobre o modelo composto por seis camadas funcionais, onde que cada camada agrupa os componentes lógicos ou físicos que possuem funcionalidades similares. Os componentes mantêm a interoperabilidade por meio de interfaces que padronizam a comunicação no ambiente, sendo baseadas na especificação WSRF (*Web Services Resource Framework*), criada pelo grupo OASIS (CZAJKOWSKI et al., 2004; OASIS, 2007).

A figura 4.1 apresenta a disposição das camadas funcionais da arquitetura do ambiente OncoGrid, sendo esta subdividida em cinco camadas: segurança, acesso do usuário, serviços de aplicação, serviços de grade, serviços de conexão de dados e recursos.

Figura 4.1: Arquitetura OncoGrid expressada sobre diagrama de camadas funcionais.

A seguir serão apresentadas as funcionalidades de cada camada.

Camada de Segurança

A padronização de segurança assumida pelo OGSA é o GSI (Grid Security Infrastructure), que também é uma especificação do OGF. Esta especificação indica o uso de uma infra-estrutura de chaves públicas (ICP) através de uma autoridade certificadora (AC). A autoridade certificadora fornece mecanismos para assinar, renovar e revogar os certificados dentro do ambiente.

As funcionalidades previstas nesta camada são: o gerenciamento de certificado, autenticação, autorização, delegação, proteção de mensagem e controle de acesso.

Camada de Acesso do Usuário

A camada de acesso do usuário é responsável por prover a interface de interação do usuário com o ambiente. Avaliamos que, no projeto OncoGrid seria útil o estabelecimento de interface Web, conhecidas como portal de grade, aplicações cliente baseadas em ambiente gráfico (KDE, GNOME, Windows) e interpretador de comandos (Linux: shell e sh,

Windows: command).

As aplicações Web possibilitam mobilidade e excluem a necessidade de instalar softwares adicionais podendo ser acessados de PCs convencionais, dispositivos móveis e estações de acesso à Internet. Neste caso, o usuário deve efetuar a autenticação por meio de uma entrada no navegador Web, possibilitando que a aplicação Web acessada realize os demais processos de segurança de usuários. Esta interface é indicada para a realização de muitas tarefas, como: entrada de pequenas quantidades de dados; execução de serviços; consultas e visualizações de informações. No entanto, o emprego desta interface para alguns processos de interação, como a carga manual de grandes quantidades de registros textuais, é inadequado devido às limitações dos navegadores Web.

As aplicações clientes não apresentam muita mobilidade, pois necessitam de um ambiente controlado e de instalação prévia para sua utilização. A vantagem de utilizar este tipo de interface para interação em grade se dá pelo fato desta apresentar melhor eficácia na execução de processos repetitivos.

A interface por interpretador de comando é bastante utilizada para pesquisadores e desenvolvedores de aplicativos, sendo bastante flexível para os usuários experientes.

Camada de Serviços de Aplicação

A camada de serviços de aplicação é responsável por disponibilizar de forma padronizada os serviços de aplicação disponíveis no OncoGrid.

Estruturalmente esta camada é composta por contêiner para disponibilização de serviços e de bibliotecas de desenvolvimento. Este conjunto forma um ambiente de execução comum, onde a comunicação entre as aplicações é sempre padronizada pelas especificações utilizadas nos serviços Web.

É possível a inclusão e exclusão de serviços dinamicamente nesta camada, sem que ocorram alterações no funcionamento do ambiente.

Camada de Serviços de Grade

A camada de serviços de grade é a responsável por integrar os componentes que realizam o gerenciamento e organização funcional do ambiente, executando operações de localização de serviços e de recursos, monitoramento dos recursos integrantes, alocação e desalocação de recursos, submissão de tarefas para processamento, meta-escalonamento.

Para suprir os requisitos necessários ao OncoGrid utilizamos as ferramentas *Grid Resource Allocation Management* (GRAM) para executar as tarefas de gerenciamento de execução de processos computacionais e *Monitoring and Discovery System* (MDS) para realizar as tarefas relacionadas a localização e indexação dos recursos e serviços.

As duas ferramentas são distribuídas em conjunto com o Globus, porém podemos acoplar outros componentes para agregar funcionalidades, como sistemas de monitoramento (Ganglia, MonALISA), meta-escalonadores (GridWay, CSF) e ambientes de programação distribuída (GridSolve, Ninf-G, MPICH-G2).

Camada de Serviços de Conexão de Dados

A camada de serviços de conexão de dados é a responsável por uniformizar o modo de acesso aos dados no ambiente. No caso do OncoGrid, é necessário transportar arquivos, acessar bancos de dados relacionais e setores de informações, como as bases de informações genéticas.

Para suprir os requisitos elencados foi necessário o emprego de duas ferramentas para manipulação de dados, ambas distribuídas no Globus: o GridFTP e o OGSA-DAI (*OGSA-Data Access and Integration*). O modo de operação destas ferramentas foi apresentado na seção 3.5.1.

Camada de Recursos

A camada de recursos é responsável por manter os recursos primários disponibilizados no ambiente. Seus componentes se encontram sobre a administração das instituições que integram o integrum. O conjunto de componentes desta camada é formado por recursos como bases de dados em diferentes distribuições (Oracle, DB2, PostgreSQL, MySQL, Microsoft SQL Server), recursos de processamento (aglomerados, computadores, micro-computadores) e setor de dados em sistemas de arquivos.

A componentização fundamental para a operação do OncoGrid é apresentada na figura 4.2. Os itens da figura que se encontram na camada de recursos são ilustrativos, sendo que o ambiente não necessita necessariamente conter estes recursos físicos específicos para estar em funcionamento.

Figura 4.2: Componentes fundamentais do OncoGrid

4.2 Proposta do Ambiente de Processamento Distribuído em Grade Aplicado ao OncoGrid

O ambiente proposto nesta seção tem como finalidade suprir o processamento em tarefas computacionalmente complexas ou que necessitem de recursos específicos para aplicação no campo da bioinformática em oncologia, podendo ser estendido a outras áreas que possuem características similares.

Devemos considerar que a arquitetura que será abordada a seguir possui compatibilidade com o ambiente OncoGrid, assim estendendo suas funcionalidades e requisitos já apresentados.

Para a apresentação desta proposta detalhamos os requisitos adicionais necessários para o suporte de processamento distribuído no OncoGrid, as estratégias para distribui-

ção de tarefas de processamento e por fim a modularização adicional necessária para o estabelecimento deste ambiente. Na sequência será abordada a descrição da aplicação elaborada para a identificação de similaridades genéticas em bases de expressões genéticas de *microarrays*.

4.2.1 Requisitos do Ambiente de Processamento do OncoGrid

O ambiente de processamento distribuído agrega os seguintes requisitos ao OncoGrid:

1. Suporte a Delegação: O ambiente deve possibilitar a delegação de credenciais a fim de que um processo em execução possa acessar outros recursos necessários em nome do usuário que solicitou o processo;
2. Suporte a tarefas distintas: O ambiente deve possibilitar a execução de tarefas de diferentes tamanhos, complexidades e modelos de programação (paralela, tarefa única, BoTs);
3. Suporte a processamento em ambiente heterogêneo: O ambiente deve possibilitar a execução de tarefas distribuídas sobre plataformas de hardware e sistema operacional heterogêneo, onde seja possível indicar as necessidades ou limitações das aplicações a serem processadas;
4. Suporte a escalonamento multinível: O ambiente deve suportar escalonamento em dois níveis: escalonamento global, permitindo a distribuição de tarefas no ambiente multi-institucional e escalonamento local, possibilitando a distribuição de tarefas em ambiente institucional;
5. Suporte a re-submissão de tarefas e tolerância a falhas: O sistema deve executar as suas tarefas mesmo estando em estado de falha, mantendo assim a qualidade no oferecimento de seus serviços;.
6. Suporte de operações em *Pipe*: O ambiente deve possibilitar que a saída de um processo alimente o próximo, assim facilitando a execução das sequências de operações que são realizadas em estudos genéticos pela bioinformática.

4.2.2 Arquitetura do Ambiente de Processamento do OncoGrid

Vamos descrever o funcionamento da arquitetura detalhando os componentes responsáveis pelos processos de execução de tarefas computacionais e apresentando os métodos

de distribuição de processos computacionais previstos.

Para suprir as necessidades requisitadas para o ambiente, devemos assumir que o ambiente possua as seguintes características e componentes:

- **Escalação e Distribuição:** O escalonamento de tarefas poderá ser realizado em dois níveis: meta-escalonamento, que é a distribuição de tarefas em ambiente global e escalonamento local, que é a distribuição das tarefas internamente dentro das instituições participantes. Para a distribuição de tarefas globais será utilizada a ferramenta GridWay. O escalonamento de tarefas nas redes locais das instituições participantes, pode ser realizado utilizando o próprio gerenciador de tarefas dos sistemas operacionais das extrações, ou gerenciadores de aglomerados de computadores que possuem compatibilidade com ambientes baseados no Globus, como o Torque-PBS por exemplo;
- **Submissão de Tarefas:** A ferramenta GRAM fica responsável pelo gerenciamento das tarefas nos recursos de processamento, sendo a principal interface do meta-escalonador com os recursos. Todos os comandos direcionados aos processos computacionais executados nas estações são enviados por meio deste componente;
- **Bibliotecas de Programação:** o ambiente deve possuir suporte a bibliotecas para programação paralela, como o MPICH-G2, que é a implementação da especificação da biblioteca MPI com suporte ao sistema de segurança empregado no Globus. As máquinas que participam do ambiente devem estar equipadas com o ambiente de execução do JAVA o JRE (*Java Runtime Environment*), devido ao fato de ser uma plataforma de desenvolvimento utilizada para implementação de sistema distribuído;
- **Localização de Recursos:** O ambiente deve possuir um sistema de informações que disponibilize os dados sobre o estado e localização de recursos e serviços para o meta-escalonador. Optamos por empregar o MDS provido pelo Globus. Para complementar as suas funcionalidades, estabelecemos que o monitoramento das estações será realizado de forma dinâmica, empregando componentes de monitoramento nos recursos que disponibilizarão as informações do estado das estações da grade em tempo real. Para realizar esta função, empregamos o monitor Ganglia;
- **Acesso a Dados:** As operações que realizam movimentação de dados devem utilizar as ferramentas padrão empregadas no Globus (GridFTP, OGSA-DAI). Adicionalmente a estas se faz necessário o emprego do sistema para transferência de dados

confiável RFT, por ser requisitado nas funções de gerenciamento de processos realizadas pelo GRAM.

Mediante as funcionalidades apresentadas, nos cabe esclarecer o motivo para a adoção das ferramentas para gerenciamento de execução e de meta-escalonamento GridWay e Torque-PBS.

O Torque-PBS é uma versão do *Portable Batch System*. Esta ferramenta adere ao sistema proposto porque apresenta facilidades de gerenciamento e possui compatibilidade com o GRAM.

O GridWay é uma ferramenta desenvolvida pela *Globus Alliance* que é o mesmo consórcio que desenvolve o Globus, assim possuindo forte integração com os ambientes nela baseados. Outro fato favorável a esta adoção é a possibilidade de utilizar aplicações desenvolvidos em JAVA. Por ser uma ferramenta desenvolvida especialmente para o ambiente de grade computacional, apresenta políticas de escalonamento melhor adaptadas para este tipo de ambiente, quando relacionadas às políticas utilizadas pelos escalonadores adaptados de ambientes de aglomerados de computadores para grades computacionais, como é o caso do Condor-G.

Existe a perspectiva da utilização de ferramentas como o Condor que permite explorar os recursos computacionais que se encontram ociosos nas organizações que participam do OncoGrid de forma oportunista. Esta atividade é vista por muitas instituições como um fator negativo, por ser um modelo de computação distribuída extremamente pervasivo.

Para esclarecer a distribuição dos componente do ambiente de processamento abordamos a arquitetura OncoGrid novamente sob o modelo de camadas funcionais. Na figura 4.3 apresentamos a arquitetura do projeto OncoGrid, agora contemplando adicionalmente os componentes utilizados para o estabelecimento do ambiente de processamento distribuído. Os componentes que participam do processamento distribuído estão destacados na cor cinza.

A disposição das camadas indica a organização funcional, mas não implica que uma determinada camada somente se comunique com a imediatamente superior ou inferior, por exemplo; uma estação de trabalho conectada ao ambiente pode submeter uma tarefa para processamento, assim acessando as camadas de segurança, acesso do usuário, serviços de grade e de recursos.

Os blocos dispostos nas camadas indicam os componentes do ambiente. Na figura 4.3 é apresentado apenas um componente de cada, porém estes podemos possuir N réplicas.

Figura 4.3: Arquitetura do ambiente de processamento do OncoGrid representada no modelo de camadas funcionais.

Por exemplo, cada instituição participante pode ter um contêiner de serviço, e um serviço que está no contêiner da instituição **A** pode requisitar um serviço que está no contêiner da instituição **B** para realizar suas tarefas, assim como o serviço de informação da grade pode identificar os recursos que estão em **A** e em **B**. Por sua vez, o gerenciador de recursos pode utilizar recursos físicos para processamento das duas localidades para executar uma determinada tarefa.

4.2.3 Métodos de Distribuição de Tarefas

São previstos três métodos para a distribuição de tarefas de processamento no ambiente: a distribuição de tarefas independentes, a distribuição de lotes de tarefas sobre conjunto de dados distintos e distribuição de tarefas paralelas. Para abordar o funcionamento destes processos no ambiente serão apresentados três diagramas que descrevem as etapas destes processos.

Assumimos que para realizar algum dos procedimento que serão descritos a seguir, o usuário realizou todas as etapas relacionadas com a segurança, ou seja que tenha um certificado de usuários válido, que tenha gerado um certificado Proxy e se autenticado no ambiente.

Tarefas Independentes

A submissão de uma tarefa independente para processamento em ambiente de grade é o método mais simples entre os apresentados nesta proposta. O benefício de executar uma tarefa independente no ambiente de grade ao invés de localmente é visto quando uma tarefa requisita um recurso específico para seu processamento, que não existe na instituição de origem. Outro fato favorável é a possibilidade de aproximar a execução dos dados a serem processados.

No processo apresentado na figura 4.4, o usuário envia uma tarefa para execução por meio de um arquivo de descrição de tarefa (1); o meta-escalador GridWay consulta o sistema de informações do ambiente MDS para descobrir qual recurso será disponibilizado para a execução da tarefa(2); após receber a identificação do recurso, o GridWay prepara a execução da tarefa no recursos, processo que consiste em enviar o código binário a ser executado e os dados caso necessário, por meio do GridFTP (3, 4); após o preparo para a execução da tarefa o GridWay solicita para o gerente de execução do ambiente GRAM a coordenação da tarefa junto ao recurso determinado(5); o GRAM inicia a execução e mantém a comunicação com o recurso para possuir a informação do estado de execução do processo computacional (6), repassando estas informações para o GridWay (5); caso seja necessário, o recurso pode acessar dados para processamento em outros recursos de dados por meio do GridFTP (7,8); o estado do recurso de processamento é sempre atualizado junto ao MDS (9); o resultado da execução da tarefa é enviado para o usuário por meio do GridFTP (4, 3). É comum que o resultado seja um arquivo contendo os dados de saída da execução, cujo o nome é indicado na descrição da tarefa.

Figura 4.4: Submissão de uma tarefa independente no OncoGrid.

No caso de todos os recursos estarem ocupados a tarefa fica em um fila, aguardando a disponibilização de algum recurso.

Lotes de Tarefas

A submissão de tarefas em lote pode ser iniciada de duas formas. A primeira é utilizando um único arquivo de descrição da tarefa, indicando para o meta-escalonador quantas instâncias idênticas ele deve iniciar. Cada tarefa iniciada possui um identificador de processo que pode ser utilizado pela aplicação para controlar a sequência do processamento e os dados que serão utilizados pelo processo. Por exemplo, se iniciarmos quatro tarefas segundo este método podemos dividir o segmento de dados a ser processado em quatro partes iguais e distribuir um quarto da porção do processamento para cada processo.

A segunda forma é utilizar um conjunto de arquivos de descrição da tarefa, alterando apenas a entrada de dados. Por exemplo, se possuímos um conjunto de sequências genéticas Z que devemos comparar com quatro banco genéticos A, B, C, D, então necessitamos

criar quatro arquivos de descrição de tarefas com entradas de dados distintas, sendo, Z-A, Z-B, Z-C e Z-D.

Quando abordamos o processo (1) da figura 4.5 o usuário poderá estar utilizando uma destas duas estratégias para submeter as tarefas ao ambiente.

Figura 4.5: Submissão de um lote de tarefa no OncoGrid.

No processo apresentado na figura 4.5, o usuário submete um lote de tarefa para execução por meio de arquivo(s) de descrição(ões) de tarefa(s) (1); após receber o conjunto de descritores das tarefas, o GridWay consulta o sistema de informações do ambiente MDS para descobrir quais os recursos disponíveis para a execução das tarefas (2); realizada a identificação do conjunto de recursos, o GridWay prepara as tarefas nos recursos por meio do GridFTP (3, 4); após a preparação das tarefas nos recursos, o GridWay solicita ao GRAM a coordenação da tarefa junto ao recurso determinado (5); o GRAM inicia a execução e mantém a comunicação com os recursos para conhecer o estado de execução das tarefas (6), sempre atualizando estas informações junto ao GridWay (5); caso seja neces-

sário, as tarefas podem acessar dados em recursos de dados por meio do GridFTP (7,8); os estados dos recursos são atualizados junto ao MDS periodicamente (9); o conjunto de arquivos contendo os resultados das execuções das tarefas são retornados ao usuário por meio do GridFTP (4, 3).

No caso deste modelo de distribuição de tarefas, o emprego de parâmetros ou requisitos para a execução das tarefas pode proporcionar ganho de desempenho, como selecionando arquiteturas mais apropriadas para a execução das tarefas ou otimizando a estratégia de acesso aos dados, de modo que as estações selecionadas para processamento adquiram os dados que estejam mais próximos a ela. Este processo não é automático e deve ser indicado no arquivo descritor de tarefas. Para maiores detalhes sobre os parâmetros de configuração consulte a apêndice B.

Abordando os recursos de processamento apresentados na figura 4.5, devemos considerar que estes equipamentos são heterogêneos, como, conjunto de estações de trabalho, aglomerados com escalonadores locais, sistemas operacionais e arquitetura de processadores distintas. Se torna imprescindível assim especificar os requisitos da tarefa no seu arquivo descritor, como a escolha do sistema operacional e da arquitetura do processador.

Tarefas Paralelas

Uma tarefa paralela é executada como um único processo, no instante que o processo é submetido a execução, é gerado, a partir deste, um conjunto determinado de sub-processos. Cada item deste conjunto fica responsável por realizar uma parte determinada dos cálculos. É uma característica deste modelo de distribuição de processamento a troca de mensagens entre os processos em execução. Após o término da execução de todos os sub processos, o processo primário que originou os demais tem a responsabilidade de agrupar os resultados em um único.

Esta tarefa é normalmente executada por um aglomerado de computadores ou computador paralelo, projetados especificamente para estes fins. A comunicação entre os processos é realizada por interfaces para programação paralela, como o MPI, sobre infra-estruturas de redes de alta velocidade. Em um ambiente de grade computacional podemos disponibilizar este tipo de equipamento como recurso da grade. O intercâmbio de mensagens entre estes recursos é realizados por meio da infra-estrutura de comunicação do ambiente de grade.

Figura 4.6: Submissão de uma tarefa paralela no OncoGrid.

No processo apresentado na figura 4.6, o usuário submete uma tarefa paralela para execução por meio de um arquivo de descrição de tarefa (1); o GridWay recebe a descrição da tarefa e realiza a consulta ao MDS descobrindo quais os recursos com as características apropriadas estão disponíveis (2); após identificar os recursos que serão utilizados, o GridWay prepara a execução das tarefas nos recursos por meio do GridFTP (3, 4); após a preparação da tarefa para execução o GridWay solicita para o gerente de execução do ambiente GRAM a coordenação da tarefa junto ao recurso determinado (5); o GRAM inicia a execução da tarefa nos recursos (6); a próxima etapa consiste na criação dos sub-processos e o envio dos mesmos aos componentes de processamento (nós) dos aglomerados por comunicação MPI, esta etapa é realizada pela própria tarefa, e mantém a integração com os outros aglomerados que participam do processamento (10); o estado da execução da tarefa principal é mantido pelo GRAM(6); caso seja necessário, a tarefa pode aces-

sar dados em recursos de dados por meio do GridFTP (7,8); os estados dos recursos de processamento são sempre atualizados junto ao MDS (9); após o término da execução de todos os sub-processos a tarefa principal agrupa os resultados e o envia para o usuário por meio do GridFTP (4, 3).

Quando se trata da execução de aplicações paralelas, devemos considerar a capacidade e a qualidade dos serviços de comunicações envolvidas, assim como, a velocidade dos processadores e a quantidade de memória disponível para cada processador, porque estes fatores delimitam a execução dos sub-processos paralelos. Neste caso, o recurso com menor capacidade de processamento ou o segmento de rede com menor capacidade de comunicação determina o tempo total de execução da tarefa.

4.3 Concepção de Aplicação de Bioinformática para Validação do Ambiente

A validação experimental do ambiente de processamento aplicado ao OncoGrid foi realizada mediante a utilização de uma aplicação de bioinformática. Esta aplicação tem como objetivo localizar os indivíduos que possuem perfis genéticos semelhantes a partir da análise das suas expressões genéticas.

O estudo das expressões genéticas de um grupo de pacientes, associadas às avaliações clínicas dos mesmos, pode responder algumas questões biológicas bastante relevantes, quando abordando a Oncologia, tais como: “O estudo do perfil genético consegue determinar o prognóstico dos pacientes?”, “O perfil genético, comparado com a evolução do tratamento auxiliado pela análise de imagens radiológicas consegue determinar novos tratamentos ou indicar predições clínicas?”; “Existe um padrão genético entre pacientes com o mesmo diagnóstico?”.

O benefício da aplicação proposta é atribuída à possibilidade de identificar qual paciente de um grupo possui o perfil genético mais semelhante em relação ao estudado. Conseqüentemente, ela também possibilita identificar quais são os pacientes do grupo avaliado que possuem os perfis genéticos mais semelhantes, para o que precisamos realizar o cruzamento de todas as expressões contra todas.

A aplicação proposta consiste de dois módulos de software. O primeiro é responsável por realizar a comparação entre duas expressões genéticas. Este módulo pode ser executado distribuídamente, como um lote de tarefas onde que cada tarefa compara um par de expressões, sendo denominado de Identificador de Semelhanças Genéticas Local

(ISGL). O segundo módulo é o responsável pela análise dos resultados provindos do ISGL, consolidando-os em um resultado classificatório global, sendo denominado Identificador de Semelhanças Genéticas Global (ISGG).

4.3.1 Conjunto de Dados Utilizado para Validação

O conjunto de dados utilizado como base de testes foi adquirido de uma base de dados pública disponibilizada pelo Projeto para identificação do perfil molecular de Linfomas e Leucemias (*Lymphoma/Leukemia Molecular Profiling Project*) (ALIZADEH et al., 2000).

A extração dos dados das imagens foi realizada pelo projeto originário, com o auxílio do software ScanAlyze, que é disponibilizado gratuitamente (EISEN, 1999). Os arquivos dos experimentos foram adquiridos em formato DAT, que é um arquivo texto formatado com tabulações.

As informações utilizadas dos arquivos DAT neste trabalho são: o endereço do *spot* e os valores dos canais ch1 correspondente à cor verde, e ch2, correspondente à cor vermelha (Cy3 e Cy5, respectivamente).

Os módulos da aplicação responsáveis pela leitura de arquivos foram projetados para compreender apenas os arquivos DAT, necessitando de adaptações para adquirir dados de arquivos com outros formatos.

4.3.2 Módulo Identificador de Semelhança Genética Local

Este módulo é projetado para identificar as semelhanças entre duas expressões genéticas provindas de experimentos similares, ou seja, expressando os mesmos genes no mesmo endereço de *spots*.

Para o desenvolvimento da lógica empregada no módulo ISGL consideramos os seguintes pontos chave:

- A expressão do canal Cy3 (canal verde, que representa a amostra do mRNA extraído do tecido saudável) representa o eixo das abcissas, representado pela variável X;
- A expressão do canal Cy5 (canal vermelho, que representa a amostra do mRNA extraído do tecido doente) representa o eixo das ordenadas, representado pela variável Y;

- Desta forma temos dois níveis de expressão para cada gene identificado por um *spot*: (X_1, Y_1) , que são as coordenadas da expressão original e (X_2, Y_2) , que são as coordenadas da expressão comparada.

Com esta informação podemos calcular a distância entre os pontos no eixo cartesiano. Assim, quanto menor é a distância entre os pontos, maior é a similaridade entre as expressões comparadas. A figura 4.7 apresenta os pontos de referência da expressão de um gene em duas expressões genéticas e a distância euclidiana entre estes dois pontos.

Figura 4.7: Distância euclidiana de um gene expressados em dois experimentos de *micro-array*.

A distância euclidiana (dE) entre os pontos A e B é dada pela equação 4.1.

$$dE(A,B) = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \quad (4.1)$$

Para esclarecer o funcionamento deste módulo de software, a tabela 4.1 apresenta o resultado da execução do algoritmo comparando uma expressão indicada pelo usuário com outras duas. Os dados contidos na tabela são: *spot*, que indica o endereço do *spot* na lâmina, possibilitando localizar o gene de referência; Cy3o, equivalente a X_1 da equação, que indica a incidência do gene no tecido saudável na expressão genética original; Cy3c,

equivalente a X_2 da equação, que indica a incidência do gene no tecido saudável na expressão genética comparada; Cy5o, equivalente a Y_1 da equação, que indica a incidência do gene tecido afetado pelo câncer na expressão genética original; Cy5c, equivalente a Y_2 da equação, que indica a incidência do gene no tecido afetado pelo câncer na expressão genética comparada; e por fim a distância calculada, indicada por dEc_1 e dEc_2 .

<i>spot</i>	Cy3o	Cy5o	Cy3c ₁	Cy5c ₁	Cy3c ₂	Cy5c ₂	dEc_1	dEc_2
0	99	50	350	238	78	154	313,6	106,1
1	87	46	407	265	87	171	387,76	125
2	75	40	512	338	88	177	528,94	137,62
3	89	45	537	360	99	192	547,66	147,34
4	98	57	394	262	129	300	360,06	244,97
5	89	53	468	330	137	342	469,44	292,96
6	69	34	490	301	113	254	498,53	224,36
7	63	32	405	187	108	205	375,49	178,76
8	65	35	642	265	140	232	621,15	210,79
9	65	34	614	253	138	239	591,07	217,61

Tabela 4.1: Cálculo das distâncias euclidianas entre a expressão genética original **o** e as comparadas **c₁** e **c₂**.

Desta forma, para cada expressão comparada, será retornado um vetor contendo as distâncias euclidianas entre os genes da expressão original e das comparadas, indicado na tabela pela coluna dEc_1 e dEc_2 .

Podemos observar na tabela 4.1 que a expressão genética **c₂** possui valores mais próximos aos da expressão original que a **c₁**. Ou seja, menores distâncias indicam maior similaridade genética pontual.

4.3.3 Módulo Identificador de Semelhança Genética Global

O módulo identificador de semelhança genética global ISGG é um componente de software disposto de forma centralizada no ambiente, recebendo os resultados dos cálculos das distâncias genéticas providas das análises realizadas pelo módulo ISGL. O ISGG realiza processos para a classificação das distâncias calculadas entre um conjunto de expressões genéticas em relação a uma expressão indicada pelo usuários, denominada expressão original. A figura 4.8 apresenta o modo de operação do módulo ISGG utilizando os dados de saída do módulo ISGL.

Figura 4.8: Modo de operação do módulo ISGG recebendo os dados de saída do módulo ISGL.

Para melhor esclarecer, dispomos de E expressões genéticas distribuídas no OncoGrid, cada uma contendo a quantidade de G genes expressados. Assim, a resposta do identificador de semelhança genética local será uma matriz de tamanho (G, E) , aqui nomeada de matriz resposta e representada por $R(G, E)$, a tabela 4.2 apresenta um exemplo de uma matriz resposta, onde cada coluna E representa as distâncias genéticas de uma expressão e cada linha G representada as distância de um gene em todas as expressões. Neste sistema utilizamos a matriz como dois conjuntos de vetores, as linhas G_i e as colunas E_j , porque todos os cálculos seguintes serão realizados na direção horizontal ou vertical.

	Exp 1	Exp 2	Exp 3
Gene 1	799	220	178
Gene 2	673	779	105
Gene 3	940	725	672
Gene 4	580	727	2
Gene 5	240	451	110
Gene 6	235	825	899

Tabela 4.2: Matriz resposta.

Uma vez obtida a matriz $R(G_i, E_j)$, realizamos a classificação em cada linha G_i . Este processo consiste em atribuir a classificação seqüencial iniciando em 1, indicando o gene mais próximo a i , e o mais distante, formando assim a matriz classificação representada por $C(G, E)$. A tabela 4.3 apresenta a matriz classificação obtida através da matriz resposta apresentada na tabela 4.2.

	Exp 1	Exp 2	Exp 3
Gene 1	3	2	1
Gene 2	2	3	1
Gene 3	3	2	1
Gene 4	2	3	1
Gene 5	2	3	1
Gene 6	1	2	3

Tabela 4.3: Matriz classificação.

Sobre todos os valores da matriz $C(G, E)$, aplicamos a equação 4.2 para fortalecer os resultados classificatórios. Com este processo conseguimos elevar o valor de classificação dos genes mais próximos ao comparado, e enfraquecer quanto mais distante este seja, assim gerando a matriz $P(G, E)$.

$$P(G, E) = \frac{1}{C(i, j)^4} \quad (4.2)$$

A tabela 4.4 apresenta a aplicação da equação 4.2 nos dados da matriz $C(G, E)$ apresentada na tabela 4.3. Por fim aplicamos a somatória de todos os elementos das colunas E da matriz $P(G, E)$, os maiores valores indicam as expressões com maior grau de semelhança.

	Exp 1	Exp 2	Exp 3
Gene 1	0,11	0,25	1
Gene 2	0,25	0,11	1
Gene 3	0,11	0,25	1
Gene 4	0,25	0,11	1
Gene 5	0,25	0,11	1
Gene 6	1	0,25	0,11
Σ	1,97	1,08	5,11

Tabela 4.4: Matriz pontuação.

A somatória apresentada na última linha da tabela 4.4 indica as expressões do grau de semelhança geral da expressão, sendo a expressão Exp 3 a mais semelhante, seguida pela Exp 1 e Exp 2.

4.3.4 Modelo de Distribuição do Algoritmo

A disposição da aplicação apresentada na seção 4.3 no ambiente OncoGrid engloba a utilização de diferentes recursos e camadas deste ambiente. O módulo identificador de semelhança genética local ISGL é distribuído para processamento como um lote de tarefas conforme apresentado na figura 4.5. O módulo ISGG, é disposto como uma serviço na camada de serviços de aplicação.

Figura 4.9: Exemplo de distribuição da aplicação no ambiente OncoGrid

Na figura 4.3 expomos um diagrama que representa a execução distribuída da aplicação no ambiente proposto. Consideramos que usuário que solicita a execução do serviço está autenticado no ambiente, assim cumprindo todos os requisitos fundamentais de acesso.

O primeiro passo consiste da entrada da expressão original para ser comparada com

os demais. Posteriormente é criado um lote de tarefas ISGL para realizar o cruzamento das informações da expressão genética original com todas as comparadas. As tarefas ISGL são submetidas por meio do meta-escalador, que consulta as informações sobre os recursos disponíveis e as envia para processamento no ambiente. Os resultados das tarefas retornam para o meta-escalador, que os retorna para o usuário ou para o serviço que solicitou a execução. A partir dos resultados obtidos na execução distribuída do módulo ISGL o módulo ISGG realiza a classificação geral indicando as expressões comparadas por ordem de semelhanças.

4.4 Resumo do Capítulo

A arquitetura base do ambiente de grade computacional OncoGrid foi abordada sob um modelo contendo seis camadas funcionais. Cada camada engloba os componentes lógicos que atuam no mesmo nível de prestação de serviço.

As camadas estabelecidas foram: camada de segurança, camada de acesso do usuário, camada de serviços de aplicação, camada de serviços de grade, camadas de conexão de dados e camada de recursos. Utilizando este modelo de especificação, propomos um ambiente para processamento de dados em larga escala, com suporte à distribuição geográfica.

O ambiente de processamento distribuído integra funcionalidades dos componentes para realização dos seus serviços, incluindo identificação de informação de recursos, submissão de tarefas e compatibilidade com a segurança do ambiente. O modo de operação do ambiente de processamento suporta as distribuições de três tipos de tarefas: tarefas independentes, que consistem em tarefas únicas enviadas para processamento no ambiente; lotes de tarefas, que consistem em um grupo de tarefas similares executando operações sob um conjunto de dados distintos; e tarefas paralelas, que consistem em tarefas que são seccionadas em sub-processos que no final da sua execução tem seus resultados consolidados pelas tarefas principais.

Para a validação do ambiente foi projetada uma aplicação no campo da bioinformática em oncologia que realiza a identificação de similaridades genéticas comparando uma única expressão com um grupo de expressões distribuídas na grade, providas de experimentos que utilizaram lâminas com configurações similares. Esta aplicação é composta por dois módulos: o identificador de semelhanças genéticas local (ISGL), que realiza as comparações entre um par de expressões genéticas e o identificador de semelhanças genéticas global (ISGG), que utiliza os dados calculados pelo ISGL para ordenar por grau de semelhança

o grupo de expressões.

A distribuição desta aplicação é realizada como um lote de tarefas ISGL. A partir dos resultados obtidos o ISGG realiza a classificação geral de modo centralizado.

5 *Avaliação Experimental*

Para apresentar os desenvolvimentos realizados neste trabalho vamos abordar a implementação da arquitetura base do ambiente OncoGrid. Na sequência abordaremos o estabelecimento do ambiente de processamento em grade computacional proposto, que é acoplado à arquitetura do OncoGrid. Por fim vamos apresentar a implementação dos módulos de aplicação de identificação de similaridades genéticas entre grupos de expressões genéticas de *microarray*.

A implementação completa do ambiente computacional proposto exige uma grande demanda de esforços, o que foge do escopo deste trabalho. Por este motivo, optamos por construir apenas a parte da proposta em ambiente simulado que ofereça o suporte necessário para a execução da aplicação de bioinformática, sendo o suficiente para a realização das validações necessárias.

A seção 5.1 apresenta a implementação do ambiente OncoGrid. A seção 5.2 apresenta a implementação realizada do ambiente de processamento proposto para o OncoGrid. Por fim a seção 5.3 apresenta como foi implementada a aplicação de bioinformática utilizada para a avaliação do ambiente.

5.1 Implementação do Ambiente OncoGrid

O projeto OncoGrid foi iniciado como um projeto piloto que foi desenvolvido entre o Laboratório de Sistemas Integráveis (LSI-EPUSP) e o Núcleo de Telessaúde da Universidade Federal de Pernambuco (NUTES). O trabalho de implementação realizado neste projeto consistiu no estabelecimento das principais camadas funcionais e componentes que proporcionaram o correto funcionamento do ambiente.

Para a implementação deste ambiente, optamos por utilizar a ferramenta Globus Toolkit 4.0.X, porque esta ferramenta possui maior flexibilidade, modularidade e escalabilidade em relação às demais pesquisadas (veja seção 3.5.4, página 47).

O projeto OncoGrid consistiu na componentização das camadas funcionais abordadas na seção 4.1.2, como é descrito a seguir:

- Camada de Segurança: Foi implementada uma autoridade certificadora própria denominada OncogridCA para gerenciamento dos certificados de usuários e servidores. Também foi realizada a integração do servidor Myproxy para gerenciamento dos certificados Proxy temporários, permitindo a recuperação destes certificados de qualquer máquina integrada à arquitetura. O processo de delegação é realizado a partir dos certificados Proxy dos usuários. As credenciais são delegadas no início dos processos que envolvem o usuário e a responsabilidade da delegação fica a cargo dos serviços de grade, como o GRAM, por exemplo;
- Camada de Acesso do Usuário: A interface por submissão por linha de comando é adicionada por padrão no ambiente porém a máquina que o usuário utiliza deve ser integrante do ambiente. Adicionalmente, incluímos um servidor de aplicação externo com acesso ao ambiente para prover a interface Web aos usuário e não empregamos nenhuma interface como aplicação cliente que realiza interação com a grade;
- Camada de Serviços de Grade: Nesta etapa da implementação não incluímos nenhum componente nesta camada;
- Camada de Serviços de Conexão de Dados: Nesta camada implementamos o GridFTP para transferência de dados compatível com a infra-estrutura de segurança do ambiente. Diversos processos da grade utilizam este componente para a realização de suas atividades, como é o caso da submissão de tarefas pelo meta-escalonador;
- Camada de Recursos: Nesta camada incluímos sistemas de arquivo para trabalhar como recursos de dados, que foram utilizados para validações. Posteriormente incorporamos um sistema gerenciador de base de dados para validação da integração de setores de registros de dados por meio da infra-estrutura.

A figura 5.1 apresenta a disposição dos componentes empregados na implementação inicial do projeto OncoGrid, novamente utilizando a especificação em camadas funcionais. Ainda neste figura podemos verificar a distribuição dos componentes nas instituições participantes (LSI-EPUSP e NUTES-UFPE).

Figura 5.1: Implementação inicial da arquitetura OncoGrid apresentada sob o modelo de camadas funcionais.

A tabela 5.1 complementa as informações apresentada na figura 5.1, abordando a configuração básica dos equipamentos utilizados para a implantação da arquitetura de testes. Para interconexão externa entre as instituições utilizamos a infra-estrutura da Rede Nacional de Ensino e Pesquisa. As duas extremidades receberam uma porta de rede com a velocidade de 1 Gigabit (Gb).

	LSI-EPUSP	Nutes-UFPE
Nº de Servidores	2	2
Processador	2 x Xeon 3.0 GHz	Pentium D 2.8 GHz
Memória RAM	2 Gb	2 Gb
Discos	2 x 80 SATA 1	2 x 80 SATA 2
Interface de Rede	2 Gb Ethernet	2 Gb Ethernet

Tabela 5.1: Componentes físicos aplicados na implementação piloto do OncoGrid

5.1.1 Autenticação no OncoGrid

Para que um usuário realize qualquer operação no ambiente é indispensável que este esteja devidamente autenticado. Por este motivo dedicamos esta seção para abordar como foi estabelecido o sistema de autenticação no OncoGrid.

O processo de autenticação de usuários no ambiente OncoGrid exige que os usuários tenham um certificado válido e que a estação utilizada seja participante do ambiente, possuindo também um certificado de estação válido. No caso de acesso via um portal de grade, a estação que hospeda o portal deve possuir integração com a grade. Em ambientes baseados no Globus podemos contratar uma Autoridade Certificadora, como Thawte ou VeriSign, ou mater uma própria AC. Neste caso optamos por manter uma autoridade certificadora denominada OncoGridCA devido ao fato do ambiente ser utilizado apenas para validações.

O ambiente OncoGrid prevê a funcionalidade de autenticação única. Uma vez autenticado, o usuário não necessitam redigitar a sua chave para acessar os recursos do ambiente, isso dentro de um limite de tempo pré determinado. Para que ocorra o processo de autenticação única é gerado um certificado temporário com tempo de vida curto (por padrão 12 horas), denominado certificado Proxy, já abordado na seção 3.5.1. O usuário pode destruir a credencial Proxy a qualquer momento, impedindo o uso inapropriado. O processo de delegação consiste na possibilidade do usuário delegar a sua credencial Proxy para um serviço ou estação da grade, onde este poderá executar procedimentos em nome do usuário.

Para o gerenciamento dos certificados Proxy dos usuários foi agregada ao ambiente do OncoGrid uma ferramenta para armazenamento e gerenciamento destes certificados denominada MyProxy. Esta ferramenta armazena o certificado Proxy do usuário permitindo que o mesmo possa ser recuperado de qualquer ponto do ambiente. As credenciais armazenadas na base de dados do MyProxy têm a validade de sete dias. Quando o usuário solicita uma credencial ao servidor é gerada uma nova credencial baseada na armazenada, porém com validade de doze horas. Os parâmetros de validade dos certificados Proxy podem ser alterados, mas nesta implementação foram mantidas as padronizações originais.

Figura 5.2: Processo de autenticação estabelecido no OncoGrid (ALVES et al., 2008).

A figura 5.2 apresenta o modo de operação para autenticação no ambiente OncoGrid. O processo acontece como descrito: o usuário solicita sua autenticação ao ambiente por meio de um portal da grade ou de uma estação integrada ao ambiente. Quando a credencial Proxy não existir no servidor MyProxy, é gerada uma nova credencial que é armazenada em uma base de dados do MyProxy e o certificado Proxy é devolvido à estação ou portal que o usuário está acessando no momento. A partir deste ponto o usuário passa a ter acesso ao ambiente. Caso já exista a credencial Proxy deste usuário armazenada, o servidor MyProxy apenas a recupera da sua base de credenciais e entrega à estação.

5.2 Arquitetura de Processamento do OncoGrid

A partir deste ponto do texto vamos relatar as implementações realizadas referentes à proposta do ambiente de processamento distribuído em grade computacional, que consistiu na integração dos componentes necessários para: gestão das informações da grade, gerenciamento de transferência de dados, controle da execução das tarefas nos recursos e estabelecimento do escalonamento de tarefa de forma global.

Para a implementação da infra-estrutura do ambiente foi necessário instalar em todos os equipamentos a ferramenta Globus, visto que todos eles necessitam acessar os componentes da camada de segurança, assim necessitando das ferramentas cliente para acesso de usuários e dos componentes para suportar as conexões e executar os serviços de gestão da grade. Assim, devemos considerar que todas as estações possuem um certificado válido no OncoGrid, e que a ferramenta Globus esteja implantada.

Assumimos como base a arquitetura e os componentes empregados no desenvolvido do projeto piloto OncoGrid, que foi executado pela parceria entre o LSI-EPUSP e o NUTES-UFPE. Para concretizar esta implementação, adicionamos os componentes descritos na seção 4.2.2, página 64, formando a arquitetura apresentada na figura 4.3, página 67.

Na implementação realizada para validação não integramos o ambiente de processamento implementado com o desenvolvido no projeto piloto OncoGrid. No entanto utilizamos os componentes de segurança e padrões empregados no desenvolvimento do projeto piloto OncoGrid. Devido a estes fatos, torna-se possível integrar estes ambientes, basta estabelecer a conexão entre os sistemas de informações da implementação realizada neste trabalho com a do projeto piloto OncoGrid e configurar as permissões de usuários.

Para facilitar a compreensão desta seção, abordamos em primeira instância os recursos físicos empregados para o ambiente de processamento. Baseado nas informações sobre a componentização física, vamos indicar como foi realizada a organização dos componentes de software empregados para o estabelecimento do ambiente de processamento do OncoGrid.

5.2.1 Recursos Físicos

Para a implementação dos testes e validações do ambiente de processamento do OncoGrid utilizamos quatro estações de trabalho convencionais como recursos físicos. Os equipamentos se encontram nos núcleos de telemedicina do LSI-EPUSP e são interconec-

tadas por um chaveador de pequeno porte com velocidade de 100 Mb. A descrição dos equipamentos está apresentada na tabela 5.2.

Nome	grid01	grid02	grid03	aedes-aegipty
CPU	Pentium 4 HT x86 3.00GHz	Pentium 4 HT x86 3.00GHz	Pentium 4 x86 2.40GHz	Intel Core 2 Duo x86_64 2.00GHz
MEM RAM	1 GB	2 GB	768 MB	2 GB
Discos	30 GB IDE	30 GB IDE	30 GB IDE	120 GB SATA
NIC	100Mb	100Mb	100Mb	100Mb
SO	Linux i686 2.6.22-14 SMP	Linux i686 2.6.22-14 SMP	Linux i686 2.6.22-14 SMP	Linux i686 2.6.24-19 SMP

Tabela 5.2: Componentes físicos aplicados no ambiente de processamento implementado.

Os processadores das estações grid01 e grid02 são equipados com a tecnologia HT (*hyperthreading*), sendo identificados no sistema operacional com dois núcleos cada. A estação aedes-aegipty é equipada com um processador de dois núcleos. Sendo assim, o ambiente é composto por sete unidades de processamento, possibilitando a execução de sete tarefas simultâneas.

5.2.2 Sistema de Informações

O sistema de informações do ambiente aplicado ao OncoGrid é composto por um gerenciador de informações e um agente de monitoramento das estações.

O processo de gerenciamento de informações consiste na coleta das informações sobre os serviços e recursos em cada estação que compõe o ambiente e na disponibilização destas informações aos usuários e a outros serviços.

O gerente de informações utilizado foi o MDS, que é uma ferramenta distribuída em conjunto com o Globus. É necessário implantar esta ferramenta em todos os recursos do ambiente, sendo que todos os MDS locais reportam as suas informações para um MDS centralizador, que é o responsável por gerenciar a hierarquia da grade, fornecendo as informações de forma a abranger todo o ambiente. Este componente é conhecido como indexador padrão da organização e realiza a função de organizar a topologia lógica da

OV.

Para coletar as informações sobre o estado dos recursos físicos utilizamos a ferramenta Ganglia Monitor. Esta ferramenta não é distribuída em conjunto com o Globus, por este motivo foi necessário realizar a integração do monitor com o MDS em cada uma das estações.

A integração destas ferramentas consistiu na configuração de um componente interno do MDS, conhecido como provedor de propriedades de recursos (*Resource Property Provider*), indicando que este utiliza as informações disponibilizadas pelos monitores Ganglia.

A figura 5.3 apresenta esquematicamente como foi realizada a implementação do sistema de informações nos recursos de processamento do OncoGrid.

Figura 5.3: Representação esquemática do sistema de informação do OncoGrid.

Adicionalmente foi implantado um gerenciador Web das informações da grade, denominado WebMDS. Apesar desta ferramenta fornecer outras funcionalidades, a utilizamos somente para visualizar as informações do ambiente. A figura 5.4 apresenta a visualização Web dos recursos de processamento do OncoGrid.

ServiceGroup Overview

This page provides a brief overview of Web Services and/or WS-Resources that are members of a WS-ServiceGroup.

This WS-ServiceGroup has 6 direct entries, 15 in whole hierarchy.

Resource Type	ID	Information	
ServiceGroup	grid03	This WS-ServiceGroup has 3 direct entries, 3 including descendants.	detail
RFT	grid03	0 active transfer resources, transferring 0 files. 0 B transferred in 0 files since start of database.	detail
GRAM	grid03	1 queues, submitting to 1 cluster(s) of 1 host(s).	detail
Unknown	grid03	Aggregator entry with no content from https://grid03:8443/wsrp/services/ManagedJobFactoryService	detail
GRAM	grid01	1 queues, submitting to 1 cluster(s) of 1 host(s).	detail
Unknown	grid01	Aggregator entry with no content from https://grid01:8443/wsrp/services/ManagedJobFactoryService	detail
ServiceGroup	aedes-aegipy	This WS-ServiceGroup has 3 direct entries, 3 including descendants.	detail
GRAM	aedes-aegipy	1 queues, submitting to 1 cluster(s) of 1 host(s).	detail
RFT	aedes-aegipy	0 active transfer resources, transferring 0 files. 982 B transferred in 1 files since start of database.	detail
Unknown	aedes-aegipy	Aggregator entry with no content from https://10.0.165.213:8443/wsrp/services/ManagedJobFactoryService	detail
RFT	grid01	0 active transfer resources, transferring 0 files. 156.8 KB transferred in 13 files since start of database.	detail
ServiceGroup	grid02	This WS-ServiceGroup has 3 direct entries, 3 including descendants.	detail
Unknown	grid02	Aggregator entry with no content from https://grid02:8443/wsrp/services/ManagedJobFactoryService	detail
RFT	grid02	0 active transfer resources, transferring 0 files. 0 B transferred in 0 files since start of database.	detail
GRAM	grid02	1 queues, submitting to 1 cluster(s) of 1 host(s).	detail

Please report bugs and feature requests into the [Globus Bugzilla](#).

Figura 5.4: Página do WebMDS para a visualização das informações do ambiente de processamento OncoGrid.

5.2.3 Gerenciamento de Execução de Tarefas

Em cada estação foi integrado um gerenciador de execução GRAM. Este componente é fundamental para o ambiente de processamento, porque é o responsável por fazer a tradução das informações em formato WSRF, provinda dos serviços de grade, para o sistema operacional dos recursos físicos.

O GRAM é ativado automaticamente quando é instalado o Globus, mas é necessário implantar o transferidor de arquivos confiável (*Reliable File Transfer* - RFT) e a configurar as permissões de acesso dos usuários. A seção 5.2.4 aborda em detalhes da implementação do RFT.

O processo de aplicar as permissões de acesso dos usuários é fundamental para a gestão da OV. Assim podemos controlar quais grupos de usuários e usuários individuais poderão executar tarefas de processamento nas estações.

5.2.4 Movimentação de Dados

A movimentação de dados realizada pelo ambiente de execução utiliza basicamente a ferramenta GridFTP para transferência de dados para processamento e também dos executáveis. Esta ferramenta não necessita de configurações adicionais para o seu funcionamento, sendo distribuída no Globus.

O GRAM realiza o estagiamento de arquivos para manter a confiabilidade em nível de comunicação, quando realizando a gerência de tarefas computacionais. Por este motivo implantamos o serviço RFT nas estações. Este serviço utiliza um sistema gerenciador de banco de dados (SGBD) implantado nos recursos para armazenar o estado das transferências.

5.2.5 Sistema de Meta-Escalonamento

O sistema de meta-escalonamento implantado no ambiente foi o GridWay. Esta ferramenta utiliza as funcionalidades oferecidas pela maioria dos outros serviços de grade abordados nesta seção, assim possibilitando adicionalmente avaliar o funcionamento do ambiente de forma abrangente e integrada.

O GridWay foi implantado apenas na estação **grid03**, pelo motivo que a instalação do Globus nesta estação ter sido realizada a partir dos códigos fonte, assim possuindo todas as bibliotecas necessárias para a implantação da ferramenta em questão.

Para a integração do GridWay com o ambiente, foi necessário a configuração das conexões do mediador de acesso a dispositivos MAD. Este trabalho consistiu em indicar o modo de operação que o GridWay deveria utilizar para adquirir as informações do ambiente, transferir dados e submeter tarefas para execução. Devido às características dos componentes de grade já instalados (MDS, GRAM, GridFTP), o meta-escalonador foi configurado seguindo as informações apresentadas na tabela 5.3.

Mediador de	Serviço de Referência	Mediador (<i>Driver</i>)	Provedor de Informações
Informações	MDS4	gw_im_mad_mds4_thr	grid01
Execuções	Web Services	gw_em_mad_ws	
Transferências	GridFTP	gw_tm_mad_ftp	

Tabela 5.3: Configurações aplicadas no GridWay para o estabelecimento da integração do meta-escalonador com as ferramentas de informações, gerência de execução e transporte de dados do OncoGrid.

Após o estabelecimento das configurações para comunicação e interação com o ambiente OncoGrid, o GridWay disponibiliza ao usuário a configuração física dos recursos que poderão ser utilizados para a execução de tarefas. A tabela 5.4 apresenta a saída emitida pelo meta-escalonador quando realiza a consulta sobre os recursos de processamento disponíveis no ambiente.

PRI	OS	ARCH	MHZ	%CPU	MEM(F/T)	DISK(F/T)	N(U/F/T)	LRMS	HOSTNAME
1	Linux2.6.22-	x86	2992	192	714/1003	22760/30235	0/2/2	Fork	grid01
1	Linux2.6.22-	x86	2390	100	485/757	23730/30235	0/1/1	Fork	grid03
1	Linux2.6.24-	x86	2001	200	989/2017	25302/105707	0/2/2	Fork	aedes-aegypty
1	Linux2.6.22-	x86	2992	190	1691/2019	24950/30292	0/2/2	Fork	grid02

Tabela 5.4: Informações sobre os recursos de processamento identificadas pelo GridWay.

As colunas da tabela 5.4 expressam as seguintes informações:

- PRI: prioridade do recurso;
- OS: nome e versão do sistema operacional;
- ARCH: arquitetura do processador;
- MHZ: velocidade do processador em MHz;
- %CPU: porcentagem de CPU disponível para o uso pelo meta-escalonador;
- MEM(F/T): quantidade de memória, F = livre, T = total;
- DISK(F/T): capacidade de armazenamento em disco no recurso, F = livre, T = total;
- N(U/F/T): número de unidades de CPU no recurso, U = utilizados pelo GridWay, F = livre, T = total;
- LRMS: sistema gerenciador de recursos local (fork, PBS, SGE);
- HOSTNAME: nome do recurso registrado na rede.

A implantação da ferramenta contempla a configuração das políticas de escalonamento. Podemos parametrizar diferentes variáveis a fim de personalizar o ambiente conforme as necessidades da OV. O ambiente implementado utilizou algumas parametrizações para otimização do tempo de escalonamento apenas.

As políticas de escalonamento aplicadas ao GridWay são adaptativas, porque utilizam informações sobre execuções anteriores para determinar as decisões de escalonamento futuras. As políticas são divididas em políticas de escalonamento de tarefas e políticas de escalonamento de recursos. A combinação destas forma a política de escalonamento na grade. As tabelas 5.5 e 5.6, apresentam os parâmetros configurados no ambiente relacionados às políticas para escalonamento de tarefas e referentes ao escalonamento de recursos respectivamente.

Parâmetro	Descrição	Valor Utilizado
DISPATCH_CHUNK	Número máximo de tarefas que podem ser disparadas por ação de escalonamento	50
MAX_RUNNING_USER	Número máximo de tarefas em execução simultâneas por usuário	30
FP_WEIGHT	Peso para a política de prioridade fixada	1
FP_USER[DEFAULT]	Prioridade para execução de usuários padrão	1
SH_WEIGHT	Peso para política compartilhada	5
SH_USER[DEFAULT]	Cota de criação de tarefas pela política compartilhada pelos usuários padrão	50
SH_WINDOW_DEPTH	Intervalo de histórico de submissões realizadas pelos usuários	5
SH_WINDOW_SIZE = 1	Intervalo em dias que o sistema armazena o histórico de submissões	1

Tabela 5.5: Parâmetros de escalonamento para configuração das políticas relacionadas com as tarefas.

Parâmetro	Descrição	Valor Utilizado
MAX_RUNNING_RESOURCE	Numero máximo de tarefas em execução por recurso	10
RP_IM[DEFAULT]	Prioridade para todos os recursos descobertos pelo MAD	1
UG_HISTORY_WINDOW	Número de dias que o meta-escalonador armazena o histórico de execução dos recursos	3
FR_MAX_BANNED	Tempo máximo que um recurso é banido por motivo de falha na execução (segundos)	3600

Tabela 5.6: Parâmetros de escalonamento para configuração das políticas relacionadas com os recursos.

Para demonstrar o modo de operação utilizado pelo GridWay, executamos um teste

com um lote de dezesseis tarefas. A tabela 5.7 apresenta como o GridWay representa para o usuários os estados de execução das tarefas.

USER	JID	DM	EM	START	END	EXEC	XFER	EXIT	NAME	HOST
oper	0	done	—	16:17:54	16:19:49	0:01:10	0:00:45	0	job.tp.test_m	grid01/Fork
oper	1	done	—	16:17:54	16:19:58	0:01:12	0:00:52	0	job.tp.test_m	grid01/Fork
oper	2	done	—	16:17:54	16:19:52	0:01:14	0:00:44	0	job.tp.test_m	grid03/Fork
oper	3	done	—	16:17:54	16:19:27	0:00:46	0:00:47	0	job.tp.test_m	aedes-aegipty/Fork
oper	4	done	—	16:17:54	16:19:52	0:00:48	0:01:10	0	job.tp.test_m	aedes-aegipty/Fork
oper	5	done	—	16:17:54	16:19:52	0:01:11	0:00:47	0	job.tp.test_m	grid02/Fork
oper	6	done	—	16:17:54	16:19:52	0:01:12	0:00:46	0	job.tp.test_m	grid02/Fork
oper	7	epil	—	16:17:54	—:—	0:00:27	0:01:21	—	job.tp.test_m	aedes-aegipty/Fork
oper	8	epil	—	16:17:54	—:—	0:00:21	0:01:26	—	job.tp.test_m	aedes-aegipty/Fork
oper	9	epil	—	16:17:54	—:—	0:00:59	0:00:27	—	job.tp.test_m	grid01/Fork
oper	10	epil	—	16:17:54	—:—	0:01:01	0:00:24	—	job.tp.test_m	grid02/Fork
oper	11	epil	—	16:17:54	—:—	0:01:02	0:00:23	—	job.tp.test_m	grid02/Fork
oper	12	epil	—	16:17:54	—:—	0:01:02	0:00:22	—	job.tp.test_m	grid01/Fork
oper	13	epil	—	16:17:54	—:—	0:01:03	0:00:20	—	job.tp.test_m	grid03/Fork
oper	14	wrap	actv	16:17:54	—:—	0:00:04	0:00:49	—	job.tp.test_m	aedes-aegipty/Fork
oper	15	wrap	pend	16:17:54	—:—	0:00:02	0:00:45	—	job.tp.test_m	aedes-aegipty/Fork

Tabela 5.7: Informações sobre os processos computacionais (tarefas) em execução no meta-escalonador GridWay.

As colunas da tabela 5.7 expressam as seguintes informações:

- USER: usuário proprietário da tarefa;
- JID: identificação da tarefa;
- DM: estado da submissão da tarefa;
- EM: estado do gerenciamento de execução;
- START: instante em que a tarefa foi submetida para processamento;
- END: instante em que o meta-escalonamento recebeu a saída padrão da tarefa;
- EXEC: tempo total de execução;
- XFER: tempo total de transferência de arquivos, incluindo arquivos de entrada e de saída;
- EXIT: código de saída retornado pela tarefa (0 indica sucesso);
- NAME: nome da tarefa;
- HOST: recurso que executa a tarefa.

No instante em que foram extraídas as informações apresentadas na tabela 5.7, as colunas DM e EM indicavam cinco estados distintos. O estado **done**, coluna na DM, expresso nas tarefas com JID de 0 a 6, indica que as tarefas estão encerradas. O estado **epil** (abreviatura de *epilogue*), coluna DM, tarefas com JID de 7 a 13, indica que as tarefas estão em processo de conclusão. O estado **wrap**, coluna DM, tarefas com JID 14 e 15 indica o estado de preparação da tarefa para o início da execução. O estado **actv**, coluna EM, tarefa com JID 14, indica que a tarefa está ativa em processamento. O estado **pend**, coluna EM, tarefa com JID 15, indica que a tarefa está pendente, aguardando o término do processo de preparação para iniciar a ativação.

Na lista de itens a seguir apresentamos os dados de desempenho obtidos a partir do teste realizado com as dezesseis tarefas.

- Instante do início da execução: 16:17:54;
- Instante do fim da execução: 16:21:03;
- Tempo total de execução: 189 s;
- Tempo médio de preparação da tarefa: 45,5 s;
- Tempo médio de CPU: 52,75 s;
- Tempo médio de execução de uma tarefa local: 47,171 s.

5.2.6 Tolerância a Falhas no Processamento de Tarefas

O sistema de meta-escalonamento deve possuir tolerância a falhas, por ser estabelecido em um ambiente fracamente acoplado, conseguindo operar mesmo em situação de falha. O ambiente implementado consegue se livrar do estado de falha nas seguintes situações:

1. Quando a tarefa é finalizada na própria estação antes do seu termino, a atitude assumida pelo meta-escalador é retornar a saída padrão para o usuário e não re-submeter ou migrar a mesma. Este procedimento é realizado por diferentes motivos, por exemplo: a aplicação pode estar em um laço infinito, e se for reenviada para processamento, poderá apresentar o problema novamente; o processo computacional pode estar travado; o equipamento pode não ser o indicado para a execução necessitando um aprimoramento das descrições dos parâmetros de execução, entre outros.

2. Quando ocorre um defeito físico em um recurso ou falha na conexão de rede, a atitude assumida é migrar os processos para um outro recurso com características similares. O recurso falho não receberá mais tarefas para processamento dentro de um intervalo de tempo determinado. No OncoGrid, o intervalo de tempo é de uma hora;
3. Na etapa de preparação da tarefa o GridWay armazena todas as informações sobre esta em arquivos de log. Durante o processamento da tarefa estas informações são atualizadas continuamente. Quando ocorre falha física ou lógica na estação hospedeira do meta-escalador e o sistema é restabelecido, o meta-escalador reenvia os processos que não foram terminados até o instante da falha.

5.3 Desenvolvimento da Aplicação de Bioinformática para Validação do Ambiente

Para a codificação dos módulos de aplicação ISGL e ISGG foi utilizada a linguagem C, adicionalmente foi utilizado a biblioteca MATH, que possibilita a utilização de funções matemáticas mais avançadas. A implementação da linguagem escolhida foi o Gnu C Compiler 4.2, porque é disponibilizado gratuitamente e possui código livre, sendo parte integrante de todas as distribuições do sistema operacional Linux.

Devido ao fato da utilização da linguagem C, foi empregado o modelo de programação estruturada para as implementações dos módulos de aplicação envolvidos.-

Nas seções 5.3.2 e 5.3.1 vamos apresentar os diagrama de fluxo, estruturas de dados e as principais funções implementadas dos módulos ISGL e ISGG, respectivamente.

5.3.1 Implementação do Módulo Identificador de Semelhanças Genéticas Local

Em cada análise realizada pelo módulo ISGL são carregados dois arquivos de expressão genética na memória. Para armazenar as informações extraídas a partir da leitura seqüencial destes arquivos criamos uma estrutura de dados denominada **spot**. As variáveis que compõem a estrutura são:

- `endereco_lamina[2]`: vetor de duas posições de números inteiros que armazena o endereço do *spot* na lâmina do experimento. Este dado é armazenado por precaução, caso haja a necessidade de consultar a imagem para verificações;

- `n_ref_spot`: O número de referência do *spot* na lâmina do experimento. Este dado é utilizado efetivamente para localização dos *spot* em todos os cálculos realizados neste modulo;
- `ch1l`: expressão do canal `cy3` do *spot* de referência;
- `ch2l`: expressão do canal `cy5` do *spot* de referência.

As informações dos arquivos das expressões genéticas são carregadas em dois vetores. Cada posição do vetor é formada por uma estrutura do tipo **spot**.

As operações do ISGL são basicamente duas: a leitura seqüencial dos arquivos de expressões genéticas, adquirindo as informações dos canais expressos na lâmina do experimento, o cálculo das distâncias euclidianas entre todos os valores dos *spots*, que suporta expressões que tenham até 20.000 *spots*, uma vez que os arquivos de experimento utilizados para os testes possuem dezoito mil quatrocentos e trinta e dois *spots*.

A figura 5.5 apresenta o fluxograma utilizado para a codificação do módulo ISGL.

O resultado da execução deste módulo retorna a lista de distâncias, que é utilizada para os cálculos de classificação global e a definição do nome do experimento que é utilizado para localizar as expressão comparadas vencedoras dentro do grupo de expressões.

Figura 5.5: Fluxograma do módulo de Identificação de Semelhanças Genéticas Local - ISGL.

5.3.2 Implementação do Módulo Identificador de Semelhanças Genéticas Global

O identificador de semelhanças genéticas global (ISGG) suporta como entrada até três mil arquivos de resultados do módulo ISGL. Cada arquivo de entrada se torna uma coluna da matriz resposta (veja tabela 4.2 para maiores detalhes).

O ISGG realiza quatro operações fundamentais: a carga dos arquivos provindo do ISGL na matriz resposta $R(G,E)$, a geração da matriz classificação $C(G,E)$ a partir da matriz resposta (veja tabela 4.3), a criação da matriz pontuação a partir da matriz classificação $P(G,E)$ (veja tabela 4.4) e por fim o cálculo da classificação final a partir da somatória dos pontos das colunas da matriz pontuação.

A figura 5.6 apresenta o fluxograma utilizado para a codificação do módulo ISGL.

Figura 5.6: Fluxograma do módulo de Identificação de Semelhanças Genéticas Global - ISGG.

O algoritmo 1 descreve o processo para gerar a matriz classificação.

Para gerar a matriz pontuação aplicamos a equação 4.2 apresentada na seção 4.3.3 página 78 em todos os endereços da matriz classificação. Para gerar a classificação final, empregamos o mesmo algoritmo utilizado para gerar a matriz classificação em um vetor que possui a somatória das colunas da matriz pontuação.

Algoritmo 1 Processo para gerar a matriz classificação

```

Score  $\leftarrow$  Node expressões analisadas
para todas linhas G faça
  para i de 1 até número de colunas faça
    para j de 1 até o número de colunas faça
      se  $R(L, i) < R(L, j)$  então
        Score  $\leftarrow$  Score - 1
      fim se
    fim para
     $C(i, j) \leftarrow$  Score
  fim para
fim para

```

5.4 Resumo do Capítulo

A arquitetura implementada para a distribuição de processamento no ambiente de grade computacional utilizou as mesmas especificações sobre o modelo de camadas funcionais empregadas no ambiente OncoGrid.

No projeto piloto OncoGrid implementamos a componentização fundamental para o funcionamento do ambiente (a autoridade certificadora OncoGrid-CA e o servidor MyProxy na camada de segurança, a interface por linha de comando e a interface Web na camada de usuários, o GridFTP na camada de conexão de dados) e disponibilizamos diretórios em sistemas de arquivos na camada de recursos para a realização de teste de conexão e de transporte de dados.

A implementação adicional realizada neste trabalho agregou à arquitetura do OncoGrid os componentes necessários para a gestão distribuída geograficamente dos recursos para processamento (o gerente de informações MDS, o gerente de execução GRAM e o meta-escalonador GridWay na camada de serviços de grade) e adicionalmente disponibilizamos equipamentos como de recursos físicos de processamento na camada de recursos.

Para melhorar as capacidades na gestão dos processos distribuídos na grade, implantamos o Ganglia para monitoramento do estado dos recursos e o integramos com o ambiente de informações da grade, possibilitando que o ambiente tenha o conhecimento dos estados dos recursos no instante desejado. Esta funcionalidade permite maior dinamismo no escalonamento das tarefas, evitando sobrecarga nos recursos.

Para avaliar o uso do ambiente de processamento do OncoGrid aplicado na execução de tarefas relacionadas à bioinformática, foram desenvolvidos os módulos de aplicação ISGL e ISGG, ambos codificados em linguagem C utilizando o modelo estrutural. O módulo

ISGL realiza os cálculos das distâncias genéticas entre os *spots* de duas expressões, o processamento deste módulo é distribuído no ambiente de grade. O módulo ISGG agrupa os resultados provindos das execuções das tarefas ISGL e os consolida em uma classificação ordenada indicando as expressões genéticas mais semelhantes.

6 *Análises de Resultados*

Neste capítulo serão apresetadas as análises e testes relacionados com esta pesquisa, avaliando a sua validade experimental. O objetivo destas validações é a comprovação experimental das decisões técnicas empregadas para a elaboração da proposta realizada neste trabalho.

Para realizar as validações estruturamos duas baterias de testes. A primeira realiza a comparação de uma expressões genética contra um grupo de outras 52 expressões sendo denominado de teste “um contra todos”. Neste teste variamos a quantidade de processadores disponibilizados para execução das tarefas. A segunda realizou a comparação entre todas as expressões contra todas as expressões, sendo denominado de teste “todos contra todos”. A intenção deste teste foi avaliar o comportamento do ambiente em situação de sobrecarga e obter o par de expressões genéticas do grupo todo, possibilitando avaliar as características de escalonamento e dos recursos de processamento utilizado. Para este teste o ambiente esteve configurado com sua capacidade total a todo tempo.

A seção 6.1 apresenta os resultados do teste um contra todos. A seção 6.2 apresenta os resultados do teste todos contra todos. Por fim a seção 6.3 apresenta as avaliações e relatos sobre os resultados das implementações e testes realizados.

6.1 Resultados do Teste Um Contra Todos

O teste um contra todos consiste em escolher uma expressão genética de nossa base de expressões e executar o processo ISGL contra todas as outras restantes. Utilizamos a comparação da expressão analisada com ela mesma para validar o funcionamento de ambos os módulos.

O processamento deste teste foi submetido quatro vezes, variando a quantidade de processadores em cada execução. Os itens a seguir indicam o número do teste de validação quantos processadores atuaram na sua execução:

1. 1 CPU, estação grid03;
2. 3 CPUs, estações grid01 e grid03;
3. 5 CPUs, estações grid01, grid02 e grid03;
4. 7 CPUs, capacidade total do ambiente.

O processamento da tarefa enviado para execução no ambiente, por meio do escalonador global, consiste de cinco etapas. Em primeiro lugar a tarefa é colocada em uma fila de processos do meta-escalonador. Na segunda fase o meta-escalonador identifica um recurso de processamento disponível para entregar a tarefa para execução. O terceiro passo é a preparação da tarefa no recurso associado a ela, que consiste em enviar os dados envolvidos na execução da tarefa, incluindo os códigos binários e arquivos de entrada. A quarta etapa consiste no trabalho de iniciar e gerenciar o processo de execução. Por fim, é realizada a transmissão do resultado da tarefa provindo do recurso para o meta-escalonador.

Neste teste experimental os dados enviados para o recurso consistem em arquivos de controle do processo, totalizando cerca de 30 KB, e dois arquivos de expressões genéticas, cada um ocupando aproximadamente 3,5 MB. Os dados retornados após o processamento são arquivos com tamanho de 208KB cada. O total de em bytes transportados para a execução de cada tarefa é de aproximadamente 7,2 MB. O montante de dados transitados neste experimento fica próximo a 374,82 MB. Durante a execução das tarefas o meta-escalonador também se comunica com os serviços de grade e com os recursos de processamento, no entanto estas comunicações não foram mensuradas nas avaliações.

A figura 6.1 apresenta o gráfico que indica os tempos de execução das cinquenta e duas tarefas, considerando o tempo de processamento e o tempo de transferência de dados, assim abordando o processo como um todo.

Abordando os dados apresentados no gráfico da figura 6.1, podemos observar que a execução do teste em um processador durou 972 segundos. Disponibilizando mais duas CPUs ao ambiente a operação durou 504 segundos. Acrescentando mais duas CPUs, o lote de tarefas foi processado em 246 segundo. Finalmente a execução do teste utilizando a capacidade total do ambiente (7 CPUs) durou 207 segundos.

A figura 6.2 apresenta o gráfico contendo as informações relacionadas ao tempo médio de execução e ao tempo médio de utilização de rede obtidos a partir do processamento dos lotes de tarefas testados.

Figura 6.1: Gráfico representando o tempo total de processamento das cinquenta e duas tarefas ISGL nos testes realizados com 1, 3, 5, e 7 processadores.

No gráfico apresenta na figura 6.2 podemos observar que quanto maior é o número de CPU envolvidas na execução das tarefas maior é o tempo de comunicação de rede por tarefa. Este fato é atribuído ao motivo do meta-escalonador estar enviando os dados para processamento e recebendo os resultados de todas as tarefas processadas.

Devemos considerar que a estação que apresenta o maior tempo de execução médio (aproximadamente 16 s), possui a menor capacidade de processamento e de memória RAM (757 MB), quando observando todos os recursos envolvidos no teste. Esta estação acumula a responsabilidade de escalonar todas as tarefas e realizar o processamento de algumas delas, chegando a esgotar sua capacidade de memória e de processamento.

A figura 6.3 apresenta o acréscimo de desempenho (*speedup*), medido a partir das execuções dos lotes de tarefas deste teste.

Figura 6.2: Gráfico representando o tempo médio por tarefa de utilização de CPU e de comunicações de rede nos testes com 1, 3, 5 e 7 CPUs.

Os resultados retornados pela execução do módulo ISGL são 52 vetores de distâncias, comparando o experimento de *microarray* cujo seu identificador é lc8n006rex2 com todos os outros, incluindo ele mesmo. Os vetores são utilizados pelo módulo de aplicação ISGG que calcula a classificação dos experimentos baseado nas distâncias. Quando comparamos um experimento com ele mesmo o valor de suas distâncias serão sempre nulos, assim, esta comparação sempre estará em primeiro lugar na classificação geral. A comparação da expressão estudada com ela mesma foi utilizada para validação do correto funcionamento do módulo ISGL.

A tabela 6.1 apresenta a saída gerada pelo módulo ISGL. Podemos observar que a expressão mais semelhante à comparada é a lc8n012rex, sem considerar a expressão estudada (lc8n006rex2).

Figura 6.3: Gráfico representando o acréscimo de desempenho obtido na execução da bateria de testes um contra todos.

6.2 Resultados do Teste Todos Contra Todos

Para validar o funcionamento do ambiente em situação de carga elevada, elaboramos um teste que realiza o cruzamento entre todos os cinquenta e dois experimentos *microarray* disponíveis em nossa base de testes. Este processo é denominado como cruzamento de todos contra todos.

O resultado desta execução proporciona as evidências dos casos clínicos mais semelhantes geneticamente dentro do grupo avaliado, obviamente que considerando como variável apenas os genes expressos nos experimentos analisados.

A tabela 6.2 apresenta como são estipuladas as operações ISGL para a realização do cruzamento todos contra todos, por meio de um exemplo que cruza cinco expressões. Os cruzamentos de nosso interesse são os marcados na tabela com **X** os que não são utilizados para o experimentos estão marcados com **0**.

A quantidade de tarefas necessárias para contemplar todas as análises é dada pela equação 6.1, onde $Z(n)$ é a quantidade necessária de tarefas para cobrir o experimento e n é o número de expressões genéticas disponíveis para avaliação.

O conjunto de dados utilizado para validação possui 52 expressões genética, assim totalizando 1326 tarefas ISGL.

$$Z(n) = \frac{n^2 - n}{2} \quad (6.1)$$

Experimento	Ponto	Classificação		Experimento	Ponto	Classificação
lc8n006rex	18432.0000	1		lc8n100rex	10.752057	27
lc8n012rex	132.363859	2		lc8n107rex	10.747903	28
lc8n077rex	127.795769	3		lc8n092rex	9.626945	29
lc8n128rex	125.737558	4		lc8n109rex	9.416366	30
lc8n051rex	101.949059	5		lc8n091rex	8.743385	31
lc8n126rex	87.415304	6		lc8n110rex	7.439937	32
lc8n037rex	81.059916	7		lc8n078rex	6.943068	33
lc8n129rex	68.454167	8		lc8n114rex	6.873630	34
lc8n011rex	63.087439	9		lc8n136rex	6.471833	35
lc8n093rex	62.631197	10		lc8n135rex	5.687924	36
lc8n132rex	54.064576	11		lc8n133rex	5.593657	37
lc8n096rex	52.493780	12		lc8n119rex	5.552891	38
lc8n125rex	48.894552	13		lc8n121rex	5.088502	39
lc8n127rex	43.320862	14		lc8n118rex	5.087886	40
lc8n097rex	39.034758	15		lc8n122rex	4.903959	41
lc8n083rex	38.938859	16		lc8n120rex	4.508050	42
lc8n057rex	38.615228	17		lc8n134rex	4.162519	43
lc8n108rex	33.044384	18		lc8n112rex	3.296384	44
lc8n106rex	32.687362	19		lc8n105rex	2.906851	45
lc8n076rex	30.408520	20		lc8n116rex	2.412833	46
lc8n098rex	28.533966	21		lc8n115rex	2.116108	47
lc8n095rex	26.275906	22		lc8n111rex	1.533157	48
lc8n130rex	21.378321	23		lc8n103rex	1.507385	49
lc8n085rex	15.192539	24		lc8n113rex	1.118962	50
lc8n101rex	13.911371	25		lc8n117rex	1.079353	51
lc8n131rex	13.298657	26		lc8n104rex	0.623778	52

Tabela 6.1: Classificação geral do teste comparando a expressão lc8n006rex contra todas as outras.

A estratégia de manipulação de dados utilizada para estes testes foi o acesso de arquivos locais, ou seja, a tarefa acessa os dados para processamento no próprio sistema de arquivo do recurso que a executa. Este fato se tornou possível porque replicamos o conjunto de informações entre todas as estações.

Para cada tarefa foi criado um arquivo descritor de tarefas que é interpretado pelo GridWay para realizar a submissão. Os arquivos foram gerados por meio de *script* (programa Shell). A seguir apresentamos o arquivo que descreve a submissão da tarefa que calcula as distâncias entre as expressões lc8n006rex2 e lc8n011rex2.

```

EXECUTABLE = isgl
ARGUMENTS  = "/home/oper/genexp/DAT/lc8n006rex2.DAT \
              /home/oper/genexp/DAT/lc8n011rex2.DAT"
STDERR_FILE = err.lc8n006rex2.DAT.lc8n011rex2.DAT.alltoall
STDOUT_FILE = out.lc8n006rex2.DAT.lc8n011rex2.DAT.alltoall

```

Os parâmetros utilizados são EXECUTABLE, que indica qual executável deve ser enviado e executado na estações remota; ARGUMENTS, que indica os argumentos de entrada do programa executável; STDERR_FILE, que indica qual arquivo receberá a

	Ex 1	Ex 2	Ex 3	Ex 4	Ex 5
Ex 1	0	X	X	X	X
Ex 2	0	0	X	X	X
Ex 3	0	0	0	X	X
Ex 4	0	0	0	0	X
Ex 5	0	0	0	0	0

Tabela 6.2: Processo de criação das tarefas para execução do teste de todos contra todos.

saída de erro padrão; e `STDOUT_FILE`, que indica o nome do arquivo que receberá a saída padrão do processamento.

Para não sobrecarregar o meta-escalonador no momento da submissão das tarefas, assumimos que os envios dos processos solicitados pelo usuário seriam realizados no intervalo de 3 segundos.

Os parâmetros de escalonamento que restringiam a quantidade de tarefas que um usuário poderia submeter no ambiente foram alterados para possibilitar que a execução do teste fosse realizada de forma contínua.

A tabela 6.3 apresenta os dados sobre a execução do teste em questão, abordando os recursos individualmente e o ambiente integrado.

Para compreender os dados apresentados na tabela 6.3, devemos considerar que os recursos de processamento participantes são heterogêneos e que muitas das operações estão sendo realizadas simultaneamente, como por exemplo, uma estação que está executando o processamento da tarefa pode preparar a próxima para execução. As estações que possuem dois núcleos de processamento ou estão equipadas com tecnologia HT conseguem executar duas tarefas simultaneamente e compartilham a interface de rede para transmissão dos resultados. Quando os recursos se tornam sobrecarregados, como aconteceu com o `grid03` e o `aedes-aegypty`, eles são temporariamente ignorados pelo escalonador, até que o seu estado transite de ocupado para disponível. Por fim, devemos considerar que as políticas de escalonamento do ambiente são adaptativas, e por este fato não é linear a quantidade de execuções que cada recurso do ambiente deve processar.

Por estes motivos não é correto afirmar que os tempos de execução e de transferência de dados do ambiente OncoGrid são correspondentes à somatória dos tempos atribuídos

aos recursos.

	grid01	grid02	grid03	aedes-aegipy	OncoGrid
Tarefas executadas	607	564	23	132	1326
Tempo total em transferência de dados	3225 s	6346 s	761 s	3479 s	X
Tempo médio de transferências por tarefa	5,31 s	11,25 s	33,09 s	26,36 s	X
Tempo total de uso de CPU	11114 s	11311 s	111 s	3889 s	X
Tempo médio de CPU por tarefa	9,15 s	10,03 s	3,47 s	14,73 s	X
Tempo de serviço no processamento das tarefas	7909 s	7876 s	1035 s	4500 s	7909s
Tempo médio de processamento total por tarefa	13,03s	13,96s	45s	34,09 s	5,96s
Status do equipamento no instante da execução	1	1	1 e 2	1 e 3	

Tabela 6.3: Dados referente a execução do teste todos contra todos (1 recurso de processamento, 2 escalonador de tarefa, 3 estação de usuário).

Recebemos como resposta da execução do processamento distribuído do módulo ISGL 1326 arquivos contendo os vetores de distâncias do teste todos contra todos. Para conseguir a classificação geral, executamos o aplicativo ISGG utilizando como entrada todos os arquivos resultante. A tabela 6.4 apresenta os dez pares de expressões genéticas que apresentaram maior grau de semelhança. Indicamos alguns resultados classificatórios intermediários para oferecer a visão do resultado deste teste de validação.

Para validar os dados de saída do teste, realizamos observação em sete dos vetores de distância resultantes que correspondem aos cruzamentos classificados em primeiro a quinto, ducentésimo quinquagésimo e milésimo colocados. A figura 6.4 apresenta o gráfico representando as médias das distâncias euclidianas expressas pela análise ISGL. Podemos observar nos dados apresentados que o primeiro colocado, que analisa as expressões lc8n092rex2 e lc8n085rex2, possui a média entre as suas distância próxima à casa dos 200 pontos. Nas demais acontece um aumento gradativo nos valores de suas médias.

Realizamos as avaliações das distâncias de dois experimentos, o primeiro e o quinto colocados. Calculamos quantos *spots* apresentaram distancias de 0 a 10, de 0 a 20, de

Experimentos	Pontos	Classificação
lc8n092rex2 X lc8n085rex2	297.650459	1
lc8n126rex2 X lc8n098rex2	268.268735	2
lc8n093rex2 X lc8n085rex2	266.529313	3
lc8n037rex2 X lc8n012rex2	224.949537	4
lc8n098rex2 X lc8n097rex2	201.232399	5
lc8n083rex2 X lc8n057rex2	200.743664	6
lc8n098rex2 X lc8n051rex2	198.493570	7
lc8n126rex2 X lc8n051rex2	194.339142	8
lc8n127rex2 X lc8n051rex2	186.584111	9
lc8n129rex2 X lc8n126rex2	183.293341	10
...	...	...
lc8n108rex2 X lc8n012rex2	21.542085	250
...	...	...
lc8n096rex2 X lc8n037rex2	8.110683	500
...	...	...
lc8n112rex2 X lc8n011rex2	1.057445	1000
...	...	...
lc8n104rex2 X lc8n096rex2	0.000000	1326

Tabela 6.4: Resultados dos teste de avaliação todos contra todos, apresentando os dez primeiros classificados.

0 a 30, de 0 a 40 e de 0 a 50 unidades de medida euclidiana. Esta observação permitiu validar o fato do experimento melhor classificado possuir menores distâncias, indicando que a equação aplicada para a pontuação atuou de forma satisfatória, distanciando as expressões genéticas menos semelhantes e aproximando as mais semelhantes.

Apresentamos na figura 6.5 o gráfico representando os dados obtidos a partir desta contagem. É nítido que as menores distâncias são atribuídas ao experimento melhor colocado.

6.3 Análises e Discussões dos Resultados

O ambiente OncoGrid possibilita a integração multi-institucional utilizando as interconexões físicas das redes de computadores mundiais, sendo elas de pesquisa ou comerciais.

O emprego das ferramentas para estabelecimento do ambiente segue a padronização estipulada pela OGF, viabilizando a interoperabilidade com outros ambientes de grade computacional que utilizem o mesmo padrão.

A infra-estrutura de segurança empregada nas ferramentas aplicadas no desenvolvimento do ambiente são baseadas na padronização já utilizada na Internet, como autoridade certificadora, certificados de usuários, padrões de segurança aplicados a serviços Web e protocolos para transferência de dados seguros como HTTPS (*HyperText Transfer Protocol Secure*). No entanto os componentes de segurança da grade estendem as funci-

Figura 6.4: Gráfico representando as médias das distâncias euclidianas de alguns dos cruzamentos testados.

onalidades das ferramentas aplicadas à Internet para proporcionar o compartilhamento coordenado de recursos e serviços computacionais. Alguns exemplos destas extensões são a utilização dos certificados Proxy e a possibilidade de realizar a delegação de credenciais.

O ambiente OncoGrid foi concebido sob um projeto modular baseado no modelo de camadas funcionais. A implementação utilizando este modelo de especificação apresenta a vantagem de possibilitar a realização do projeto de forma gradativa e modularizada, integrando apenas os componentes de software necessários para a função que o ambiente irá realizar.

A escalabilidade da arquitetura implementada é um fator relevante, pois em um ambiente com essas características torna-se inevitável incluir ou excluir usuários, serviços e recursos de forma dinâmica. Constatamos diversas características que possibilitam atender este requisito: A camada de segurança possibilita a utilização de uma ou mais autoridades certificadoras e o gerenciamento de forma distribuída das credenciais Proxy, por meio do servidor MyProxy, permitindo que o usuário possa recuperar o seu certificado a partir de qualquer equipamento conectado no ambiente. Também é possível replicar o quanto necessário os componentes da camada de serviços de grade. Para incluir uma instituição em uma OV, tornando esta uma participante da grade, o administrador de sistema necessita

Figura 6.5: Gráfico representando as médias das distância euclidianas obtidas dos experimentos classificados em primeiro e quinto lugar.

realizar a integração do seu sistema de informação com a da grade que se deseja participar, e possuir certificados compatíveis com a os da grade para os seus recursos físicos e usuário. A inclusão de um recurso físico de uma instituição na grade implica na disponibilização deste equipamento para os demais usuários e instituições participantes. As informações sobre este recurso são automaticamente disponibilizadas no ambiente.

O ambiente pode proporcionar diferentes interfaces para o usuário, como, portais de grade, aplicações cliente ou a interação por linha de comando, neste trabalho nós utilizamos a interação por linha de comando, pelo motivo desta interface proporcionar maior facilidade de interação com as ferramentas de gerenciamento do ambiente como é o caso do meta-escalador.

O ambiente de processamento do OncoGrid foi concebido utilizando o meta-escalador GridWay e quatro recursos de processamento, totalizando sete unidades de CPUs. Com este ambiente, conseguimos realizar a validação de dois métodos de distribuição de tarefas: processamento de tarefas únicas, que foi avaliado durante a implementação do ambiente, e a execução de lotes de tarefas, que foi validada durante a execução dos testes um contra todos e todos contra todos. Em ambos os casos o ambiente apresentou comportamento estável, agregando de forma significativa as capacidades computacionais dos recursos.

Quando observamos o gráfico apresentado na figura 6.3, podemos constatar o ganho de desempenho de aproximadamente próximo a 5 vezes.

Observamos que o sistema de informações de uma grade computacional é fundamental para as atividades de meta-escalonamento. Quanto mais abrangente for este sistema, melhor será a gerência do ambiente de processamento, porque o meta-escalonador utiliza as informações de estado dos recursos para distribuir as tarefas no ambiente. Assim informações mais precisas implicam melhores resultados destas atividades.

O OncoGrid deve atuar sobre um ambiente heterogêneo de *hardware* e *software*. Este requisito é atendido de duas formas. Em primeiro lugar as políticas utilizadas para distribuir as tarefas são adaptativas e se comportam adequadamente sobre a plataforma heterogênea de *hardware*, conseguindo distribuir as tarefas evitando sobrecarga nos recursos, como pode ser verificado no tabela 6.3, as estações que estavam sobrecarregadas não receberam mais tarefas para processamento. Em relação à heterogeneidade de *software*, podemos indicar no arquivo descritor de tarefas qual é o sistema operacional ou versão do sistema operacional que deverá executar a tarefa, assim como as bibliotecas necessárias para execução, como a MPI, por exemplo.

Considerando os recursos de processamento, o ambiente possibilita a inclusão de recursos centralizados, como estações convencionais, multiprocessadas e multicomputadores, possibilitando o escalonamento para diferentes implementações de sistemas de gerenciamento de recursos locais, como fork, PBS, Condor e Sun Grid Engine.

O sistema de tolerância à falhas em nível de processamento migra os processos em casos de falhas na comunicação ou nos componentes físicos dos recursos. O equipamento que apresentou falha fica inativo no ambiente por um tempo determinado, configurável pelo administrados do ambiente. Caso ocorra uma falha no equipamento que executa a função de meta-escalonador, após o restabelecimento do funcionamento deste equipamento as tarefas que ficaram pendentes ou que estavam na fila de execução são reenviadas para processamento e as informações sobre as tarefas pendentes são recuperadas por meio de arquivos de log.

O ambiente possui limitações em relação à localização dos dados. Não encontramos nenhum método ou ferramenta que possibilitasse a localização dos setores de dados que armazenavam os experimentos. No caso do teste um contra todos, os dados para processamento foram enviados para as estações pela interface oferecida pelos arquivos descritores de tarefas, no caso do teste todos contra todos, os diretórios que armazenavam os experimentos foram replicados em todos os recursos de processamento, assim acessando

localmente os dados a serem processados.

6.4 Resumo do Capítulo

Para validar o emprego do ambiente de grade computacional aplicado a saúde, elaboramos dois testes utilizando a aplicação de identificação de semelhanças genéticas. O primeiro realizou a comparação de uma expressão contra um conjunto de cinquenta e duas expressões, sendo denominado teste um contra todos. O segundo teste consistiu no cruzamento de todas as cinquenta e duas expressões contra todas, totalizando 1326 tarefas.

Na realização dos testes um contra todos, variamos o número de CPUs disponibilizadas no ambiente (1, 3, 5 e 7 CPUs), assim verificando o acréscimo de desempenho. Comparando a utilização de sete CPUs em relação à de apenas uma, o ambiente conseguiu uma redução de aproximadamente 80 % do tempo de processamento total do lote de tarefa no ambiente.

Na realização do teste todos contra todos verificamos o comportamento do ambiente em estado de sobrecarga. Este teste nos permitiu verificar as opções assumidas pelo meta-escalonador em situação adversas, como por exemplo quando utilizamos um dos recursos para executar outra tarefa. O ambiente de processamento realizou as adequações necessárias para processar as tarefas evitando a sobrecarga nos recursos.

Os resultados dos cálculos das distâncias genéticas obtidos a partir do processamento distribuído das tarefas ISGL foram submetidos para classificação global no aplicativo ISGG. Partindo destes resultados classificatórios, avaliamos algumas das expressões genéticas para comprovação experimental do funcionamento da aplicação. A análise destes resultados forneceu as indicações de que a aplicação encontra, de fato, os experimentos que possuem maiores semelhanças entre os seus genes.

A avaliação dos resultados mostrou que o OncoGrid atendeu os requisitos elencados na fase de projeto e indicou as deficiências do ambiente relacionadas na localização dos dados em sistemas de arquivo.

7 Conclusões, Trabalhos Futuros e Considerações Finais

Os resultados obtidos e as experiências adquiridas durante os estudos e implementações realizadas neste trabalho nos levam a expressar reflexões conclusivas e a sugestão de trabalhos futuros.

7.1 Conclusões

Verificamos que as atividades de pesquisa, prevenção, diagnóstico e tratamento do setor de saúde aplicados na atenção à oncologia estabelece um domínio de colaboração multi-institucional, envolvendo unidades de tratamento e centros de pesquisa, e de conhecimentos interdisciplinares. Estas atividades são responsáveis por gerar grandes quantidades de informações distribuídas em seus centros de pesquisa e atenção. Muitos estudos em oncologia relacionam tais dados, como é o caso dos trabalhos realizados no campo da bioinformática. Por este motivo algumas atividades deste setor necessitam da integração e do processamento destas informações.

Mediante estes fatos acreditamos que aplicar a tecnologia de grades computacionais no cenário da pesquisa e atenção a saúde em oncologia é uma alternativa desafiadora do ponto de vista da pesquisa e desenvolvimento, porque esta tecnologia aborda a complexidade de gerenciar, integrar e processar dados em larga escala utilizando compartilhamento de recursos computacionais de forma colaborativa sobre ambientes multi-institucionais geograficamente distribuídos.

No âmbito da pesquisa envolvida nesta dissertação foi desenvolvido um projeto piloto denominado OncoGrid-Grade Computacional em Oncologia para avaliar as alternativas e possibilidade de aplicação deste tipo de arquitetura computacional no setor de saúde em oncologia. Este projeto resultou na especificação de uma arquitetura base utilizando o modelo de camadas funcionais. A implementação realizada neste projeto abordou a

integração de dados e o estabelecimento do sistema de segurança, porém apenas estes pontos não são suficientes para atender o setor da saúde, considerando as atividades de pesquisa, como é o caso da bioinformática. Ainda se fazia necessário gerar alternativas para proporcionar a realização de processamento de alto desempenho no ambiente. Desta forma, este trabalho focou a pesquisa de alternativas e soluções para atender esta demanda.

Este trabalho levantou os métodos para aquisição da expressão genética por meio da tecnologia de *microarrays* de DNA complementar por hibridização. Levantou o estado da arte sob a tecnologia de grade computacional sobre uma visão abrangente e aprofundou o estudo no campo de processamento computacional distribuído, a fim de obter soluções tecnológicas para atender a demanda por aplicações no campo da bioinformática em oncologia que sejam classificadas como de larga distribuição de dados, alta complexidade computacional ou que necessitem de recursos computacionais específicos. Propôs e modelou uma plataforma para processamento distribuído aderente ao ambiente OncoGrid sob a abordagem de camadas funcionais, agregando a este a capacidade de distribuir processamento. Desenvolveu uma aplicação no campo da bioinformática com características de sistema de telessaúde que realiza a identificação de similaridades genéticas entre um grupo de experimento de *microarray* de DNA complementar. Realizou as validações do ambiente utilizando a aplicação desenvolvida, mediante a execução de testes de validações sobre uma implementação da arquitetura proposta em laboratório.

7.2 Trabalhos Futuros

Devido ao fato do ambiente não conseguir indicar a localização física dos arquivos de expressões genéticas que estão armazenadas na grade, propomos como trabalhos futuros pesquisas direcionadas ao estabelecimento de um sistema para catalogação de informações sobre os dados em sistema de arquivos distribuído no ambiente, como um serviço de metadados global, por exemplo, que possa ser integrado com o sistema de meta-escalonamento e com os sistemas de informações da grade, assim possibilitando que as tarefas sejam distribuídas de forma que sejam processadas o mais próximo possível dos dados ou otimizando a transferência de informações nas etapas de preparação da execução da tarefa, assim reduzindo o tempo total de execução das tarefas e evitando sobrecarga nos canais de comunicação.

Tratando especificamente dos dados de *microarrays*, consideramos necessário o emprego de uma padronização comum das informações dos experimentos, possibilitando a

consulta dos dados indexados diretamente pelo nome ou identificação do gene, e não pelo endereço do *spot* da lâmina como foi realizado neste trabalho. A realização deste trabalho futuro visa proporcionar a utilização dos dados obtidos a partir de experimentos realizados com diferentes tecnologias ou fabricantes de *microarrays*. Atualmente são empregados esforços para a padronização das informações destes experimentos. Um dos resultados destes trabalhos é o padrão MIAME, que indica as informações mínimas necessárias para a publicação e experimentação utilizando *microarrays*. Sugerimos o emprego de padronizações já difundidas como esta para a realização deste trabalho futuro.

Considerando a aplicação proposta, desenvolvida e avaliada, podemos observar que os cálculos realizados para a identificação de semelhanças genéticas são triviais. Portanto, a metodologia de distribuição em ambiente de grade computacional é válida, desta forma também devemos considerar o aprimoramento da aplicação possibilitando que o profissional que a utilize, e indicando quais as análises deverão ser realizadas. Também é importante considerar a disponibilização desta aplicação como um serviço de telessaúde via Web melhorando a interação com o usuário, possibilitando, assim, a sua utilização por usuários menos experientes.

Como trabalhos futuros, pretendemos estender o uso da aplicação desenvolvida de similaridade genética para possibilitar:

- Pesquisar e determinar agrupamentos de perfis genéticos similares de pacientes com mesmo diagnóstico;
- Permitir a comparação de um novo paciente com os perfil genéticos identificados, possibilitando o apoio ao diagnóstico médico e prognóstico de tratamento;
- Acompanhar a evolução clínica deste perfil genético junto à análise de imagens, para viabilizar a pesquisa de novos protocolos de tratamentos.

Está sendo realizada uma implantação de grade computacional baseada na proposta apresentada neste trabalho em colaboração com o centro de Pesquisas Rene Rachou da Fundação Osvaldo Cruz de Minas Gerais. Esta implementação pretende integrar três aglomerados de computadores totalizando oitenta e dois núcleos de processamento dispostos no ambiente. A infra-estrutura estabelecida será utilizada para o processamento de tarefas computacionais relacionadas com o alinhamento de seqüências genéticas e a identificação de proteínas.

7.3 Contribuições Oferecidas

Os estudos realizados neste trabalho formam um arcabouço de informações consolidadas em um material que indica os direcionamentos técnicos de como implementar ambientes de grade computacionais com capacidade para a distribuição de processamento.

A modelagem do ambiente em camadas funcionais possibilita a organização sistemática dos componentes da grade computacional, desta forma possibilitando o projeto e implementação de ambientes de forma modularizada e customizada, atendendo diferentes necessidades.

A disponibilização da plataforma computacional implementada neste trabalho, no âmbito da Rede ONCONET, possibilitará a realização de novas pesquisas e ferramentas voltadas ao apoio clínico a distância e atenção ao setor de saúde. Considera-se, também, a possibilidade de aperfeiçoamentos de várias das aplicações existentes no projeto ONCONET para explorar a utilização do ambiente de grade computacional projetado e testado durante o desenvolvimento desta dissertação.

Os testes realizados com a distribuição da aplicação nos traz indícios sobre como e quando utilizar os ambientes de grade computacional para processamento. Desta forma, a observação detalhada dos resultados experimentais proporciona ao leitor a possibilidade de transpor a realidade das suas necessidades.

O desenvolvimento deste trabalho proporcionou a publicação de dois artigos em congressos, sendo um nacional e um internacional.

APÊNDICE A -- Artigos Publicados

CAMPOS, M. et al. *Computação em grade na saúde: Proposta de uma arquitetura para integração de informações médicas*. Congresso Brasileiro de Informática em Saúde, 2006. Disponível em: <<http://www.sbis.org.br/cbis/anaiscbis2006.htm>>.

Artigo aceito para publicação em Abril de 2008.

ALVES, H.; CAMPOS, M. et al. *Oncogrid: A Proposal of Grid Infrastructure for the Establishment of a National Health Information System on Childhood Cancer*. IPDPS - IEEE International Parallel & Distributed Processing Symposium, 2008.

APÊNDICE B -- Parâmetros de Configuração de Ambiente de Execução no GridWay

Os parâmetros de configuração do ambiente de execução são utilizados para indicar as necessidades e requisitos das tarefas relacionadas com o ambiente de execução.

A tabela B.1 apresenta os parâmetros disponibilizados pelo GridWay que podem ser utilizados no ambiente OncoGrid.

Parâmetro	Descrição
HOSTNAME	Nome da estação (DNS)
ARCH	Arquitetura do processador
OS_NAME	Sistema operacional
OS_VERSION	Versão do sistema operacional
CPU_MODEL	Modelo da CPU da estação
CPU_MHZ	Velocidade da CPU em MHz
CPU_FREE	Percentual de CPU livre
CPU_SMP	Quantidade de SMP da CPU
NODECOUNT	Número de nós do recurso
SIZE_MEM_MB	Memória total
FREE_MEM_MB	Memória Disponível
SIZE_DISK_MB	Espaço total em disco (MB)
FREE_DISK_MB	Espaço livre em disco (MB)
LRMS_NAME	Nome do gerenciador de recursos local (Fork, PBS)
LRMS_TYPE	Tipo do gerenciador de recursos local (Fork, PBS)
QUEUE_NAME	Nome da fila de processos
QUEUE_NODECOUNT	Total de recursos de processamento da fila
QUEUE_FREENODECOUNT	Total de recursos de processamento livres da fila
QUEUE_MAXTIME	Tempo máximo de permanência na fila
QUEUE_MAXCPU TIME	Tempo máximo de CPU das tarefas da fila
QUEUE_MAXCOUNT	Número máximo de submissões permitidas na fila
QUEUE_MAXRUNNINGJOBS	Número máximo de execuções simultâneas na fila
QUEUE_MAXJOBSINQUEUE	Número máximo de tarefas na fila
QUEUE_PRIORITY	Prioridade da fila
QUEUE_STATUS	Estado da fila

Tabela B.1: Parâmetros para configuração de ambiente para submissão de tarefas no GridWay.

Referências Bibliográficas

ABBAS, A. *Grid Computing: A Practical Guide to Technology and Applications*. 1. ed. Hingham, Massachusetts, EUA: CHARLES RIVER MEDIA, INC, 2004. ISBN 1-58540-236-2.

ALEGRO, M. et al. Rhcnet: Aplicação para consolidação nacional de registros hospitalares de câncer. *CBIS 2006 - X Congresso Brasileiro de Informática em Saúde*, 2006.

ALIZADEH, A. A. et al. *Lymphoma Leukemia Molecular Profiling Project*. 2000. Lymphoma Leukemia Molecular Profiling Project.

ALVES, H. A. V. et al. Oncogrid: A proposal of grid infrastructure for the establishment of a national health information system on childhood cancer. *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium*, Miami, FL, USA, p. 1-7, 2008.

ANDRADE, N. et al. Peer-to-peer grid computing with the ourgrid community. *Proceedings of the 23rd Brazilian Symposium on Computer Networks*, May 2005.

ANTONIOLETTI, M. et al. Ogsa-dai usage scenarios and behaviour: Determining good practice. *UK e-Science All Hands Meeting*, 9 2004.

ATA. *American Telemedicine Association*. 9 2005. Acessado em março de 2006. Disponível em: <<http://www.americantelemed.org/news/definition.html>>.

BACIC, A. S. *Um Ambiente Colaborativo de Apoio ao Diagnóstico Médico Assistido por Computadores de Alto Desempenho*. Dissertação (Mestrado) — Universidade de São Paulo, São Paulo - SP, 2001.

BADER, D. A. Computational biology and high-performance computing. *Communications of the ACM*, v. 47, n. 11, nov 2004.

BARROS, A. São grandes os desafios para o sistema nacional de informações em saúde. *Ciência E Saúde Coletiva*, v. 11, n. 4, 2006. ISSN 1413-8123.

BERMAN, F.; HEY, A.; FOX, G. *Grid Computing Making the Global Infrastructure a Reality*. 1. ed. [S.l.]: Wiley, 2003. (Wiley Series in Communications Networking & Distributed Systems, v. 1). ISBN 0-470-85319-0.

BILYKH, I. Can grid services provide answers to the challenges of national health information sharing? *Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative research*, 2003.

BRASILEIRO, F.; CIRNE, W. *Portal do Projeto OurGrid*. 2005. Acessado em novembro de 2006. Disponível em: <<http://www.ourgrid.org/>>.

BRASILEIRO, F.; CIRNE, W. *Portal Laboratório de Sistemas Distribuídos*. 2005. Acessado em junho de 2007. Disponível em: <<http://www.lsd.ufcg.edu.br/>>.

BRAZMA, A. et al. Minimum information about a microarray experiment (miami)—toward standards for microarray data. *Nature Genetics*, v. 29, n. 15, 2001.

BRESNAHAN, J. et al. Globus gridftp: What's new in 2007. *First International Conference on Networks for Grid Applications (GridNets 2007)*, 2007.

CAMPOS, M. et al. Computação em grade na saúde: Proposta de uma arquitetura para integração e informações médicas. *Congresso Brasileiro de Informática em Saúde*, 2006. Disponível em: <<http://www.sbis.org.br/cbis/anaiscbis2006.htm>>.

CFM. *Portal Médico Conselho Federal de Medicina*. 9 2007. Acessado em janeiro de 2007. Disponível em: <http://www.portalmedico.org.br/resolucoes/cfm/2002/1643_2002.htm>.

CHAKRABARTI, A. *Grid Computing Security*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007. ISBN 3540444920.

CLUSTER-RESOURCE. *Cluster Resource Web Page - TORQUE Resource Manager*. 2007. Acessado em novembro de 2007. Disponível em: <<http://www.clusterresources.com/>>.

CRISTO, E. B. *Métodos estatísticos na análise de experimentos de microarray*. Dissertação (Mestrado) — Instituto de Matemática e Estatística da Universidade de São Paulo, São Paulo - SP, 2003.

CUNHA, R. Cartão nacional de saúde: os desafios da concepção e implantação de um sistema nacional de captura de informações de atendimento em saúde. *Ciência e Saúde Coletiva*, v. 7, n. 4, 2002. ISSN 1413-8123.

CZAJKOWSKI, K. et al. From open grid services infrastructure to ws-resource framework: Refactoring & evolution. *Global Grid Forum Draft Recommendation*, 2004.

DEFANTI, T. et al. Overview of the i-way: Wide area visual supercomputing. *International Journal of Supercomputer Applications*, v. 10, n. 2, p. 123–130, 1996.

DUAN, Q.; TALLEY, M.; SEETHA, N. Qos-based network service broker for high-performance grid applications. *IEEE Computer Society*, Washington, DC, USA, p. 978–985, 2007.

EISEN, M. *ScanAlizer User Manual*. Stanford, CA: [s.n.], 1999. Acessado em agosto de 2007. Disponível em: <<http://rana.lbl.gov/EisenSoftware.htm>>.

FELLER; FOSTER, I.; MARTIN, S. Gt4 gram: A functionality and performance study. *TERAGRID 2007 CONFERENCE*, 2007.

FENSTERMACHER, D. et al. The cancer biomedical informatics grid. *Engineering in Medicine and Biology 27th Conference*, 2005.

- FERREIRA, L. et al. *Introduction to Grid Computing with Globus*. [S.l.]: IBM Redbooks, 2003. ISBN 0738499889.
- FOSTER, I. What is the grid? a three point checklist. *GRIDToday*, jul 2002.
- FOSTER, I. et al. A security architecture for computational grids. ACM, New York, NY, USA, p. 83–92, 1998.
- FOSTER, I.; KESSELMAN, C.; TUECKE, S. The anatomy of the grid: Enabling scalable virtual organizations. *International J. Supercomputer Applications*, v. 15, n. 3, 2001.
- FOSTER, I. et al. *The Open Grid Services Architecture, Version 1.5*. [S.l.], July 24 2006. Disponível em: <<http://www.ggf.org/documents/GFD.80.pdf>>.
- FOSTER, I. T. Globus toolkit version 4: Software for service-oriented systems. Springer, v. 3779, p. 2–13, 2005.
- FREITAS, S. R. et al. The coupled aerosol and tracer transport model to the brazilian developments on the regional atmospheric modeling system (catt-brams). part 1: Model description and evaluation. *ACPD/European Geosciences Union MS*, 2007.
- FREY, J. et al. Condor-g A computation management agent for multi-institutional grids. *Cluster Computing*, v. 5, p. 237–246, 2002.
- GIBAS, C.; JAMBECK, P. *Developing Bioinformatics Computer Skills*. 1. ed. [S.l.]: O'Reilly, 2003. (O'Reilly, v. 1). ISBN 1-56592-664-1.
- GLOBUS Alliance. 7 2007. Acessado em abril de 2007. Disponível em: <<http://www.globus.org/>>.
- GLOBUSALIANCE. *GridWay Metascheduling Technologies for the Grid*. 2007. Acessado em dezembro de 2007. Disponível em: <<http://www.gridway.org/>>.
- GOLDCHLEGER, A. *InteGrade: Um Sistema de Middleware para Computação em Grade Oportunista*. Dissertação (Mestrado) — Unidade Instituto de Matemática e Estatística (IME) da Universidade de São Paulo, São Paulo - SP, 2004.
- HENDERSON, R. L. Job scheduling under the portable batch system. Springer-Verlag, London, UK, p. 279–294, 1995.
- HERRERA, J. et al. Execution of typical scientific applications on globus-based grids. IEEE Computer Society, Washington, DC, USA, p. 177–183, 2004.
- HIRA, A. Y. *Telessaúde Sobre Tecnologias Livres: Um Sistema de Informação em Saúde Para Gestão da Atenção em Oncologia Pediátrica*. Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo, São Paulo - SP, 2005.
- HUDO, E.; MONTEIRO, R.; LLORENTE, I. The gridway framework for adaptive scheduling and execution on grids. Practice and Experiences, p. 1–7, 2005.
- JITHESH, P. et al. Genegrid: Architecture, implementation and application. *Journal of Grid Computing*, v. 4, n. 2, p. 209–222, jun 2006. Disponível em: <<http://www.springerlink.com/content/b3318452682p2215/>>.

- JITHESH, P. et al. Genegrid: Grid service based virtual bioinformatics laboratory. *Global Grid Forum*, Workshop on Grid Applications: from Early Adopters to Mainstream Users, Chicago, USA, v. 14, p. 1–9, 2005.
- JITHESH, P. V. et al. Genegrid: grid based solution for bioinformatics application integration and experiment execution. *18th IEEE Symposium on Computer-Based Medical Systems*, p. 523–528, june 2005.
- KRAUTER, K.; BUYYA, R.; MAHESWARAN, M. A taxonomy and survey of grid resource management systems for distributed computing. *Software Practice and Experience*, v. 32, n. 2, p. 135–164, 2002.
- LING, M.; LEE, T. *Analysis of Microarray Gene Expression Data*. Boston, USA: Kluwer Academic Publishers, 2004. ISBN 0-7923-7087-2.
- LONGO, K. S. R. et al. The coupled aerosol and tracer transport model to the brazilian developments on the regional atmospheric modeling system (catt-brams). part 2: Model sensitivity to the biomass burning inventories. *atmos. chem. phys. discuss. ACPD/European Geosciences Union MS*, 2007.
- MINOLI, D. *A Networking Approach to Grid Computing*. [S.l.]: Wiley-Interscience, 2004. ISBN 0471687561.
- MUNGIOLI, A. S. *Uma proposta de tecnologia para videoconferência integrando tecnologias grid*. Tese (Doutorado) — Escola Politécnica (EP) da Universidade de São Paulo, São Paulo - SP, 2005.
- MUNIZ, J. R. *Aplicação da bioinformática no estudo dos genes e enzimas envolvidos na síntese da goma fastidiana produzida pela xylella fastidiosa*. Dissertação (Mestrado) — Instituto de Física de São Carlos-Universidade de São Paulo, São Carlos, 2003. Disponível em: <<http://www.teses.usp.br>>.
- NORONHA, C. P. et al. *Estimativa 2008, Incidência de Câncer no Brasil*. 1. ed. [S.l.]: Ministério da Saúde do Brasil, 2008.
- OASIS. *OASIS Advancing Open Standards for Information Society*. 2007. Acessado setembro de 2007. Disponível em: <<http://www.oasis-open.org/>>.
- OGF. *Open Grid Forum*. 08 2007. Acessado em fevereiro 2007. Disponível em: <<http://www.ogf.org/>>.
- PASANEN, T. et al. *DNA Microarray Data Analysis*. Finland: CSC – Scientific Computing Ltd., 2003. ISBN 952-9821-89-1.
- RATTEI, T. et al. Using public resource computing and systematic pre-calculation for large scale sequence analysis. *Springer*, v. 4360, p. 11–18, 2006.
- RIBEIRO, C.; OLIVEIRA, F.; B., B. S. Portal web da grade computacional do Incc. *Workshop em Grade Computacional e Aplicações*, 1 2003. Disponível em: <http://www.lncc.br/~jauvane/papers/wgca03_cribeiro.pdf>.
- RODRIGUES, K. E.; CAMARGO, B. de. Diagnóstico precoce do câncer infantil: Responsabilidade de todos. *Revista Associação Médica Brasileira*, v. 1, n. 49, 2003.

ROSA, A. et al. Development of a collaborative environment applied to pediatric oncology. *Of The 21st ACM Symposium On Applied Computing*, v. 21, apr 2006.

SALTZ, J. et al. cagrid: design and implementation of the core architecture of the cancer biomedical informatics grid. *Bioinformatics*, Oxford University Press, Oxford, UK, v. 22, n. 15, p. 1910–1916, 2006. ISSN 1367-4803.

SAUDE, M. da. *Portal Saúde, Ministério da Saúde*. 2008. Acessado em janeiro de 2008. Disponível em: <<http://portal.saude.gov.br/saude/>>.

SCHOPF, J. M. et al. *Monitoring and Discovery in a Web Services Framework: Functionality and Performance of the Globus Toolkit's MDS4*. [S.l.], 2005. 1–14 p.

SEBASTIANI, P. et al. Statistical challenges in functional genomics. *Statistical Science*, v. 18, n. 1, p. 33–70, 2003.

SILBERSCHATZ, A.; GAGNE, G.; GALVIN, P. B. *Sistemas Operacionais com Java*. Brasil, São Paulo, SP: Editora Campus, 2004. ISBN 8535214852.

SOUZA, I. A. et al. Direct volumetric rendering based on point primitives in opengl. *The MedicineMeets Virtual Reality Conference*, jan 2006.

THAIN, D.; TANNENBAUM, T.; LIVNY, M. Distributed computing in practice: the condor experience. *Concurrency - Practice and Experience*, v. 17, n. 2-4, p. 323–356, 2005.

TOP500, S. Top500 list statistics. *Online publication*:<http://www.top500.org/statistics>, 2007.

YANG, X.; SUN, X. Meta-analysis of several gene lists for distinct types of cancer: A simple way to reveal common prognostic markers. *BMC Bioinformatics*, v. 8, n. 1, p. 118, 2007. ISSN 1471-2105. Disponível em: <<http://www.biomedcentral.com/1471-2105/8/118>>.