MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE- 14797-TDI/1240

CLIMATOLOGIA DE MESOESCALA EM GRADE COMPUTACIONAL

Eugenio Sper de Ameida

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Haroldo Fraga de Campos Velho e Airam Jônatas Preto, aprovada em 27 de fevereiro de 2007.

INPE
São José dos Campos
2007

Publicado por: **esta página é responsabilidade do SID**

Instituto Nacional de Pesquisas Espaciais (INPE)
Gabinete do Diretor – (GB)
Serviço de Informação e Documentação (SID)
Caixa Postal 515 – CEP 12.245-970
São José dos Campos – SP – Brasil
Tel.: (012) 3945-6911
Fax: (012) 3945-6919
E-mail: pubtc@sid.inpe.br

Solicita-se intercâmbio
We ask for exchange

Publicação Externa – É permitida sua reprodução para interessados.

MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE- 14797-TDI/1240

CLIMATOLOGIA DE MESOESCALA EM GRADE COMPUTACIONAL

Eugenio Sper de Ameida

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Haroldo Fraga de Campos Velho e Airam Jônatas Preto, aprovada em 27 de fevereiro de 2007.

INPE
São José dos Campos
2007

681.3.02 : 551

Almeida, E. S.
Climatologia de mesoescala em grade computacional/ Eugênio Sper de Almeida – São José dos Campos: INPE, 2007.
149p. ; (INPE-14797-TDI/1240)

1.Computação em grade. 2.Meteorologia. 3.Modelos atmosféricos. 4.Processamento de alto desempenho. 5.Infra-estrutura de grade. I.Título.

Aprovado (a) pela Banca Examinadora em cumprimento ao requisito exigido para obtenção do Título de **Doutor(a)** em

Computação Aplicada

Dr. Stephan Stephany

Presidente / INPE / SJCampos - SP

Dr. Haroldo Fraga de Campos Velho

Orientador(a) / INPE / SJCampos - SP

Dr. Airam Jônatas Preto

Orientador(a) / INPE / SJCampos - SP

Dr. Jairo Panetta

Convidado(a) / USP / Cachoeira Paulista - SP

Dr. Philippe Olivier Alexandre Navaux

Convidado(a) / UFRGS / Porto Alegre - RS

Aluno (a): **Eugênio Sper de Almeida**

São José dos Campos, 27 de Fevereiro de 2007

“Somos o que pensamos. Tudo o que somos vem dos nossos pensamentos. Com nossos pensamentos fazemos o mundo; sustentamos o mundo com o nosso diálogo interior.”

Siddharta Gautama

A meus pais,
Milton Pires de Almeida e
Anna Esmenia Sper de Almeida.

AGRADECIMENTOS

À minha esposa Lilian e aos meus filhos Pedro Henrique e João Guilherme.

Ao Instituto Nacional de Pesquisas Espaciais – INPE, pela oportunidade de estudos e utilização de suas instalações.

Aos meus orientadores Dr. Haroldo Fraga de Campos Velho e Dr. Airam Jônatas Preto.

Aos colegas Álvaro Luiz Fazenda, Márcia Maria Schubert Dolbrowolsky, Eduardo Rocha Rodrigues, Roberto Pinto Souto e Rogério da Silva e Souza.

Expresso minha gratidão e reconhecimento a todo corpo docente do curso de Computação Aplicada do Instituto Nacional de Pesquisas Espaciais.

Ao Dr. Celso Luis Mendes, ao Dr. Jairo Panetta, ao Dr. José Paulo Bonatti, ao Dr. Júlio César Santos Chagas, a Dra. Maria Assunção Faus Dias, ao Dr. Saulo Freitas e ao Dr. Stephan Stephany pelo apoio dado.

E a todos que de maneira direta e indireta contribuíram para a realização deste trabalho, disponibilizando informações técnicas ou demonstrando amizade e respeito.

RESUMO

A determinação da climatologia de modelos meteorológicos, com técnicas de ensemble, necessita recursos computacionais com alto poder computacional devido à grande quantidade de dados que processa. Por se tratar de uma aplicação cuja execução ocorre de forma independente, é adequada para grade computacional pois possibilita que este esforço computacional possa ser dividido por outras máquinas distribuídas geograficamente. Grade computacional consiste de infra-estrutura de hardware e software que provê acesso confiável e consistente a recursos computacionais localizados em pontos geograficamente diferentes. No entanto, problemas relacionados à transferência de dados, armazenamento e escalonamento podem influenciar no desempenho desta aplicação em grade. No Projeto G-BRAMS, os objetivos foram desenvolver metodologia para geração de climatologia em grade computacional; testar 3 (três) plataformas de grade para esta aplicação: OurGrid, CIGRI/OAR e Globus; e disponibilizar pelo menos 10 anos de climatologia do modelo numérico de mesoescala "Brazilian Regional Atmospheric Modeling System" (BRAMS), de três regiões do Brasil (Norte, Nordeste e Sul/Sudeste), para a comunidade. Esta grade é composta de "clusters" instalados no Laboratório de Computação Aplicada (LAC) e no Centro de Previsão de Tempo e Estudos Climáticos (CPTEC) – pertencentes ao Instituto Nacional de Pesquisas Espaciais (INPE) - e no Instituto de Informática (II) da Universidade Federal do Rio Grande do Sul (UFRGS). No entanto, apesar de ter sido comprovado que aglutinar recursos em grade computacional ser viável e funcionar para a execução da climatologia, utilizando um portal, necessitava-se saber se o desempenho da aplicação e se a proposta de uso seriam adequadas para um uso mais extensivo. Desta forma, os objetivos desta tese foram: implementar a climatologia desenvolvida pelo "International Research Institute for Climate Prediction" (IRI) em grade computacional, compreender o funcionamento da execução da "climatologia" mensal de três anos (janeiro de 1996 a dezembro de 1998) do BRAMS (teste de conceito de 3 anos), na grade computacional do projeto G-BRAMS e fazer uma análise mais refinada do comportamento e do desempenho efetivo da climatologia no ambiente desta grade, por meio da temporização do processamento em cada nó e da transmissão dos dados. Verificou-se que a metodologia desenvolvida para a climatologia é robusta em uma grade dedicada com portal dedicado. A questão da armazenagem/transferência dos dados e das falhas ocorridas na grade devem ser melhor tratadas para garantir o desempenho da aplicação.

MESOSCALE CLIMATOLOGY IN COMPUTACIONAL GRID

ABSTRACT

Establishing the climatology of meteorological models with ensemble techniques demands high computational power due to the huge amount of data to be processed. Being an application whose execution is independent, it is suitable for computational grid since it allows the computational effort to be shared with other geographically distributed machines. Computational grids consist of hardware and software infrastructure that provides trustworthy and consistent access to computational resources located at different geographical places. However, problems related to data transferring, storage and scheduling may influence application performance in a grid. G-BRAMS Project goals have been to develop a methodology for the generation of climatology in computational grid; to test 3 (three) grid platforms for this application: OurGrid, CIGRI/OAR and Globus; and to provide at least 10 years of the mesoscale model "Regional Brazilian Atmospheric Modeling System" (BRAMS) climatology for the community, for three regions of Brazil (North, Northeast and Southeastern/South). This grid comprises "clusters" installed at the Laboratório de Computação Aplicada (LAC), at the Centro de Previsão de Tempo e Estudos Climáticos (CPTEC) - belonging to the Instituto Nacional de Pesquisas Espaciais (INPE) - and at the Instituto de Informática (II) of the Universidade Federal do Rio Grande do Sul (UFRGS). However, although it has been proven that agglutinating resources in computational grid is viable and works for climatology generation using a portal, it was unknown if the application performance and the proposed use would be adequate for more extensive use. In this way, the objectives of the experiment described here have been: to implement the climatology developed by "International Research Institute for Climate Prediction" (IRI) in the computational grid; to understand how works the execution of the three-year "monthly climatology" (January-1996 to December-1998) of the BRAMS (three-year concept test) in the computational grid of G-BRAMS project; and to make a more refined analysis of the behavior and of the effective performance of the climatology in the environment of this grid, through the temporization of application processing in each node and data transmission. The results had shown that the developed scheduling works for a dedicated grid with dedicated portal. It was observed a performance fall of the application due to non-availability of the grid and to data communication time. It was verified that the methodology developed for the climatology is robust in a dedicated grid with dedicated portal. The question of data storage/transfer and of failures occurred on the grid must be better treated to guarantee the performance of the application.

SUMÁRIO

Pág.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

AppLeS	- Application-Level Scheduling"),
APST	- AppLeS Master-Worker Application Template
BOT	- Bag-Of-Tasks
BRAMS	- Brazilian Regional Atmospheric Modeling System
CPTEC	- Centro de Previsão de Tempo e Estudos Climáticos
CA	- Autoridade Certificadora
CIGRI	- CIMENT GRID
CIMENT	- Calcul Intensif, Modélisation, Expérimentation Numérique et Technologique
COLA	- Center for Ocean, Land and Atmosphere Studies
CRC	- Cyclical Redundancy Check
DERF	- Dynamical Extended-Range Forecasting
DNS	- Domain Name System
ECMWF	- European Centre for Medium-Range Forecasts
ENSO	- El Nino/Oscilação Sul
FCFS	- First Come First Serve
GRIS	- Grid Resource Information Service
GIIS	- Grid Index Information Service
GrADS	- Grid Application Development Software Project
GrADS	- Grid Analysis and Display System
GRAM	- Globus Resource Allocation Manager
GRIP	- GRid Information Protocol
GRRP	- Grid Registration Protocol
GSI	- Grid Security Infrastructure
GuM	- Grid Machines

II	- Instituto de Informática
INPE	- Instituto Nacional de Pesquisas Espaciais
IRI	- International Research Institute for Climate Prediction
ISAN	- ISentropic ANalysis package
JDL	- Job Descripition Language
LAC	- Laboratório de Computação Aplicada
LDAP	- Lightweight Directory Access Protocol
LEAF	- Land Ecosystem-Atmospheric Feedback Model
LSF	- Load Sharing Facility
MCGA	- Modelo de Circulação Geral Atmosférico
MDS	- Monitoring and Discovery Service
MIMD	- Multiple Instruction Multiple Data
NCEP	- National Center for Environmental Prediction
NQE	- Network Queuing Environment
NWS	- Network Weather Service
PBS	- Portable Batch System
RAMS	- Regional Atmospheric Modeling System
SSL	- Secure Socket Layer
S.O.	- Sistema Operacional
SGE	- Sun Grid Engine
SIMD	- Single Instruction Multiple Data
SSiB	- Simplified Simple Biosphere Model
TSM	- Temperatura de superfície do mar
UFRGS	- Universidade Federal do Rio Grande do Sul

CAPÍTULO 1

INTRODUÇÃO

A história da computação iniciou-se com Wilhelm Schickard que desenvolveu o primeiro equipamento mecânico considerado máquina de calcular (perdida na Guerra dos Trinta Anos). Charles Babbage ficou conhecido como o "Pai do Computador" após projetar o "Calculador Analítico", equipamento mecânico que permitiria a execução de cálculos automaticamente e seria muito próximo da concepção de um computador atual. As máquinas desta época utilizavam base decimal (0 a 9), mas foram encontradas dificuldades em implementar um dígito decimal em componentes eletrônicos, pois qualquer variação provocada por um ruído causaria erros de cálculo consideráveis. O matemático inglês George Boole publicou em 1854 os princípios da lógica booleana, onde as variáveis assumem apenas valores 0 e 1 (verdadeiro e falso), que passou a ser utilizada a partir do início do século XX. A partir de meados dos anos 40, surgiram os computadores eletrônicos, que inicialmente foram construídos utilizando válvulas eletrônicas, depois transistores e atualmente é empregada a tecnologia de circuitos integrados (WIKIPEDIA, 2007).

Uma das primeiras aplicações da era da computação eletrônica digital foi a previsão numérica de tempo, que foi executada com sucesso no Eletronic Number Integrator And Computer (ENIAC, reconhecido como primeiro computador eletrônico de propósito geral (CHARNEY et al., 1950). Posteriormente em 1946, John Von Neuman organizou o Projeto Computador Eletrônico, que tinha o objetivo de projetar e construir um computador eletrônico que superasse o desempenho dos anteriores. Uma das mais importantes características deste computador era ser “paralelo”, isto é, permitiria que o processamento se realizasse sobre todo o número ao invés de bit a bit (SHUMAN, 1989). Esta característica permitiu que a previsão de 24 horas, que era executada no ENIAC em 24 horas, fosse executada neste novo computador em 5 minutos.

A pressão pelo aumento do poder de computação fez com que os projetistas de computadores aumentassem velocidade dos processadores. No entanto, as necessidades crescentes de processamento não eram mais atendidas pelo avanço da tecnologia de processadores. A solução encontrada foi o emprego de vários processadores em conjunto na obtenção de uma maior capacidade de processamento. Surge então o termo processamento paralelo para designar as várias técnicas para atender as demandas de processamento. O desenvolvimento desta área levou ao surgimento dos supercomputadores, máquinas capazes de realizar grandes quantidades de operações por segundo, e que tinha capacidade de processamento vetorial. Em seguida surgiram as arquiteturas que empregam paralelismo de memória central e distribuída. As últimas arquiteturas que apareceram foram os agregados de computadores ("clusters"), interligados por uma rede de comunicação rápida e com uma biblioteca que possibilita a execução de programas paralelos (De ROSE ; NAVAUX, 2003).

No entanto, os recursos computacionais de alto desempenho sempre estiveram vinculados a uma localidade física. A partir 1969, a intercomunicação de computadores a nível mundial torna-se possível com a criação da "Advanced Research Projects Agency Network" (ARPANET) do Departamento de Defesa dos Estados Unidos da América, que atualmente é conhecida como Internet. Esta evolução levou à pesquisa para possibilitar a utilização de recursos computacionais geograficamente distribuídos, interligados pela Internet, como outra facilidade para auxiliar no desafio de computação intensiva (computação em grade). Ian Foster e Carl Kesselman foram os pioneiros nesta área, definindo que a computação em grade difere da computação distribuída convencional devido ao fato de ser voltada para compartilhamento de recursos em larga escala (como supercomputadores, "clusters", sistema de armazenamento, dados, instrumentos e pessoas), em alguns casos, orientada ao alto desempenho (FOSTER et al., 2001). Esta tecnologia é uma nova forma de fazer computação, que possui enormes vantagens aos seus usuários e inúmeros desafios aos seus desenvolvedores.

No entanto, problemas atacados por sistemas de alto desempenho nem sempre podem ser resolvidos eficientemente pela grade. A previsão de tempo curto prazo, utilizando técnica de decomposição de domínio, não é uma solução prática em grade devido à

interdependência das informações dos subdomínios. Este fator compromete uma previsão de curto prazo dentro dos requisitos de um ambiente meteorológico operacional.

O Centro de Previsão de Tempo e Estudos Climáticos (CPTEC) é uma das Coordenadorias do Instituto Nacional de Pesquisas Espaciais (INPE) e é responsável pela previsão numérica de tempo e clima no Brasil. Produtos diários relacionados à previsão de curto prazo e mensais relativos à previsão de médio prazo são disponibilizados a toda sociedade brasileira.

A previsão de tempo é disponibilizada para sete dias (resolução global de 63 Km e regional de 20 km) e a previsão oceânica para cinco dias (resolução global de 111 Km e regional de 33 km), geradas a cada seis horas (quatro execuções diárias). A confiabilidade desses modelos é alta para os três primeiros dias. Recentemente a previsão ambiental passou a fazer parte dos produtos disponibilizados pelo CPTEC/INPE para toda a sociedade. Esta previsão é executada uma vez ao dia e fornece previsão ambiental para dois dias com resolução de 30 Km: informações sobre emissões produzidas por queimadas e dispersão de constituintes usando-se o monóxido de carbono como traçador. Modelos hidrológicos são utilizados em pesquisa para previsão de enchentes, riscos de desencadeamento de processos erosivos e deslizamentos de terra em encostas – projeto BRAMSNet.

A previsão climática operacional gerada no CPTEC/INPE para todo o país, na resolução espacial de 200 km é importante para várias aplicações, mas existem aplicações que demandam de anomalias climatológicas com resolução maior. Porém, gerar previsões climáticas globais na resolução necessária, usualmente 40 Km, é tarefa difícil de ser realizada operacionalmente, devido ao alto custo computacional de processamento desse grande volume de dados.

A alternativa é utilizar modelos que possam gerar previsões climáticas de mesoescala (maior resolução espacial), onde as condições iniciais e de contorno são obtidas a partir dos campos numéricos do modelo de previsão global.

O processo de previsão de clima necessita da climatologia do modelo empregado, que consiste na determinação de médias, para um certo período, da simulação do estado da atmosfera por este modelo por um período longo de tempo – de 30 a 50 anos. Na metodologia de determinação da climatologia empregada pelo "International Research Institute for Climate Prediction" (IRI) é realizada uma única integração longa, enquanto que na do "European Centre for Medium-Range Forecasts" (ECMWF) são realizadas múltiplas integrações distintas, iniciadas em meses distintos e com duração curta (alguns meses).

Uma técnica muita utilizada em climatologia é o uso da estratégia de ensemble. Nesta técnica são executadas múltiplas integrações, cada uma iniciando com uma condição inicial distinta.

Durante o processo de simulação de mesoescala, o modelo necessita de condições de contorno em determinados instantes de tempo e de condições iniciais, que são provenientes de saídas de climatologia de modelos globais. Após a simulação, são calculadas médias mensais sobre o ensemble. Há poucos trabalhos na literatura tratando de climatologia de mesoescala.

A determinação da climatologia, utilizando técnicas de ensemble, tipicamente utiliza um grande volume de dados e longos tempos de integração, o que torna o tempo de execução muito alto para um único computador, mesmo que possua alto desempenho computacional. É uma aplicação que possui características apropriadas para estudo e avaliação de grades computacionais na área de meteorologia. Como a execução de cada membro do ensemble pode ser realizada de forma independente, este aspecto pode ser explorado no uso de computação geograficamente distribuída, utilizando infra-estrutura de grade computacional. Grade computacional é uma das metodologias mais modernas em ciência da computação.

O objetivo da presente tese é avaliar o desempenho de uma aplicação usando a infra-estrutura da computação em grade. A aplicação é climatologia para o modelo BRAMS.

“Clusters” são recursos de alto desempenho que estão cada vez mais utilizados em centros de pesquisa e universidades. Mesmo assim, nota-se que a maioria dos centros regionais, estaduais e faculdades de meteorologia carecem de recursos computacionais substanciais para o processamento de modelos meteorológicos de tempo e clima nas escalas espacial e temporal necessárias para seus trabalhos de pesquisa. Neste caso, a solução é aglutinar os recursos computacionais das instituições que possuam mesmos objetivos em uma grade computacional, como o caso do projeto G-BRAMS. Este projeto atendeu a uma chamada ao edital de Computação em Grade - Grade-01/2004, sendo financiado com recursos da FINEP CT-INFO, e implementou uma grade de “clusters” localizados no Laboratório Associado de Computação e Matemática Aplicada (LAC) e no Centro de Previsão de Tempo e Estudos Climáticos (CPTEC) do Instituto Nacional de Pesquisas Espaciais (INPE) e no Instituto de Informática (II) da Universidade Federal do Rio Grande do Sul (UFRGS).

Definiu-se no projeto que a climatologia do “Brazilian Regional Atmospheric Modeling System” - BRAMS (INPE/CPTEC, 2006) - seria determinada para três regiões do Brasil (Norte, Nordeste e Sul/Sudeste), utilizando o método “International Research Institute for Climate Prediction” (IRI).

O projeto G-BRAMS teve os seguintes objetivos:

- Desenvolver metodologia para geração de climatologia em grade computacional;
- Testar 3 (três) plataformas de grade para esta aplicação: OurGrid (ANDRADE et al., 2003), CIGRI/OAR (CAPIT, 2004) e Globus (FOSTER e KESSELMAN, 1999);
- Disponibilizar pelo menos 10 anos de climatologia do modelo numérico BRAMS para a comunidade.

Neste experimento, empregou-se uma grade computacional para diminuir o tempo total de execução da climatologia, pois cada nó de grade contribui para a execução de uma parcela da climatologia. Uma avaliação preliminar utilizando um “cluster” de 17

processadores (sendo um deles dedicado à entrada/saída de dados), e um membro, mostrou que a climatologia de 10 anos possui o tempo de execução de 112,5 dias, comprovado durante o experimento de grade com o valor total acumulado de 117 dias (SOUTO et al, 2007).

Para a diminuir o tempo de execução da aplicação, no contexto de uma plataforma de grade computacional, é necessário não apenas compartilhar recursos computacionais poderosos mas também ter disponível um mecanismo eficiente para decidir onde executar a aplicação. Além disso, há a necessidade de gerenciamento da armazenagem e transferência de um grande volume de dados. Um conjunto de um ano de condições de fronteira, para as regiões de interesse do projeto G-BRAMS, possui tamanho de aproximadamente 2 GB, que se tornam 180 GB se considerarmos uma simulação de 30 anos utilizando três membros.

No entanto, apesar de ter sido comprovado que aglutinar recursos em grade computacional ser viável e funcionar para a execução da climatologia, utilizando um portal, para uma análise científica de desempenho é importante conhecer o comportamento da aplicação e se a proposta de uso seria adequada para uma utilização voltada para a prática operacional.

Desta forma, o objetivo deste trabalho de tese trata-se de medidas de uma aplicação de computação em grade, visando compreender o funcionamento da execução da "climatologia" mensal de três anos (janeiro de 1996 a dezembro de 1998) do BRAMS (teste de conceito de 3 anos), na grade computacional do projeto G-BRAMS e fazer uma análise mais refinada do comportamento e do desempenho efetivo da climatologia no ambiente desta grade.

Para isto, o programa para determinação da climatologia foi temporizado para coletar os tempos de processamento em cada nó e de transmissão dos dados. Os resultados obtidos durante o experimento foram analisados e mostraram que a plataforma de grade que apresentou melhor desempenho foi o OurGrid, mas este resultado não é conclusivo pois foram verificadas quedas de desempenho da climatologia devido a interrupções no processamento na grade ou dos nós da grade devido a problemas com a Internet, falhas

na rede, na administração dos recursos, e indisponibilidade dos recursos. Verificou-se também um alto tempo ociosidade da grade devido ao tempo gasto na transferência de dados entre o portal e os nós de grade.

O projeto G-BRAMS funcionou adequadamente com grade dedicada com portal dedicado. No entanto, o ideal é que mais aplicações e mais usuário possam fazer uso de uma grade computacional. Isto requer que a grade seja dedicada com portal aberto. As análises feitas a partir do experimento demonstraram que existe ineficiência na transmissão de dados. Esses dois fatores requerem que uma nova estratégia de escalonamento seja implementada.

O Capítulo 2 descreve o conceito básico de grades computacionais e explora com mais detalhe as características das três plataformas de grade que estão sendo analisadas dentro do escopo do projeto: Globus, OurGrid e CIGRI/OAR. Ao final é feita uma comparação entre as três plataformas.

No Capítulo 3 é apresentada uma revisão sobre escalonamento em grades computacionais, apresentando trabalhos na área escalonamento de aplicações paralelas em grades computacionais.

Uma descrição de modelos numéricos de previsão de tempo e clima, com ênfase no modelo BRAMS, é apresentada no Capítulo 4.

No Capítulo 5 é descrita uma revisão sobre climatologia, discutindo as metodologias DERF/ECMWF E IRI e a utilização de “ensemble” para determinação de climatologia. Nesse Capítulo também é apresentada a simulação global de 50 anos, em modo “ensemble”, e a climatologia regional.

No Capítulo 6 é apresentado o projeto G-BRAMS (acrônimo de Grid-BRAMS), o algoritmo de climatologia, o funcionamento do BRAMS em “clusters”, o acesso à grade computacional, as estratégias de escalonamento utilizadas no projeto com as plataformas de grade OurGrid, CIGRI/OAR, Globus.

No Capítulo 7 são apresentados os resultados do experimento na grade com a temporização da climatologia, assim com uma análise qualitativa e quantitativa dos mesmos para as três plataformas de grade utilizadas.

No Capítulo 8 são apresentadas as conclusões sobre os resultados do trabalho e sugeridos temas para continuidade desta pesquisa.

CAPÍTULO 2

COMPUTAÇÃO EM GRADE

Segundo Foster et al. (2001), a computação em grade difere da computação distribuída convencional devido ao fato de possibilitar compartilhamento coordenado de recursos em larga escala (como supercomputadores, “clusters”, sistema de armazenamento, dados, instrumentos e pessoas) e permitir a resolução de problemas computacionais em organizações virtuais multiinstitucionais, em alguns casos orientados ao alto desempenho. O compartilhamento deve necessariamente ser altamente controlado, com os fornecedores de recursos e consumidores definindo claramente e cuidadosamente apenas o que é compartilhado, a quem é permitido compartilhar e as condições nas quais o compartilhamento ocorre. A seguir apresentamos uma descrição detalhada das plataformas de grade a serem utilizadas neste projeto: Globus, CIGRI/OAR e OurGrid

2.1 Globus toolkit

A arquitetura de grade, proposta por Foster et al. (2001), segue os princípios do modelo ampulheta (“NATIONAL RESEARCH COUNCIL”, 1994). Neste modelo (Figura 2.1), diferentes comportamentos de alto nível (topo da ampulheta) são mapeados em um pequeno conjunto de abstrações centrais (“core abstrations”) e protocolos (parte mais estreita da ampulheta), que por sua vez são mapeados em diferentes tecnologias fundamentais (base da ampulheta).

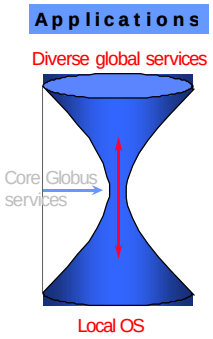


Figura 2.1 - Modelo ampulheta do Globus

Fonte: Foster e Kesselman (1999)

Na correspondência entre o modelo ampulheta e as camadas da arquitetura de grade (Figura 2.2), a parte mais estreita da ampulheta consiste dos protocolos de “Resource” e “Connectivity”, que facilitam o compartilhamento de recursos individuais. Os protocolos nessas camadas são projetados de forma que possam ser implementados sobre uma grande faixa de tipos de recursos, na camada “Fabric”, e que também possam ser usados para construir uma grande faixa de serviços globais e comportamentos específicos de aplicações na camada “Collective”.

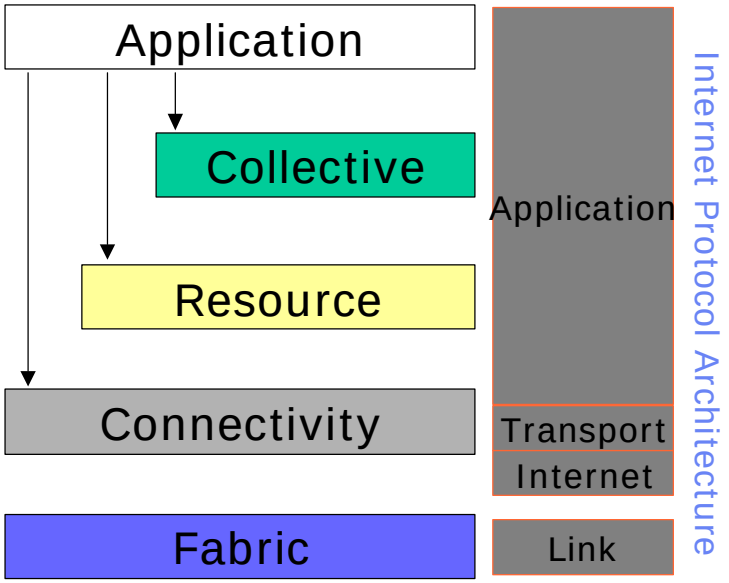


Figura 2.2 - Relacionamento da arquitetura em camadas de grade com protocolo Internet.

Fonte: Foster et al. (2001)

Atualmente não existe um padrão internacional para tecnologia de computação em grade, mas o “Globus toolkit” tem se tornado um padrão “de facto”. Ele foi desenvolvido dentro do escopo do “The Globus Project” que visa desenvolver tecnologia para a criação de grades computacionais, que são ambientes que permitem a integração de softwares com instrumentos, displays, recursos computacionais e de informação que são gerenciados por diversas organizações espalhadas em várias localizações.

Os principais componentes do “Globus toolkit” incluem “Grid Security Infrastructure” (GSI), que fornece serviços de autenticação baseada em chave pública e autorização

local; serviços de gerenciamento de recursos, que fornece uma linguagem para especificação de requisitos de aplicação, mecanismos para reserva imediata e posterior dos recursos da grade e gerenciamento remoto de tarefas; e serviços de informação que recupera e distribui informações sobre os recursos da grade. Os serviços de grade de dados complementam e ampliam esses componentes: serviço de transferência GridFTP e serviço de gerenciamento de réplica.

2.1.1 Grid Security Infrastructure (GSI)

Para autenticação e comunicação segura entre elementos de uma grade computacional, o GSI (FOSTER et al., 1998) possui serviços para autenticação mútua e assinatura única. Ele é baseado em criptografia de chave pública, certificados X.509 e protocolo de comunicação “Secure Socket Layer” (SSL).

A autenticação, com GSI, envolve o uso de certificados (Figura 2.3) codificados no formato de certificados X.509, que contém:

- Identificação do assunto – identifica a pessoa ou objeto que o certificado representa;
- Chave pública pertencente ao assunto;
- Identidade da Autoridade Certificadora (CA) – responsável por assinar o certificado de forma a certificar que tanto a chave pública quanto a identidade pertencem ao assunto;
- Assinatura digital da CA.

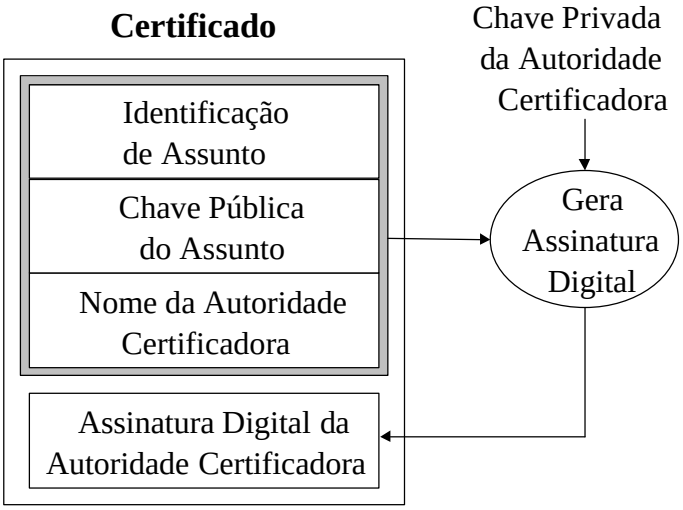


Figura 2.3 – Processo de geração de um certificado.

Uma autenticação mútua ocorre quando dois parceiros possuem certificados e ambos confiam na CA que assinou o certificado do outro. Desta forma, prova-se ao outro que ele é quem ele diz que é. Na prática os parceiros devem ter cópia do certificado gerado pela CA (que contém a chave pública da CA). Para isto, o GSI utiliza o protocolo Secure Sockets Layer (SSL). Um exemplo de certificado é apresentado no Apêndice A.

Durante o processo de autenticação mútua, a primeira pessoa (A) estabelece uma conexão com a segunda pessoa (B). Para iniciar o processo da autenticação, A dá a B seu certificado. O certificado informa a B quem A está reivindicando ser (identidade), que é a chave pública de A, e qual CA está sendo usada para certificar o certificado. B irá primeiramente assegurar que o certificado é válido, verificando a assinatura digital da CA, para se certificar de que a CA assinou realmente o certificado e de que o certificado não foi alterado (é neste ponto que B deve confiar na CA que assinou o certificado de A). A conexão é estabelecida quando existe a autenticação mútua. A partir deste ponto o GSI não atua mais, de forma que a comunicação possa ocorrer sem o “overhead” constante de codificação e decodificação.

A chave privada é normalmente armazenada em um arquivo, em uma área local do computador. Para evitar que ela seja roubada, este arquivo é cifrado utilizando uma senha (chamada de “pass phrase”). Desta forma, para utilizar o GSI é necessário decifrar este arquivo.

O GSI possui uma capacidade de delegação, isto é, uma forma de reduzir o número de vezes que o usuário deve digitar a sua “pass phrase”. Se a computação de grade necessita de vários recursos de grade (cada um necessitando autenticação mútua), ou se existe a necessidade de se ter agentes (locais ou remotos) requisitando serviços representando o usuário, a necessidade de digitar várias vezes a “pass phrase” pode ser evitada com a criação de um “proxy”.

Um “proxy” consiste de um novo certificado (com uma nova chave pública) e uma nova chave privada. O novo certificado contém a identidade do dono, com uma pequena modificação para indicar que é um “proxy”. O novo certificado é assinado pelo dono ao invés da CA. O certificado também inclui uma definição de tempo de duração do “proxy”.

Quando “proxies” são utilizadas, o processo de autenticação mútua é ligeiramente diferente. O parceiro remoto recebe não apenas o certificado do “proxy”, mas também o certificado do dono. Durante a autenticação mútua, o dono da chave pública (obtida do certificado) é utilizado para validar a assinatura no certificado do “proxy”. A chave pública da CA é então utilizada para validar a assinatura do dono do certificado. Isto estabelece um elo de confiança da CA para o “proxy” através do dono.

2.1.2 Serviços de informação

Em computação distribuída de larga escala, geralmente é necessário uma cuidadosa seleção e configuração dos computadores, redes, protocolos de aplicação e algoritmos para atingir os objetivos desejados. Para isto é necessário armazenar informações sobre o hardware, o software e o “status” do sistema. Serviços de informações são partes vitais de qualquer software de infra-estrutura de grade, pois fornecem mecanismos para a descoberta e o monitoramento, permitindo o planejamento e a adaptação do comportamento da aplicação. O monitoramento e a descoberta são atividades que são relacionadas e envolvem os mesmos tipos de informação. No entanto, a descoberta preocupa-se com as características de uma entidade em um dado instante de tempo enquanto que o monitoramento está interessado em como as características da entidade variam com o tempo (CZAJKOWSKI et al., 2001).

Em uma arquitetura de serviços de informação de grade, existem duas entidades fundamentais: fornecedores de informação e serviços de diretórios agregados. A função de um fornecedor de informação é fornecer informação sobre o número de nós, quantidade de memória, S.O. (sistema operacional), carga média, etc.. Serviços de diretórios agregados fornecem uma visão específica dos recursos, serviços, etc. de uma organização virtual.

O fornecedor de informação utiliza o GRid Information Protocol (GRIP) para acessar informações sobre entidades e o Grid Registration Protocol (GRRP) para informar ao serviço de diretórios agregados a disponibilidade de informações.

Os serviços de diretórios agregados utilizam o GRRP e o GRIP para obter informação (de um conjunto de fornecedores de informação) sobre um conjunto de entidades e então responde a perguntas relativas a essas entidades.

Para implementar os serviços de informação, está disponível no “Globus toolkit” o “Monitoring and Discovery Service” (MDS) que utiliza os protocolos GRIP e GRRP, que por sua vez utiliza o “Lightweight Directory Access Protocol” (LDAP), que é um protocolo padrão para construção de diretórios.

Para entender o funcionamento do serviço de informação é necessário compreender o significado de diretório e serviço de diretório (Von Laszewski e Foster, 2002). Diretório é utilizado para armazenar e recuperar informação, e tem como características o fato de ser projetado para efetuar mais leitura que escrita, oferecer uma visão estática dos dados e efetuar atualizações bastante simples (sem transação). O serviço de diretório fornece um diretório que pode ser acessado via protocolo de rede. Muitas vezes os serviços de diretórios incluem mecanismos para replicação e distribuição de dados. Um exemplo de serviço de diretório é o Domain Name System (DNS).

O MDS possui integração com o GSI para garantir segurança nas transações citadas anteriormente e utiliza Grid Resource Information Service (GRIS) para fornecer informações e o Grid Index Information Service (GIIS) para a construção de diretórios agregados.

O GRIS é um sistema fornecedor de informações configurável que é implementado como um servidor OpenLDAP. Atualmente estão implementadas informações estáticas (versão do S.O., tipo de CPU, número de processadores, etc.) e dinâmicas (carga média, entradas da fila, etc.) das máquinas, do sistema de armazenamento (espaço de disco disponível, espaço de disco total, etc.) e da rede (largura de banda e latência).

O GIIS é um sistema para construção de diretórios agregados. Ele aceita informações do GRIS e envia informações a um diretório agregado de nível mais alto, utilizado pela organização virtual.

A Figura 2.4 apresenta a implementação de um sistema de informação utilizando o "Globus toolkit", onde encontramos dois níveis de diretórios agregados e fornecedores de informação.

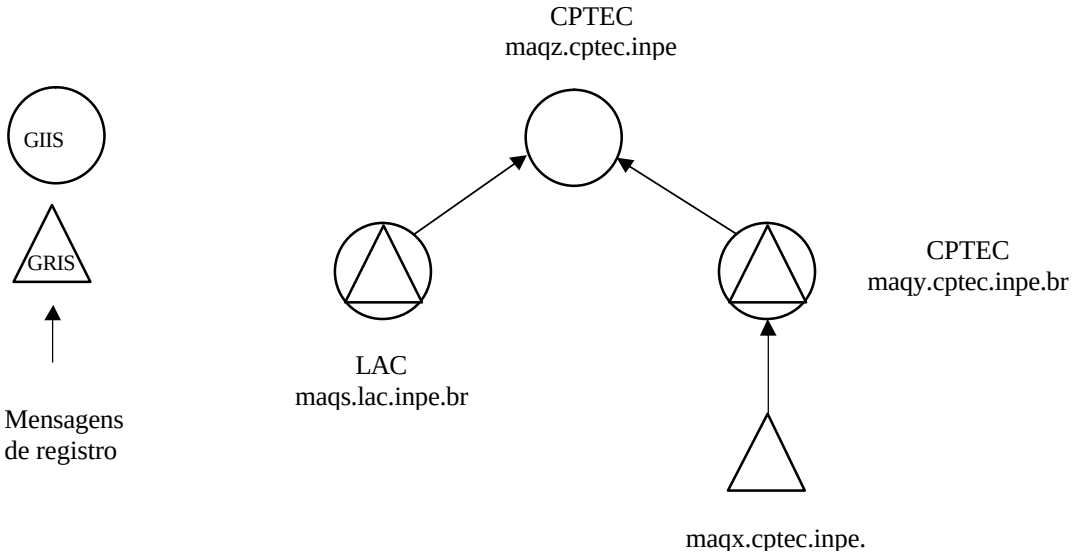


Figura 2.4 – Implementação de um sistema de informação utilizando o "Globus toolkit"
Fonte: Adaptada de Fitzgerald et al. (1997)

2.1.3 Serviços de gerenciamento de recursos

Segundo Czajkowski et al. (1998), o gerenciamento de recursos pode ser dividido em sistemas de gerenciamento local ("Networked Batch Queuing Systems") e sistemas de gerenciamento global ("Wide-Area Scheduling Systems"). Os sistemas de

gerenciamento locais têm como função manipular tarefas submetidas pelos usuários, alocando recursos dos computadores. No caso dos sistemas de gerenciamento global, o gerenciamento de recursos dependerá da forma que foi projetado, podendo suportar classes específicas de aplicações, classes gerais de programas paralelos, computação com “high-throughput”, etc..

A arquitetura do Globus distribui o problema de gerenciamento de recursos em negociadores de recursos (“broker”), co-alocadores de recursos (“co-allocator”) e gerenciadores de recurso (GRAM), conforme ilustrado na Figura 2.5. Nesta arquitetura é utilizada uma linguagem extensível de especificação de recursos (RSL) para enviar as requisições de recursos para os seus componentes: da aplicação para o negociador de recursos, depois para os co-alocadores de recursos e finalmente para os gerenciadores de recursos.

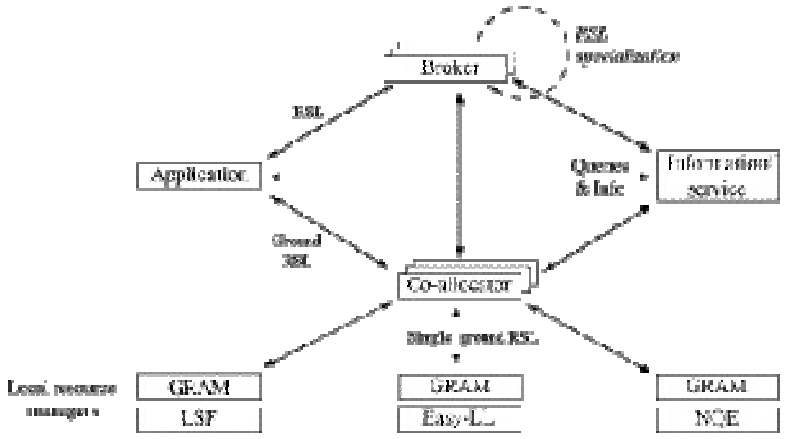


Figura 2.5 – Arquitetura de gerenciamento de recursos do Globus

Fonte: Czajkowski et al. (1998)

Os negociadores de recursos são responsáveis por receber uma especificação RSL de alto nível e transformá-la em especificações mais concretas através de um processo chamado especialização. Nesta definição estão incluídos os escalonadores de aplicação (que encapsulam informação sobre os tipos de recursos necessários para atingir um requisito particular de performance), localizadores de recursos (que mantém informação sobre disponibilidade de vários tipos de recursos) e “traders” (que criam mercados para os recursos).

Em cada caso, o negociador, para obter maior detalhamento, utiliza informação mantida localmente, obtida de um MDS, ou contida na especificação, mapeando-a em uma nova especificação que conterá mais detalhes. As requisições podem ser passadas a diversos negociadores até que a especificação seja especializada, a ponto de identificar um gerenciador de recursos específico. Esta nova especificação pode ser enviada ao GRAM apropriado ou, em caso de múltiplas requisições, a um co-alocador de recursos.

Por meio da ação de um ou mais negociadores de recursos, os requisitos de uma aplicação são refinados em uma expressão denominada “ground” RSL. Como dito anteriormente, se esta expressão consistir de uma requisição de recurso simples, ela é submetida diretamente ao gerenciador que controla aquele recurso. Se a aplicação necessitar que vários recursos sejam disponibilizados simultaneamente, o negociador de recursos produz uma requisição múltipla que é manipulada pelos co-alocadores de recursos.

A função dos co-alocadores de recursos é dividir uma requisição nos componentes que a constituem, submetendo cada componente ao gerenciador de recursos local apropriado e em seguida fornecendo meios para a manipulação do conjunto de recursos resultantes como um todo.

O gerenciador de recursos local é a camada mais baixa da arquitetura de gerenciamento de recursos, implementada com o nome de “Globus Resource Allocation Manager” (GRAM). Ele fornece os componentes locais para gerenciamento de recursos e possui as seguintes atribuições:

- Processar especificações RSL que representam requisições de recursos;
- Habilitar o monitoramento e gerenciamento remoto de tarefas geradas por requisições de recursos;
- Atualizar periodicamente o MDS com informação referente à disponibilidade e capacidade atual do recurso que gerencia.

Cada GRAM é responsável por um conjunto de recursos operando sob a mesma política específica de alocação de cada domínio administrativo, geralmente implementado por um sistema local de gerenciamento de recursos, como Portable Batch System (PBS), Sun Grid Engine (SGE), Load Sharing Facility (LSF), Condor, Network Queuing Environment (NQE), LoadLeveler ou um simples “daemon fork”. Para submissão, monitoramento e gerenciamento remoto seguro de tarefas, o GRAM utiliza-se de API. Essas API utilizam internamente também a linguagem RSL.

2.1.4 Serviços de grade de dados

O volume de dados gerado por diferentes áreas do conhecimento está atualmente na ordem de terabytes e em breve em petabytes. Esses dados estão muitas vezes distribuídos em instituições geograficamente espalhadas. Segundo Chervenak et al. (2002), aplicações distribuídas de alto desempenho, que fazem uso intensivo de dados, necessitam de dois serviços fundamentais: transferência de dados segura, confiável e eficiente; e habilidade de registrar, localizar e gerenciar múltiplas cópias de conjunto de dados. Esses serviços podem ser utilizados para construir um conjunto de capacidades de alto nível, incluindo criação de cópias confiáveis de dados em uma nova localização, seleção das melhores réplicas para operações de transferência de dados baseadas em performance e criação de novas réplicas em resposta à demanda das aplicações. Esses serviços são o GridFTP e o gerenciamento de réplica.

O protocolo de transferência de dados GridFTP é uma extensão do protocolo FTP que permite a inclusão de um superconjunto de características disponíveis por vários sistemas de armazenamento de grade. Como funcionalidades adicionais podemos citar: suporte para GSI e Kerberos, controle da transferência de dados por terceiros, transferência de dados paralela, transferência parcial de arquivos, negociação automática do tamanho do buffer/janela TCP e suporte para transferência de dados confiável e reinicializável.

O gerenciamento de réplica é responsável pelo gerenciamento da replicação de cópias completas e parciais de conjunto de dados, definidas como coleção de arquivos. Serviços de gerenciamento de réplicas incluem: criação de novas cópias de uma coleção

de arquivos completa ou parcial, registro destas novas cópias no catálogo de réplicas e permissão de consulta ao catálogo, a usuários e aplicações, para descobrir todas as cópias existentes de um arquivo particular ou coleção de arquivos.

2.2 CIGRI/OAR

Segundo Capit (2004), o CIGRI (CIMENT GRID) foi desenvolvido para atender à comunidade "Calcul Intensif, Modélisation, Expérimentation Numérique et Technologique (CIMENT") visando criar uma grade na região de Grenoble (Figura 2.6). Trata-se de uma grade leve, onde alguns problemas inerentes a uma grade não são tratados, como por exemplo, a autenticação centralizada de usuários. Ele é composto de um servidor que se comunica com todos os escalonadores de tarefa, tendo o objetivo de ser o menos intrusivo possível sobre os "clusters". O CIGRI trabalha de uma forma diferente do Globus, pois não instala nenhum programa específico nos "clusters", apenas utiliza os utilitários de sistema clássicos. Para controlar a execução de tarefas, o CIGRI utiliza o escalonador de tarefas OAR (CAPIT et al., 2005) como base, em cada organização.

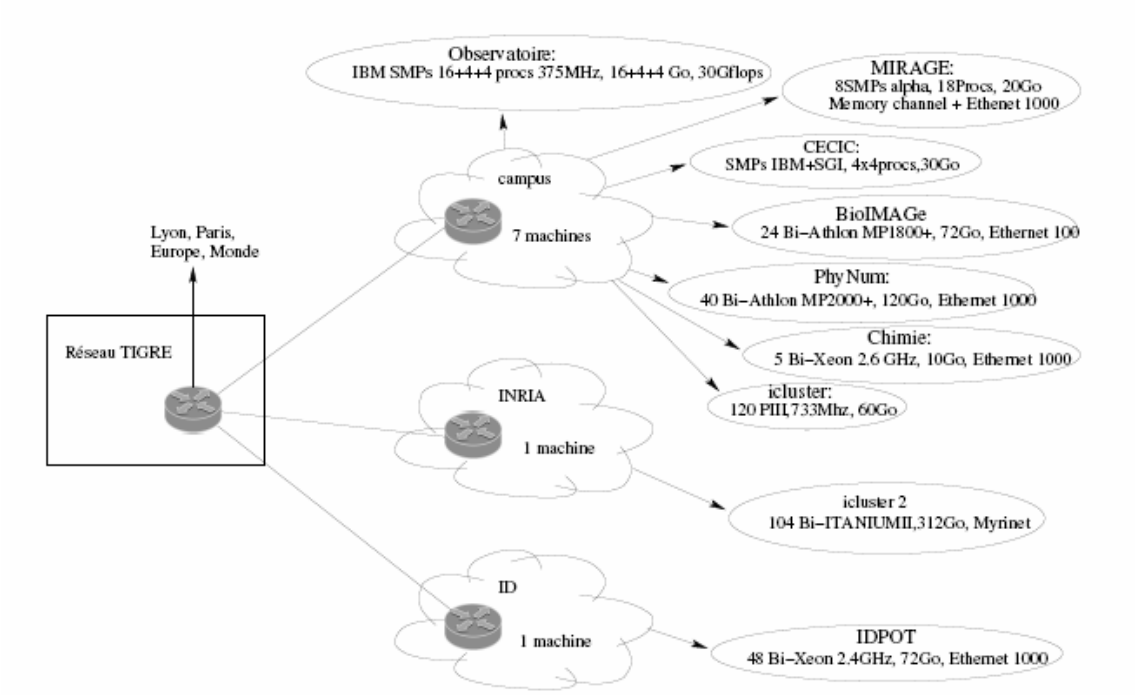


Figura 2.6- Grade do projeto CIMENT

Fonte: Capit (2004)

O OAR foi desenvolvido com o objetivo de implementar funcionalidades não existentes em escalonadores como PBS (ALTAIR GRID TECHNOLOGIES, 2006), Condor (UNIVERSITY OF WISCONSIN-MADISON, 2006) e SGE (SUN MICROSYSTEMS, 2002), de modo a permitir um gerenciamento simplificado das tarefas. Ele possui uma arquitetura (Figura 2.7) onde a parte servidora é dividida em duas partes: um agente banco de dados (Mysql) e uma parte executiva. O agente banco de dados é utilizado para fazer a correspondência dos recursos, armazenar e explorar informações de "log" e contabilização. A parte executiva é composta de diversos módulos, incluindo um para lançamento e controle da execução das tarefas e outro para escalonamento das tarefas.

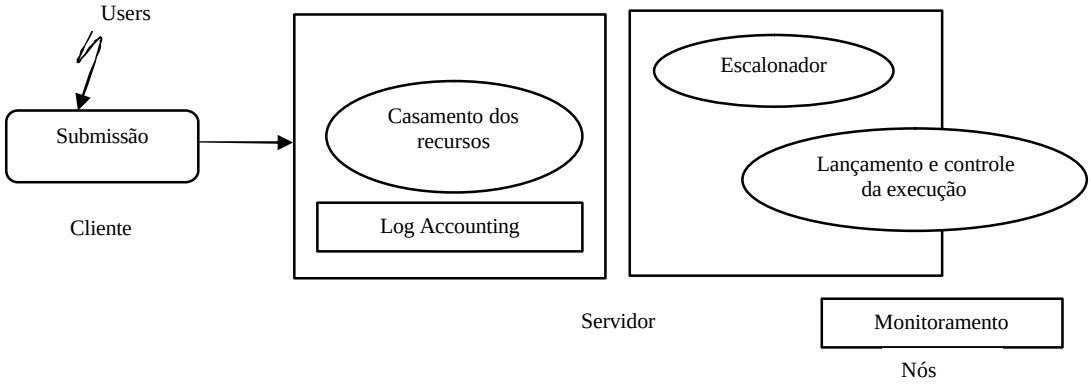


Figura 2.7 – arquitetura do OAR

O módulo central do OAR visa garantir que todas as funções importantes (escalonamento, execução e monitoramento) sejam executadas quando requisitadas (escalonamento na submissão de tarefas) e regularmente (monitoramento).

O lançamento, a apresentação e o monitoramento em escalonadores de tarefa são normalmente feitos por processos específicos executando nos nós ("daemons"). No OAR, ele é feito pelo Taktuk (MARTIN e RICHARD, 2003), que utiliza um algoritmo "work-stealing" dinâmico para distribuir trabalho e evitar desbalanceamento de carga entre os nós e escalar para milhares de nós.

O algoritmo "work-stealing" pode ser resumido da seguinte maneira: a cada etapa, cada processador vazio emite um pedido a um outro processador aleatoriamente escolhido. Cada processador Q não vazio que recebe ao menos um tal pedido seleciona um deles.

Agora cada processador vazio P cujo pedido é aceito por um processador, Q, "rouba" f(l) tarefas de Q, onde l denota a carga de Q.

A política de utilização dos “clusters” da grade do CIGRI/OAR prevê que os usuários externos ao “cluster” podem utilizar o poder computacional dos mesmos. No entanto, a prioridade de uso sempre é dos usuários internos. Desta forma, as tarefas de usuários externos são canceladas toda vez que uma tarefa de um usuário interno necessita do “cluster”. Para isto as aplicações devem ser tolerantes a falhas, isto é, as aplicações devem ser projetadas para serem re-submetidas ao “cluster”.

O OAR e o CIGRI foram concebidos independentemente, mas como foram projetados por módulos que interagem por meio de uma base de dados, as seguintes informações são compartilhadas:

- Estado de todos os nós de todos os “clusters”;
- Estado dos trabalhos submetidos (para acompanhamento);
- Eventos (erros, re-submissões, ...);
- “Log” de tudo que se passa (permite registrar eventos para facilitar o diagnóstico de uma pane; permite igualmente ordenar as estatísticas de utilização);
- Informações sobre os usuários

O CIGRI é composto de 5 (cinco módulos), conforme apresentado na Figura 2.8:

- **Updator:** permite manter atualizada a base de dados (estado dos nós) e conhecer o estado das tarefas;
- **Scheduler:** determina as tarefas a serem submetidas;
- **Runer:** lança as tarefas nos “clusters” como um usuário normal;
- **Nikita:** módulo de limpeza (destrói as tarefas);

- **Colombo:** gestão de eventos (erros, tarefas destruídas, ...), tomada de decisões (retirada de uma aplicação, de um “cluster”, ...).

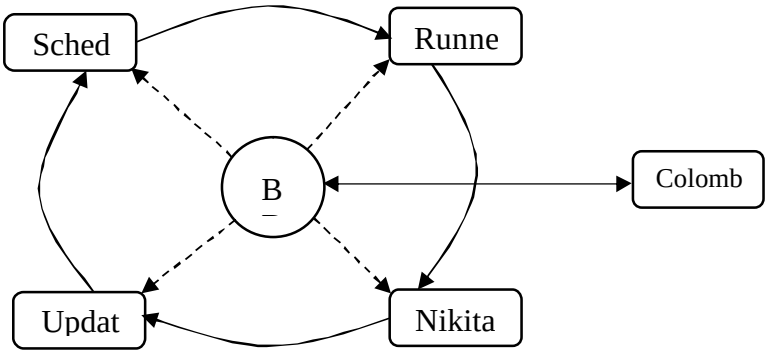


Figura 2.8 – Organização dos módulos internos do CIGRI
Fonte: Capit (2004)

Quando da submissão de uma aplicação na grade, o usuário deve descrever o conjunto de tarefas a serem executadas num arquivo chamado “Job Descripition Language” (JDL). O arquivo JDL é composto de um campo que possui parâmetros comuns a todos os domínios administrativos e campos que são relativos aos “clusters” onde a aplicação será executada (CAPIT, 2004). Na Figura 2.9 é apresentada a interface WEB para submissão e acompanhamento das tarefas no CIGRI.

Figura 2.9 - Portal CIGRI - interface de submissão e acompanhamento de tarefas
Fonte: Capit (2004)

2.3 OurGrid

O OurGrid (ANDRADE et al., 2003) foi desenvolvido na Universidade Federal de Campina Grande tendo como base o sistema MyGrid (PARANHOS et al, 2003). O MyGrid permite a execução de tarefas do tipo “bag-of-tasks” (BOT) em todas as máquinas acessíveis ao usuário. Aplicações BOT possuem a característica de serem totalmente independentes (aplicações paralelas que não necessitam de comunicação durante a sua execução). O OurGrid foi projetado sobre um modelo de compartilhamento de recursos, voltado para aplicações BOT. A idéia é que o OurGrid funcione como uma rede de recursos par-a-par (“peer-to-peer”) de uma comunidade de usuários de grade (ANDRADE et al., 2003). Para comunicar-se com a comunidade e ganhar acesso aos recursos, todos os pares utilizam o protocolo OurGrid de compartilhamento de recursos. Este possui basicamente três participantes:

- **Clientes:** programas que gerenciam acesso aos recursos da grade e executam aplicações na mesma;
- **Consumidores:** partes de um par que recebem requisições de um usuário cliente para localizar recursos;
- **Fornecedores:** são partes de um par, que gerenciam os recursos compartilhados e os provêem aos consumidores.

O OurGrid é formado por três componentes: MyGrid Broker (CIRNE at al., 2003), responsável por ser a interface do usuário com a grade; OurGrid Peer (ANDRADE et al., 2003), responsável por agrupar os recursos da grade para serem utilizados pelas instâncias MyGrid; e Swan, uma solução de “Sandboxing”, baseado no Xen (BARHAM, 2003), que garante o acesso aos recursos de maneira segura. Esses componentes do “toolkit” OurGrid e suas interações estão representados na Figura 2.10.

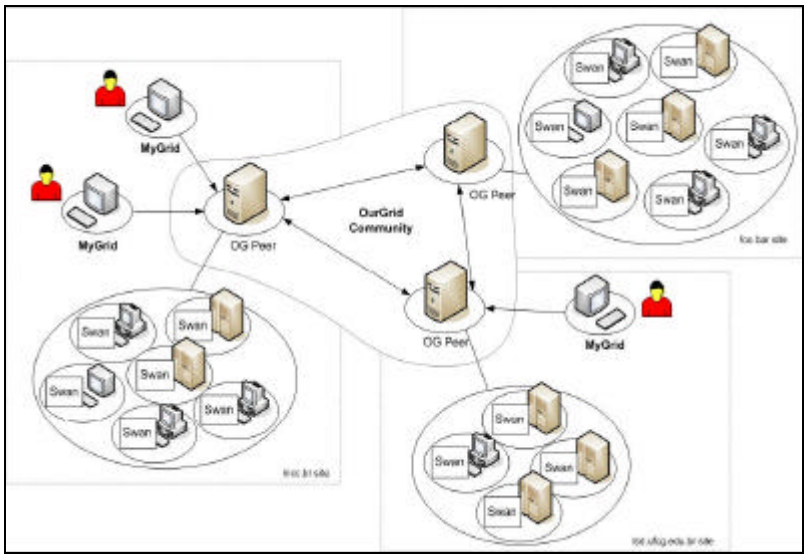


Figura 2.10 - Arquitetura OurGrid

Fonte: Andrade et al.(2003)

O MyGrid broker é responsável por prover ao usuário uma abstração de alto nível da grade. Para executar uma aplicação utilizando o OurGrid, o usuário deve descrever sua aplicação e o conjunto de recursos a que tem acesso, através de ponto de entrada na forma de um portal (Figura 2.11), para o OurGrid Peer. Esse conjunto de recursos pode ser apenas a indicação de um OurGrid Peer que tem a função de obter recursos para o usuário.

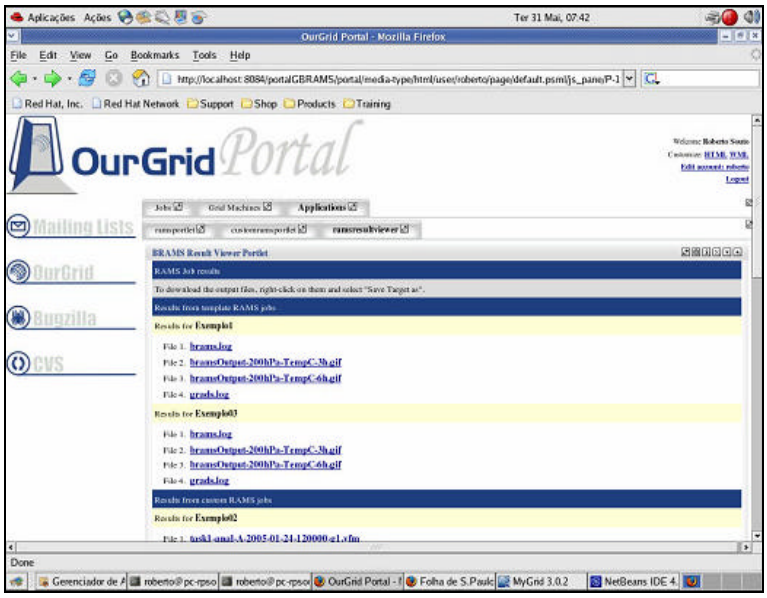


Figura 2.11 - Portal OurGrid

Uma aplicação é basicamente um conjunto de tarefas, seus arquivos de entrada, arquivos de saída e seus requisitos (S.O. necessário, mínimo de memória, arquitetura do processador, etc.). O usuário submete sua aplicação para execução no Grid através do MyGrid Broker. O componente interno do MyGrid Broker que recebe a submissão é o Scheduler. Por sua vez, o Scheduler requisita aos provedores de Grid Machines (GuM) recursos para executar a aplicação submetida pelo usuário. Esses provedores podem responder com recursos locais ou recursos obtidos na rede de favores. Para o Scheduler receber uma resposta dos provedores é necessário encontrar uma GuM que preencha os requisitos determinados na descrição da aplicação.

Segundo Cirne e Santos-Neto (2005), uma vez descoberto um recurso que possui atributos compatíveis com os requisitos da aplicação, o recurso é alocado e repassado para o Scheduler que o solicitou (somente possível se o recurso estiver disponível). Caso o recurso tenha sido descoberto através da rede de favores, o recurso pode ser tomado de volta (preemptado) pelo par que o forneceu, seguindo a dinâmica da rede de favores. A preempção é um evento natural e previsto pela arquitetura do OurGrid, uma vez que os recursos só são cedidos caso estejam ociosos. Desta forma, uma solicitação local no domínio administrativo ao qual o recurso pertence pode ocasionar a preempção. A alocação do recurso é feita no nível do MyGrid Broker, não significando que o recurso estará dedicado exclusivamente ao MyGrid Broker. Desta forma, outras aplicações que não usam a infra-estrutura do OurGrid podem estar executando concorrentemente com a aplicação submetida pelo usuário.

Na arquitetura OurGrid existem basicamente dois níveis de autenticação. Esses níveis dependem de como o usuário obteve o recurso. Primeiramente, o usuário pode ter acesso direto a alguns recursos (através das Grid Machines – GuM's - em sua rede local); neste caso, o usuário usa o esquema de autenticação tradicional; em geral, isso implica na utilização da infra-estrutura de autenticação do sistema operacional do recurso, ou seja, nome de usuário (login) e uma senha. Contudo, além das GuM's a que o usuário tem acesso direto, OurGrid permite (e promove) a obtenção de acesso a GuM's de outros domínios administrativos; isso ocorre através de um Our-Grid Peer local ao domínio administrativo do usuário.

A parte referente ao escalonamento da aplicação é mantida fora do escopo do OurGrid de forma que usuário possa escolher, entre os algoritmos de escalonamento existentes (PARANHOS et al, 2003; CASANOVA at al, 2000), o que melhor otimize sua aplicação de acordo com seu conhecimento sobre as características da aplicação.

Atualmente a solução de escalonamento utilizada pelo OurGrid é a "Workqueue" (WQ) original. Neste algoritmo, as tarefas são atribuídas às máquinas na forma "First-Come-First-Serve" (FCFS), sem se preocupar com as características de desempenho das mesmas, pois este não necessita de nenhum tipo de informação para o escalonamento de tarefas. Na forma FCFS, as tarefas são escolhidas por ordem de chegada e enviadas aos processadores. Os resultados são enviados de volta e o escalonador que atribui nova tarefa ao processador. O algoritmo Workqueue with Replication" (WQR) basicamente adiciona replicação de tarefas ao algoritmo Workqueue original. Com esta abordagem de replicação, processadores que estão livres executam réplicas das tarefas que estão sendo executadas (PARANHOS et al, 2003).

Uma vez que o foco da solução OurGrid está nas aplicações "Bag-of-Tasks", não faz parte do escopo da solução OurGrid prover mecanismos de comunicação para aplicações fortemente acopladas. No entanto, é possível usar a infra-estrutura OurGrid para executar aplicações deste tipo, desde que a execução seja interna a um domínio administrativo. Por exemplo, uma aplicação que usa MPI, quando descrita pelo usuário, pode ter especificado em seus requisitos que necessita de uma GuM (Grid Machine), que na verdade é o "front-end" de uma coleção de vários processadores ("cluster"). Essa demanda será atendida se existir uma GuM, não alocada, que possua um atributo compatível com o requisito especificado pela aplicação. Portanto, apesar de não ter uma arquitetura que provê comunicação entre as tarefas que estão sendo executadas nas GuMs, a solução OurGrid provê meios de agregar ao Grid GuM's que permitem a execução de aplicações fortemente acopladas (CIRNE ; SANTOS-NETO, 2005).

2.4 Comparação das plataformas de grade

Por ser um sistema modular, o Globus é a plataforma de grade mais versátil. Possui módulos projetados para coletar, armazenar e recuperar informações estáticas (MDS), segurança através de autenticação (GSI), transferência segura de dados (GridFTP) e para alocação de recursos remotos (GRAM) da grade. Esta modularidade permite a integração com outros sistemas. Desta forma, gerenciadores de recursos locais (SGE, PBS, etc...) podem ser integrados ao módulo GRAM e informações dinâmicas da grade também podem ser conseguidas através de sistemas com "Network Weather Service" (NWS). Dos sistemas analisados, é o que possui maior aceitação sendo empregado nas principais grades de produção (GriPhyN, 2006; NGS, 2006; TeraGrid, 2006; DataGrid, 2006; IVDGL, 2006) devido a sua versatilidade e robustez. No entanto, é necessário desenvolvimento adicional para a implementação de funcionalidades não existentes.

O OurGrid é o sistema mais fácil de implementar, porém o custo desta facilidade é a falta de informações sobre os recursos computacionais. Esta falta de informações se reflete no escalonamento, pois a atribuição das tarefas aos recursos computacionais não considera a carga do sistema. O mecanismo de escalonamento adotado considera a replicação de tarefas para acelerar o seu tempo de execução de tarefas. Foi projetado inicialmente para executar tarefas independentes (aplicações "Bag-of-Task" – BOT) em grades computacionais composta de "desktops".

O CIGRI funciona juntamente com o OAR, que tem a responsabilidade do escalonamento local. Uma das características do CIGRI/OAR é permitir que as tarefas da grade sejam canceladas quando um usuário local necessita dos recursos computacionais, o que não é desejável na maioria das vezes.

A Tabela 2.1 apresenta o quadro comparativo das funcionalidades das três plataformas de grade propostas para análise dentro do projeto G-BRAMS

Tabela 2.1 - Análise de plataformas de grade

Plataforma de grade	Globus	CIGRI/OAR	OurGrid
Facilidade de instalação		**X**	**X**
Informação da grade	**X**	**X**	
Escalonador nativo		**X**	**X**
Cancelamento arbitrário de tarefas		**X**	
Desenvolvimento adicional necessário	**X**		
Mecanismos de segurança	**GSI**	**SSH**	**SSH**
Migração de tarefas		**X**	

CAPÍTULO 3

ESCALONAMENTO EM GRADES COMPUTACIONAIS

Outro objetivo desta tese é analisar e propor estratégias para o escalonamento de aplicações paralelas em uma grade de pesquisa composta de arquiteturas paralelas. As atuais arquiteturas paralelas são da classe “Multiple Instruction Multiple Data” (MIMD). Nessas arquiteturas, cada processador executa seu próprio programa sobre seus próprios dados de forma assíncrona (TANENBAUM, 2001). De forma análoga, grades computacionais podem ser classificadas como sendo da classe MIMD.

Existem dois modelos de programação paralela que podem ser utilizados em arquiteturas paralelas da classe MIMD: paralelismo de função (ou de controle) e paralelismo de dados. No modelo paralelismo de função (ou de controle), um programa paralelo é composto por um conjunto de tarefas diferentes, que implementam funções distintas, executadas concorrentemente nos diversos processadores. No modelo paralelismo de dados, um programa paralelo caracteriza-se por um mesmo processo executando em todos processadores sobre diferentes conjuntos de dados (PLASTINO et al., 1998).

Normalmente, a arquitetura de computadores utilizada para processamento de modelos meteorológicos é paralela. O modelo meteorológico BRAMS, que possui um modelo de programação paralela do tipo de paralelismo de dados, utiliza esta arquitetura de computadores. Nela, cada máquina executa o código do BRAMS sobre dados distintos de forma assíncrona. Segundo Foster (1995), este modelo de programação paralela, quando associado a arquiteturas paralelas, é conhecido como modelo de programação “Single Program, Multiple Data” (SPMD), uma vez que cada tarefa executa o mesmo programa, mas utilizando diferentes dados.

O tempo de execução de um programa paralelo é definido pelo intervalo entre o início do processamento pelo primeiro processador e o encerramento do processamento pelo

último. Para minimizar o tempo de execução paralelo de um programa paralelo e maximizar a utilização dos recursos computacionais disponíveis é necessário distribuir adequadamente as tarefas aos processadores. O processo de distribuição de tarefas entre os processadores é denominado de balanceamento de carga, que é realizado através do escalonamento de tarefas (PLASTINO et al., 1998).

O desenvolvimento de técnicas eficientes para distribuição de tarefas (escalonamento de tarefas) em ambientes computacionais paralelos e distribuídos sempre foi um grande desafio. O escalonamento de tarefas visa minimizar o tempo de processamento da aplicação e/ou maximizar o uso dos recursos computacionais, sendo utilizados mecanismos ou algoritmos para a realização do escalonamento. Um escalonamento pode ser executado em duas fases:

- Decisão, pelo escalonador, de quais aplicações terão acesso aos recursos computacionais e a localização e quantidade de recursos que serão disponibilizados à aplicação;

- Mapeamento das tarefas que compõem a aplicação aos recursos computacionais.

Na taxonomia de Casavant e Kuhl (1998), o escalonamento em sistemas distribuídos de propósito geral envolve duas etapas: escalonamento global e local. Na primeira etapa o escalonamento global é responsável por decidir, entre as várias aplicações que são submetidas ao sistema, o conjunto de aplicações que terão acesso aos recursos computacionais e a localização e quantidade de recursos que serão disponibilizados à aplicação. Na segunda etapa, o escalonamento local é responsável pelo mapeamento das tarefas da aplicação aos processadores.

O escalonamento local é geralmente implementado pelo sistema operacional. Os programas em execução em um sistema operacional são denominados processos. Uma forma mais leve de programas em execução é denominada “threads”. A política de escalonamento mais comum é o escalonamento de curto prazo, que aumenta a prioridade de processos bloqueados e reduz a de processos que possuam grande

quantidade de tempo de execução. Esta política pode ser implementada através de um algoritmo round-robin ou “First-Come-First-Served” (FCFS).

No escalonamento global, as políticas de escalonamento podem ser em relação ao tempo (“time sharing”) ou em relação ao espaço dos processadores (“space sharing”). A primeira prevê a utilização de uma fila global compartilhada entre os processadores. Na segunda, os processadores são organizados em conjuntos disjuntos que são atribuídos para as aplicações, na forma de um conjunto por aplicação. Esses conjuntos são chamados de partições, podendo ter políticas fixas, adaptativas ou dinâmicas. Na política fixa as partições são pré-estabelecidos durante a configuração do sistema e inalteráveis durante o tempo de execução. Assim como na política fixa, na política adaptativa a partição utilizada por uma aplicação só é liberada ao término da mesma. A diferença está na definição das partições em função da carga do sistema. As políticas que utilizam partições dinâmicas definem o tamanho das mesmas à medida que as aplicações chegam ao sistema, de acordo com o estado do sistema, podendo o tamanho da partição ser alterado durante o tempo de execução da aplicação.

Segundo Senger (2002), as técnicas e políticas de escalonamento podem ser implementadas internamente à aplicação, permitindo total liberdade de como serão atribuídas as tarefas aos processadores, ou externamente à aplicação, realizada dentro do sistema operacional ou através de software sobre o sistema operacional (ambiente de escalonamento). Os ambientes de escalonamento estabelecem quais aplicações terão acesso aos recursos computacionais e quais recursos serão disponibilizados para determinadas aplicações. A maioria dos ambientes de escalonamento utiliza um escalonador mestre que gerencia múltiplas filas, que são estabelecidas através de regras específicas, cada uma criada para atender aplicações com determinadas características (tempo de execução, quantidade e tipos de recursos). Neste tipo de ambiente a responsabilidade de submissão da aplicação a uma determinada fila é do usuário. Outros ambientes buscam os processadores que são necessários à aplicação pelas características especificadas pelo usuário (quantidade de memória, espaço em disco, etc.).

Esses ambientes de escalonamento recebem o nome de escalonadores de tarefa e gerenciam recursos, que em “clusters” são denominados nós. Entre os mais conhecidos podemos citar: PBS/OpenPBS (ALTAIR GRID TECHNOLOGIES, 2006), LSF (PLATAFORM, 2006), NQS (ALBING, 1993), Loadleveler (IBM, 2003) e Glunix (Ghormley (1998). A arquitetura clássica (Figura 3.1) de um escalonador de tarefas é composta de um cliente que é um módulo para submissão de tarefas que valida requisições, e de um servidor composto de um módulo de escalonamento associado com mecanismos que efetuam a correspondência de recursos e de um módulo de execução que controla a execução das tarefas. Adicionalmente existe um módulo para a contabilização da atividade do sistema e outro para o monitoramento dos recursos (CAPIT et al, 2005).

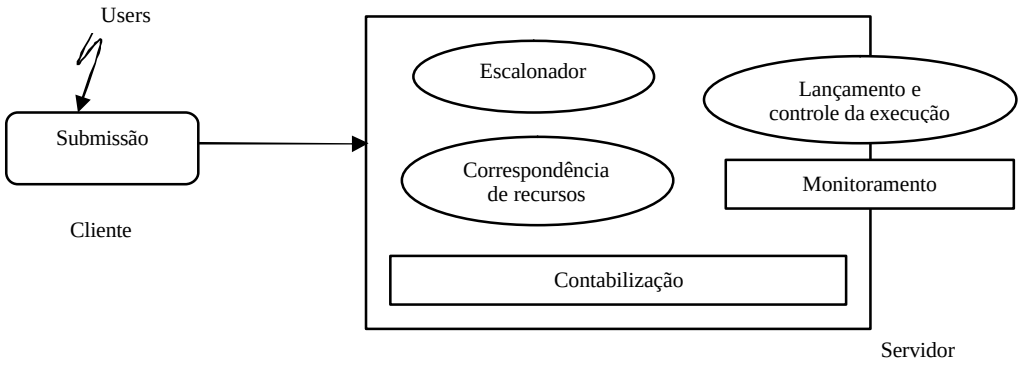


Figura 3.1 - Arquitetura do geral de um escalonador de tarefa

Fonte: Capit et al (2005)

A climatologia do BRAMS para um determinado número de anos não pode ser executada de forma contínua devido a uma restrição desta aplicação. Desta forma, cada ano deve ser executado de forma dependente do ano anterior, utilizando a característica de “checkpoint-restart” do BRAMS. Esta climatologia, em modo “ensemble”, prevê diversas execuções do BRAMS tendo cada uma dados de entrada distintos em uma grade composta de “clusters” de computadores.

A climatologia pode ser entendida como uma aplicação mista, pois internamente aos “clusters” ela é executada de forma paralela e na grade é executada de forma independente com relação a cada membro e dependente com relação a uma dada região/membro.

3.1 Técnicas de escalonamento em grades computacionais

A importância de grades computacionais na resolução de diversos problemas computacionais torna-se evidente a cada dia. A tecnologia atual permite que aplicações possam utilizar recursos computacionais distribuídos geograficamente, que podem ser agregados onde são escassos ou otimizados onde são excedentes (FOSTER et al., 2001). Dois tipos de aplicações podem ser executados em uma grade: independentes e paralelas.

A necessidade crescente de recursos computacionais torna a área de meteorologia uma candidata potencial para o uso desta tecnologia, que deve ser composta de arquiteturas paralelas para execução de modelos meteorológicos.

Os trabalhos pesquisados até o momento sobre escalonamento em grades computacionais mostram que extrair o melhor desempenho de aplicações paralelas complexas ainda constitui um desafio. A seguir são analisadas algumas abordagens para resolver o problema do escalonamento em grades computacionais.

O compartilhamento de recursos computacionais geograficamente distribuídos é possível atualmente devido às redes de alta velocidade. As aplicações paralelas que utilizam esses recursos necessitam de desempenho, mas apesar do desempenho potencial que esses sistemas distribuídos oferecem, atingir o desempenho pretendido pelo usuário pode ser difícil. Segundo Berman et al. (1996), um dos problemas fundamentais que devem ser resolvidos para se obter um bom desempenho é a determinação de um planejamento eficiente. Um planejamento efetivo envolve a integração de informações específicas da aplicação e do sistema, assim como a interação dinâmica entre a aplicação e o sistema. O escalonamento, ao nível da aplicação, é baseado nos seguintes princípios:

- **Informações sobre a aplicação e o sistema são necessárias para um bom planejamento:** os usuários determinam um bom planejamento para suas aplicações, baseado nas suas percepções da capacidade do sistema e no seu conhecimento da estrutura e dos requisitos da aplicação. Os atributos para

desenvolver um planejamento que tenha um desempenho eficiente são: tempo de comunicação e computação, tamanho de memória necessário, velocidade e largura de rede, etc.;

- **Informação dinâmica é necessária para determinar o estado da máquina:** O planejamento necessita conhecer quais máquinas estão disponíveis e quais estão com muita ou pouca carga. Esta carga varia com o tempo e com o uso dos recursos do sistema. Se a escolha da rede ou plataforma computacional estiver disponível, o usuário combinará seu conhecimento de como a aplicação usará o sistema com a carga atual ou que foi prevista nos seus recursos;

- **Um bom planejamento envolve algumas predições do desempenho da aplicação e do sistema:** Previsão fornece a base para a maioria dos escalonamentos. O usuário prevê como sua aplicação será executada no sistema e utiliza esta previsão para escolher o planejamento mais eficiente para obter desempenho. Tais previsões são difíceis de se serem precisas, uma vez que o sistema varia com o tempo devido a disputas por recursos. O desempenho da aplicação será dependente dos dados e da carga do sistema;

- **Todos os recursos podem ser avaliados estritamente em termos do desempenho que eles entregam para a aplicação:** Os usuários definem diferentes critérios para desempenho (velocidade, custo, etc..), mas a decisão sobre quais e quando utilizar os recursos computacionais é baseada na maneira em que eles atuarão quando executarem a aplicação do usuário.

Neste trabalho Berman et al. (1996) apresentam o software “Application-Level Scheduling” (AppLeS), que tem como proposta facilitar e melhorar as atividades de escalonamento do usuário. Com a utilização do AppLeS, obteve-se um ganho de desempenho do programa paralelo Jacobi2D (que utiliza o método Jacobi) em ambiente de grades computacionais. Para determinar o mapeamento das tarefas nos computadores, foi utilizada eliminação gaussiana simples para a resolução do sistema de equações lineares. Foi proposto neste trabalho formular o problema como um problema

de minimização baseado em restrições, onde técnicas de programação linear podem ser utilizadas. No entanto, somente foi validado para aplicação paralela com carga previsível.

Schopf e Berman (1999), propõem uma técnica de escalonamento (escalonamento estocástico) para melhorar o tempo de execução da aplicação que possua variação do desempenho representada por uma distribuição normal. Os autores empregaram a mesma metodologia desenvolvida por Berman et al. (1996) para determinar o mapeamento das tarefas aos computadores. A novidade encontrada neste trabalho é o acréscimo de um fator de ajuste nas equações de mapeamento. A técnica de escalonamento proposta é útil apenas para as aplicações paralelas não determinísticas que possuam distribuição normal dos tempos de iteração.

Os passos que devem ser seguidos para escalonar aplicações foram descritos detalhadamente por Schopf (2002), que apresentou uma arquitetura geral para escalonamento em grades computacionais que possui três etapas: descoberta de recursos, seleção de recursos e execução da aplicação. Neste trabalho são apontadas algumas características que deveriam existir em diversos escalonadores disponíveis na atualidade.

Para Berman et al. (2003), o fato das grades serem compostas de coleções de recursos heterogêneos (com diferentes níveis de desempenho) e possuírem variação do desempenho fornecido (devido à competição dos recursos pelos usuários, falha, troca dos recursos, etc...) pode dificultar a extração do desempenho potencial desta nova tecnologia. Neste trabalho os autores apresentam diversos trabalhos que utilizam as ferramentas do projeto AppLeS, que fornece metodologia e tecnologia para escalonar aplicações em grades computacionais. Com base na aceitação desta tecnologia, este projeto desenvolveu “templates” para facilitar sua utilização. Para aplicações paralelas que adotam o modelo mestre-escravo, foi desenvolvido o “template” “AppLeS Master-Worker Application Template” (APST).

Os trabalhos acima utilizaram o “National Weather Service” (NWS), desenvolvido por Wolski et al. (1999), como previsor do comportamento da grade computacional o que

possui internamente diversas estratégias de previsão. Yang et al. (2003b) apresentam estratégias de previsão melhores que as do NWS e que podem ser incluídas nele para melhorar a previsão. Essas estratégias são baseadas no histórico de dados previamente medidos em intervalos de tempo constantes.

Recentemente Yang et al. (2003a) propuseram o escalonamento conservativo. Este utiliza informações sobre a média e a variância da capacidade da CPU para definir o mapeamento apropriado dos dados aos recursos. Esta idéia provém do fato que um recurso com grande capacidade poderá apresentar uma alta variância no seu desempenho, que como conseqüência executará uma aplicação em um tempo maior que um recurso com menor variância. Este trabalho é uma extensão dos trabalhos de Yang et al. (2003b) na área de previsão e de Schopft e Berman (1999) na área de algoritmos de escalonamento. A eficiência da técnica de escalonamento conservativo foi validada apenas para a classe de aplicações paralelas que são iterativas e com acoplamento fraco.

Dail et al. (2003) propõem uma técnica para aumentar o desempenho de aplicações paralelas em grades computacionais dentro do "Grid Application Development Software Project" – (GrADS) (BERMAN et al., 1996), que atribui ao escalonador a tarefa de descobrir os recursos disponíveis, selecionar um conjunto apropriado de recursos à aplicação e mapear os dados ou tarefas aos recursos selecionados. Diferentemente das propostas de Berman et al. (1996) e Schopf e Berman (1999), onde o problema do escalonamento é resolvido dentro da aplicação, neste trabalho existe a separação entre os requisitos e as características da aplicação em um modelo de desempenho (métrica analítica para um desempenho esperado da aplicação em um dado conjunto de recursos) e o mapeador (diretivas para mapeamento lógico dos dados/tarefas da aplicação nos recursos físicos).

Lee e Schopf (2003) acreditam que não é sempre possível obter modelos de desempenho de aplicações complexas. Desta forma, neste trabalho é proposta uma técnica de predição em tempo real para aplicações paralelas que utilizam métodos de regressão e técnicas de filtragem para derivar o tempo de execução da aplicação. Isto é feito através da descoberta da relação entre variáveis que afetam o tempo de execução

da aplicação e o histórico dos tempos de execução anteriores. No entanto, este tipo de técnica não considera aplicações paralelas com tempo de execução que não são determinísticos.

Uma nova abordagem para a decidir sobre a alocação dos processos, utilizando lógica "fuzzy", é proposta pelo sistema Autopilot (RIBLER et al., 2001). No entanto, outros componentes precisam ser acrescentados à estrutura da grade computacional, que podem dificultar sua utilização.

O SEA (SIRBU ; MARINESCU, 1997) representa a aplicação como grafos, utilizados como entrada para um sistema especialista que avalia quais tarefas estão aptas a serem executadas, abordagem difícil de ser implementada no estudo de caso analisado.

No IOS (BUDENSKE, 1997), os mapeamentos são gerados por algoritmo genético para diferentes configurações de parâmetros do programa antes da execução, não suportando alterações dinâmicas não previstas.

O Ninf (MATSUOKA et al., 1996) é um ambiente para resolução de problemas que utiliza agentes para acessar informações de rede. No entanto seu uso é para otimizar aplicações cliente-servidor.

3.2 Considerações sobre escalonamento em grade

Analisando a taxonomia de algoritmos de balanceamento de carga para aplicações SPMD, proposta por Plastino et al. (1998), a climatologia do BRAMS em grades computacionais pode ser inserida no contexto da política de distribuição: algoritmos dinâmicos e sob demanda. Com respeito ao momento em que as tarefas são distribuídas aos recursos computacionais, os algoritmos dinâmicos permitem a alocação de tarefas aos recursos computacionais ao longo da execução da aplicação. Com respeito à forma em que as tarefas são distribuídas aos recursos computacionais, os algoritmos sob demanda distribuem as tarefas aos recursos computacionais através de um coordenador. Uma tarefa será enviada quando do encerramento da execução de uma das tarefas pelos recursos computacionais.

O grande desafio é escalonar a aplicação neste ambiente, pois os trabalhos pesquisados até o momento sobre escalonamento em grades computacionais mostram que extrair o melhor desempenho de aplicações mistas (paralelas e independentes/dependentes), ainda constitui um desafio.

Os exemplos apresentados mostram que é possível fazer o escalonamento de aplicações conhecendo ou não a situação dos recursos computacionais. A literatura mostrou que é possível obter bom desempenho da aplicação, mesmo quando as informações sobre os recursos computacionais não estão disponíveis ou são difíceis de se conseguir (OurGrid). No entanto, quando as informações sobre os recursos computacionais estão disponíveis, o desempenho se torna efetivo quando existe uma grande quantidade de recursos computacionais.

Das plataformas de grade analisadas, verifica-se que o CIGRI/OAR e o OurGrid já possuem mecanismos de escalonamento embutidos. Isto facilita sua utilização pelo usuário, pois são mais amigáveis. No Globus, esta facilidade não existe e o usuário terá que desenvolver seu próprio mecanismo de escalonamento.

O OurGrid possui a facilidade de uso e não existe tempo adicional para o escalonamento, devido ao fato de não consultar informações da grade para o escalonamento. A desvantagem aparece quando o número de tarefas é maior que o número de recursos computacionais. Como não existem mecanismos para verificar o estado dos recursos computacionais, muitos processos em um mesmo recurso computacional levam o mesmo a uma saturação de CPU e/ou de memória.

Uma das características do CIGRI/OAR é priorizar o uso local, interrompendo a execução de aplicações não locais. Para resolver problemas decorrentes desta característica, os algoritmos utilizados pelo CIGRI/OAR possuem mecanismos para migração de tarefas. As aplicações que são tolerantes a falhas são migradas para outras máquinas que não possuam uso e são executadas novamente. As aplicações tolerantes a falhas (caso do BRAMS) possuem características de reinicialização a partir do ponto que foi interrompida. Este mecanismo permite uma utilização mais eficiente da grade.

O Globus apresenta a desvantagem de não ser uma solução já preparada para uso imediato. Ele requer que sejam desenvolvidos ou adaptados mecanismos para executar o escalonamento da aplicação. Como vantagem temos a modularidade do Globus, que permite a agregação de novos módulos ou módulos já existentes desenvolvidos pela comunidade de usuários do Globus, como módulos para a integração com gerenciadores de tarefa.

CAPÍTULO 4

MODELOS NUMÉRICOS DE PREVISÃO DE TEMPO E CLIMA

Nos dias atuais, a previsão de tempo e clima está fortemente relacionada com a análise dos resultados gerados por modelos numéricos de previsão de tempo, que são programas complexos que representam o movimento e os processos físicos da atmosfera através de equações matemáticas. Esses modelos recebem como parâmetros de entrada dados observacionais, dados derivados de imagens de satélite e dados gerados por modelos de dias anteriores. O processamento desses dados, utilizando-se as equações embutidas dentro dos modelos, gera arquivos de previsão numérica de tempo e clima. Esses arquivos são representados na forma de matrizes, sendo cada matriz relacionada a uma determinada variável física, a um nível atmosférico e a um instante de tempo. Atualmente o CPTEC possui dois modelos em uso para previsão de tempo e clima: Modelo de Circulação Geral Atmosférico (MCGA), que cobre todo o globo, e ETA (η), que é de área limitada, cobrindo a América do Sul e parte dos oceanos adjacentes.

O MCGA em uso no CPTEC é proveniente da versão 1.7 do “Center for Ocean, Land and Atmosphere Studies” (COLA) e denominado versão CPTEC/COLA. O MCGA é usado operacionalmente para previsão de tempo e, com as devidas modificações, para previsão de clima. Este modelo utiliza métodos espectrais de solução das equações de previsão de tempo, que consistem da expansão das variáveis dependentes em uma série de funções. Para previsão de tempo o modelo é executado com as resoluções T62L28 e T126L28, onde T refere-se ao tipo de truncamento espectral utilizado, denominado triangular, nas ondas zonais 62 e 126, e L refere-se ao número de camadas na vertical, neste caso, 28. As resoluções horizontais T62 e T126 equivalem respectivamente, a uma resolução aproximada de 200x200 Km e 100x100 Km próximo à linha do Equador. O MCGA é executado para previsão de sete dias nos horários 00 e 12 UTC (BONATTI, 1996).

O modelo ETA é um modelo de mesoescala de equações primitivas, que utiliza o método de ponto de grade. Neste método, um conjunto de pontos (grade) é introduzido na região de interesse e variáveis dependentes são inicialmente definidas e em seguida computadas nesses pontos. A versão do modelo ETA que é executada operacionalmente no Centro de Previsão de Tempo e Estudos Climáticos (CPTEC) é hidrostático e cobre a maior parte da América do Sul e oceanos adjacentes. A resolução horizontal atual é de 40 kmx40 km e a vertical de 38 camadas. As previsões são fornecidas duas vezes ao dia 00 e 12 UTC. A condição inicial é proveniente da análise do "National Center for Environmental Prediction" (NCEP) e as condições de contorno lateral são provenientes das previsões do MCGA do CPTEC e atualizadas a cada 6 horas. O prazo de integração é de 72 horas (CHOU, 1996).

Outro modelo de área limitada que vem sendo utilizado no CPTEC/INPE é o BRAMS, que será discutido mais detalhadamente a seguir.

4.1 BRAMS

O "Brazilian Regional Atmospheric Modeling System" (BRAMS) é o resultado de um projeto de pesquisa financiado pela Financiadora de Estudos e Projetos (FINEP) na área de computação de alto desempenho em Meteorologia, visando produzir uma versão do "Regional Atmospheric Modeling System" (RAMS) ajustada aos trópicos, ser utilizado em produção pelos centros regionais de meteorologia e ser utilizado em pesquisa por universidades brasileiras. Apesar de portabilidade de software ter sido um dos objetivos, neste projeto visou-se principalmente o uso em "clusters" de computadores. Na área computacional, este projeto visava melhorias no RAMS em: documentação; modularidade do software; robustez; remoção de opções e aspectos obsoletos; criação de novas opções algorítmicas; e desempenho seqüencial e paralelo. Neste projeto estava previsto o desenvolvimento de um sistema de hardware nacional de alto desempenho, conduzido inicialmente pela ELEBRA Sistemas e depois pela Itautec, que em sua concepção final tornou-se o sistema Infocluster: Cluster de computadores bi-processados com tecnologia X86 (BARROS, 1998; MENDES; PANETTA,1999). Segundo Tremback e Walko (1997), o RAMS foi desenvolvido pelo Departamento de

Ciência Atmosférica da Universidade do Estado do Colorado a partir da junção dos modelos: CSU de nuvens/mesoescala (TRIPOLI ; COTTON, 1982), versão hidrostática do modelo de nuvens (Tremback, 1990) e do modelo de brisa do mar (MAHRER ; PIELKE, 1977). Para Pielke et al. (1992), dentre todas as melhorias introduzidas no RAMS com a junção desses modelos, a capacidade de aninhamento de grades é a mais importante, pois permite a criação de uma grade detalhada (maior resolução) a partir de uma grade de menor resolução (é possível aninhar mais de uma grade dentro de uma grade de menor resolução). O RAMS possui uma série de características e opções (discutidas detalhadamente por Walko e Tremback, 1991), que podem ser configuradas em tempo de execução.

Para simulações de previsão de tempo, o RAMS necessita de dados de prognósticos para as fronteiras laterais (condição de contorno) provenientes de um modelo global e de dados de análise para as condições iniciais. Os dados a serem utilizados pelo BRAMS são provenientes do RAMS/ISAN - ”ISentropic ANalysis package” (TREMBACK, 1990). Este pacote converte o formato da análise proveniente de um modelo global para o formato do RAMS, ou combina e processa dados observacionais, gerando uma análise de superfície e de ar superior.

Para a geração de suas previsões, o RAMS utiliza como condição de contorno as informações fornecidas por modelos globais de previsão de tempo. Esta fronteira altera seus valores a cada passo de tempo através de uma média ponderada entre os valores iniciais e finais das saídas do modelo global para uma dada previsão do modelo RAMS. A cada passo de tempo o RAMS calcula a previsão de tempo forçando os dados de sua fronteira com os dados da média ponderada do modelo global, através de uma técnica de assimilação de dados (“nuddging”).

O resultado de uma previsão, gerada por um modelo meteorológico para um dado instante de tempo, é armazenado em um conjunto de matrizes tridimensionais (cada uma representando uma variável). Esses campos também podem ser vistos como colunas verticais sobre pontos de grade.

O código do BRAMS permite execução paralela, para isto o conjunto de pontos de grade que compõem a análise é transformado em subconjuntos retangulares de pontos de grade (subdomínios). O diagrama de uma grade (com subdomínios) possui uma semelhança com fileiras de blocos empilhadas, com linhas contínuas separando cada fileira. As linhas que separam os blocos terminam geralmente na parte de cima e de baixo de cada fileira, mas em alguns casos pode coincidir com as linhas adjacentes de cada fileira. A Figura 4.1 apresenta a decomposição em subdomínios, conforme observado em experimento conduzido por Mendes e Panetta (1999).

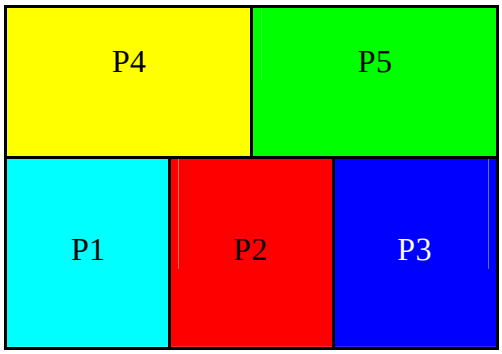


Figura 4.1 - Decomposição de domínio para cinco processadores.
Fonte: Mendes e Panetta (1999).

Segundo Tremback e Walko (1997), nesta estrutura de subdomínios emprega-se o modelo mestre-escravo. Neste modelo, um processo mestre controla a execução dos demais processos (escravos), que são efetivamente os responsáveis pela execução das tarefas. No BRAMS, o processo mestre manipula a inicialização e a saída e os processos escravos executam a computação do modelo.

Cada nó escravo recebe um subdomínio, com a respectiva região de fronteira, e executa a computação. A cada passo de tempo (“timestep”), os nós trocam as informações sobre as fronteiras dos subdomínios, utilizando a biblioteca de passagem de mensagens MPI (GROPP et al., 1999) e o processamento continua até a geração de todos os campos para os horários previstos.

O BRAMS é um modelo meteorológico projetado para execução em “clusters”, que são compostos de um conjunto de computadores interligados por uma rede de comunicação. Durante a execução paralela do BRAMS existe uma divisão estática do domínio, em

função do número de processadores, que ocorre durante o processo de inicialização. A distribuição de cada subdomínio aos processadores é realizada de forma estática, sendo cada subdomínio atribuído a um processador deste conjunto. Tanto a divisão dos subdomínios, como a atribuição dos mesmos aos computadores, não é alterada durante a execução do BRAMS. A figura 4.1 mostra a divisão de domínios do BRAMS para 5 processadores escravos. Nesta divisão, o BRAMS considera a utilização de uma arquitetura homogênea de computadores para a execução da aplicação.

Esta divisão estática do domínio causa um desbalanceamento de carga conforme nos mostra a Figura 4.2. Nesta figura é apresentada a variação do tempo de processamento do BRAMS em cinco processadores de uma arquitetura homogênea, durante a integração do modelo para a geração de um prognóstico para 24 horas. Ela é referente às diferenças no tempo de processamento dos subdomínios, quando da utilização de um nível maior de análise microfísica, devido ao fato que as regiões que possuem nuvens necessitam mais processamento do que as regiões sem nuvens. Nesta figura é clara a diferença do tempo de processamento entre os processadores 1 e 3 e demonstra o comportamento não determinístico desta aplicação. Mendes e Panetta (1999) já tinham observado que esta divisão estática do domínio causa um desbalanceamento de carga, conforme nos mostra a figura 4.2, e concluíram que não é adequada.

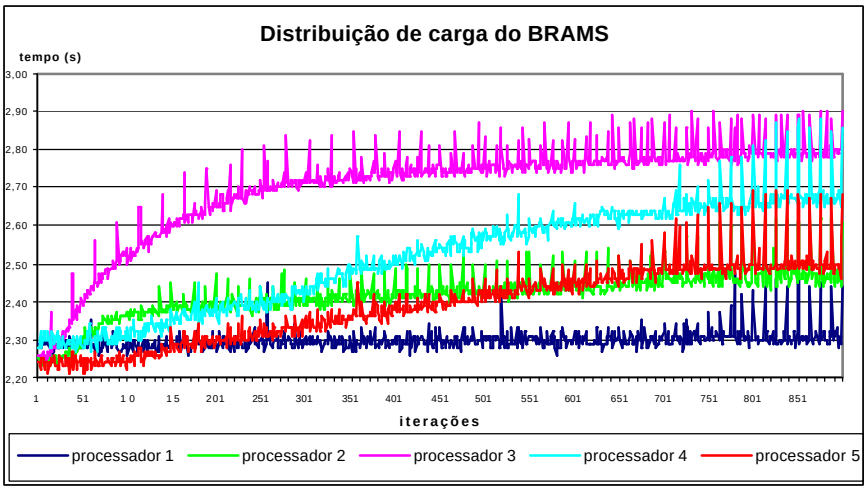


Figura 4.2 – Distribuição de carga para integração do BRAMS durante 24h em 5 processadores.

Analisando a Figura 4.3, nota-se a evolução da integração da precipitação durante um período de 24h. Para o subdomínio referente ao processador 3 existe uma variação grande da precipitação durante o período de integração que ocasiona uma variação grande do tempo de processamento deste domínio. Devido ao fato que não existe variação da precipitação no subdomínio do processador 1, quase não existe variação do tempo de processamento. Nos outros subdomínios nota-se uma variação do tempo de processamento em função da distribuição de precipitação.

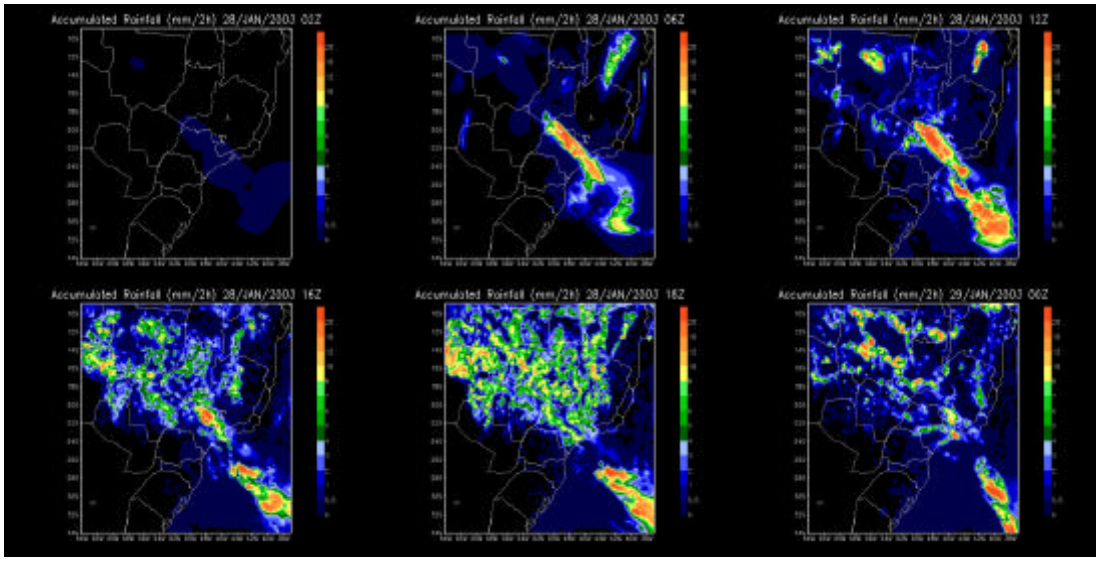


Figura 4.3 – Distribuição da precipitação durante a integração do BRAMS durante 24h

4.2 Utilização de modelos para previsão de clima

Segundo IRI (2006), as previsões de tempo são geralmente razoavelmente exatas em termos de predizer as características significativas do tempo para os próximos 1 a 2 dias. A exatidão das previsões de tempo diminui quando o tempo de integração aumenta para 3, 4, 5 ou mais dias. Uma previsão por 4 dias, por exemplo, necessita freqüentemente ser revisada com a aproximação desses dias, e em alguns casos a revisão pode ser grande. Para tempos de integração de 5 a 7 dias existe pouca exatidão para previsões de tempo, e após 7 dias não há quase nenhuma.

Para tempos de integração sazonal, os modelos não possuem destreza para prever em qual dia uma dada localidade terá ocorrência de precipitação, tempestades, extremos da temperatura, passagens frontais, etc.. Isto é devido à queda rápida da destreza após

diversos dias. No entanto, há alguma destreza em prever anomalias na média sazonal do tempo (anomalias do clima). Esta destreza está presente apesar do ajuste diário dos principais eventos de tempo dentro do período. A precipitação total, por exemplo, pode ser predita para ser mais elevada do que a média climatológica devido a uma freqüência prevista maior do que a normal de um padrão de circulação atmosférico específico que contribua para a chuva na localidade em questão. Novamente, o ajuste dos eventos de chuva permanece desconhecido. As previsões da probabilidade de aumento ou diminuição de chuva ou de temperaturas inferiores ou superiores a média, durante uma estação, tem um nível da exatidão que está longe de ser perfeito, mas visivelmente acima do nível da possibilidade aleatória. Este nível da destreza para médias ou totais sazonais pode ser útil para os setores impactados pela variabilidade do clima, tal como a produção de energia, a agricultura, a saúde, etc..

Muita da destreza em predizer o desvio das normais sazonais totais ou médias, freqüentemente associadas com padrões de circulação atmosférica, tem sua origem na mudança lenta das condições da superfície terrestre que podem influenciar o clima. A condição mais importante que afeta o clima é a temperatura de superfície do mar (TSM), particularmente a TSM nas zonas tropicais. Outras condições de superfície como a umidade do solo e a cobertura de neve são geralmente menos influentes. A característica das condições de superfície que dá a habilidade de influenciar a média das condições de tempo sobre um período futuro prolongado é a lentidão com que podem mudar, e conseqüentemente o período prolongado sobre os quais podem exercer sua influência consistente. Quando a TSM é maior do que o normal, ela geralmente mantém-se por vários meses, e às vezes por um ano ou mais, como durante os episódios El Niño ou La Niña (fases quentes e frias do ENSO - o El Niño/Oscilação Sul) da TSM do Pacífico tropical. Similarmente, quando há uma elevada umidade do solo, ou cobertura de neve, freqüentemente leva pelo menos diversas semanas para esta situação retornar ao normal, porque em cada dia o Sol pode somente evaporar ou derreter uma parcela limitada do excesso. Quando o solo está muito seco, pode levar de 4 a 8 eventos significativos de chuva para trazer a umidade do solo a seu normal, uma vez que a água de uma chuva pesada freqüentemente escorre pelo solo e não reabastece a umidade do

solo mais do que superficialmente. As anomalias da TSM são particularmente lentas à mudança por causa da elevada capacidade de calor da água com relação à atmosfera, tanto por causa de sua densidade mais elevada como devido ao fato que as anomalias podem se estender a muitas dezenas dos metros abaixo da superfície do oceano. Variações lentas da TSM implicam que os desvios significativos em relação à normal (anomalias) da TSM observada são prováveis de persistir nos meses seguintes. Isto também significa que se anomalias de TSM podem ser previstas com alguma confiabilidade (caso de determinados exemplos importantes), então o clima que é associado dinamicamente com as anomalias de TSM pode também ser previsto com alguma confiabilidade.

4.3 Parametrizações do MCGA e do BRAMS

Segundo Melo e Marengo (2004), o MCGA (Modelo de Circulação Global Atmosférica) com resolução T042L28 do CPTEC-COLA é derivado do modelo NCEP (KINTER et al. 1988). Ele possui um truncamento triangular, caracterizando uma resolução horizontal, na direção meridional e zonal, de 42 ondas na coordenada horizontal (2,8º de latitude por 2,8º de longitude) e 28 camadas em coordenadas sigma. O MCGA utiliza um módulo de superfície, o "Simplified Simple Biosphere Model" (SSiB), que considera a influência da vegetação de uma forma mais sofisticada (Xue et al., 1991).

As parametrizações ativadas no MCGA são:

- radiação de ondas curtas segundo Lacis e Hansen (1974), modificada por Davies (1982);
- ·radiação de ondas longas por Harshvardhan et al. (1987);
- a interação radiação-nuvens considera o esquema híbrido do Hou (1990), o qual é baseado no método de previsão de nuvens de Slingo (1987);
- convecção profunda do tipo Kuo (1974);

- convecção rasa segundo Tiedtke (1983);

- difusão vertical turbulenta aplicada a camada limite planetária, e difusão tipo bi-harmônica, para difusão horizontal, a qual é necessária para controlar o ruído de pequena escala, segundo Mellor e Yamada (1982).

Para a geração de previsões de clima as seguintes parametrizações são ativadas no BRAMS:

- Módulo de superfície LEAF 3, Land Ecosystem-Atmospheric Feedback Model, com formalismo de "patches"; prognostica temperatura e umidade do solo, da vegetação e do ar do dossel. Usando teoria de similaridade, prognostica o fluxo de calor sensível, latente e de momentum entre a superfície e a atmosfera(Walko et al., 2000; Lee et al., 1992);

- Radiação de onda curta (solar) e longa (terrestre), baseadas no trabalho de Chen e Cotton (1983), inclui espalhamento e absorção por água de nuvem;

- Turbulência baseada no formalismo de Mellor e Yamada (1982), prognostica a energia cinética turbulenta e as difusividades de momentum e de escalares com fechamento de ordem 2,5;

- A convecção profunda é baseada no trabalho de Grell e Devenyi (2002), possuindo um conjunto de hipóteses, da função gatilho e de fechamentos para determinar o valor ótimo dos parâmetros de aquecimento e umedecimento da atmosfera e da precipitação.

4.4 Modelos e grades computacionais

Neste capítulo foram apresentados modelos que necessariamente precisam ser executados em computadores com grande poder computacional: supercomputadores vetoriais e "clusters" de PC. Esses computadores possuem arquiteturas distintas, que requer que a adaptação do código desses modelos para que consiga ser executada com eficiência em uma dessas arquiteturas.

Os modelos MCGA, ETA e BRAMS são utilizados para tempo e clima no CPTEC/INPE. Seus códigos estão adaptados para execução tanto no supercomputador vetorial NEC-SX6 como em clusters de PC. O modelo ETA encontra-se paralelizado utilizando as linguagens de programação OpenMP (paralelismo intra-nó) e MPI (paralelismo entre-nós), enquanto que nos modelos MCGA e BRAMS foi implementado apenas paralelismo com MPI.

Supercomputadores são máquinas extremamente caras, que se torna inviável a composição de uma grade de supercomputadores no Brasil. Neste caso, construir uma grade computacional com “clusters” é mais viável devido ao custo e a grande disseminação desta tecnologia.

Como aplicação para ser executada em grade computacional foi escolhido o modelo BRAMS por ter uma grande comunidade no Brasil, utilizando-o tanto em produção como em pesquisa. A tecnologia de clusters é amplamente difundida no país, desta forma a grade de pesquisa será composta de clusters localizados em regiões geograficamente distintas executando o modelo meteorológico BRAMS.

Além disto, a utilização de grade computacional para determinação de climatologia de mesoescala do modelo BRAMS é uma nova área de pesquisa que possui desafios relacionados ao escalonamento de tarefas e adaptação do próprio código. Como conseqüência, construir uma grade computacional composta de “clusters” de computadores é a melhor solução para utilização do BRAMS em climatologia.

CAPÍTULO 5

CLIMATOLOGIA

Climatologia de modelos consiste na determinação de médias, para um certo período, da simulação do estado da atmosfera por um determinado modelo por um período longo de tempo. O conhecimento da climatologia mensal das variáveis de um modelo permite que a execução desse modelo seja usada para a geração de previsões de clima.

Basicamente existem dois métodos básicos para a determinação da climatologia de modelos numéricos:

- Uma única integração longa, metodologia empregada pelo “International Research Institute for Climate Prediction” (IRI) e;
- Múltiplas integrações distintas, iniciadas em meses distintos e com duração curta (alguns meses), adotado pelo “European Centre for Medium-Range Forecasts” (ECMWF).

Em termos computacionais, a climatologia IRI possui um custo menor que a do ECMWF.

Uma forma de melhorar a qualidade da climatologia é simular o estado da atmosfera em modo “ensemble”. Desta forma, para a determinação da climatologia de um modelo, para uma determinada área geográfica, três parâmetros necessitam ser definidos:

- Data de início da integração;
- Comprimento da integração (dias, meses ou anos);
- Número de membros.

5.1 ENSEMBLE

Em física, um “ensemble” estatístico é um conjunto de cópias de um sistema, considerados todas de uma vez. Cada cópia do sistema representando uma possibilidade diferente para realização do sistema, consistente com as propriedades macroscópicas do sistema observado (WIKIPEDIA, 2006).

Geralmente, o alvo de rodar um “ensemble” é tratar as incertezas de um sistema. Um “ensemble” climatológico consiste em utilizar o mesmo modelo em termos dos mesmos parâmetros físicos atmosféricos e forçantes, mas executando com uma variedade de datas iniciais diferentes. O resultado de cada execução é denominado membro. Devido ao fato que o sistema do clima é caótico, pequenas mudanças em variáveis (como temperatura, ventos, e umidade em um lugar) podem levar o sistema a resultados muito diferentes para um todo.

Segundo Mendonça e Bonatti (2002), um “ensemble” gera uma grande quantidade de informações e pode, em princípio, surpreender a um usuário que ainda não teve contato com este método. Condensar as informações contidas nestas previsões e extrair delas as características mais importantes é um dos desafios. A forma mais simples de apresentar estas informações seria plotar todas as previsões dos membros em uma única página. Entretanto, isto poderia causar interpretações subjetivas e discussões improdutivas; além disso, à medida que o número de membros aumentasse seria difícil ter uma visão global de todos os padrões previstos. Esforços significativos têm sido dedicados ao desenvolvimento de produtos que sintetizem as informações do “ensemble” e que auxiliem sua interpretação.

A forma mais condensada de obter informações da previsão por “ensemble” é denominada “ensemble” médio. Ela consiste em calcular a média das previsões considerando-se que todos os membros sejam igualmente prováveis de ocorrerem, desta forma, não se atribui peso a nenhuma previsão específica. O cálculo pode ser feito para cada ponto de grade j conforme apresentado na Equação 5.1:

$$EM_j = \frac{1}{N}\sum_{i=1}^{N} F_j^i \qquad (5.1)$$

onde N é o número de membros do “ensemble” e F_j^i são as previsões de cada membro.

5.2 Climatologia – métodos DERF/ECMWF e IRI

O método DERF/ECMWF para determinação da climatologia prevê que inicialmente sejam executadas simulações do modelo, em modo “ensemble”, para cada membro separadamente. Para determinar a climatologia de um dado mês, cada simulação considera uma integração do modelo por três meses. Para todo período considerado da climatologia este processo se repete (com exceção dos dois últimos meses, devido a inexistência de dados para a simulação do modelo nos três meses subseqüentes). Desta forma, este método possui o custo computacional apresentado na Equação 5.2:

Custo = ((n_anos * n_meses) –2)) * per_integr* n_membros (5.2)

onde:

n_anos – número de anos
n_meses – número de meses
per_integr – período de cada integração
n_membros – número de membros

A climatologia mensal do método DERF/ECMWF é calculada a partir da média sobre todas as integrações iniciadas em determinado mês. Com isto são geradas doze climatologias, sendo cada uma de três meses.

No método para determinação de climatologia do IRI, executam-se integrações do modelo para a totalidade do período estipulado, para cada membro. O custo computacional deste método é apresentado na Equação 5.3:

Custo = ((n_anos * n_meses) * n_membros (5.3)

onde:

n_anos – número de anos
n_meses – número de meses
n_membros – número de membros

Gera-se uma climatologia por mês. A média é calculada sobre todas as incidências do mês ao longo das integrações.

Para uma climatologia mensal de três anos do modelo BRAMS, o método IRI necessita realizar três integrações de 36 meses cada. Considerando que são três membros, o custo computacional é de 108 meses de integração. O método DERF/ECMWF necessita de 34 integrações ((3 x 12) - 2) de três meses, totalizando 102 meses. O custo final deste método é igual 306 meses de integração, considerando os três membros. Conclui-se que o custo computacional da metodologia DERF/ECMWF é aproximadamente três vezes maior que a do IRI.

5.3 Simulação global de 50 anos em modo "ensemble"

A simulação global de 50 anos, em modo "ensemble", foi gerada no supercomputador vetorial NEC SX-6 a partir da execução de três integrações de 50 anos do MCGA/T062L28 (aproximadamente 200 km de resolução horizontal) do CPTEC-COLA. As reanálises do NCAR/NCEP para os dias dezoito, dezenove e vinte de janeiro de 1950 foram utilizadas para a inicialização do modelo. Como condição de contorno, foi utilizado o conjunto de TSM observada, mês a mês, para o período de janeiro de 1950 a dezembro de 2001. As variáveis de superfície são temperatura da superfície do solo, umidade do solo, albedo da superfície e profundidade da neve, as quais são introduzidas no início da integração com valores climatológicos e são ajustadas durante a integração. O albedo é função do ângulo zenital solar sobre o oceano, no entanto, sobre a superfície é predito pelo SSiB. Para a concentração de dióxido de carbono considera-se um valor constante de 345 ppm.

Esta simulação gerou saídas dados de:

- Membro 1 – 18 de janeiro de 1950 a 21 de dezembro de 2001;
- Membro 2 – 19 de janeiro de 1950 a 22 de dezembro de 2001;
- Membro 3 – 20 de janeiro de 1950 a 23 de dezembro de 2001;

Os primeiros meses de integração são desprezados por serem considerados como tempo necessário para que as variáveis do modelo possam atingir um certo estágio de equilíbrio.

Durante pós-processamento do modelo global é executada a conversão entre os estados da atmosfera (do MCGA para o BRAMS), que gera a cada seis horas um arquivo de saída (arquivo iniciado com prefixo gamrams). O arquivo é específico para a área de interesse, cobrindo toda a América do Sul, conforme representado na Figura 5.1 .

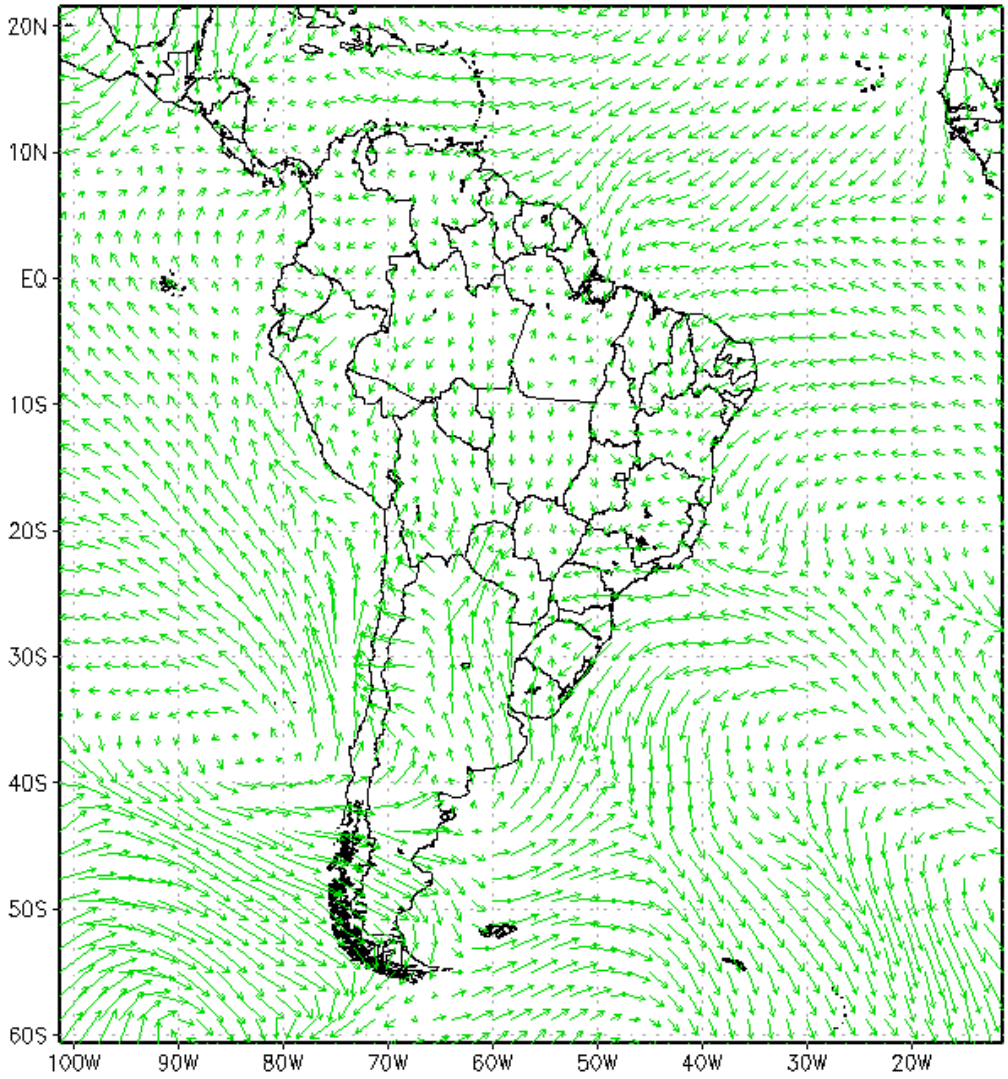


Figura 5.1 – Campo de vento - referente ao recorte da simulação do MCGA

No MCGA do CPTEC, o estado da atmosfera é definido por cinco variáveis prognósticas (componentes u e v da velocidade, temperatura, umidade relativa e pressão na superfície). Ele utiliza coeficientes espectrais na horizontal e níveis sigma na vertical, abrangendo todo o globo terrestre. No BRAMS, o estado da atmosfera é definido por variáveis prognósticas equivalentes às do MCGA do CPTEC (com geopotencial substituindo a pressão na superfície), mas em outra representação. São campos completos, com grade horizontal regular (igualmente espaçada em latitudes e em longitudes) e níveis de pressão na vertical, abrangendo apenas a região de interesse.

5.4 Climatologia regional

A previsão climática global não é muito útil ao usuário em seu processo de tomada de decisão pois ele necessita na maioria das vezes uma previsão climática regionalizada. A técnica de "downscaling" dinâmico permite regionalizar a previsão climática global.

A técnica de "downscaling" dinâmico (ou ampliação da resolução) é baseada em modelos numéricos e consiste no aninhamento de um modelo regional atmosférico a um Modelo de Circulação Global Atmosférico (MCGA), sendo o primeiro forçado pelas informações geradas pelo MCGA. A característica desta técnica é ampliar a resolução espacial da previsão, com incorporação dos efeitos da topografia, vegetação, contrastes entre continente/oceano, entre outros.

No caso deste trabalho, esta técnica é aplicada à simulação de 50 anos do MCGA do CPTEC para obter a climatologia do BRAMS para três regiões do Brasil (Norte, Nordeste e Sul/Sudeste).

O conjunto de TSM observada para o período de novembro de 1995 a dezembro de 1998, é utilizado como condição de contorno para o modelo regional. O BRAMS atualiza, a cada seis horas de integração, o estado da atmosfera na região de interesse inserindo fenômenos advindos de outras regiões do globo. Conseqüentemente, a cada seis horas de integração o BRAMS deve ler um arquivo gerado pelo modelo global do CPTEC e preparado para a região de interesse.

A climatologia regional do modelo BRAMS é determinada em duas fases. Inicialmente executa-se a simulação regional do BRAMS para um período de três anos, em modo "ensemble" (três membros). Isto é feito executando três integrações de três anos do BRAMS, utilizando os dados da simulação MCGA/T042L28 do CPTEC-COLA para três dias consecutivos (dias primeiro, dois e três de novembro de 1995) como condições iniciais para do modelo BRAMS. Na fase seguinte, determina-se o "ensemble" médio, que é o resultado do cálculo da média mensal de todos os membros de cada mês.

A climatologia regional executada no ambiente de grade é um dos temas da tese. Uma descrição mais detalhada da grade computacional, nos seus aspectos de hardware, softwares básicos e dos protocolos de comunicação (plataforma de grade) utilizados para o experimento deste estudo são apresentados no Capítulo 6. No capítulo 7 também são apresentados os resultados da aplicação G-BRAMS em climatologia de mesoescala.

CAPÍTULO 6

O PROJETO G-BRAMS

O Projeto G-BRAMS é financiado pela FINEP CT-INFO através do edital de Grid Computing “Grade-01/204”. Este projeto foi proposto pelo Laboratório de Computação Aplicada (LAC), pelo Centro de Previsão de Tempo e Estudos Climáticos (CPTEC) – pertencentes ao Instituto Nacional de Pesquisas Espaciais (INPE) - e pelo Instituto de Informática (II) da Universidade Federal do Rio Grande do Sul (UFRGS) para avaliar a tecnologia de grade na área de meteorologia. Neste projeto foi proposta a utilização de simulação de climatologia de 10 anos do modelo de mesoescala BRAMS em modo “ensemble”. Três regiões do Brasil foram selecionadas e para cada área selecionada a simulação foi executada para três condições iniciais diferentes, resultando em uma simulação com três membros por área. Este tipo de simulação requer diversas execuções independentes do BRAMS. Três tecnologias de grade (Globus, CIGRI/OAR e OurGrid), foram utilizadas na grade composta de “clusters” (nós de grade) instalados nas instituições envolvidas no projeto, para a comparação das tecnologias de grade.

6.1 Descrição do ambiente de grade

A grade implantada é composta por três nós de grades: 1 (um) “cluster” com 18 processadores Xeon 3.0 GHz e outros 2 (dois) “clusters” CRAY/XD1, com 12 processadores Opteron 2.6GHz, localizados respectivamente no CPTEC/INPE (Cachoeira Paulista), no LAC/INPE (São José dos Campos) e II/UFRGS (Porto Alegre). Esta é uma grade homogênea com relação ao sistema operacional (Linux) e heterogênea com relação à arquitetura e ao poder computacional de cada nó de grade.

Para a criação da grade, foram instaladas e configuradas no computador “front-end” de cada nó de grade, três plataformas de grade: Globus, OurGrid e CIGRI/OAR. Estas ferramentas permitem a execução remota de programas em computadores localizados em diferentes localizações geográficas.

Adicionalmente, foi instalada a biblioteca de passagem de mensagem MPICH (GROPP et al., 1996). Esta biblioteca permite que aplicações paralelas possam ser executadas de forma paralela em “clusters”, como é o caso do BRAMS.

O Globus possui um esquema de segurança mais rígido que o CIGRI/OAR e OurGrid. Neste caso, uma autoridade certificadora foi instalada no computador “front-end” do nó de grade localizado no LAC/INPE. Os certificados assinados pela autoridade certificadora foram instalados para cada usuário no computador “front-end” de cada nó de grade, que foram validados através de testes específicos executados para verificar o funcionamento do sistema de autenticação.

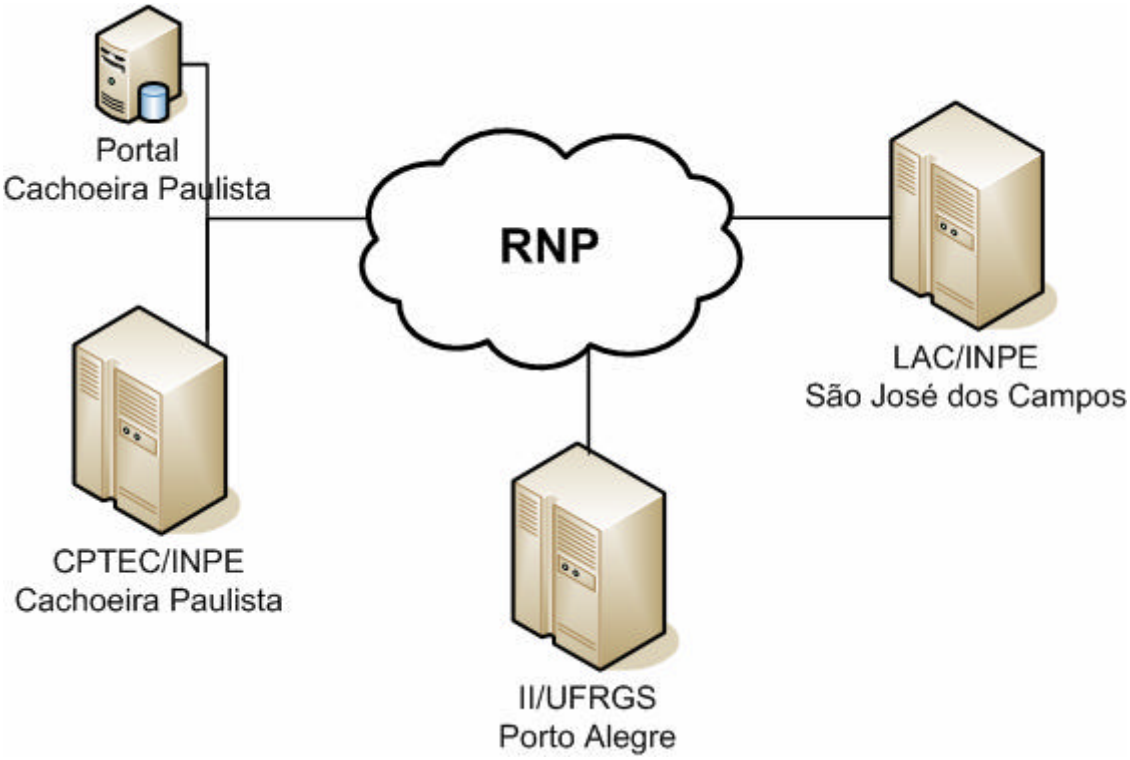


Figura 6.1 – Arquitetura da grade computacional

6.2 Algoritmo de climatologia em grades computacionais

A climatologia de mesoescala do modelo BRAMS é determinada a partir da execução da metodologia do IRI na grade computacional apresentada na seção 5.2, utilizando o Grid Analysis and Display System (GrADS).

O sistema GrADS (DOTY, 1995) é uma ferramenta de “desktop” interativa, que possui um conjunto de funções embutido. Possui uma interface programável, fornecida na forma de uma linguagem interpretada (“script”). Esta linguagem permite a criação de interfaces gráficas e automatização de cálculos complexos ou exibições. As operações podem ser realizadas diretamente e interativamente nos dados. Este pacote de pós-processamento tem amplo uso e permite a análise e exibição de dados de Ciência da Terra. O GrADS está implementado em várias plataformas e opera sobre dados de modelos em 4-D, as 3 variáveis espaciais mais o tempo.

Na metodologia desenvolvida para determinação de grades computacionais, o modelo BRAMS foi integrado para o período de estudo (Novembro-1995 a Dezembro-1998) e foram geradas saídas a cada 6 horas. Após esta integração, executou-se o programa desenvolvido para o cálculo das médias mensais dos campos gerados pelo BRAMS, utilizando funções do sistema GrADS. O resultado é uma saída média mensal, denominada de climatologia daquela integração (Figura 6.2).

Para cada variável de saída do BRAMS, a climatologia média mensal foi calculada para cada horário sinótico (00, 06, 12 e 18 UTC) e também para o período total, no computador escolhido para esta função.

Em modo “ensemble”, este procedimento tem que ser repetido em função do número de membros a ser considerado. Desta forma, utilizando condições iniciais diferentes, esses procedimentos são executados em “clusters” distintos (nós da grade). Ao término da execução de todos os membros, os resultados são enviados ao computador que disponibilizará a climatologia.

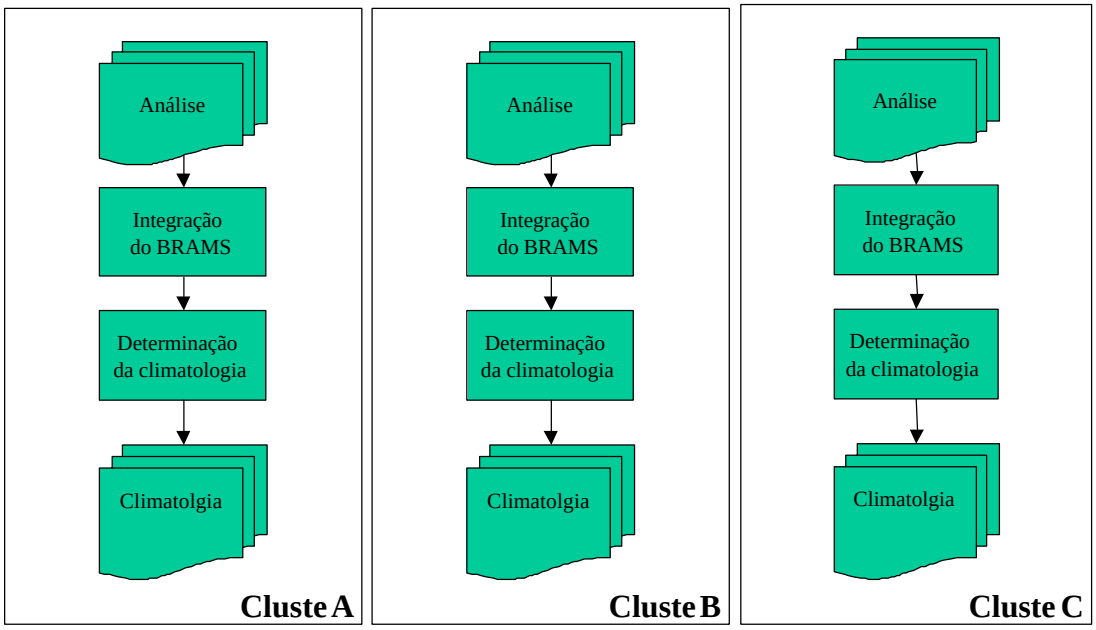


Figura 6.2 – Determinação da climatologia.

6.3 Descrição de um experimento em uma grade de pesquisa

O ponto de partida deste experimento são os dados da simulação de 50 anos (19-janeiro-1950 a 22-dezembro-2001) do MCGA do CPTEC na resolução T062L28 (aproximadamente 200 km de resolução horizontal). Esses dados contem o estado da atmosfera global e são armazenados em arquivos com o tamanho de 5.456.464 bytes (5,2 Mb). A partir deste arquivo gera-se um subconjunto de dados (“recorte”) para a área referente à América do Sul (coordenadas geográficas entre as longitudes 101º 14’ e 9º 22’ O e entre as latitudes 23º 45’ N e 60º 02’ S). Esses arquivos iniciam com o nome de “*gamrams*” e o tamanho de cada arquivo é de 652.680 bytes (637,4 Kb). Na Tabela 6.1, apresenta-se o nome das variáveis, o número de níveis e a unidade dos dados armazenados neste arquivo de saída, que é o resultado da simulação de 50 anos do MCGA/CPTEC.

Tabela 6.1 – Nome, número de níveis e unidades das variáveis do arquivo gamrams

variável	n°. níveis	nome da variável	unidade
topo	0	TOPOGRAPHY	(M)
lsmk	0	LAND SEA MASK	(NO DIM)
psnm	0	SEA LEVEL PRESSURE	(Mb)
prec	0	TOTAL PRECIPITATION	(Kg M**-2 Day**-1)
uvel	14	ZONAL WIND (U)	(M/Sec)
vvel	14	MERIDIONAL WIND (V)	(M/Sec)
zgeo	14	GEOPOTENTIAL HEIGHT	(M)
temp	14	ABSOLUTE TEMPERATURE	(K)
umrl	14	RELATIVE HUMIDITY	(No Dim)

A simulação de 10 anos do BRAMS em grade computacional utiliza como condição de inicialização e de contorno os campos de temperatura do ar (apresentado na Figura 6.4), de geopotencial, de umidade do ar (temperatura do ponto de orvalho ou razão de mistura ou umidade relativa do ar ou, ainda, diferença psicrométrica) e de vento (componentes zonal - u e meridional - v ou direção e intensidade) em diferentes níveis da atmosfera. Esses dados são originados dos resultados da simulação dos 50 anos do MCGA, apresentados anteriormente, e possuem as seguintes características:

- Grade:
  - Resolução espacial de 200kmx2000km;
  - Numero de pontos: 49x45;
  - Coordenadas geográficas: longitudes 101.2º O a 11.2º O e das latitudes 60.6º S to 21.4º N.

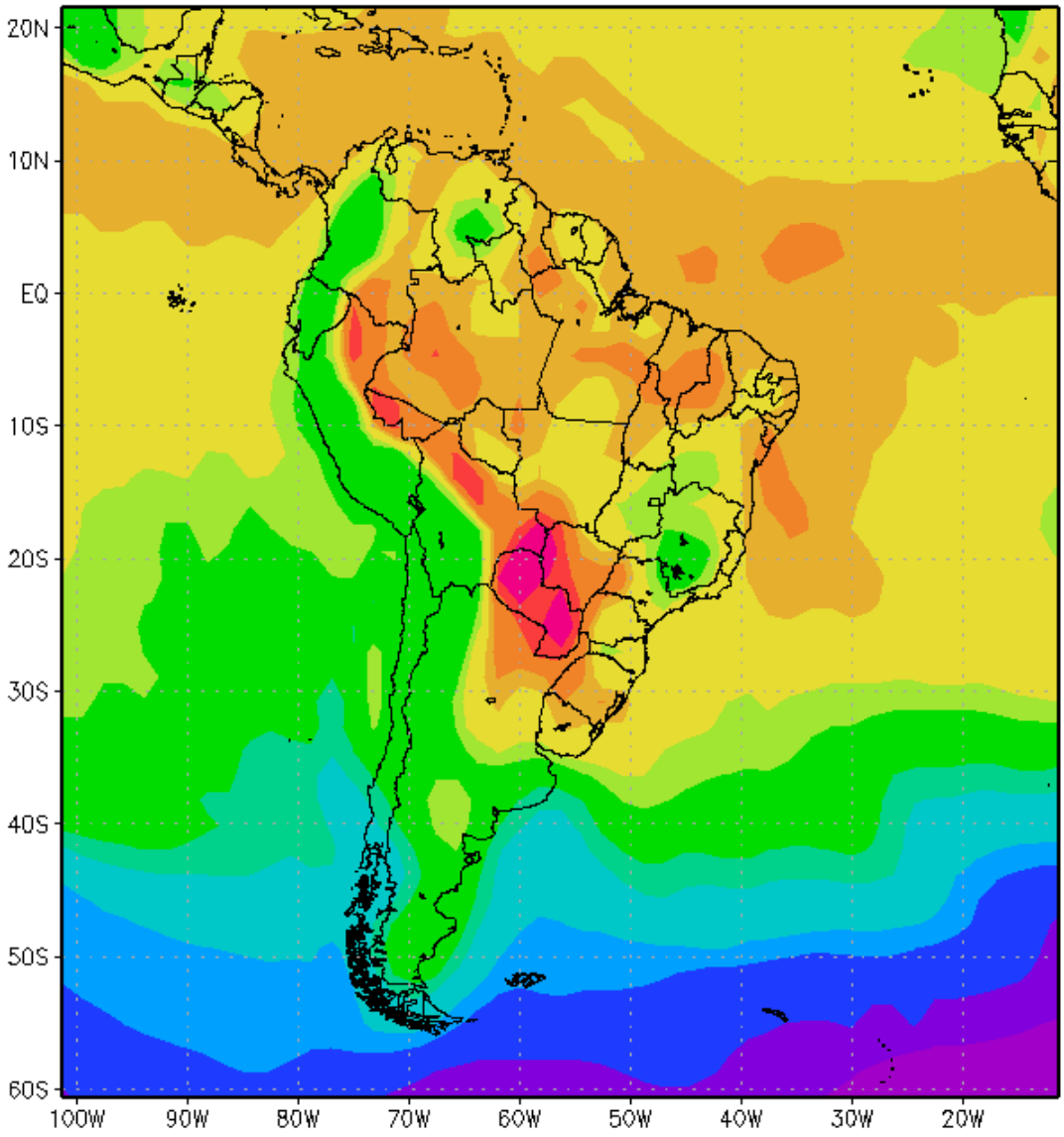


Figura. 6.3 – Campo de temperatura referente ao recorte da simulação do MCGA

Neste experimento, foram definidas três regiões (grades) para determinar a climatologia: Norte, Nordeste e Sul/Sudeste. A simulação regional do BRAMS, para cada região foi executada para duas grades aninhadas utilizando os arquivos recortados do MCGA. As características das grades são apresentadas a seguir.

- Região Norte:
  - Grade aninhada 1:
    - Resolução espacial de 160kmx160km;
    - Numero de pontos: 44x44;

- Coordenadas geográficas: longitudes 75.4º O a 46.3º O e das latitudes 14.1º S a 5.9º N;

o Grade aninhada 2:

- Resolução espacial de 40kmx40km;
- Numero de pontos: 83x58;
- Coordenadas geográficas: longitudes 75.4º O a 46.3º O e das latitudes 14.1º S a 5.9º N;
- Ampliação de 4 vezes.

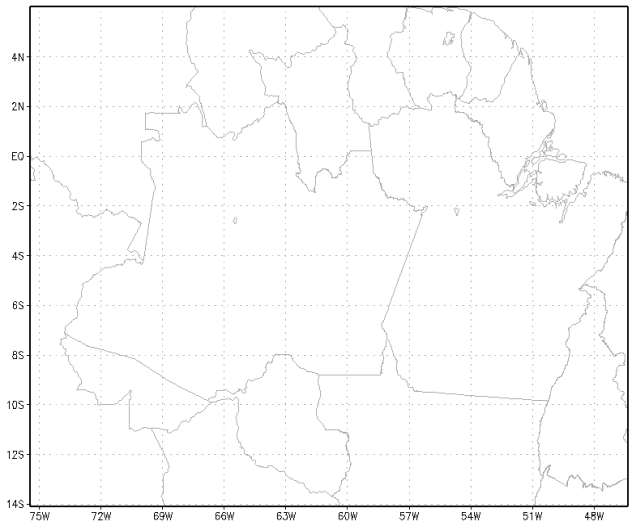


Figura 6.4 – Área referente à região Norte.

➢ Região Nordeste:

o Grade aninhada 1:

- Resolução espacial de 160kmx160km;
- Numero de pontos: 64x64;
- Coordenadas geográficas: longitudes 50.7º O a 28.0º O e das latitudes 21.3 º S a 0.94 º N;

o Grade aninhada 2:

- Resolução espacial de 40kmx40km;
- Numero de pontos: 64x64;
- Coordenadas geográficas: longitudes 50.7º O a 28.0º O e das latitudes 21.3 º S a 0.94 º N;
- Ampliação de 4 vezes.

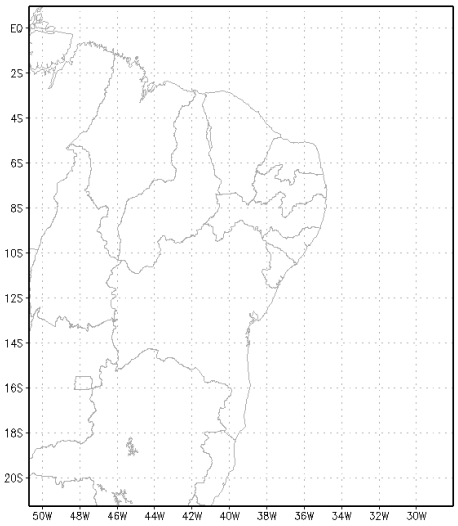


Figura 6.5 – Área referente à Região Nordeste

- Região Sul/Sudeste:
  - Grade aninhada 1:
    - Resolução espacial de 160kmx160km;
    - Numero de pontos: 42x42;
    - Coordenadas geográficas: longitudes 64.4º O a 36.2º O e das latitudes 37.5º S a 11.8 º S;
  - Grade aninhada 2:
    - Resolução espacial de 40kmx40km;
    - Numero de pontos: 81x83;
    - Coordenadas geográficas: longitudes 64.4º O a 36.2º O e das latitudes 37.5º S a 11.8 º S;
    - Ampliação de 4 vezes.

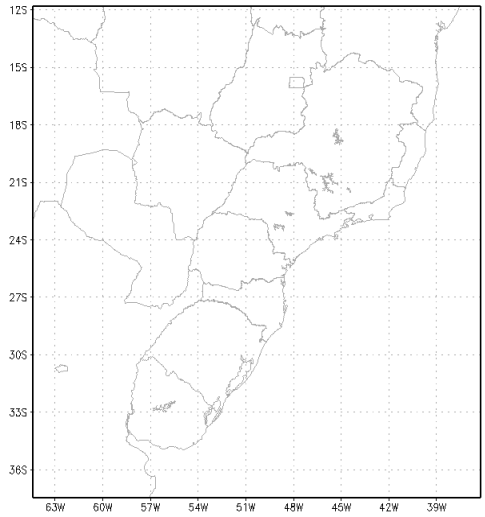


Figura 6.6 – Área referente à região Sul/Sudeste

O objetivo do primeiro aninhamento (grade aninhada 1) é gerar uma grade com aproximadamente o mesmo número de pontos da grade (44x44 pontos) do MGA, mas com uma maior resolução (160 Km) do que o MCGA. No Apêndice C são apresentadas as variáveis de superfície e de nível referentes às saídas das simulações paras as regiões Norte, Nordeste e Sul/Sudeste. No segundo aninhamento, o objetivo é gerar a simulação para as áreas e resoluções propostas. Neste processo de "downscaling", as variáveis de fronteira são interpoladas antes da execução do BRAMS, gerando dados para a integração do modelo no tempo de integração definido.

No processo de "downscaling" houve um aumento de quatro vezes da resolução dos dados da simulação do modelo global. A Tabela 6.2 apresenta a variação da resolução durante o processo de downscaling.

Tabela 6.2 – Resolução das regiões processadas: Norte, Nordeste, Sul/Sudeste

Resolução dos modelos (Km)		
Inicial	**Intermediária**	**Final**
200 x 200	160 x 160	40 x 40

Durante este processo de detalhamento da climatologia, houve também uma seleção das áreas alvo (Norte, Nordeste, Sul/Sudeste). Isto implicou na definição das coordenadas geográficas, que se encontram apresentadas na Tabela 6.3.

Tabela. 6.3 – Coordenadas geográficas das regiões processadas

Região	**Intervalo de coordenadas geográficas dos modelos**					
	Inicial		**Intermediária**		**Final**	
	longitude	**latitude**	**longitude**	**latitude**	**longitude**	**latitude**
Norte	101.2º O 11.2º O	60.6º S 21.4º N	75.4º O 46.3º O	14.1º S 5.9º N	75.4º O 46.3º O	14.1º S 5.9º N
Nordeste	101.2º O 11.2º O	60.6º S 21.4º N	50.7º O 28.0º O	21.3º S 0.94º N	50.7º O 28.0º O	21.3º S 0.94º N
Sul/Sudeste	101.2º O 11.2º O	60.6º S 21.4º N	64.4º O 36.2º O	37.5º S 11.8º S	64.4º O 36.2º O	37.5º S 11.8º S

Para cada região de interesse, a simulação de três anos foi executada com três membros, para o período de novembro de 1995 a dezembro de 1998. A simulação do BRAMS foi

executada para cada um dos três membros, considerando como data de início da integração de cada simulação os dias 1, 2 e 3 de novembro de 1995, respectivamente.

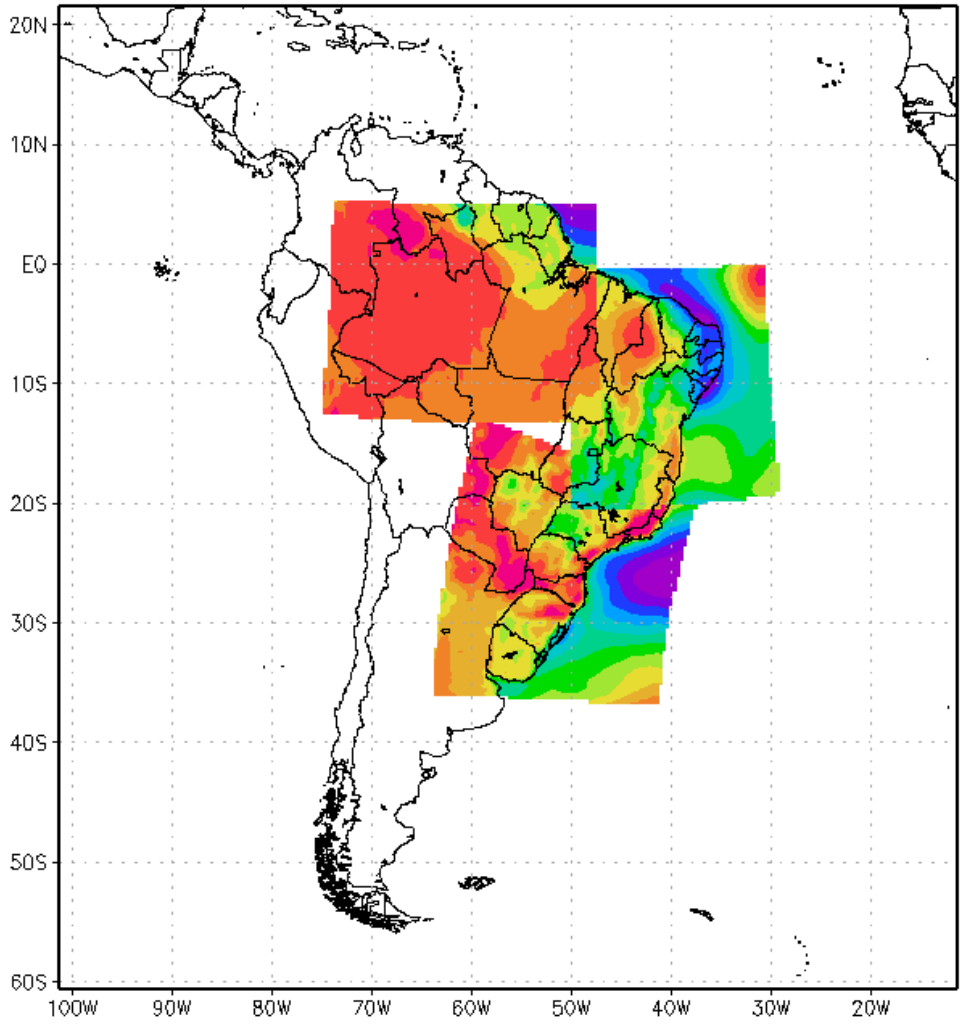


Figura. 6.7 – Áreas referente às regiões para determinação da climatologia.

6.4 Acesso à grade computacional

Para viabilizar a utilização da grade computacional, foi desenvolvido um portal Web, dentro do escopo do projeto G-BRAMS (CAMPOS VELHO et al, 2006; PANETTA et al., 2006, SOUTO et al., 2007), para se ter um ambiente único para acesso às três tecnologias de grade definidas no escopo do projeto. Este portal foi desenvolvido utilizando GridSphere Portal Framework (RUSSELL et al., 2005), baseado no Portlet Java Specification Request - JSR 168 (JAVA COMMUNITY PROCESS, 2004). Portlet

JSR define uma interface de programação com a aplicação (API) e um modelo para empacotamento e apresentação de conteúdos Web como portlets (classes Java que possuem uma interface bem definida).

O portal desenvolvido para o projeto G-BRAMS possui um banco de dados acoplado, modelado para conter informações sobre o estado das tarefas submetidas às tecnologias de grade (Figura 6.9). O estado das tarefas armazenado no banco de dados, apresenta os seguintes estados:

- EDICAO: A tarefa encontra-se no banco de dados, mas ainda não foi enviado à grade. O botão "Executar" libera a tarefa para execução;
- EDICAO*: Ocorre quando uma tarefa é resultado de uma rodada fracionada em meses. Quando a tarefa referente ao mês anterior é enviada para execução, o estado desta tarefa passa para EDICAO;
- LIBERADA: Estado da tarefa que estava com estado EDICAO e foi enviada para ser executada na grade;
- AGUARDANDO: Estado da tarefa que estava com estado EDICAO e foi enviada para ser executada na grade, mas ainda não há máquinas disponíveis para ser executá-la;
- EXECUTANDO: Estado da tarefa que estava como LIBERADA e um nó da grade ficou disponível para executá-la;
- CONCLUÍDA: Estado da tarefa que estava com estado EXECUTANDO e terminou de forma normal;
- ACEITA: Estado da tarefa que estava com estado CONCLUIDA e teve os resultados analisados e aceitos pelo meteorologista;
- ERRO: Estado da tarefa que estava em EXECUTANDO e foi cancelada pela plataforma de grade por ocorrência de erro durante a execução.

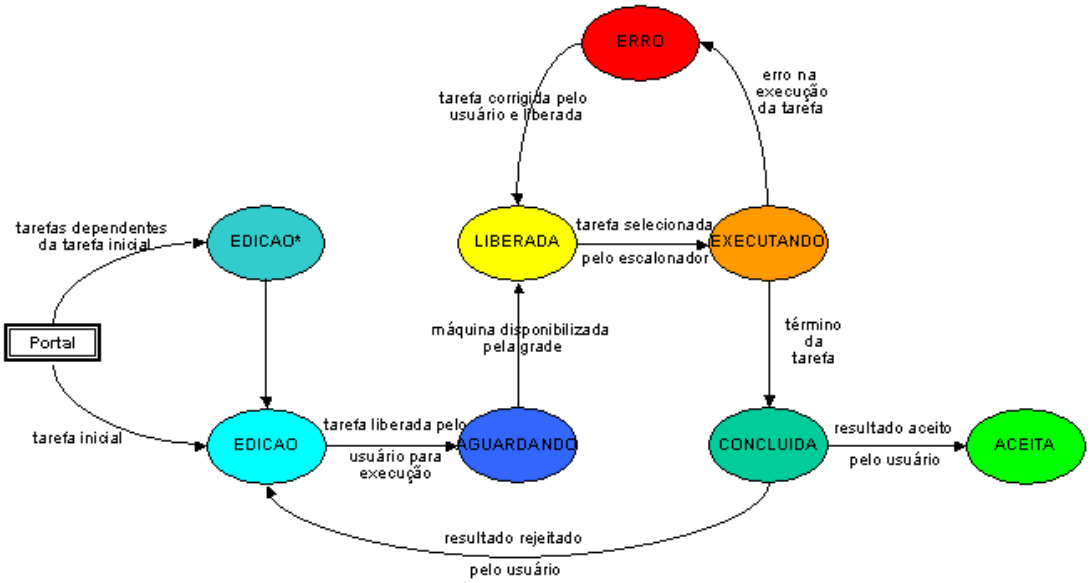


Figura 6.8 – Diagrama de estados

Cada tecnologia de grade possui sua própria interface, que tem o mesmo aspecto para o usuário para criação (Figura 6.10), para submissão/acompanhamento das tarefas (Figura 6.11) e para visualização de resultados (Figura 6.12), para o caso em estudo a simulação de climatologia do modelo de mesoescala BRAMS. A interface de submissão/acompanhamento das tarefas permite acompanhar e alterar o estado das tarefas. Uma tarefa criada possui o estado EDICAO, que informa que está pronta a ser executada. O estado desta tarefa pode ser alterado, permitindo sua execução ou edição. Os estados das tarefas apresentados na interface de submissão/acompanhamento das tarefas são armazenados em banco de dados acoplado à interface.

Na interface para criação das tarefas encontram-se todos os parâmetros para a execução do BRAMS. Esta interface possui os mesmos parâmetros que o arquivo RAMSIN, que é o arquivo necessário para execução do BRAMS no modo linha de comando.

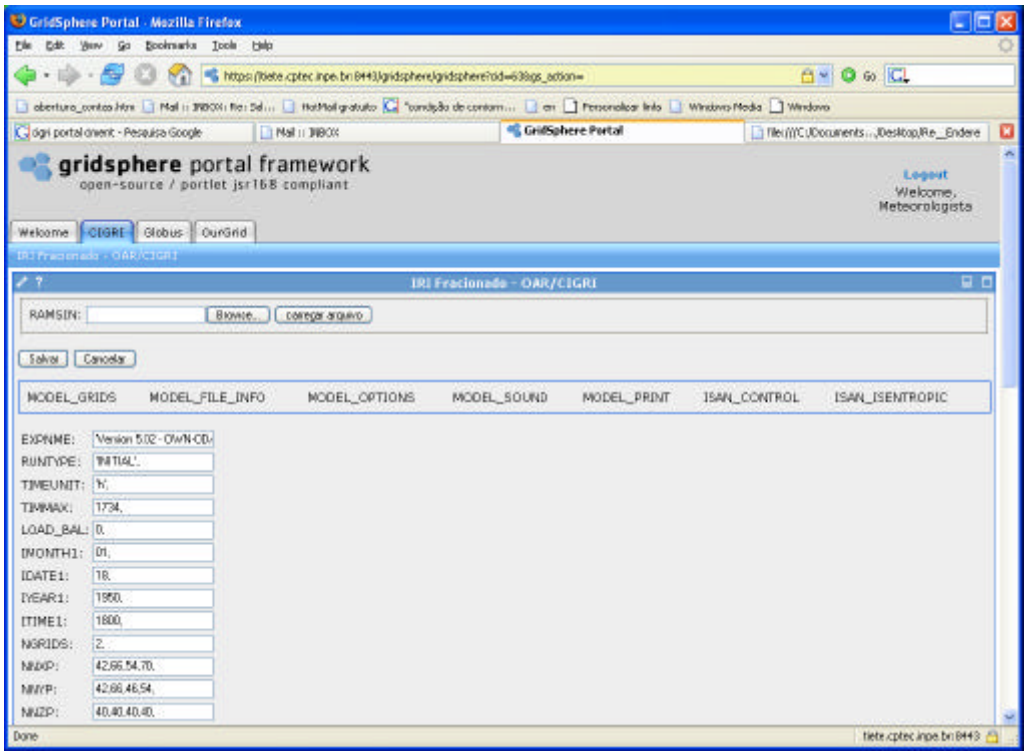


Figura 6.9 - Portal G-BRAMS – interface para criação de tarefas

Na interface de submissão/acompanhamento das tarefas o usuário possui três opções (botões): Criar Job, Executar e Resultados. Nesta interface são apresentadas as seguintes informações sobre a tarefa:

- ID – número que indica a seqüência de criação de tarefas;
- Criação – Data e hora da criação da tarefa;
- Início da Execução - Data e hora do início da execução da tarefa;
- Fim da Execução - Data e hora do término da execução da tarefa;
- Estado – Informação sobre o estado da tarefa;
- Host – nó de grade onde a tarefa foi executada.

A primeira opção a ser utilizada é Criar Job, que altera para a interface para criação das tarefas, explicada anteriormente. Ela permite que o usuário defina os parâmetros para a execução do BRAMS e salve-os no banco de dados. Após a criação da tarefa, o estado

EDICAO é atribuído para esta tarefa no entrada do banco de dados. A partir deste momento, o usuário pode alterar o estado da tarefa, selecionando a mesma e utilizando botão Executar, que submete a aplicação para execução na grade. Após a execução, o resultado pode ser visualizado através do botão Resultados na interface de submissão/acompanhamento das tarefas. Este botão alterna para a interface de visualização

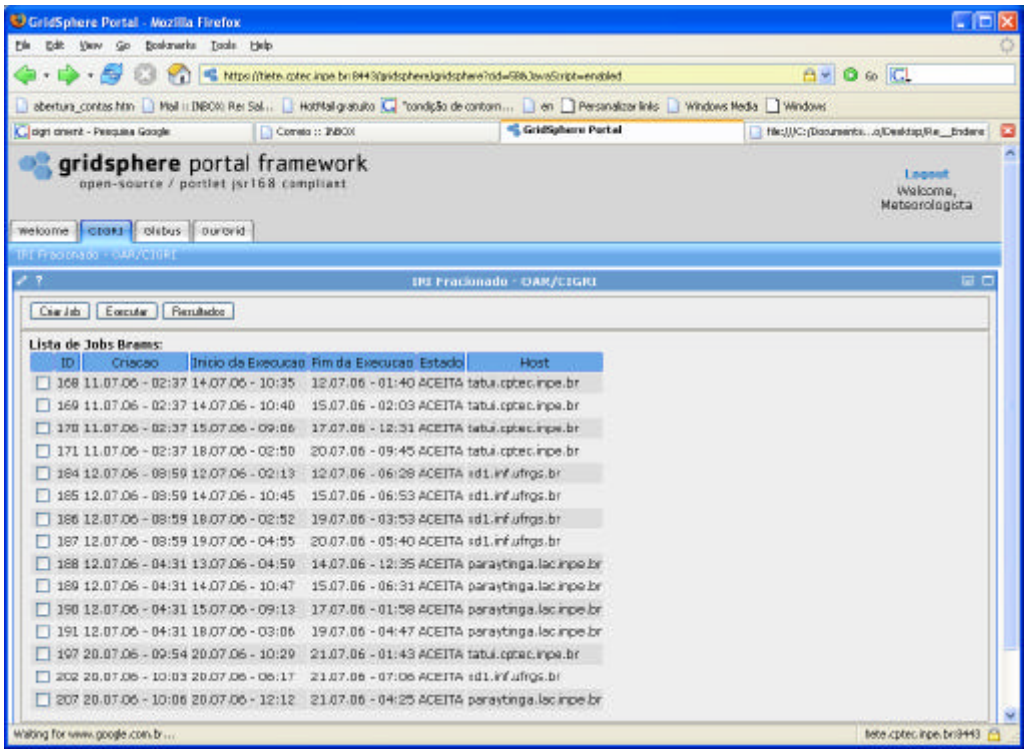


Figura 6.10 - Portal G-BRAMS: interface de submissão e acompanhamento de tarefas

O usuário possui duas ações sobre o resultado apresentado na interface de visualização de resultados: ACEITA ou REJEITA. Ao concordar com o dado apresentado, o usuário aciona o botão ACEITA e a próxima tarefa é liberada para execução. No caso de rejeitar o resultado, a tarefa é retornada para o modo edição.

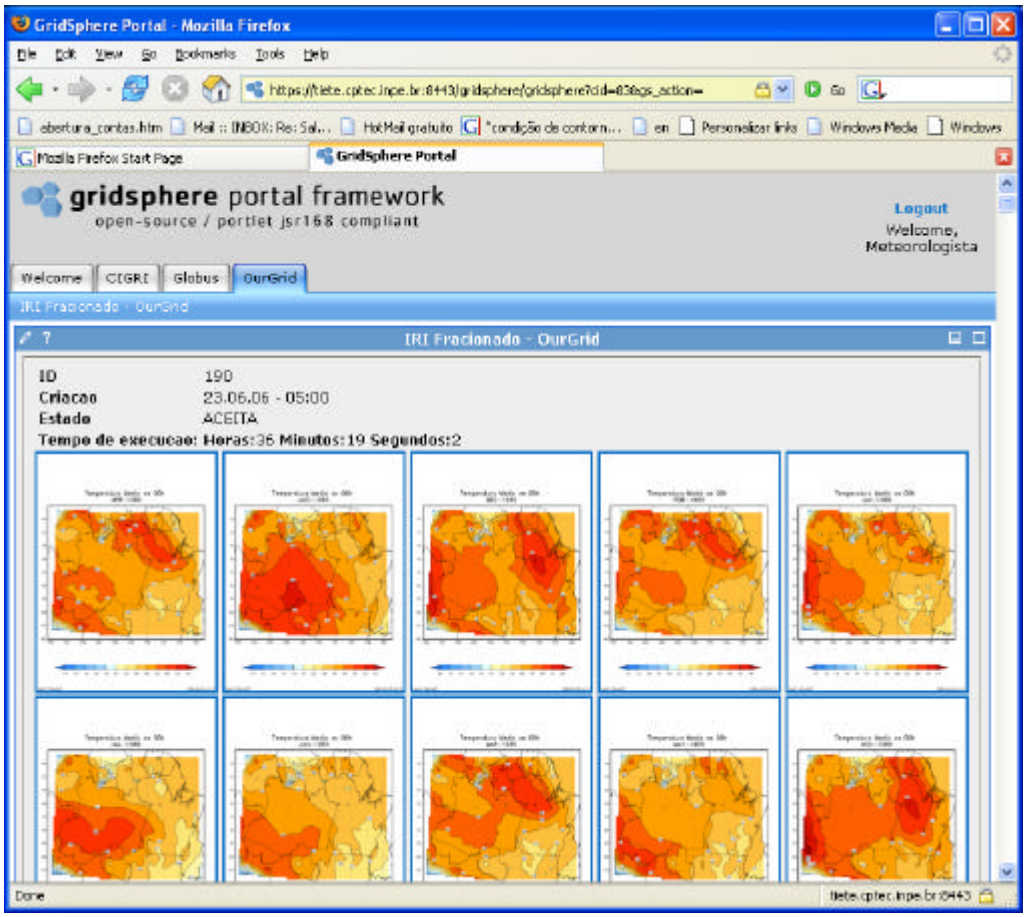


Figura 6.11 – Portal G-BRAMS: interface para visualização e análise dos dados

6.5 Estratégias de escalonamento utilizadas no projeto

A utilização de um único modelo meteorológico com diferentes dados de entrada por um período longo de tempo é uma aplicação classificada como computação intensiva de dados. Neste tipo de aplicação torna-se mais barato pesquisar máquinas ociosas ou com menor carga computacional do que submeter a aplicação a um supercomputador saturado ou investir na compra de recurso computacional que permita executar a aplicação. A grade computacional para este objetivo pode consistir de “clusters” interconectados através de uma conexão confiável.

Na perspectiva do usuário, o mais importante é o desempenho. O desempenho de qualquer aplicação paralela é altamente dependente da forma em que as tarefas são atribuídas aos processadores (balanceamento de carga). Para obter desempenho de uma aplicação, o escalonamento da aplicação é fundamental. Escalonadores são utilizados

para atribuir tarefas e dados aos recursos, com o objetivo de tornar disponível o desempenho potencial daquela plataforma alvo.

Em ambiente de grade, segundo Foster e Kesselman (1999), tanto as aplicações como os componentes devem ser escalonados para atingir o desempenho com sucesso. No entanto, cada mecanismo de escalonamento possui diferentes objetivos de desempenho.

Os **escalonadores de recursos** controlam os recursos do sistema. Eles coordenam requisições múltiplas para o acesso a um dado recurso otimizando critérios de justiça (assegurando que todas as requisições sejam satisfeitas) ou utilização de recursos (para medir o montante de recursos utilizados). Uma característica importante dos escalonadores de recurso é que eles recebem solicitações de vários usuários e, portanto, tem que arbitrar entre estes vários usuários o uso dos recursos que controlam. Por exemplo, o sistema operacional controla o computador no qual roda, decidindo quando e aonde (no caso de multiprocessadores) cada processo executa. Tradicionalmente, há um escalonador que controla os recursos do sistema (i.e., não há como usar os recursos sem a autorização do escalonador). **Escalonadores de tarefas** são componentes de software comumente integrados a sistemas operacionais paralelos e/ou distribuídos e que têm como função a distribuição de trabalho computacional para as unidades de processamento integrantes do sistema, de modo a maximizar o desempenho global do processamento realizado, isto é, promover o balanceamento de carga entre as unidades de processamento envolvidas. Promovem o desempenho do sistema (medido pelo desempenho agregado das tarefas) através da otimização da vazão (medida pelo número de tarefas executadas pelo sistema), também são denominados de escalonadores de alta vazão. Ambos privilegiam o desempenho do sistema ao invés do desempenho das aplicações. Esses objetivos conflitam com os objetivos dos **escalonadores de aplicação** (escalonadores de alto desempenho), que promovem o desempenho da aplicação através da otimização das medidas de desempenho como tempo mínimo de execução, resolução, “speedup” ou outras medidas de custo centradas na aplicação. Para isto, o escalonador de aplicação (i) escolhe quais recursos serão utilizados na execução da aplicação, (ii) estabelece quais tarefas cada um destes recursos realizará, e (iii) submete solicitações aos escalonadores de recurso apropriados para que estas tarefas sejam

executadas. Escalonadores de aplicação não controlam os recursos que usam. Eles obtêm acesso a tais recursos submetendo solicitações para os escalonadores que controlam os recursos (SLTI, 2006).

Como a noção de desempenho difere, usuários de grades não podem acreditar em escalonadores de recursos ou outros componentes do sistema para extrair o desempenho da aplicação, pois tanto os escalonadores de tarefas como os escalonadores de recursos promoverão o desempenho do sistema ao invés do desempenho das aplicações. O objetivo no caso em estudo foi promover o desempenho da aplicação, no caso representada pelo BRAMS, em grades computacionais através da minimização do seu tempo de execução.

No entanto, a tecnologia de grades computacionais possui alguns fatores que podem dificultar a obtenção do desempenho desejado da aplicação:

- Grades computacionais são compostas de computadores com desempenhos distintos e redes de interconexão operando com diferentes bandas e latências;
- Computadores podem estar, ou não, executando outras aplicações, consumindo desta forma ciclos de máquina da CPU;
- Computadores podem ser dinamicamente incorporados ou desincorporados da grade computacional;
- Existência de tarefas na grade exigindo banda da rede;
- A carga é dependente das características dos dados de entrada e não da quantidade de dados. O BRAMS é uma aplicação paralela que possui uma distribuição de carga não previsível (Mendes e Panetta, 1999), pois o tempo de processamento de cada subdomínio é dependente do estado da atmosfera.

As técnicas para escalonamento da climatologia de mesoescala do BRAMS nas plataformas de grade OurGrid, CIGRI/OAR e Globus com duas estratégias distintas de escalonamento são apresentadas com mais detalhes a seguir.

6.5.1 Estratégia para definir tarefas a escalonar

Quando o usuário do portal submete a aplicação para a execução, os estados EDICAO* e EDICAO são atribuídos (no banco de dados) a todas tarefas que possuem dependências a resolver e resolvidas, respectivamente.

As tarefas criadas necessitam de autorização do usuário para execução e passam para o estado LIBERADA. Existe um programa que verifica constantemente quando o portal ou usuário liberou as tarefas para execução.

Este programa foi desenvolvido pelo analista do projeto para resolver o problema de atribuição de tarefas aos nós de grade. Constitui-se de um escalonador de tarefas simples, que se baseia na fila de tarefas armazenadas no banco de dados e na disponibilidade de recursos livres (nós de grades disponíveis).

As tarefas criadas pelo usuário e que se encontram no banco de dados são submetidas à grade ou escalonador da grade, utilizando um algoritmo de escalonamento adaptativo. Inicialmente, este algoritmo submete as tarefas aos nós de grade em uma ordem pré-definida. As submissões seguintes são dependentes da existência de um recurso computacional disponível. Uma vez disponibilizado o recurso, o algoritmo percorre o campo de estados no banco de dados à procura de tarefas com estado LIBERADA. Se existe uma tarefa com estado LIBERADA e existe recurso computacional disponível, a tarefa é submetida para execução utilizando o escalonador ou aplicativo de submissão de cada plataforma de grade.

6.5.2 Escalonamento no OurGrid

Em uma grade composta de um reduzido número de recursos computacionais, a execução da climatologia diretamente no OurGrid pode gerar ineficiência na execução das tarefas na grade e/ou pode saturar com o aumento de usuários.

A estratégia para definir as tarefas a escalonar, apresentada em 6.5.1, utilizada em conjunto com o escalonador do OurGrid, garante a submissão e a execução das mesmas com sucesso numa grade com reduzido número de recursos computacionais. O OurGrid

permite que a escolha da técnica de escalonamento, uma vez que a mesma está fora do escopo do OurGrid, seja feita pelo projetista da solução de grade. O objetivo é escolher entre o algoritmo de escalonamento que melhor otimize a aplicação, de acordo com o conhecimento sobre as características da aplicação.

O algoritmo de escalonamento adotado dentro do projeto G-BRAMS foi o Workqueue original. Este algoritmo não necessita informação para o escalonamento de tarefas. No escopo deste projeto, as tarefas foram selecionadas a partir da sequência definida pelo banco de dados e enviadas aos nós da grade. Ao término do processamento, os resultados são enviados de volta ao nó de grade mestre e o escalonador atribui nova tarefa ao nó da grade. Na Figura 6.13 é apresentado o algoritmo utilizado como estratégia de escalonamento na execução do experimento com a infra-estrutura de grade OurGrid.

Segundo Yu et al. (2005), o desempenho do algoritmo WQR depende do número de máquinas para executar as tarefas. Para um ambiente homogêneo, constatou-se que o algoritmo WQR supera o WQ apenas quando o número de máquinas é duas vezes superior ao número de tarefas. Devido ao número reduzido de nós de grades na grade e ao elevado número de tarefas, optou-se por não utilizar replicação de tarefas do algoritmo.Workqueue original.

O envio dos parâmetros para a execução do BRAMS nos nós de grade e o retorno dos resultados do processamento dos nós de grade é realizado através de um mecanismo seguro para transferência de dados (sftp).

```
while (T ≠ 0)
    if status=(EDICAO E EDICAO*)
        aguarda_liberacao( )
    elseif status=(LIBERADA E LIBERADA_AGUARDANDO)
        recebe_entrada( )
        submete_job( )
    elseif status=(EXECUTANDO)
        espera( )
    elseif status=(CONCLUIDA | CANCELADA |ERRO | NÃO_DEFINIDA)
        envia_saida()
        procura_outro_job( )
    endif
end while
```

Figura 6.12 – Algoritmo de escalonamento no OurGrid

6.5.3 Escalonamento no CIGRI/OAR

Diferentemente do OurGrid, o CIGRI/OAR está preparado para resolver o problema de ineficiência na execução das tarefas na grade e/ou saturação da grade com o aumento de usuários, por possuir escalonador local.

Os mecanismos do sistema de submissão de tarefas do CIGRI/OAR são acionados a partir da estratégia para definir as tarefas a escalonar, apresentada em 6.5.1. Ao verificar uma tarefa com estado LIBERADA ou LIBERADA_AGUARDANDO, um programa gera automaticamente um arquivo “Job Description Language” (JDL) que, como informado anteriormente, possui um campo comum a todos domínios administrativos e campos que são relativos aos nós de grade onde a aplicação será executada. Uma vez identificado um recurso computacional que atenda à aplicação, esta é submetida para execução. O CIGRI possui um mecanismo que regularmente coleta os dados de saídas das tarefas em cada nó de grade e os retorna à máquina que o escalonou.

Em caso de falha da execução da tarefa, o CIGRI/OAR possui mecanismos que possibilita a re-execução da mesma. Na Figura 6.14 é apresentado o algoritmo utilizado como estratégia de escalonamento na execução do experimento com a infra-estrutura de CIGRI/OAR.

O envio dos parâmetros para a execução do BRAMS nos nós de grade e o retorno dos resultados do processamento dos nós de grade é realizado através de um mecanismo seguro para transferência de dados (sftp).

```
while (T ≠ 0)
    if status=(EDICAO E EDICAO*)
        aguarda_liberacao( )
    elseif status=(LIBERADA E LIBERADA_AGUARDANDO)
        cria_arquivo_JDL( )
        recebe_entrada( )
        submete_job( )
    elseif status=(EXECUTANDO)
        espera( )
    elseif status=(CONCLUIDA | CANCELADA | ERRO | NÃO_DEFINIDA)
        envia_saida()
        procura_outro_job( )
    endif
end while
```

Figura 6.13 – Algoritmo de escalonamento no CIGRI/OAR

6.5.4 Escalonamento no Globus

A versão do Globus (3.2) adotada neste projeto não possui mecanismos para gerenciamento de tarefas e para balanceamento de carga em grades computacionais.

Como o Globus não possui escalonador nativo, a submissão de tarefas é feita diretamente com a utilização do programa de execução remota segura (globus-job-run) que acompanha a distribuição, utilizando a estratégia para definir as tarefas a escalonar apresentada em 6.5.1.

O envio dos parâmetros para a execução do BRAMS nos nós de grade e o retorno dos resultados do processamento dos nós de grade é realizado através de um mecanismo seguro para transferência de dados do Globus (GridFtp), também fornecido junto com a distribuição. Na Figura 6.15 é apresentado o algoritmo utilizado como estratégia de escalonamento na execução do experimento com a infra-estrutura de Globus.

```
while (T ≠ 0)
    if status=(EDICAO E EDICAO*)
        aguarda_liberacao( )
    elseif status=(LIBERADA E LIBERADA_AGUARDANDO)
        recebe_entrada( )
        submete_job( )
    elseif status=(EXECUTANDO)
        espera( )
    elseif status=(CONCLUIDA | CANCELADA |ERRO |NÃO_DEFINIDA)
        envia_saida()
        procura_outro_job( )
    endif
end while
```

Figura. 6.14 – Algoritmo de escalonamento no Globus

CAPÍTULO 7

RESULTADOS

A metodologia apresentada foi validada na grade computacional descrita no capítulo 6, utilizando as plataformas de grade CIGRI/OAR, OurGrid e Globus. Para cada infra-estrutura de grade, a metodologia foi executada para três membros e três regiões geográficas diferentes do Brasil, totalizando 27 (vinte e sete) execuções do programa de geração de climatologia mensal.

Cada execução do BRAMS possui 3 fases distintas, que são executadas usando três tipos de opções de execução:

- **MAKESFC:** conversão da resolução da topografia, da temperatura de superfície e tipo de solo à resolução do saída de modelo desejada;
- **MAKEVFILE:** geração das condições de fronteira;
- **INITIAL:** execução do modelo BRAMS (paralela ou seqüencial) a partir dos dados iniciais.

O BRAMS, nos modos MAKESFC e MAKEVFILE, é executado necessariamente em modo seqüencial e o tempo de execução é muito menor que a execução no modo INITIAL. Devido ao fato do modo INITIAL necessitar de maior computação, optou-se por executá-lo em modo paralelo para minimizar o tempo de execução.

Na primeira execução da simulação do BRAMS, notou-se que existia uma limitação no código que não permitia a execução para o período de três anos. A solução encontrada foi utilizar a característica de checkpoint-restart do BRAMS, que permite que a interrupção do programa para posterior execução. Neste caso, o estado da computação é salvo em um arquivo e depois o programa é executado novamente. Para reiniciar a execução do ponto na qual o programa havia sido interrompido, o BRAMS utiliza a

opção de execução HISTORY. Devido a esta limitação, o período de três anos foi divido em quatro execuções do BRAMS (Figura 7.1). Esta divisão necessitou que 108 (cento e oito) tarefas paralelas fossem processadas na grade, 36 (trinta e seis) tarefas por plataforma de grade. A divisão das execuções é apresentada abaixo:

- 01nov1995 a 31dez1995 – 1464 horas (61 dias);
- 01jan1996 a 31dez1996 - 8784 horas (366 dias);
- 01jan1997 a 31dez1997 - 8760 horas (365 dias);
- 01jan1998 a 31dez1998 – 8760 horas (365 dias).

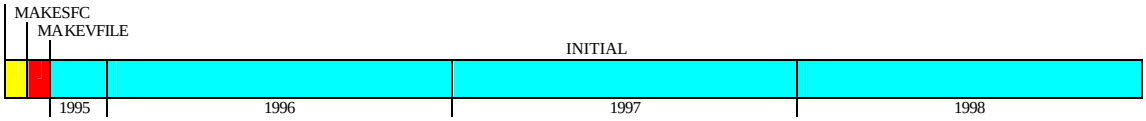


Figura 7.1 – Simulação de três anos do BRAMS

O programa para a geração da climatologia é executado ao término da execução da simulação do BRAMS de cada período de doze meses, gerando climatologias mensais. Esse programa foi executado, para cada ano considerado, em um mesmo nó de grade em instantes de tempo diferentes. Ao final da execução, acontece a transferência dos dados gerados para o nó de grade principal, onde os dados são disponibilizados. Os dados da simulação referente ao período de estabilização do modelo, dois primeiros meses, não foram utilizados para a determinação da climatologia. Uma análise quantitativa e qualitativa dos resultados obtidos neste experimento é apresentada nas seções 7.1. e 7.2. As Figuras 7.2, 7.3 e 7.4 apresentam a climatologia mensal gerada para um dado mês e membro, para as regiões Norte, Nordeste e Sul/Sudeste respectivamente.

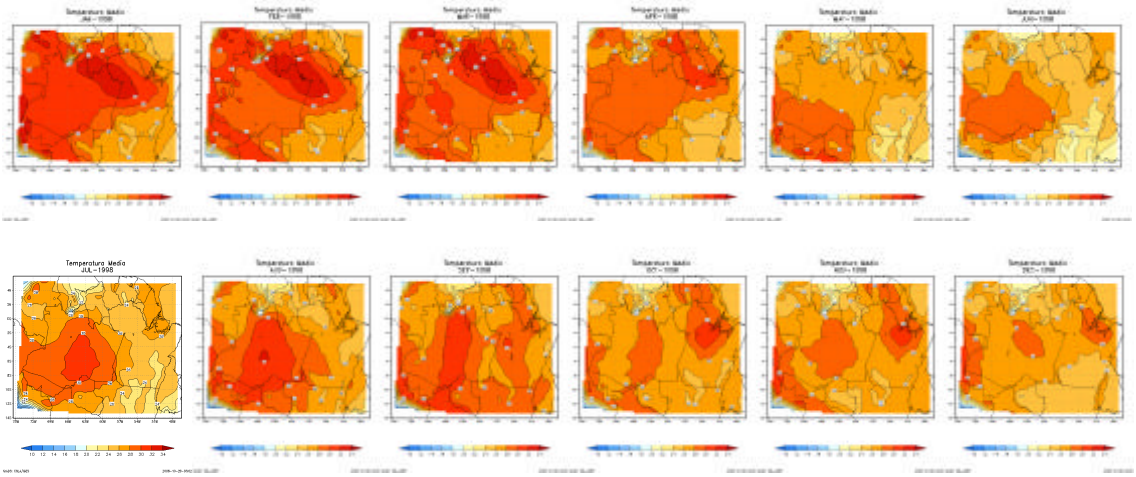


Figura 7.2 – Climatologia de temperatura média - 1998 (Norte).

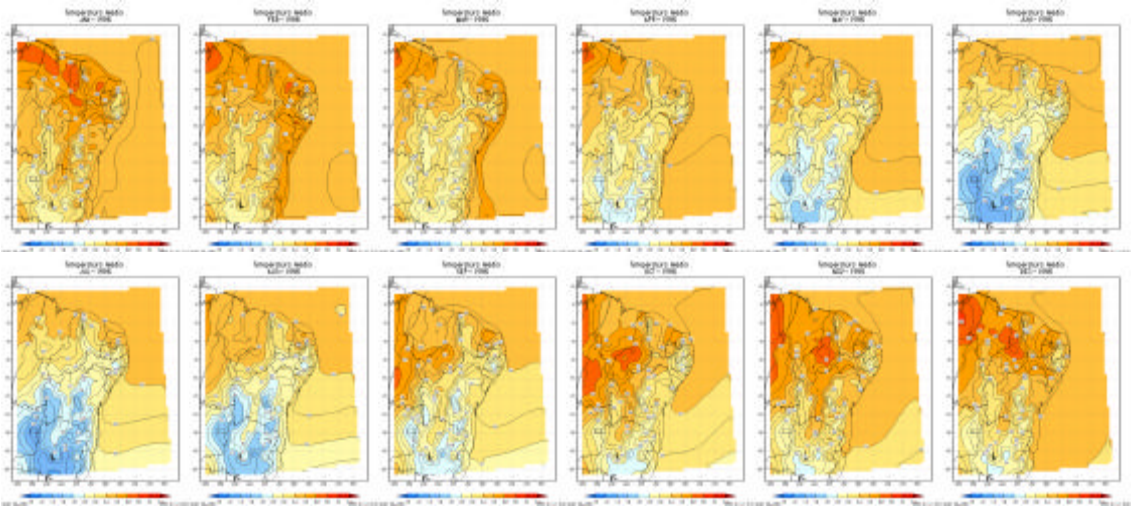


Figura 7.3 – Climatologia de temperatura média - 1996 (Nordeste).

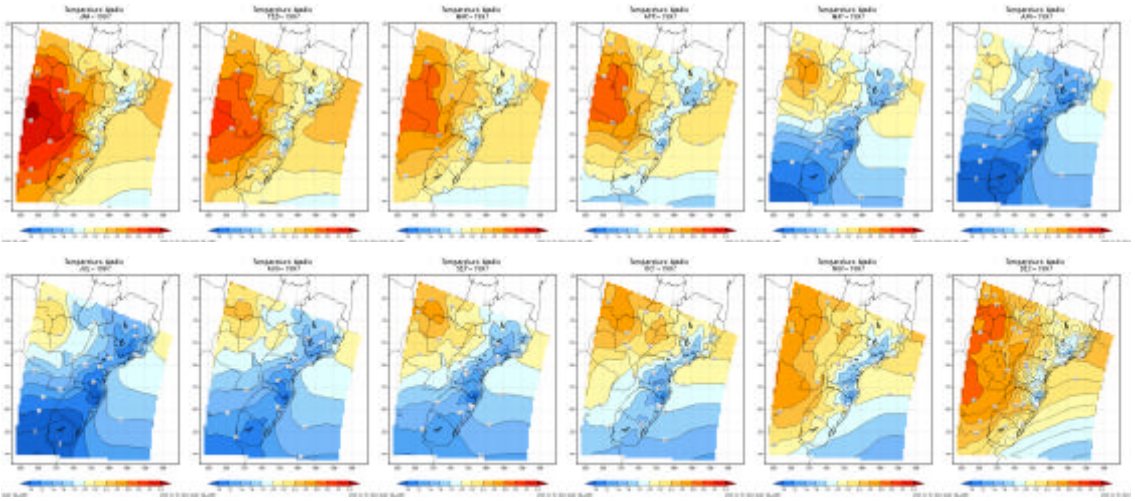
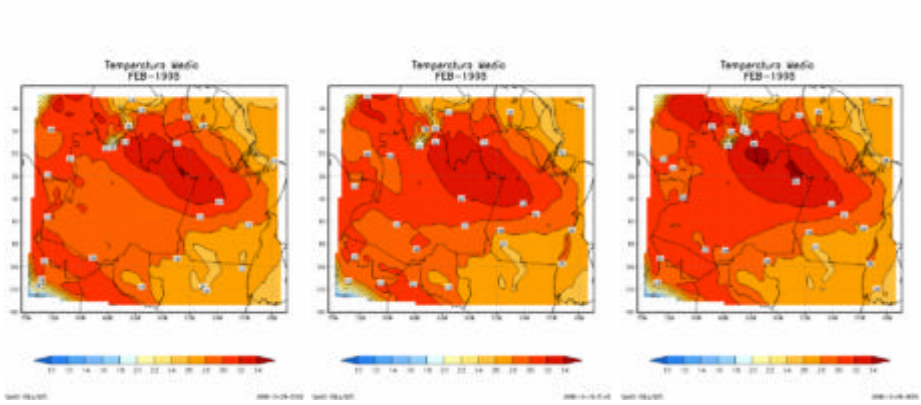


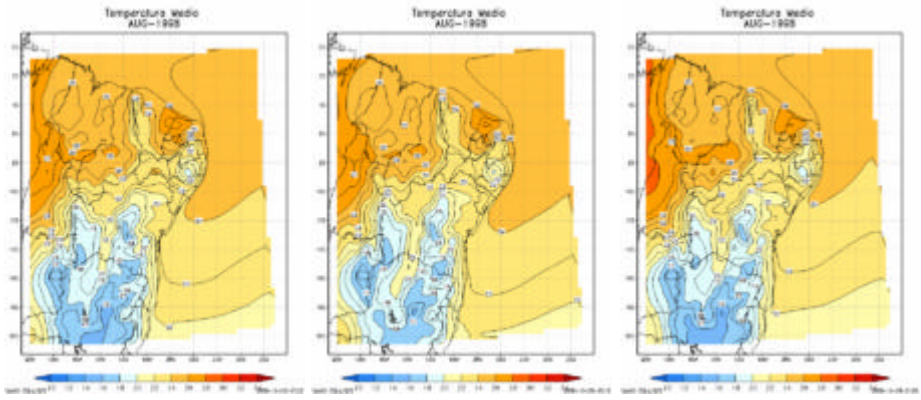
Figura 7.4 – Climatologia de temperatura média - 1997 (Sul/Sudeste).

Cada simulação de três anos da climatologia foi gerada a partir das condições iniciais e de fronteira de um dado membro do modelo global. As diferenças relativas às execuções de cada membro podem ser observadas nas Figuras 7.5, 7.6 e 7.7, que apresentam a climatologia mensal gerada para um dado mês, para as regiões Norte, Nordeste e Sul/Sudeste, respectivamente.



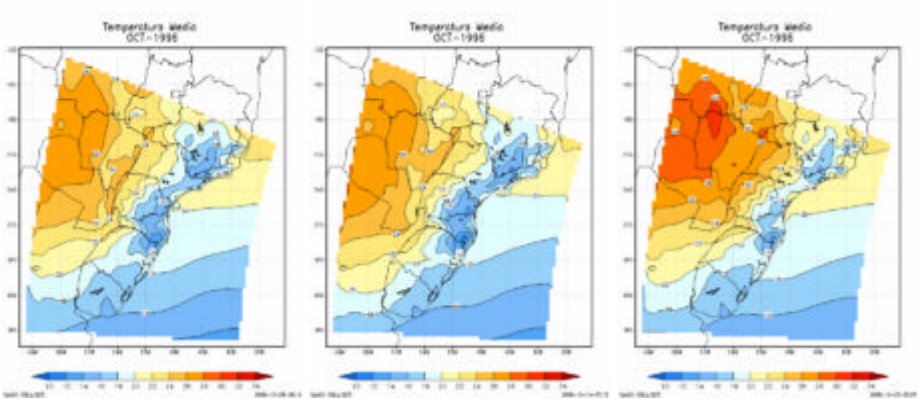
(a) (b) (c)

Figura 7.5 – Climatologia mensal – Fev/1998 (Norte): membros: (a) 1, (b) 2 e (c) 3



(a) (b) (c)

Figura 7.6 – Climatologia mensal – Ago/1998 (Nordeste): membros: (a) 1, (b) 2 e (c) 3



(a) (b) (c)

Figura 7.7 – Climatologia mensal – Out/1997 (Sul/Sudeste): membros: (a) 1, (b) 2 e (c) 3

Para fazer a comparação das três plataformas de grade utilizadas para a determinação da climatologia do BRAMS foi feita uma análise qualitativa e quantitativa. Como parâmetro para análise qualitativa, utilizou-se a resposta da plataforma de grade a uma falha da aplicação ou do nó da grade.

No caso da análise quantitativa, os parâmetros utilizados foram:

- Tempo de execução de cada ano (por região);
- Tempo de execução total da climatologia;
- Tempo de transferência dos dados

7.1 Análise quantitativa das soluções propostas

Os resultados referentes à execução da climatologia foram analisados utilizando como critérios o tempo de execução de cada ano (por região) e do total da climatologia e o tempo de transferência dos dados. Na Tabela 7.1 são apresentados os tempos de execução da climatologia para as regiões Nordeste, Norte e Sul/Sudeste do Brasil, respectivamente, utilizando as diferentes tecnologias de grade adotadas no projeto.

Os dados apresentados representam apenas o tempo de computação da simulação em um dado nó de grade. Apenas para a plataforma de grade Globus o tempo de

transmissão dos dados está inserido no tempo de computação, devido a uma opção de implementação para esta plataforma de grade.

Para o cálculo do tempo total de computação para uma dada plataforma de grade, considerou-se o início da execução da primeira tarefa e o fim da execução da última tarefa. Observou-se tempos de execução de 33,3 dias para o CIGRI/OAR e de 22,3 dias para o Globus e 20,6 dias para o OurGrid. Não se pode afirmar com o resultado dessas medidas que uma plataforma possui um desempenho superior à outra devido aos problemas discutidos na seção 7.1. Verificou-se que devido a esses problemas, as grades apresentaram uma ociosidade de 17,4 dias, 10,1 dias e 4,7 dias para as plataformas de grade CIGRI/OAR, Globus e OurGrid, respectivamente.

Tabela. 7.1 – Desempenho da climatologia paras as regiões NE, N e S/SE (tempo em h:m)

	CIGRI/OAR			**Globus**			**OurGrid**		
	NE	**N**	**S/SE**	**NE**	**N**	**S/SE**	**NE**	**N**	**S/SE**
Membro									
1	6:38	5:15	8:01	6:28	7:26	8:24	4:16	7:14	10:16
	24:35	31:10	46:12	26:30	34:30	32:30	23:41	28:40	30:22
	24:26	30:54	28:26	27:06	32:46	32:34	24:14	29:58	51:58
	67:29	30:44	40:24	26:28	30:40	30:36	24:06	30:50	28:24
2	4:16	5:14	4:43	6:14	7:24	7:48	4:21	5:09	11:57
	38:18	29:23	32:23	25:04	30:26	30:36	24:17	42:19	44:12
	39:02	30:44	28:00	26:46	34:44	30:42	37:25	31:04	44:43
	39:02	30:36	47:52	25:22	34:50	32:12	51:16	90:41	118:49
3	4:14	7:32	4:44	6:34	7:28	7:44	6:01	11:01	7:19
	39:24	32:18	56:13	25:44	29:12	31:58	25:36	31:02	27:08
	24:14	44:05	47:42	26:30	31:54	30:14	24:44	46:54	46:18
	28:46	90:37	92:24	25:42	34:42	31:54	25:00	31:02	59:25
total	340:24	368:32	437:04	254:28	316:02	307:12	274:57	384:54	480:51

Na Figura 7.8 são apresentados os gráficos dos tempos de execução da climatologia na grade para as regiões Sul/Sudeste, Nordeste e Norte do Brasil utilizando as diferentes tecnologias de grade adotadas no projeto.

Na proposta inicial da determinação da climatologia, considerou-se que os arquivos contendo as condições de fronteira estariam localizados em cada um dos discos locais dos nós de grade. Como não foi possível adquirir a quantidade de discos para armazenar os dados localmente em todos os nós de grade, a opção foi armazenar os dados na máquina do portal e enviar os dados para o nó de grade escalonado para execução.

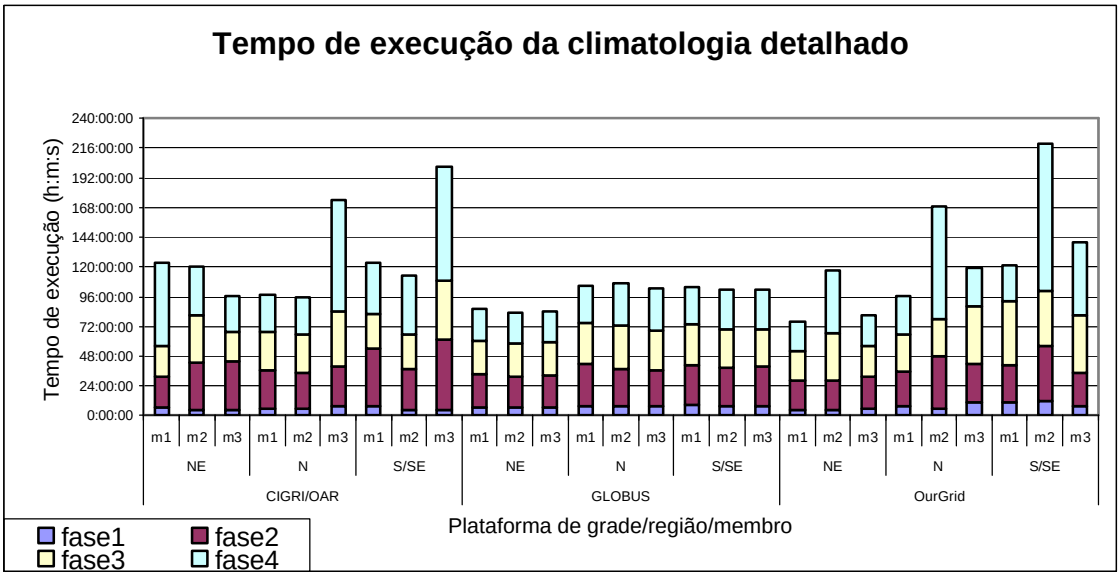


Figura 7.8 – Desempenho da climatologia em grade para as três plataformas de grade.

Na Figura 7.9, observamos que o tempo de execução de cada fase da climatologia no CIGRI/OAR varia entre 24h14m e 92h24m. Neste gráfico é possível verificar os elevados tempos de execução dos “clusters” tatuí e paraytinga, relacionadas as falhas que serão apresentadas na seção 7.2. No caso do CIGRI/OAR, a característica de ressubmissão de tarefas garante a execução da aplicação, em caso de falhas.

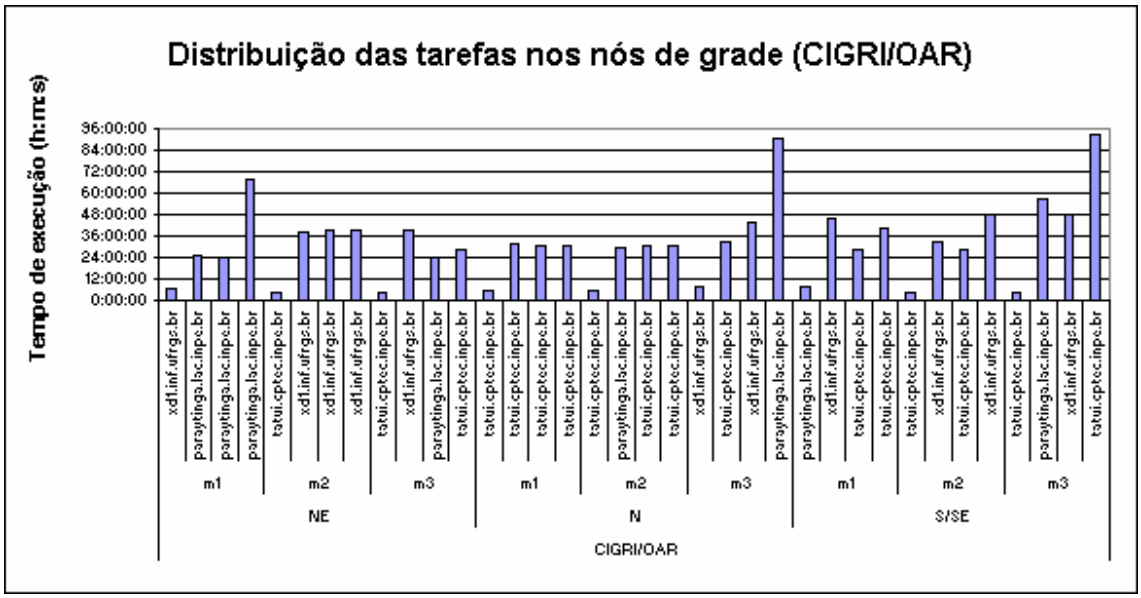


Figura 7.9 – Distribuição de tarefas nos nós de grade para o CIGRI/OAR.

Na Figura 7.10, observamos que o tempo de execução de cada fase da climatologia no OurGrid varia entre 23h41m e 118h49m. Neste gráfico é possível verificar os elevados tempos de execução dos "clusters" tatuí e xd1, ocasionados pelas falhas que serão apresentadas na seção 7.2. No caso do OurGrid, o esquema de ressubmissão de tarefas que está implementado no portal garante a execução da aplicação, em caso de falhas.

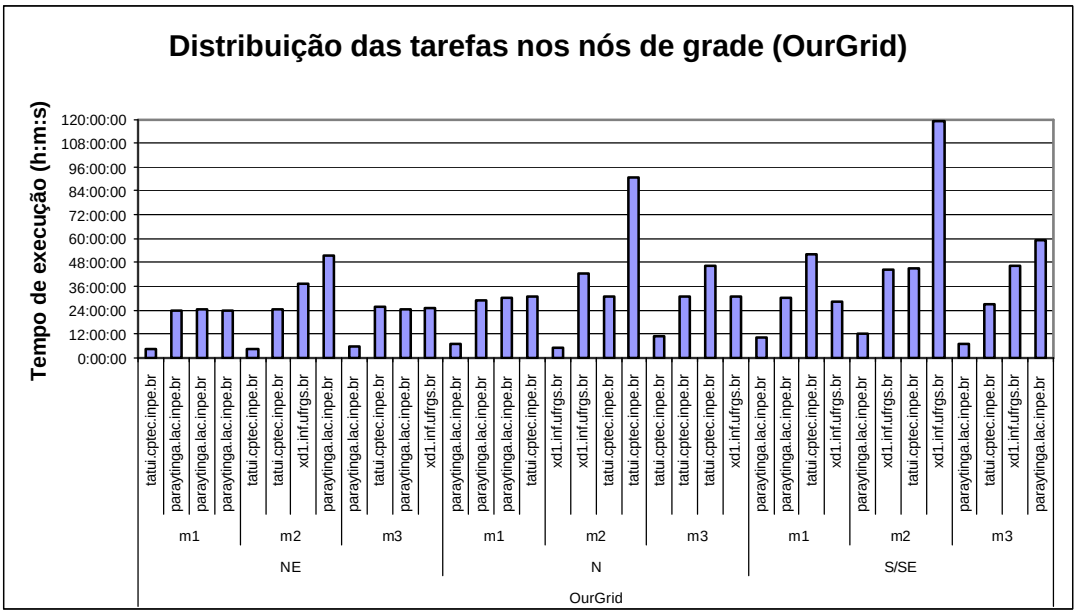


Figura 7.10 – Distribuição de tarefas nos nós de grade para o OurGrid.

Como a plataforma de grade Globus não possui as características de ressubmissão de tarefas, a variação de tempo de execução encontrada (entre 25h04m e 34h50m) está relacionada apenas ao tamanho de grade de cada região (Figura 7.11). A ressubmissão de tarefas é manual e não aparece nos tempos de computação, apenas é computada como tempo de ociosidade da grade.

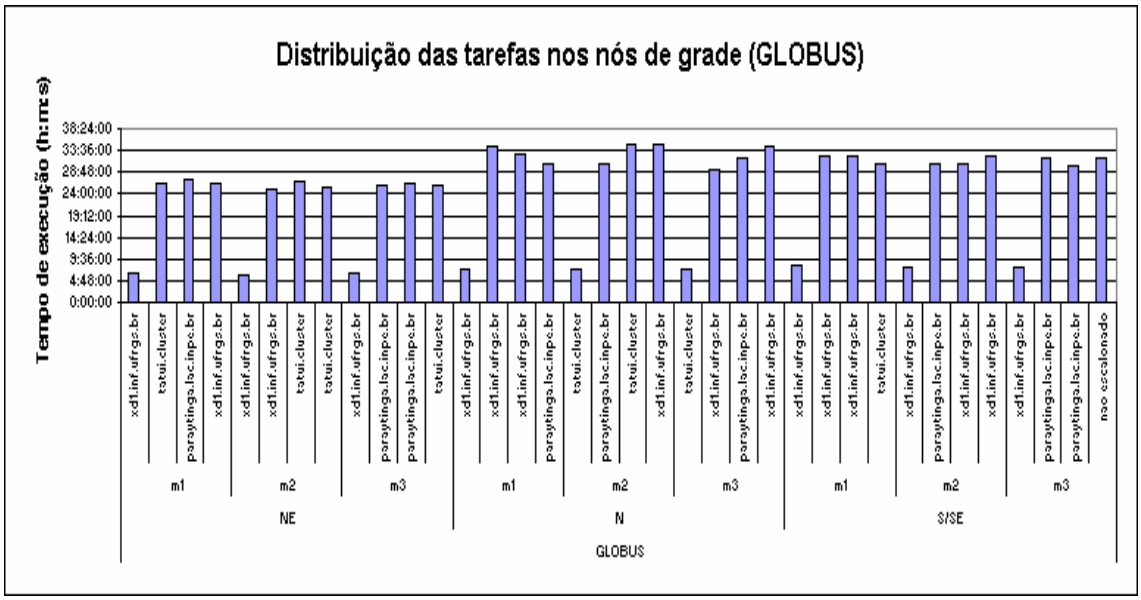


Figura. 7.11 – Distribuição de tarefas nos nós de grade para o Globus.

Neste experimento, as análises do MCGA trafegaram da máquina de portal aos nós de grade e os as climatologias mensais trafegaram dos nós de grade à máquina de portal. sendo a relação de tamanho entre eles de aproximadamente 12:1. A Tabela 7.3 apresenta o tamanho de cada dado, relativo a um ano, trafegado na rede durante o experimento.

Tabela. 7.2 – Tamanho dos dados trafegados na grade

Enviados (Análises)	**Recebidos (Climatologia)s**		
	Norte	***Sul/Sudeste***	***Nordeste***
2033853 KB	175136 KB	166468 KB	154796 KB

Observando a Tabela 7.3, nota-se que os tempos de transmissão das análises são muito superiores aos tempos de transmissão dos dados de climatologia. No melhor caso, o

tempo de comunicação pode representar apenas 1% do tempo de computação, enquanto que no pior caso pode chegar a 12% do tempo de computação.

A variação encontrada nos tempos de transmissão de dados da máquina do portal aos nós de grade e de recepção de dados dos nós de grade pela máquina do portal (Tabela 7.3) é devido a dois fatores:

- Tamanho das análises que são enviadas aos nós de grade e o dos dados, resultantes do processamento da climatologia em cada grade, recebidos dos nós de grade;
- Variabilidade do tráfego de dados na Internet ao longo do dia.

Tabela 7.3 – Tempo de transferência dos dados entre portal e nós de grade (em h:m:s)

Máquina	**Protocolo**		**Envia**	**Recebe**		
				Norte	**Nordeste**	**Sul/Sudeste**
Tatuí (local) INPE/ CPTEC	SFTP	mínimo	00:14:42	00:01:59	00:02:07	00:02:08
		máximo	00:27:20	00:06:15	00:05:22	00:05:01
	GSIFTP	mínimo	00:14:29	00:02:48	00:03:23	00:02:57
		máximo	00:18:00	00:07:01	00:06:33	00:03:09
paraytinga INPE/LAC	SFTP	mínimo	00:17:00	00:01:43	00:01:41	00:01:42
		máximo	00:46:19	00:25:15	00:03:38	00:04:09
	GSIFTP	mínimo	00:22:30	00:03:04	00:03:17	00:03:10
		máximo	00:51:26	00:05:13	00:04:31	00:04:56
xd1 II/UFRGS	SFTP	mínimo	00:34:19	00:03:25	00:05:31	00:03:20
		máximo	03:00:27	00:07:47	00:09:5	00:04:36
	GSIFTP	mínimo	00:59:20	00:06:26	00:07:29	00:06:34
		máximo	03:33:34	00:29:11	00:32:56	00:27:44

Verificando a Tabela 7.3, nota-se que as variações do tempo de comunicação das análises são muito maiores entre xd1-portal do que entre tatui-portal e paraytinga-portal. A mesma tabela indica que os tempos de comunicação entre tatui-portal e paraytinga-portal são bastante próximos, com um ligeiro aumento da variação do tempo de comunicação entre paraytinga-portal.

O gráfico da Figura 7.12 apresenta a distribuição de tarefas em relação as plataformas de grade (CIGRI/OAR, Globus e OurGrid) por nó de grade. O experimento realizado com a grade configurada com o Ourgrid foi o que obteve maior sucesso, constatado pela

menor ociosidade desta grade. O fato do “cluster” xd1 ter executado um menor número de tarefas, está fato está associado aos maiores tempos e as maiores variações de tempo de comunicação entre xd1-portal (Tabela 7.3). Na figura 7.12, verifica-se uma variação do número de tarefas executadas por cada nó de grade para as grades configuradas com CIGRI/OAR e Globus, que foram ocasionadas pelos motivos que serão discutidos na seção 7.2. No experimento com a plataforma de grade CIGRI/OAR, o “cluster” paraytinga deveria apresentar um número de tarefas processadas superior ao do “cluster” xd1, fato que não ocorreu devido a problemas apresentados pelo primeiro. No experimento com o Globus, um conjunto de fatores, também citados na seção 7.2, levou que o número de tarefas processadas pelo “cluster” tatuí fosse inferior aos “clusters” xd1 e paraytinga, apesar dos dados estarem disponíveis localmente. A infra-estrutura de grade Globus apresentou um aumento de até 17% do tempo de execução com relação as infra-estrutura de OurGrid e CIGRI/OAR, devido ao fato que no tempo de computação esta considerado o tempo de transferência de dados.

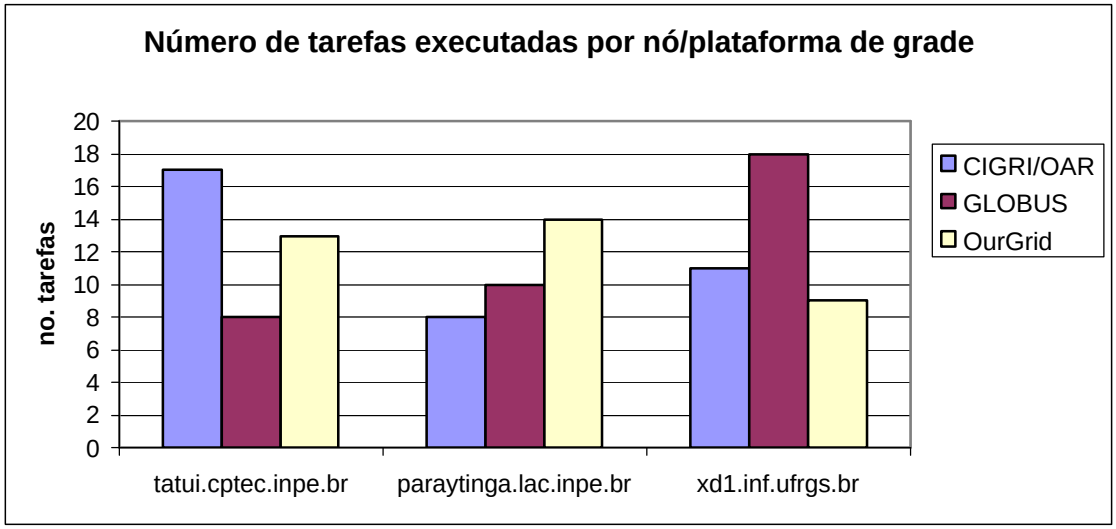


Figura 7.12 – Número de tarefas executadas por nó de grade.

7.2 Análise qualitativa das soluções propostas

Para a execução da climatologia do G-BRAMS, dois tipos de grade podem ser utilizados:

- **Grade dedicada:** permite apenas a execução de aplicações através da grade. Essas aplicações são previamente disponibilizadas através de um portal e não concorrem com aplicações locais para uso dos recursos computacionais. O portal para a submissão das tarefas poderá ser dedicado ou aberto:
    - **Portal dedicado:** uma ou mais aplicações são disponibilizadas através de programação prévia do portal. O escalonamento é mais simples devido ao reduzido número de aplicações;
    - **Portal aberto:** o portal é programado para atender qualquer tipo de aplicação. A complexidade do escalonamento aumenta devido à diversidade de aplicações;
- **Grade aberta:** permite a submissão de diferentes tipos de aplicação, podendo os usuários de grade concorrer com usuários locais. Na grade aberta, a submissão das tarefas possa ser realizada tanto localmente como através da grade, ambas concorrendo para a utilização dos recursos computacionais. A complexidade do escalonamento neste tipo de grade é muito alta.

A estratégia de escalonamento de tarefas adotada considera que a grade é dedicada e com a utilização de um portal dedicado. Este tipo de estratégia privilegia a aplicação, pois garante que as execuções do BRAMS no menor tempo possível, garantindo que o desempenho da aplicação seja atingido. O algoritmo de escalonamento prevê que uma dada execução do BRAMS será submetida aos nós de grade quando da detecção de um nó de grade sem utilização.

Como verificado anteriormente, a aplicação G-BRAMS é dividida em tarefas que possuem dependência entre si. As tarefas criadas apresentam o estado EDICAO (com dependências resolvidas) e EDICAO* (com dependências a resolver). Apenas as tarefas que possuem dependências resolvidas (estado EDICAO) podem ser submetidas à grade. A partir desta premissa, duas estratégias de escalonamento foram desenvolvidas para as plataformas de grade analisadas:

- **CIGRI/OAR e OurGrid:** necessitam de interação com o usuário com relação à dependência das tarefas. Ao término da criação das tarefas no portal, as tarefas passam para o estado EDICAO e EDICAO*. As tarefas em estado EDICAO estão aptas à execução, pois não possuem dependência. As tarefas com estado EDICAO* são dependentes da tarefa anterior e somente passam ao estado EDICAO após a conclusão da tarefa anterior e aceitação do resultado gerado. A tarefa finalizada altera para o estado CONCLUIDA e a tarefa dependente para o estado EDICAO. Novamente o usuário deve interagir com o portal, selecionando a tarefa e submetendo-a para a execução. Para cada plataforma de grade, temos inicialmente nove tarefas a serem escalonadas.

- **Globus:** A estratégia de escalonamento empregada no CIGRI/OAR e OurGrid foi aperfeiçoada, não necessitando de interação com o usuário após a criação e submissão de tarefas. O escalonador foi modificado de modo a entender as dependências das tarefas e alocar as tarefas aos nós de grade quando da identificação de ociosidade. Desta forma, as tarefas que possuem dependência passam para o estado EDICAO, assim que a tarefa da qual eram dependentes é concluída (recebendo o estado de CONCLUIDA), tornando aptas a serem escalonadas aos nós de grade. No entanto, de modo a facilitar a validação da climatologia pelo meteorologista, a mudança do estado CONCLUIDA para LIBERADA é feita interativamente.

As estratégias desenvolvidas para o CIGRI/OAR e OurGrid utilizam o sftp para transferência de dados de entrada, de saída e de estado da computação. Na estratégia do Globus, o gsiftp é utilizado para esta finalidade. As estratégias utilizadas podem ser empregadas apenas com grades dedicadas.

No CIGRI/OAR e no OurGrid a transferência dos dados de entrada da simulação, localizados na máquina de portal, somente é realizada quando a tarefa é liberada manualmente para execução em um dado nó de grade. O processo de liberação de tarefas no Globus é automático, logo a transferência dos dados da máquina de portal a um dado nó de grade é realizada por demanda. A transferência de dados processados de

climatologia (saída) só é iniciada após o aceite dos resultados pelo usuário do portal (recebendo o estado ACEITA).

A grade dedicada tem a vantagem de garantir que a aplicação seja executada no menor tempo possível. Como desvantagem, pode-se ter uma grade ociosa por falta de usuários/aplicações em um dado período de tempo.

A estratégia de escalonamento proposta utiliza o GLOBUS como plataforma de grade e pode ser empregada tanto em grades dedicadas como abertas. Para isto, prevê que em cada nó de grade exista um escalonador de tarefas que será responsável pelo controle das execuções localmente.

A grade dedicada com portal aberto ou aberta possui a vantagem do uso mais racional dos recursos computacionais em detrimento da aplicação. Sua desvantagem é não garantir ao usuário o momento da alocação de recursos às tarefas, e como conseqüência o momento de início e/ou término da execução da aplicação.

Comprovou-se o funcionamento adequado da grade proposta para execução das tarefas a ela endereçadas dentro do escopo do projeto. No entanto foram verificados alguns problemas que elevaram o tempo de execução (seção 7.1) da climatologia.

O sistema operacional fornecido pela Itautec (Linux Fedora Core), apresentou um problema de deixar indisponível um de seus nós durante a execução da climatologia. Este problema levou a tempos de execução altos no CIGRI/OAR, devido às tentativas de re-submissão das tarefas por parte desta plataforma de grade. No caso do Globus, este problema levava ao término da execução da tarefa submetida e das seguintes que estavam liberadas pelo portal para escalonamento neste nó. Este problema, ocasionado pela instabilidade do sistema operacional, pode ser resolvido substituindo o mesmo por uma versão mais recente que não apresente este problema ou trocando-o por uma distribuição que apresente uma confiabilidade maior.

A falta de energia elétrica para alimentar um determinado nó de grade foi outro fator de indisponibilidade de um nó da grade. Um fim de semana sem energia elétrica significa

um aumento de aproximadamente 5% no tempo de execução da climatologia. Este problema é mais difícil de resolver pois implica na viabilização de uma infra-estrutura elétrica de alta disponibilidade que possui um custo alto.

Outro fator que levou a indisponibilidade da grade durante o experimento **foi à expiração dos certificados** necessários para o funcionamento do Globus. Para resolver este problema, novos certificados para os nós de grade tiveram que ser gerados e assinados pela autoridade certificadora. Soluções mais definitivas seriam: (i) emitir um certificado com maior duração (no caso do projeto G-BRAMS foi de 1 ano), (ii) adotar uma estratégia automática, em que antes da data de expiração do certificado, o sistema gera uma mensagem para alguma agente certificador que renovará o certificado por um novo período (poder-se-ia realizar tal atividade pelo grupo local, ou ainda contatar os desenvolvedores do Globus).

A inconsistência do dado transmitido ou recebido pelo/do nó de grade pode levar a não execução da próxima tarefa ou a indisponibilização dos resultados. Uma forma de solucionar este problema é incluir mecanismos para verificação de dados, utilizando um código enviado na transmissão. No Unix, existe uma ferramenta chamada "cksum" que gera tanto um CRC de 32 bit como um contador de byte para qualquer arquivo. “Cyclical Redundancy Check” (**CRC**) é um método de correção de erros, onde é enviada uma quantidade relativamente grande de dados e em seguida os bits de verificação. Encontrado algum erro, todo o pacote de dados precisa ser retransmitido. Um erro de CRC significa justamente que, por qualquer motivo, os dados estão chegando corrompidos ao destino.

A falha em um dos nós do nó de grade devido a problemas de disco, memória, rede de comunicação de dados, etc. é um problema mais difícil de ser endereçado, pois se trata de problema físico que requer a intervenção humana para solucionar o problema.

Falhas nos componentes de software, como o escalonador e o portal também foram fatores que colaboram para o aumento no tempo de execução da climatologia. O escalonador é um processo que está sempre em execução na máquina do portal. Algumas vezes este processo não estava ativo por motivos desconhecidos, que

ocasionava a não submissão das tarefas aos nós de grade. Para endereçar este problema, o portal pode verificar se este processo está ativo e, no caso de detecção de inatividade ativá-lo.. Com relação ao portal, houve falhas na apresentação das páginas de acesso à grade computacional que até o momento não foi possível detectar as causas. O mecanismo encontrado para resolução foi instalar a versão cópia do portal.

Muitas das falhas verificadas podem ser minimizadas com o monitoramento constante do estado da grade computacional, que pode agilizar o processo de detecção do ponto de falha. A exata detecção do ponto de falha é muito importante para agilizar o restabelecimento do problema. Para isto as informações devem ser coletadas e disponibilizadas no portal, que pode ser feita com utilitários do Linux ou com a incorporação de ferramentas de monitoramento de domínio público disponíveis na Web. Num primeiro momento os problemas podem ser resolvidos com a atuação humana, desde que os procedimentos para o restabelecimento do sistema sejam conhecidos. No entanto é altamente desejável que os problemas possam ser resolvidos sem a necessidade de atuação humana. Mecanismos para isolar um determinado nó de um "cluster" também podem ser utilizados para minimizar o problema. Uma reavaliação do ambiente de grade encontra-se em andamento visando tornar o ambiente mais tolerante a falhas. Os pontos de falhas estão sendo analisados e melhorias serão propostas.

Com base nas constatações acima relacionadas, conclui-se que é difícil determinar uma política de escalonamento otimizada *a priori*, pois os eventos ocorrem de forma não determinística. Neste caso, uma política adaptativa de escalonamento é mais indicada.

A metodologia empregada para a determinação da climatologia em modo "ensemble" utiliza como condição inicial do BRAMS as análises do MCGA com datas diferentes. A utilização de modelos diferentes para a geração da climatologia é uma outra maneira de calcular a climatologia que pode ser explorada, pois permite que as melhores características desses modelos possam ser extraídas.

Krishnamurti et al. (2000) verificaram que a melhoria da previsão de trajetórias de furacões e precipitação pode ser obtida com o conceito de "multimodel ensemble", que consiste na melhoria das previsões através da combinação de previsões individuais,

utilizando métodos estatísticos apropriados. Yun et al. (2003) descobriram que o sistema de regressão linear é o método que fornece a escolha ótima de parâmetros para combinar as previsões numéricas disponíveis.

Um exemplo da utilidade deste novo paradigma é o esforço mundial (TIGGE – "THORPEX Interactive Grand Global Ensemble") conduzido pela "World Meteorological Organization" (WMO) para acelerar as melhorias na precisão das previsões de tempo de 1 a 14 dias, utilizando previsões "ensemble" geradas rotineiramente em diferentes centros meteorológicos do mundo.

7.3 Nova proposta de escalonamento no Globus/G-BRAMS

A partir de experimentos na grade do projeto G-BRAMS, constatou-se a ocorrência de eventos que diminuem o desempenho da grade, devido ao não funcionamento de alguns nós da grade. Tais eventos são devidos a várias causas, descritas na Seção anterior, como por exemplo: queda de energia – que torna o nó da grade não disponível, interrupção do acesso pela internet, entre outras. Assim, ficou evidente que uma estratégia de escalonamento deveria levar em conta eventos desta natureza. Nesta Seção será proposta uma nova técnica para o escalonamento de tarefas, voltada para a aplicação de determinação da climatologia do modelo BRAMS em grades computacionais. A proposta de novo escalonamento visa a concepção de uma grade com maior número de usuários e recursos computacionais. A política de escalonamento de tarefas proposta visa maximizar a utilização dos recursos computacionais da grade, garantindo a execução das aplicações através de uma distribuição a mais justa possível da carga (tarefas) entre os nós de grade.

O escalonamento apresentado para as plataformas de grade OurGrid, CIGRI/OAR e Globus parte da premissa que somente a aplicação climatologia estará sendo executada nos nós de grade. Este modelo funcionou de maneira adequada para as três plataformas de grade, uma vez que a grade era dedicada para a aplicação. A submissão das tarefas foi feita diretamente aos nós de grade, sem a utilização de escalonamento local e sem o conhecimento do desempenho de cada nó de grade.

Duas características do escalonamento atual levaram a uma reflexão sobre esta nova proposta de escalonamento. Numa grade com muitos usuários, aplicações e recursos computacionais, existe a possibilidade de escolha de um recurso computacional menos indicado para o escalonamento devido:

- A não utilização de informações dos recursos computacionais para o escalonamento;
- A sobrecarga de um nó de grade, devido à inexistência de escalonamento local em ambientes com muitos usuários/aplicações.

Na prática verifica-se que a utilização de "clusters" sem gerenciadores de tarefas pode levar ao esgotamento dos recursos computacionais (tamanho de memória, uso de CPU, etc.) ou à perda de desempenho das aplicações com o aumento dos usuários ou de aplicações.

Para o caso do experimento reportado neste trabalho, poderia ocorrer uma diminuição do desempenho da aplicação, com a configuração das plataformas de grade OurGrid, CIGRI/OAR e Globus utilizadas neste experimento, caso outras aplicações estivessem sendo executadas ao mesmo tempo nos nós de grade.

O problema da disputa das aplicações de grade pelos recursos computacionais locais leva a uma diminuição de desempenho das aplicações. Para o caso de uma grade dedicada com portal aberto ou grade aberta, apenas as plataformas de grade CIGRI/OAR e Globus resolveriam este problema, por disponibilizarem escalonadores locais. O CIGRI/OAR apresenta a desvantagem de priorizar as tarefas locais em detrimento das tarefas de grade, característica que pode ser alterada com o apoio dos desenvolvedores da plataforma. No caso do OurGrid, não se tem conhecimento, até o momento, de mecanismo para resolver o problema do escalonamento local neste tipo de grade para a aplicação alvo deste estudo. Desta forma, a plataforma de grade Globus é a que melhor resolve o problema de escalonamento local. Apesar de ser um sistema complexo, ele é bem aceito na comunidade de grade. Fato que levou ao

desenvolvimento paralelo de ferramentas complementares ao Globus por diversas instituições.

Outro fator analisado é a configuração das grades de produção, onde as tarefas só podem ser atribuídas aos nós através de gerenciadores de tarefas. A premissa de escalonamento do experimento, de que apenas as tarefas da simulação estejam sendo executadas nos nós de grade, não é fato verdadeiro em grades de produção.

Também foi observado que a análise do MCGA, a ser transmitida a um nó de grade que irá processar a climatologia, para um dado ano/membro de todas as regiões é a mesma. Constata-se também que devido à transmissão de dados, o tempo de ociosidade da grade foi de 22,8% (OurGrid) para o melhor caso. Essas constatações mostram que é necessário otimizar o tempo de transmissão dos dados para minimizar o tempo total de computação da climatologia.

Duas abordagens para solucionar este problema foram consideradas neste trabalho:

- Executar a computação considerando a localização do dado para definir o local de processamento;
- Enviar o dado enquanto o nó de grade estiver processando o dado anterior.

Isto indica que é necessário encontrar uma alternativa para minimizar esta ociosidade e evitar o problema da disputa das aplicações pelos recursos computacionais. A abordagem que será apresentada a seguir considera este fator, assim como o desempenho dos nós de grade. Apesar da sua complexidade, o Globus foi escolhido como infra-estrutura de grade devido a sua flexibilidade e modularidade. A premissa colocada neste trabalho considera que cada nó de grade seja homogêneo.

Para a determinação da climatologia de um determinado número de anos (num_anos) de uma dada região/membro, o número de tarefas com dependências resolvidas e com dependências a resolver (Td) é dado por:

$$Td = num\_anos + 1 \qquad (7.1)$$

Logo o total de tarefas da climatologia para um dado número de membros (num_membros) e para um determinado número de regiões é dado por:

$$T = Td * num\_membros * num\_regioes \quad (7.2)$$

Uma das primeiras etapas na implementação de uma metodologia de distribuição dinâmica de carga é a descoberta dos recursos computacionais disponíveis da grade. No Globus existe um sistema de informações baseado em LDAP (MDS), que coleta e armazena informações estáticas dos nós da grade: memória, clock da CPU, etc..

Uma segunda etapa é ordenar e selecionar os recursos computacionais, baseando no tamanho da fila e no desempenho de cada nó de grade. O desempenho de um dado nó de grade pode ser modelado em função do número de processadores que compõem uma determina fila e da velocidade de processamento do processador. Logo, o modelo de desempenho de um nó de grade (MD) pode ser determinado através da Equação 7.3.

$$MD = np \times vc \quad (7.3)$$

onde:

np = número de processadores da fila

vc = velocidade computational (MFlops aproximado de um benchmark simples).

Utilizando o modelo de desempenho dos nós de grade e considerando o número de tarefas submetidas à fila do nó de grade em questão, os nós de grade são ordenados. A aplicação é submetida ao gerenciador de tarefas do nó de grade classificado como de melhor desempenho e o processo de escalonamento é reiniciado. Após cada submissão da aplicação a um dado nó de grade, o processo se repete com a ordenação dos nós de grade e a nova submissão. O algoritmo para a submissão das tarefas à grade computacional é apresentado na Figura 7.13.

Optou-se por utilizar o “Sun Grid Engine” (SGE) como gerenciador de tarefas por ser gratuito e ser bastante difundido. O SGE, anteriormente conhecido como “COmputing in DIstributed Networked Environments” (CODINE) ou “Global Resource Director” (GRD), é um software livre para gerenciamento de tarefas suportado pela Sun

Microsystems. A Sun também comercializa um produto baseado no SGE: N1 Grid Engine (N1GE).

Softwares como SGE são tipicamente utilizados em “clusters” e são responsáveis por aceitar, escalonar, lançar e gerenciar a execução remota de um grande número de tarefas de usuários independentes, paralelas ou interativas. Ele também gerencia e escalona a alocação de recursos distribuídos, como processadores, memória, espaço de disco e licenças de software.

O envio dos parâmetros para a execução do BRAMS nos nós de grade e o retorno dos resultados do processamento dos nós de grade é realizado através de um mecanismo seguro para transferência de dados do Globus (GridFtp). Na Figura 7.13 é apresentado o algoritmo de escalonamento do Globus/G-BRAMS.

```
while (T ≠ 0)
    if status=(EDICAO | EDICAO*)
        aguarda_liberação ( )
    else if status = (LIBERADA | LIBERADA_AGUARDANDO)
          nome_maq=determina_maq_grade(MDS_list)
          numero_maq=size(MDS_list)
          for (i=1, numero_maq)
                MD=model_desempenho(nome_maq)
          end for
          for (i=1, numero_maq)
                tam_fila(nome_maq)
          end for
          if (análise nao enviado)
            ordena (MD, tam_fila, dist)
            recebe_entrada ( )
            envia_portal_nograde ()
          endif
          submete_job( )
    else if status = (EXECUTANDO)
            espera( )
    else if status = (CONCLUIDA | CANCELADA | ERRO)
            envia_nograde_portal ( )
            procura_outro_job ( )
    endif
end while
```

Figura 7.13 – Algoritmo de escalonamento proposta com o Globus

Este algoritmo evita a sobrecarga dos nós de grade devido ao fato que o escalonador local libera apenas uma aplicação para execução. O fluxograma do algoritmo de escalonamento proposto para Globus para a submissão das tarefas à grade computacional é apresentado na Figura 7.14.

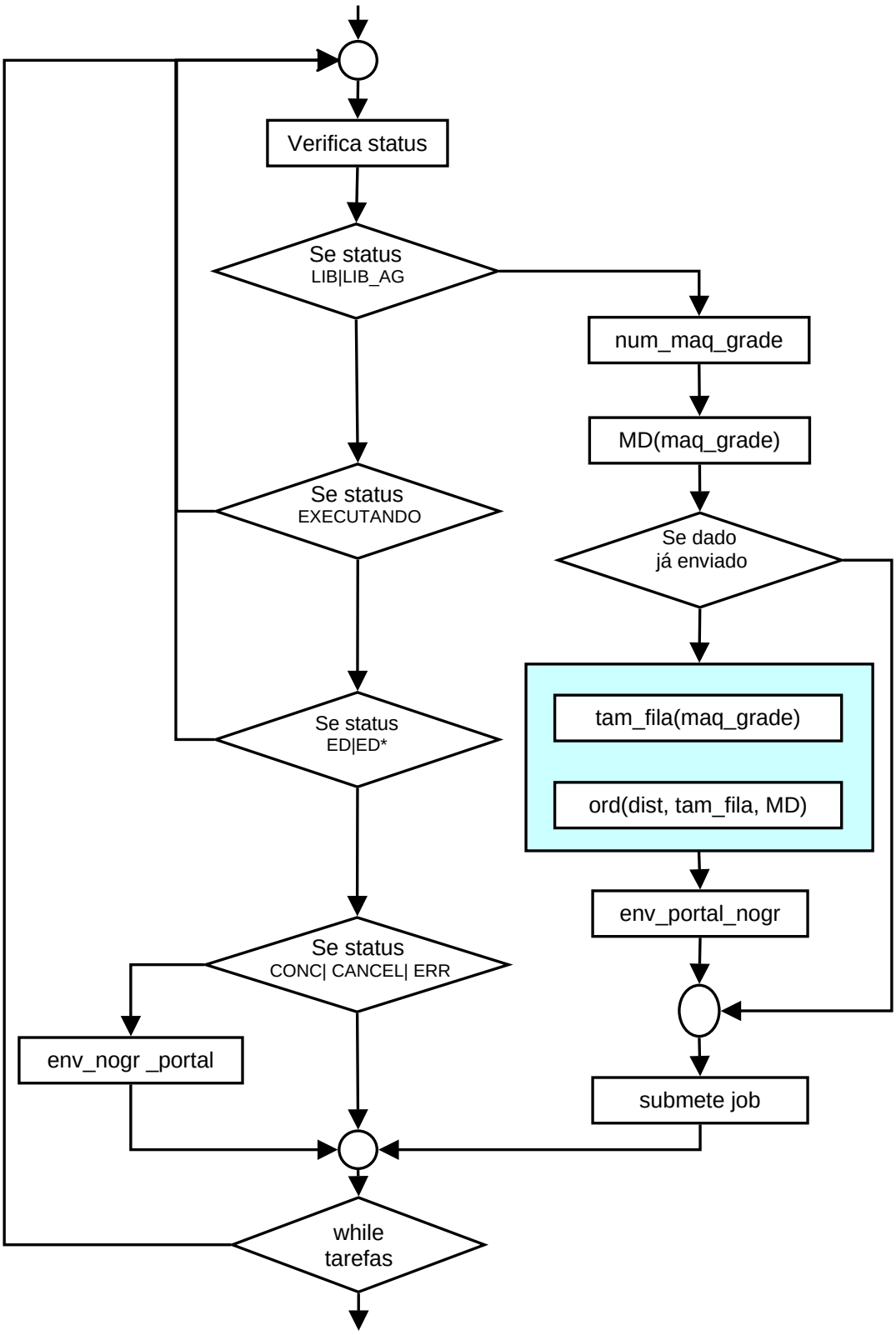


Figura. 7.14 – Fluxograma do algoritmo de escalonamento no Globus

CAPÍTULO 8

CONCLUSÕES

Uma das principais motivações deste trabalho foi analisar um ambiente que serve de embrião para a utilização de aplicações meteorológicas em grades computacionais, nos quesitos desempenho e tolerância à falhas.

Verificou-se que a estratégia de escalonamento adotada atingiu os objetivos do projeto G-BRAMS, funcionando como proposto. Esta estratégia de escalonamento é adequada para uma grade dedicada e com um portal dedicado pois garante a execução da climatologia no menor tempo possível, mesmo considerando as diferentes características das plataformas de grade:

A partir de uma análise qualitativa da grade, a primeira conclusão é que a grade deve ser dedicada com portal aberto para permitir a inclusão de novas aplicações no portal e uso por mais de um usuário.

Os tempos de execução da climatologia de 3 anos foram de 33,3 dias, de 22,3 dias e 20,5 e os tempos ociosidade das grades de 17,4 dias, de 10,1 dias e de 4,7 dias, obtidos neste experimento para as plataformas de grade CIGRI/OAR, Globus e OurGrid respectivamente. O experimento demonstrou a dificuldade da execução de uma aplicação por um longo período de tempo em uma grade de pesquisa. Notou-se que infra-estrutura de grade OurGrid apresentou melhor desempenho, mas esta informação não é conclusiva pois foram identificados alguns fatores que levaram a descontinuidade da execução da aplicação nas grades analisadas:

- Falha em um dos servidores de um nó de grade devido a problemas no hardware;
- Inconsistência dos dados transmitidos/recebidos;
- Sistema operacional instável;
- Falha no escalonador ou do portal;

- Falta de energia elétrica para alimentar um determinado nó de grade e;
- Certificados expirados.

Ficou clara a importância do monitoramento constante do estado da grade computacional, com o objetivo de detectar com maior rapidez os pontos de falha. As falhas podem ser sanadas com a atuação humana, desde que sejam conhecidos os procedimentos para o restabelecimento do sistema; ou sem atuação humana, desde que o sistema seja reprojetado para ser tolerante a falhas.

Observou-se que os tempos de transmissão de dados afetam diretamente o desempenho da aplicação e que é dependente da localização física do nó de grade. No melhor caso, o tempo de comunicação pode representar apenas 1% do tempo de computação, enquanto que no pior caso pode chegar a 12% do tempo de computação da simulação de um ano do BRAMS. Notou-se também que os tempos de transmissão das análises são muito superiores aos tempos de transmissão dos dados de climatologia. Conclui-se que a transmissão dos dados tem um peso razoável no desempenho desta aplicação. Duas abordagens devem ser consideradas: processar o dado onde o dado estiver localizado e enviar os dados ao seu local de processamento enquanto a aplicação está sendo executada.

A estratégia adotada no projeto G-BRAMS foi revista considerando o uso como uma grade dedicada com portal aberto, a ociosidade da grade atual devido as vulnerabilidade à falhas e o tempo de comunicação dos dados.

A partir das constatações acima, propõe-se uma estratégia de escalonamento que resolva os problemas relacionados à transferência de dados e a queda de desempenho que será testada em experimento futuro, utilizando o Globus como infra-estrutura de grade devido a sua flexibilidade e modularidade. Para solucionar o primeiro problema, o escalonador deve continuamente solicitar informações sobre os recursos de grade para determinar os recursos ativos. A solução ideal para o segundo problema é manter as réplicas dos dados de entrada disponíveis localmente nos nós de grade para processamento. Esta solução não é sempre é possível devido ao grande volume dos dados a ser armazenado e às limitações de armazenamento local. Para contornar esta

situação, apenas os dados necessários ao processamento em dado nó de grade devem ser transmitidos aos locais indicados para processamento. Duas abordagens para solucionar este problema deverão ser consideradas neste escalonador:

- Executar a computação considerando a localização do dado para definir o local de processamento;
- Enviar o dado enquanto o nó de grade estiver processando o dado anterior.

O escalonamento de uma aplicação em grades computacionais ainda é um desafio e um tema atual de pesquisa, pois tanto a aplicação como os computadores podem apresentar características de desempenho distintas; os recursos podem ser compartilhados por usuários; e as redes de comunicação de dados, computadores e dados podem estar localizados em domínios administrativos distantes (FOSTER E KESSELMAN, 1999).

A estratégia de escalonamento proposta utiliza o gerenciador de tarefas SGE para gerenciar as tarefas localmente em cada "cluster". Esta estratégia privilegia a seleção dos recursos computacionais com maior desempenho e com melhor interconexão (melhor relação largura de banda/tráfego). Ao mesmo tempo em que seleciona os melhores recursos ela garante a disponibilidade dos elementos de uma grade, fator muito importante para atingir o desempenho desejado. É ideal para grades dedicadas com portal aberto, pois garante de uma forma mais justa o acesso aos recursos computacionais. Ela busca ser eficiente, garantindo uso dos recursos computacionais com o aumento do número de usuários e dentro da filosofia de grade.

No entanto, devido ao fato desta grade ser composta por um reduzido número de recursos computacionais, os algoritmos analisados possuem a desvantagem de não garantir ao usuário o momento da alocação de recursos às tarefas, e como conseqüência o momento de início e/ou término da execução da aplicação, quando mais usuários utilizam a grade. Neste caso, recomenda-se avaliar mecanismos para reserva antecipada de recursos computacionais para aplicações prioritárias.

Nota-se uma dificuldade na utilização à distância de recursos computacionais por usuários de aplicações meteorológicas. A idéia implementada pelo projeto G-BRAMS para climatologia pode ser expandida para outros modelos (ETA, Global, BRAMS,

WRF, etc.) e outras finalidades (tempo, clima, ondas, etc.) na forma de portais temáticos para meteorologia. Esses portais temáticos, desenvolvidos utilizando sistemas baseados em aplicações Web, podem colaborar para que resolver problemas computacionais da área de meteorologia.

Recomenda-se para trabalhos futuros:

- Avaliação da nova proposta de escalonamento, utilizando escalonadores locais;
- Implementação da climatologia do ECMWF e avaliação de seu desempenho;
- Utilização de “multimodel ensemble”, técnica que emprega análises de modelos diferentes, para determinação da climatologia. Esta técnica permite que as melhores características desses modelos possam ser extraídas.

REFERÊNCIAS BIBLIOGRÁFICAS

Albing, C. **Cray NQS: production batch for a distributed computing world**. In: SUN USER GROUP CONFERENCE AND EXHIBITION, 11., 1993, Brookkline, MA. **Proceedings…**Brookline: [s.n], 1993, p. 302–309

Almeida, E. S.; Campos Velho, H. F.; Preto, A. J.; Stephany, S. **Metodologia para determinação de climatologia de mesoescala do modelo BRAMS em grade computacional.** In: Workshop dos Cursos de Computação Aplicada do INPE (WORCAP-2005), 5., 2005S. José dos Campos.**Anais...** São José do sCampos: INPE, 2005.

Altair Grid Technologies **Portal batch system**. Disponível em: <http://www.openpbs.org/> Acesso em 10 dec2006.

Andrade, N.; Cirne,W.; Brasileiro, F.; Roisenberg, P. **Ourgrid:** an approach to easily assemble grids with equitable resource sharing**.** In: Workshop on Job Scheduling Strategies for Parallel Processing, 9.,2003, Siattle. **Proceedings…**Siattle: [s.n], 2003. v. 9. p. 61-86. (Lecture Notes in Computer Science).

Barham, P.; Dragovic, B.; Fraser, K.; Hand, S.; Harris,T.; Ho, A.; Neugebar, R.; Pratt, I.; Warfield, A. **Xen and the art of virtualization.** In: ACM Symposium on Operating Systems Principles (SOSP), October 2003, Bolton Landing, NY. **Proceedings…** Bolton Landing, NY: ACM, 2003, p. 164-177.

Barros, S. R.M. **Towards the RAMS-FINEP parallel model: load balancing aspect**. towards teracomputing: ECMWF – WORKSHOP ON THE USE OF PARALLEL PROCESSORS IN METEOROLOGY, 8., 1998, Garching/Munich, Germany. **Proceedings…** Garching/Munich, Germany: [s.n], 1998

Berman, F.; Wolski, R.; Casanova, H.; Cirne,W.; Dail, H.; Faerman, M.; Figueira, S.; Hayes, J.; Obertelli, G.; Schopf, J.; Shao, G.; Smallen, S.; Spring, N.; Su, A. Zagorodnov, D. Adaptive computing on the Grid Using AppLeS. **IEEE Transactions on Parallel and Distributed Systems**, v. 14, n. 4, p.369--382, Apr 2003.

Berman, F.; Wolski, R.; Figueira, S.; Schopf, J.; Shao,G. **Application level scheduling on distributed heterogeneous networks.** In: Supercomputing'96, 1996, Pittsburgh, Nov., 1996. **Proceedings…**Pittsburg: [s.n], 1996. UCSD CS Tech Report #CS96-482.

Bonatti, J.P. Modelo de circulação geral atmosférico do CPTEC**. CLIMANÁLISE**: Boletim de monitoramento e análise climática, edição especial comemorativa de 10 anos. Disponível em: http://www.cptec.inpe.br/products/climanalise/cliesp10a/bonatti.html Acesso em: 16 ago 2007, 1996.

Budenske, J.; Ramanujan, R.; Siegel, H. **On-line use of off-line derived mappings for iterative automatic target recognition tasks and a particular class of hardware platforms.** In:HETEROGENEOUS COMPUTING WORKSHOP, 1997, Los Alamitos. **Proceedings…**Los Alamitos, CA: IWWW, , Computer Society Press, 1997, p. 96-110.

Campos Velho Preto, A. J.; Stephany, S.; Rodrigues; Panetta, J.; Almeida, E. S.; Souto, R. P.; Navaux, P. O.; Diverio; Maillard, N.; Dias, P. L. S. Grid computing for mesoscale climatology: experimental comparison of three platforms. In: INTERNATIONAL MEETING ON HIGH PERFORMANCE COMPUTING FOR COMPUTATIONAL SCIENCE (VECPAR'06), 7., 2006, Rio de Janeiro. **Proceedings…**, Rio de Janeiro: ABMEC, 2006. : Lecture Notes in Computer Science

Capit, N. OAR/CIGRI : outils pour grappe et grille légère. In: GridUSe-2004, Metz, France, 21-24 Juin, 2004, p. 61-84..

Capit, N.; Da Costa, G.; Georgiou, Y.; Huard, G.; Martin, C.; Mounié, G.; Neyron, P.; Richard, O. A batch scheduler with high level components**.** In: CLUSTER COMPUTING AND GRID 2005 (CCGrid05), 2005. [S.l] **Proceedings…** [S.l]: IEEE, 2005.

Casanova, H.; Legrand, A.; Zagorodnov, D.; Berman, F. Heuristics for Scheduling Parameter Sweep Applications in grid Environments**.** In: HETEROGENEOUS COMPUTING WORKSHOP, 9., 2000, Cancun. **Proceedings…** Cancun: IEEE, 2000, 349-363.

Casavant, T. L.; Kuhl, J. G. A Taxonomy of Scheduling in General-purpose Distributed Computing Systems**. IEEE Transactins on Software Engineering**, v. 14, n. 2, p.:141-154, IEEE CS Press, Los Alamitos, 1988.

Charney, J. G; Fjortof, R.; Von Neuman, J. Numerical integration of the barotropic vorticity equations**. Tellus**, v. 6, p. 309-318, 1950.

Chen, C.; Cotton, W. R. A one-dimensional simulation of the stratocumulus-capped mixed layer**. Bound-Layer Meteor** , v. 25, p. 289–321, 1983.

Chou, S. C. Modelo regional ETA**. CLIMANÁLISE**: Boletim de monitoramento e análise climática, edição especial comemorativa de 10 anos. Disponível em: <http://www.cptec.inpe.br/products/climanalise/cliesp10a/27.html >Acesso em: 16 ago 2007.

Chervenak, A. L.; Foster, I.; Kesselman, C.; Meder, S.; Nefedova, V.; Quesnal, D.; Tuecke,S. Data management and transfer in high performance computational grid environments. **Parallel Computing Journal**, v. 28, n. 5, p. 749-771, May 2002.

Cirne, W.; Santos-Neto, E. Grids computacionais: da computação de alto desempenho a serviços sob demanda**.** In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES (SBRC), 18., 2000, Belo Horizonte. **Anais...** Belo Horizonte: SBC, 2000

Cirne, W.; Paranhos, D.; Costa, L.; Santos-Neto, E.; Brasileiro, F.; Sauvé, J.; da Silva, F. A. B.; Barros, C. O.; Silveira, C. Running bag-of-tasks applications on computational grids: The mygrid approach. In: INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING, (ICCP'2003), Oct 2003, Kaohsiung, Taiwan. **Proceedings…** Kaohsiung, Taiwan: ROC, 2003.

Czajkowski, K.; Fitzgerald, S.; Foster, I.; Kesselman, C. Grid Information services for distributed resource sharing**.** IEEE INTERNATIONAL SYMPOSIUM ON HIGH-PERFORMANCE DISTRIBUTED COMPUTING (HPDC-10), 10., Aug. 2001,San Francisco. **Proceedings…** San Francisco: IEEE Press, 2001.

Czajkowski, K.; Foster, I.; Karonis, N.; Kesselman, C.; Martin, S.; Smith, W.; Tuecke, S. A resource management architecture for metacomputing systems**.** WORKSHOP ON JOB SCHEDULING STRATEGIES FOR PARALLEL PROCESSING, (IPPS/SPDP '98), 9., 1998, Orlando. **Proceedings…** Orlandol: editora, , 1998, p. 62-82.

Dail, H.; Berman, F.; Casanova, H. A decoupled scheduling approach for Grid application development environments**. J. Parallel Distributed Computing**, v. 63, n. , p. 505–524, 2003.

Data Grid. **European union dataGrid project**. Disponível em: <http://eu-datagrid.web.cern.ch/eu-datagrid/>. Acesso em 29 out 2006.

Davies, R. Documentation of the solar radiation parameterization in the GLAS climate model**.** Washington: Nasa, 57 p, 1982. NASA Tech Memo. 83961

De Rose, C. A. F.; Navaux, P. O. A. **Arquiteturas paralelas.** Porto Alegre: , Ed. Sagra Luzatto, Instituto de Informática da UFRGS ,2003

Doty, B. **Grid Analysis and Display System (GrADS).** Maryland: Center for Ocean-Land-Atmosphere Studies (COLA). Disponível em: <http://grads.iges.org/grads/head.html> , Acesso em: 23-nov1995.

Fitzgerald, S.; Foster, I.; Kesselman, C.; Von Laszewski, G.; Smith, W.; Tuecke, S. A Directory Service for Configuring High-Performance Distributed Computations**.** In: IEEE SYMPOSIUM ON HIGH-PERFORMANCE DISTRIBUTED COMPUTING, 6., 1997, Las Vegas. **Proceedingas…** Las Vegas: IEEE, 1997, p. 365-375.

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.; Kesselman, C. **The Grid:** blueprint for a new computing infrastructure**.** San Francisco: Morgan Kaufman Publishers, Inc., 1999.

Foster, I;. Kesselman, C.; Tsudik, G., Tuecke, S..A security architecture for computational grids. In: ACM CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY CONFERENCE, 5., San Francisco. **Proceedings….** San Francisco: ACM, 1998, p. 83-92.

Foster, I. Designing and Building Parallel Programs. **On-line book**, 1995. Disponível em: <http://www-unix.mcs.anl.gov/dbpp/text/book.html> Acesso em: 16 ago 2007.

Ghormley, D. R.; Petrou, D.; Rodrigues, S. H.; Vahdat, A. M**.** GLUnix: a global layer unix for a network of workstations. **Software Practice and Experience**, v. 28, n. 9, 1998.

Grell, G., Devenyi, D. A generalized approach to parameterizing convection combining ensemble and data assimilation techniques**. Geophys. Res. Lett**. v. 29, n. 14, Art. n. 1693, July 15 2002.

GriPhyN. **Grid physics network**. Disponível em: <http://www.griphyn.org/> Acesso em: 13-ago2006.

Gropp, W.; Lusk, E,; Skjellum, A. **Using MPI:** portable parallel programming with the message-passing interface**.** 2. ed. Cambridge: MIT Press, 1999.

Gropp, W.; Lusk E.; Doss,N.; Skjellum,A. A high-performance, portable implementation of the MPI message passing interface standard. **Parallel computing**, v.22, n 6, p. 789 –828, 1996

Harshvardhan, K. M., Bromwich, D. H.; Corsetti, T. G. A fast radiation parameterization for general circulation models. **Journal Geophysical Research**, v.92, p. 1009-1016, 1987.

Hou, Y. T. **Cloud-radiation dynamics interation.** 209 p. 1990. Ph.D. Thesis, University Maryland, Maryland, 1990..

IBM **LoadLeveler for AIX 5L Version 3.2** using and administering**.** Disponível em: <http://www-03.ibm.com/systems/clusters/software/loadleveler.html > Acesso em: 02 nov2006.

Instituto Nacional de Pesquisas Espapciais. Centro de Previsão de Tempo e Estudos Climáticos (INPE/CPTEC) **Brazilian Regional Atmospheric Modeling System (BRAMS)**. São José dos Campos. Disponível em: <http://www.cptec.inpe.br/brams>, Acesso em: 25 out2006.

International Research Institute for Climate Prediction (IRI) **The science and practice of seasonal climate forecasting at the IRI.** Tutorial. Disponível em: <http://iri.columbia.edu/climate/forecast/tutorial2/>, Acesso em 31 maio2006.

IVDGL International **Virtual data grid laboratory** Disponível em:<http://www.ivdgl.org/> Acesso em 22 set 2006.

Java Community Process **JSR 168 portlet specification**. Disponível em: <http://www.jcp.org/jsr/detail/168.jsp> Acesso em: 05 out 2006.

Kinter, J. L.; Shukla, J.; Marx, L.; Schneider, E. K. A simulation of winter and summer circulations with the NMC global spectral model**. J.Atm.Sci**., v. 45, p.2486-2522, 1988.

Krishnamurti, T. N.; Kishtawal, C. M.; Zhang, Z.; LaRow, T.; Bachiochi, D.; Williford, E.; Gadgil, S.; Surendran, S. Multimodel Ensemble Forecasts for Weather and Seasonal Climate**. J. Climate**, v. 13, n.23, p. 4196-4216, 2000.

Kuo, H. L. Further studies of the parameterization of the influence of cumulus convection on large-scale flow**. Journal of the Atmospheric Sciences**, v. 31, p. 1232-1240, 1974. (AMS, Boston, USA)

Lacis, A. A;. Hansen, J. E. A parameterization of the absorption of the solar radiation in the earth's atmosphere**. Journal of the Atmospheric Sciences**, v. 31, p. 118-133, 1974.

Lee, B.; Schopf, J. M. Run-time prediction of parallel applications on shared environment**.** In: Proceedings of Clusters2003, 2003, [S.l]. **Proceedings…** Disponível em: http://www-unix.mcs.anl.gov/~schopf/Pubs/ . Acesso em: 16 ago 2007.

Lee, T. J. **The impact of vegetation on the atmospheric boundary layer and convective storms.**137p. Ph.D. (Dissertation, Department of Atmospheric Science) - Colorado State University, Fort Collins, Colorado, 1992

Mahrer, Y., Pielke, R. A. A numerical study of the airflow over irregular terrain. **Beitr. Phys. Atmos**., v. 50, p. 98-113, 1977.

Martin, C.; Richard, O. Algorithme de vol de travail appliqu au dploiement d'applications parallles**.** In Soumis RenPar'15, 2003, Nice. **Proceedings…**Nice: [s.n], 2003.

Matsuoka, S.; Nagashima, U.; Nakada, H. Ninf: Network based information library for globally high performance computing**.** In: PARALLEL OBJECT-ORIENTED METHODS AND APPLICATIONS WORKSHOP (POOMA), 1996, Santa Fe, New Mexico. **Proceedings…**Santa Fe:[s.n], 1996.

Melo, M. L. D.; Marengo, J. A. Climatologia global do MCGA do CPTEC/COLA TO42L28: simulação em modo "ensemble". In: CONGRESSO BRASILEIRO DE METEOROLOGIA, 13., 2004, Fortaleza. **Anais…** São José dos Campos: SBMET, 2004.

Mellor, G. L.; Yamada, T. Development of a turbulence closure model for geophysical fluid problems**. Rev. Geophysics Space Physics**. v. 20, p. 851 – 875, 1982

Mendes, C. L.; Panetta, J. Selecting Directions for Parallel RAMS Performance Optimization. In: SYMPOSIUM ON COMPUTER ARCHITECTURE AND HIGH PERFORMANCE COMPUTING, 11., 1999, Natal, RN, **Proceedings…** Natal: 1999, p.85-92.

Mendonça, A. M.; Bonatti, J. P. **O sistema de previsão de tempo por ensemble do CPTEC.** CONGRESSO BRASILEIRO DE METEOROLOGIA, 12., 2002, Foz do Iguaçú, PR. **Anais…** São José dos Campos: INPE, 2002.

National Research Council. **Realizing the rnformation future:** the Internet and beyond**.** [S.l]: National Academic Press, 1994. Disponível em: <http://stills.nap.edu/readingroom/books/rtif> Acesso em: 01 maio2006.

National Grid Service (NGS) **UK's National Grid Service** Disponível em: < http://www.grid-support.ac.uk> Acesso em: 13 jul 2006.

Campos , P. L. S. Paranhos, D.; Cirne, W.; Brasileiro, F. Trading cycles for information: Using replicaton to schedule bag-of-tasks applications on computational grids. In: EURO-PAR 2003: INTERNATIONAL CONFERENCE ON PARALLEL AND DISTRIBUTED COMPUTING, 2003, Klagenfurt,Austria. **Proceedings…** Klagenfurt: editora, 2003.

Pielke, R. A.; Cotton, W. R; Walko, R. L.; Tremback, C. J.; Lyons, W. A.; Grasso, L. D.; Nicholls, M.E.; Moran, M. D.; Wesley, D. A.; Lee, T. J.; Copeland, J.H. **A.** Comprehensive meteorological modeling system - RAMS. **Meteor. Atmos. Phys.,** n. 49, p. 69-91, 1992.

Plastino, A.; Ribeiro, C. C.; Rodriguez, N. L. R. **Uma taxonomia de algoritmos de balanceamento de carga para aplicações SPMD**. 1998. 33p. Monografia (Ciência da Computação) – Pontifícia Católica, (PUC), Rio de Janeiro: 1998.

Plataform **Plataform LSF Family**. Disponível em: http://www.platform.com/Products/Platform.LSF.Family/ . Acesso em: 02-ago2006.

Ribler, R. L.; Simitci, H.; Reed, D. A. The Autopilot performance-directed adaptative control system. **Future Generation Computer System**, v. 18, n. 1, p. 175-187, 2001.

Russell, M.; Novotny, J.; Wehrens, O. GridSphere's Grid Portlets. In: PORTALS, PORTLETS AND GRIDSPHERE WORKSHOP. 2005, Louisiana State, USA. **Proceedings…** Luisiana: [s.n], 2005.

Schopf, J. M. A General Architecture for Scheduling on the Grid. Special Issue on the Grid of the **Journal of Parallel and Distributed Computing**, 2002.

Schopf, J. M.; Berman, F. Stochastic scheduling. In: SuperComputing'99, 1999, Portland**.. Proceedings…**Portland: ACM**,** 1999

Senger, L. S. **Obtenção e utilização do conhecimento sobre aplicações paralelas no escalonamento em sistemas computacionais distribuídos.** Monografia (Qualificação de Doutorado em Computação Aplicada), Universidade de São Paulo, São Carlos, 2002

Shao, G. **Adaptive scheduling of master/worker applications on distributed computational resources**. PhD thesis, Dept. of Computer Science, University Of California at San Diego, 2001.

Shuman, F. G. History of numerical weather prediction at the National Meteorological Center. **Weather and Forecasting**, v.4, p. 286-296, 1989.

Sirbu, M; Marinescu, D; **A scheduling expert advisor for heterogeneous environments**. In: HETEROGENEOUS COMPUTING WORKSHOP, 1997, Los Alamitos. **Proceedings…**Los Alamitos, CA: IWWW Computer Society Press, 1997, p. 96-110.

Slingo, J. M. Development of verification of a cloud prediction scheme for the ECMWF model. **Quart. J. Roy. Meteorological Society**, v. 113, p. 889-927, 1987.

Secretaria de Logística e Tecnologia da Informação (SLTI). **Guia de estruturação e administração do ambiente de cluster e grid.** Brasília: 2006, 450 p.

Souto, R. P.; Ávila, R. B.; Navaux, P. A. O.; Py, M.; Maillard, N.; Diverio, T. A.; Velho, H. F. C.; Stephany, S.; Preto, A.; Panetta, J.; Rodrigues, E.; Almeida, E. Processing mesoscale climatology in a grid environment**.** In: IEEE INTERNATIONAL SYMPOSIUM ON CLUSTER COMPUTING AND THE GRID - CCGRID'07, 7., 2007, Rio de Janeiro. **Proceedings…**Rio de Janeiro: IEEE, 2007.

Sun Microsystems **Sun ONE grid engine -** administration and user's guide. Disponível em: http://gridengine.sunsource.net/project/gridengine/documentation.html. Acesso em: 23 fev2006.

Tanenbaum, A. S. **Organização estruturada de computadores**. Rio de Janeiro: Livros Técnicos e Científicos Editora, 2001.

TeraGrid **Teragrid** Disponível em: http://www.teragrid.org/ . Acesso em: 02 nov 2006.

Tiedtke, M. **The sensitivity of the time mean large scale flow to cumulus convection in the ECMWF model.** In: WORKSHOP ON CONVECTION IN LARGE SCALE NUMERICAL MODELS, 1983, Reading. **Proceedings…** Reading: ECMWF, 1983, p. 297-316.

Tremback, C. J.; Walko, R. L. RAMS: The Regional Atmospheric Modeling System development of parallel processing computer architectures. In: RAMS USERS WORKSHOP, 3., 1997, Echuca, Victoria, Australia. **Proceedings…** Echuca: [s.n], 1997.

Tremback, C. J. **Numerical simulation of a mesoscale convective complex model development and numerical results.** 1990, 247p Ph.D. Dissertation, (Department of Atmospheric Science) Colorado State University, FortCollins, CO, 1990. 80523. (Atmos. Sci. Paper No. 465).

Tripoli, G. J.; Cotton, W. R. The Colorado State University three-dimensional cloud mesoscale model, 1982: Part I: General theoretical framework and sensitivity experiments. **J. Rech. Atmos**, v.16, p.185-220, 1982.

University of Wisconsin-Madison **Condor project homepage**. Disponível em: http://www.cs.wisc.edu/condor/. Acesso em: 02 set 2006.

Von Laszewski, G.; Foster, I. **Usage of LDAP in Globus**. Argonne, IL: Computer Science Division. Argonne National Laboratory, 2002. Disponível em: http://www.globus.org/mds/globus_in_ldap.html. Acesso em: 16 ago 2007.

Walko, R. L.; Band, L. E.; Baron. J.; Kittel, T. G. F.; Lammers, R.; Lee, T. J.; Ojima, D.; Pielke, R. A.; Taylor, C.; Tague, C.; Tremback, C. J.; Vidale, P. J. Coupled atmosphere-biophysics hydrology models for environmental modeling. **Journal of Applied Meteorology, v.** 39, n. 6, p. 931-944, 2000.

Walko, R. L.; Tremback, C. J. **RAMS - The Regional Atmospheric odeling System Version 2C:** User's guide. Fort Collins, Colorado: ASTeR, Inc.,. , 1991, 86p.

Wikipedia. **História da computação.** Disponível em: http://pt.wikipedia.org/wiki/> Historia_da_computação . Acesso em: 23 mar 2007.

Wikipedia. **Climate ensemble** Disponível em: http://en.wikipedia.org/wiki/Climate_Ensemble. Acesso em: 02 set 2006.

Wolski, R.; Spring, N.; Hayes, J. Network weather service: a distributed resource performance forecasting service for metacomputing**. Future Generation Computing Systems**, v.. 15, n.. 5-6, p. 757-768, 1999.

Xue, Y., Sellers, P. J., Kinter III,J.L.; Shukla,J. A simplified biosphere model for global climate studies**. Journal of. Climate**, v. 4, n. 3, p. 345-364, 1991.

Yang, L.; Schopf, J. M.; Foster, I. Conservative scheduling: using predicted variance to improve scheduling decisions in dynamic environments. In: SUPERCOMPUTING, 2003 ACM/IEEE CONFERENCE, 2003, [S.l.]. **Proceedings**…[S.l.]: **ACM/IEEE,** 2003a.

Yang, L.; Foster, I.; Schopf, J. M**.** Homeostatic and tendency-based CPU load predictions. INTERNATIONAL PARALLEL AND DISTRIBUTED PROCESSING SYMPOSIUM (IPDPS 2003), 2003, Nice, France. **Proceedings…** Nice: IEEE,2003b of

Yu, C.; Chen, P.; Wang, S. Adaptive task scheduling algorithms for master-worker applications in grid computing. In: WORKSHOP ON COMPILER TECHNIQUES FOR HIGH-PERFORMANCE COMPUTING, 2005. Tunghai. **Proceedings…** Tunghai: University, Taichung, Taiwan, 2005.

Yun, W. T., Stefanova, L., Krishnamurti, T. N. improvement of the multimodel superensemble technique for seasonal forecasts. **Journal of Climate**, v.16, n. 22, p. 3834-3840, 2003.

APÊNDICE A

CERTIFICADO PARA UTILIZAÇÃO COM GLOBUS TOOLKIT

Enclosed is your Globus Certificate for:

"/O=Grid/O=Globus/OU=cptec.inpe.br/CN=Eugenio Sper de Almeida"

The above string is known as your certificate subject, and it
uniquely identifies you.

To start using Globus services you must perform two steps.

1. Have your subject name added to the gridmap file at
each resource you intend to do. You will need to make
a request to the system administrator at each of these
sites.

2. Save this e-mail message into the following file.

/home/eugenio/.globus/usercert.pem

You need not edit this message in any way. Simply
save this e-mail message to the file.

If you have any questions about the certificate contact
the Globus CA at ca@globus.org

issuer :/C=US/O=Globus/CN=Globus Certification Authority
subject:/O=Grid/O=Globus/OU=cptec.inpe.br/CN=Eugenio Sper de Almeida
serial :0D5C

```
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 3420 (0xd5c)
    Signature Algorithm: md5WithRSAEncryption
    Issuer: C=US, O=Globus, CN=Globus Certification Authority
    Validity
      Not Before: Jan 19 17:03:19 2001 GMT
      Not After : Jan 19 17:03:19 2002 GMT
    Subject: O=Grid, O=Globus, OU=cptec.inpe.br, CN=Eugenio Sper de Almeida
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public Key: (1024 bit)
        Modulus (1024 bit):
          00:c6:18:9a:02:92:de:9c:96:b6:8d:b9:b6:6d:8c:
          c1:0c:3f:80:39:95:98:54:6c:87:50:0a:29:d6:75:
          9e:e6:ea:7f:09:f8:d4:d6:03:d4:a3:ad:21:93:ea:
          62:ff:2c:39:e9:c6:7a:26:b1:fb:1c:d0:89:e9:2c:
          6d:30:68:e0:11:44:a0:70:c2:61:1d:2d:b5:b8:b6:
          2e:ff:a0:0b:7a:c2:3f:c5:7a:38:72:8d:59:f7:bf:
          cc:a8:87:39:6a:16:55:9d:fb:4c:03:40:22:e2:17:
          36:49:92:b9:48:11:55:67:91:e7:91:81:ac:bd:64:
          53:16:39:b0:e7:1a:c7:36:dd
        Exponent: 65537 (0x10001)
    X509v3 extensions:
      Netscape Cert Type:
        0xC0
  Signature Algorithm: md5WithRSAEncryption
    1e:96:a7:96:44:fe:0e:57:c4:10:3b:04:25:45:ff:27:3d:d1:
    f4:85:9d:fd:65:78:75:57:c0:1d:08:14:0b:4e:2c:2f:43:65:
    39:03:4c:27:ac:5e:40:e2:bb:03:ea:a5:af:f5:10:35:bf:ef:
    19:f8:a3:99:64:33:ce:4c:01:c2:3d:5e:8b:66:a7:0d:d8:29:
    4d:e3:8b:67:90:00:8c:aa:e8:35:d2:ee:a7:bf:5c:4c:a0:77:
    a9:d9:d4:9b:66:a4:b9:d8:86:2d:ad:3b:e4:85:c4:58:62:67:
    e6:93:2f:99:10:b7:22:8f:c0:a6:c1:cb:ca:f8:ed:d2:5b:6d:
    5e:a4
```

```
-----BEGIN CERTIFICATE-----
MIICLTCCAZagAwIBAgICDVwwDQYJKoZIhvcNAQEEBQAwRzELMAkGA1UEBhMCVVMx
DzANBgNVBAoTBkdsb2J1czEnMCUGA1UEAxMeR2xvYnVzIENlcnRpZmljYXRpb24g
QXV0aG9yaXR5MB4XDTAxMDExOTE3MDMxOVoXDTAyMDExOTE3MDMxOVowWjENMAsG
A1UEChMER3JpZDEPMA0GA1UEChMGR2xvYnVzMRYwFAYDVQQLEw1jcHRlYy5pbnBl
LmJyMSAwHgYDVQQDExdFdWdlbmlvIFNwZXIgZGUgQWxtZWlkYTCBnzANBgkqhkiG
9w0BAQEFAAOBjQAwgYkCgYEAxhiaApLenJa2jbm2bYzBDD+AOZWYVGyHUAop1nWe
5up/CfjU1gPUo60hk+pi/yw56cZ6JrH7HNCJ6SztMGjgEUSgcMJhHS21uLYu/6AL
esI/xXo4co1Z97/MqIc5ahZVnftMA0Ai4hc2SZK5SBFVZ5HnkYGsvWRTFjmw5xrH
Nt0CAwEAAaMVMBMwEQYJYIZIAYb4QgEBBAQDAgDAMA0GCSqGSIb3DQEBBAUAA4GB
AB6Wp5ZE/g5XxBA7BCVF/yc90fSFnf1leHVXwB0IFAtOLC9DZTkDTCesXkDiuwPq
pa/1EDW/7xn4o5lkM85MAcI9Xotmpw3YKU3ji2eQAIyq6DXS7qe/XEygd6nZ1Jtm
pLnYhi2tO+SFxFhiZ+aTL5kQtyKPwKbBy8r47dJbbV6k
-----END CERTIFICATE-----
```

APÊNDICE B

ARQUIVO RAMSIN PARA PROCESSAMENTO DA REGIÃO NORTE

```
$MODEL _GRIDS

  EXPNME   = 'Version 5.02 - OWN-
ODA-CUINV',

  RUNTYPE  = 'MAKESFC',

  TIMEUNIT = 'h',

  TIMMAX    = 27768,

  LOAD_BAL = 0,

  IMONTH1  = 11,
  IDATE1   = 1,
  IYEAR1   = 1995,
  ITIME1   = 0000,

  NGRIDS   = 2,

  NNXP     = 44,78,54,58,
  NNYP     = 44,54,46,58,
  NNZP     = 33,33,40,40,
  NZG      = 9,
  NZS      = 1,

  NXTNEST  = 0,1,1,1,

  IF_ADAP  = 0,

  IHTRAN   = 1,
  DELTAX   = 160000.,
  DELTAY   = 160000.,
  DELTAZ   = 100.,
  DZRAT    = 1.1,
  DZMAX    = 1000.,

   ZZ       = 0.0,
    20.0,     46.0,     80.0,     120.0,
165.0,
   220.0,    290.0,    380.0,    480.0,
590.0,
   720.0,    870.0,    1030.0,   1200.0,
1380.0,
   1595.0,   1850.0,   2120.0,   2410.0,
2715.0,
   3030.0,   3400.0,   3840.0,   4380.0,
5020.0,
   5800.0,   6730.0,   7700.0,   8700.0,
9700.0,
  10700., 11700., 12700., 13700.,
14700., 15700., 16700.,
  17700., 18700., 19700.,

   DTLONG   = 120.,
   NACOUST  = 3,
   IDELTAT  = 0,

   NSTRATX  = 1,4,4,4,
   NSTRATY  = 1,4,4,4,
   NNDTRAT  = 1,3,3,3,

   NESTZ1   = 0,
   NSTRATZ1 = 2,2,2,1,
   NESTZ2   = 0,
   NSTRATZ2 = 3,3,2,1,

   POLELAT  = -15.0,
   POLELON  = -53.0,

  CENTLAT  = -15.0, -4.0, -7.0, -24.0,
  CENTLON  = -53.0, -61.0, -40.0, -
51.0,
```

```
  NINEST   = 1,0,0,0,
  NJNEST   = 1,0,0,0,
  NKNEST   = 1,1,1,1,

  NNSTTOP  = 1,1,1,1,
  NNSTBOT  = 1,1,1,1,


  GRIDU    = 0.,0.,0.,0.,
  GRIDV    = 0.,0.,0.,0.,


 $END

 $MODEL_FILE_INFO



  INITIAL  = 2,

  NUD_TYPE = 2,

  VARFPFX  = './ivar/iv-clima',
  VWAIT1   = 0.,
  VWAITTOT = 0.,

  NUD_HFILE = './h_teste-H-2002-11-
16-060000-head.txt',



  NUDLAT   = 5,
  TNUDLAT  = 3600.,
  TNUDCENT = 86400.,
  TNUDTOP  = 10800.,
  ZNUDTOP  = 15000.,

  WT_NUDGE_GRID =  1., 0.6, 0.7,
0.5,



  WT_NUDGE_UV = 1.,
  WT_NUDGE_TH = 1.,
  WT_NUDGE_PI = 1.,
  WT_NUDGE_RT = 1.,

  NUD_COND    = 0,

  COND_HFILE = './hist/a-H-2001-07-
21-000000-head.txt',

  TCOND_BEG=0.,
TCOND_END=21600.,
  T_NUDGE_RC = 3600.,
  WT_NUDGEC_GRID =  1., 0.8, 0.7,
0.5,


  IF_ODA = 0,
  ODA_UPAPREFIX = './obs/dp-r',
  ODA_SFCPREFIX = './obs/dt-s',

  FRQODA=300.,
  TODABEG=0.,
TODAEND=99999999.,

  TNUDODA= 900.,
  WT_ODA_GRID =  1., 0.8, 0.7, 0.5,

  WT_ODA_UV = 1.,
  WT_ODA_TH = 1.,
  WT_ODA_PI = 1.,
  WT_ODA_RT = 1.,

  RODA_SFCE =
50000.,100.,100.,100.,
  RODA_SFC0 =
100000.,100000.,100000.,100000.,
  RODA_UPAE =
100000.,200.,200.,200.,
  RODA_UPA0 =
200000.,2000.,2000.,2000.,

  RODA_HGT =
3000.,3000.,3000.,3000.,

  RODA_ZFACT =
100.,100.,100.,100.,

  ODA_SFC_TIL=21600.,
  ODA_SFC_TEL=900.,
  ODA_UPA_TIL=43200.,
  ODA_UPA_TEL=21600.,
```

```
  IF_CUINV = 0,
  CU_PREFIX = './t5-C-',

  TNUDCU=900.,
  WT_CU_GRID=1., 1., .5,

  TCU_BEG=0., TCU_END=7200.,
  CU_TEL=3600.,
  CU_TIL=21600.,

  TIMSTR   = .,
  HFILIN   = 'h_teste-H-2002-11-16-
060000-head.txt',

  IPASTIN   = 0,

  PASTFN    = 'a-A-2000-01-09-
000000-head.txt',

  IOUTPUT  =  2,
  HFILOUT  = './H/clima',
  AFILOUT  = './A/clima',
  ICLOBBER = 1,
  IHISTDEL = 1,
  FRQHIS   = 86400.,
  FRQANL   = 21600.,
  FRQLITE  = 0.,
  XLITE    = '/0:0/',
  YLITE    = '/0:0/',
  ZLITE    = '/0:0/',
  NLITE_VARS=4,

LITE_VARS='UP','VP','WP','THETA',
  AVGTIM   = 0.,
  FRQMEAN  = 0.,
  FRQBOTH  = 0.,
  KWRITE   = 0,
  FRQPRT   = 21600.,
  INITFLD  = 1,

  TOPFILES = './data/toph-clima',
  SFCFILES = './data/sfc-clima',
  SSTFPFX  = './data/sst-clima',
  NDVIFPFX = './data/ndvi-clima',

  ITOPTFLG = 1,1,1,1,
  ISSTFLG  = 1,1,1,1,
  IVEGTFLG = 1,1,1,1,
  ISOILFLG = 1,1,1,1,
  NDVIFLG  = 2,2,2,2,
  NOFILFLG = 2,2,2,2,

  IUPDNDVI  = 0,
  IUPDSST  = 1,

   ITOPTFN  =
'$BRAMS2/topo10km/H',
          '$BRAMS2/topo10km/H',
          '$BRAMS2/topo10km/H',
          '$BRAMS2/topo10km/H',

   ISSTFN   =
'$BRAMS2/sst_mensal/M',
         '$BRAMS2/sst_mensal/M',
         '$BRAMS2/sst_mensal/M',
         '$BRAMS2/sst_mensal/M',

   IVEGTFN  =
'$BRAMS2/veg_new/VEGv2_',

'$BRAMS2/veg_new/VEGv2_',

'$BRAMS2/veg_new/VEGv2_',

'$BRAMS2/veg_new/VEGv2_',

   ISOILFN  =
'$BRAMS2/soil_FAO/FAO',
          '$BRAMS2/soil_FAO/FAO',
          '$BRAMS2/soil_FAO/FAO',
          '$BRAMS2/soil_FAO/FAO',

   NDVIFN  =
'/shared/tools/dados_brams/ndvi/N',

'/shared/tools/dados_brams/ndvi/N',

'/shared/tools/dados_brams/ndvi/N',

'/shared/tools/dados_brams/ndvi/N',
```

```
  ITOPSFLG = 0,0,0,0,

  TOPTENH  = 1.,1.,1.,1.,
  TOPTWVL  = 3.,2.,2.,2.,

  IZ0FLG   = 0,0,0,0,
  Z0MAX    = 5.,5.,5.,5.,
  Z0FACT   = 0.005,
  MKCOLTAB = 0,
  COLTABFN =
'$BRAMS2/micro/ct2.0',

 $END

 $MODEL_OPTIONS

  NADDSC   = 0,
  ICORFLG  = 1,
  IBND     = 1,
  JBND     = 1,
  CPHAS    = 20.,
  LSFLG    = 0,
  NFPT     = 0,
  DISTIM   = 400.,
  ISWRTYP  = 1,
  ILWRTYP  = 1,
  RADFRQ   = 1200.,
  LONRAD   = 1,
  NNQPARM  = 2,2,2,2,
  CLOSURE_TYPE = 'EN',
  NNSHCU   = 1,1,1,1
 CONFRQ   = 900.,
  SHCUFRQ  = 900.,
  WCLDBS   = .0005,
  NPATCH   = 5,
  NVEGPAT  = 4,
  ISFCL    = 1,
  NVGCON   = 6,
  PCTLCON  = 1.,
  NSLCON   = 6,
  ZROUGH   = .05,
  ALBEDO   = .2,
  SEATMP   = 298.,
  DTHCON   = 0.,
  DRTCON   = 0.,
  SOIL_MOIST = 'n',
  SOIL_MOIST_FAIL = 'l',
  USDATA_IN  =
'$BRAMS2/umid/us',
  USMODEL_IN =
'$BRAMS2/umid/us',

  SLZ     =  -2.0, -1.75, -1.50, -1.25, -
1.00, -0.75, -0.50, -0.25, -0.1,
  SLMSTR  =  0.40, 0.37, 0.35, 0.33,
0.32, 0.31, 0.30, 0.29, 0.28,
  STGOFF=  0.0,  0.0,  0.0,   0.0, 0.0,
0.0,  0.0,   0.0, 0.0,

  IF_URBAN_CANOPY =0,

  IDIFFK  = 1,1,1,1,
  IHORGRAD = 1,
  CSX     = .2,.2,.2,.2,
  CSZ     = .35,.35,.35,.35,
  XKHKM   = 3.,3.,3.,3.,
  ZKHKM   = 3.,3.,3.,3.,
  AKMIN   = 0.8,0.8,0.8,0.8,
  LEVEL   = 3,
  ICLOUD  = 4,
  IRAIN   = 2,
  IPRIS   = 5,
  ISNOW   = 2,
  IAGGR   = 2,
  IGRAUP  = 2,
  IHAIL   = 2,
  CPARM   = .1e9,
  RPARM   = 1e-3,
  PPARM   = 0.,
  SPARM   = 1e-3,
  APARM   = 1e-3,
  GPARM   = 1e-3,
  HPARM   = 3e-3,
  GNU     = 2.,2.,2.,2.,2.,2.,2.,

 $END

 $MODEL_SOUND

  IPSFLG   = 1,
  ITSFLG   = 0,
  IRTSFLG  = 3,
```

```
  IUSFLG   = 0,
  HS       = 0.,

 PS =
1010.,1000.,2000.,3000.,4000.,6000.,80
00.,11000.,15000.,20000.,25000.,

  TS =    25., 18.5,  12., 4.5, -11., -24., -
37., -56.5, -56.5, -56.5, -56.5,

 RTS =
70.,70.,70.,70.,20.,20.,20.,20.,10.,10.,10
.,
  US = 3.,3.,3.,3.,3.,3.,3.,3.,3.,3.,3.,
  US = ?US?
  US = ?US?
  US = ?US?
  VS = 0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
  VS = ?VS?
  VS =  ?VS?

 $END

$MODEL_PRINT

  NPLT     = 0,
  IPLFLD   =
'UP','THP','THETA','RT','TOTPRE',
  IXSCTN    = 3,3,3,3,3,3,
  ISBVAL    = 2,2,2,2,2,2,2,

$END

 $ISAN_CONTROL

  ISZSTAGE   = 1,
  IVRSTAGE   = 1,
  ISAN_INC = 0600,
  GUESS1ST = 'PRESS',
  I1ST_FLG = 1,
  IUPA_FLG = 3,
  ISFC_FLG = 3,
  IAPR    = '/usr/local/jakarta-tomcat-
5.0.28/webapps/portalourgrid/html/mem
bro1_novo/dp',
  IARAWI  = '',
  IASRFCE = '',
  VARPFX    = './ivar/iv-clima',
  IOFLGISZ  = 0,
  IOFLGVAR  = 1,

 $END

 $ISAN_ISENTROPIC

----------------------------------
  NISN     = 43,
  LEVTH    =
280,282,284,286,288,290,292,294,296,
298,300,303,306,309,312,

315,318,321,324,327,330,335,340,345,
350,355,360,380,400,420,

440,460,480,500,520,540,570,600,630,
670,700,750,800,


  NIGRIDS  = 1,
  TOPSIGZ  = 20000.,
  HYBBOT   = 4000.,
  HYBTOP   = 6000.,
  SFCINF   = 1000.,
  SIGZWT   = 1.,
  NFEEDVAR = 1,
  MAXSTA   = 150,
  MAXSFC   = 1000,
  NOTSTA   = 0,
  NOTID    = 'r76458',
  IOBSWIN = 1800,
  STASEP   = .1,
  IGRIDFL  = 3,
  GRIDWT   = .01,.01,
  GOBSEP   = 5.,
  GOBRAD   = 5.,
  WVLNTH   = 1200.,900.,
  SWVLNTH  = 750.,300.,
  RESPON   = .90,.9,

 $END
```

APÊNDICE C

NOME DAS VARIÁVEIS DA SIMULAÇAO REGIONAL

Variável (superf.)	n°. níveis
um	1
um0h	1
um6h	1
um12h	1
um18h	1
vm	1
vm0h	1
vm6h	1
vm12h	1
vm18h	1
tempcm	1
tempcm0h	1
tempcm6h	1
tempcm12h	1
tempcm18h	1
geom	1
geom0h	1
geom6h	1
geom12h	1
geom18h	1
rhm	1
rhm0h	1
rhm6h	1
rhm12h	1
rhm18h	1
wm	1
wm0h	1
wm6h	1
wm12h	1
wm18h	1
t2mm	1
t2mm0h	1
t2mm6h	1
t2mm12h	1
t2mm18h	1
dewptcm	1
dewptcm0h	1
dewptcm6h	1
dewptcm12h	1
dewptcm18h	1
hm	1
hm0h	1
hm6h	1

Variável (superf.)	n°. níveis
hm12h	1
hm18h	1
lem	1
lem0h	1
lem6h	1
lem12h	1
lem18h	1
rshortm	1
rshortm0h	1
rshortm6h	1
rshortm12h	1
rshortm18h	1
acccon	1
totpcp	1
preciptot	1
tempcmtot	10
Variável (ar superior)	**n°. níveis**
tempcm0h	10
tempcm6h	10
tempcm12h	10
tempcm18h	10
umtot	10
um0h	10
um6h	10
um12h	10
um18h	10
vmtot	10
vm0h	10
vm6h	10
vm12h	10
vm18h	10
wmtot	10
wm0h	10
wm6h	10
wm12h	10
wm18h	10
Rvmtot	10
rvm0h	10
rvm6h	10
rvm12h	10
rvm18h	10

PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE

Teses e Dissertações (TDI)

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

Manuais Técnicos (MAN)

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

Notas Técnico-Científicas (NTC)

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programa de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

Relatórios de Pesquisa (RPQ)

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

Propostas e Relatórios de Projetos (PRP)

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

Publicações Didáticas (PUD)

Incluem apostilas, notas de aula e manuais didáticos.

Publicações Seriadas

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Constam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

Programas de Computador (PDC)

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. São aceitos tanto programas fonte quanto executáveis.

Pré-publicações (PRE)

Todos os artigos publicados em periódicos, anais e como capítulos de livros.