

Trabalho de Iniciação Científica*

Suporte para Aplicações com Conhecimento da Rede através de Pesquisa por Pacotes

Aluno: Daniel de Angelis Cordeiro

Orientador: Prof. Alfredo Goldman vel Lejbman

{danielc,gold}@ime.usp.br

1 de abril de 2008

O objetivo deste trabalho é pesquisar formas de obtenção do estado da rede de forma dinâmica, isto é, queremos obter as características do ambiente de rede onde o sistema se encontra. O foco desta pesquisa são ambientes de grade computacional (*grid computing*) que são, de forma simplificada, um conjunto de recursos computacionais (aglomerados de PCs e supercomputadores) interconectados.

A análise de sistemas de grade computacional tradicionais nos mostra que a rede não é considerada um recurso para as tarefas da grade. Descobrimos também que sistemas de sondagem especializados produzem resultados mais precisos, mas não são adequados para o ambiente de grade computacional que é formado, tipicamente, por um sistema heterogêneo.

Sistemas de sondagem por pacotes resolvem o problema da heterogeneidade da rede, mas esbarram no problema dos equipamentos de rede que não dão

*Este projeto é fomentado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq
Proc. No. 55.2028/02-9)

suporte a camada três (TCP/IP) de rede. Podemos utilizar o protocolo ICMP para resolver o problema, mas equipamentos de rede tratam pacotes ICMP e IP de maneiras diferentes, o que produz um resultado que não é preciso.

Sumário

1	Introdução	2
2	Objetivo	3
3	Trabalhos Relacionados	3
3.1	Sistemas de Grade Computacionais	4
3.1.1	Legion	4
3.1.2	OurGrid	6
3.2	Sistemas Especializados	7
3.2.1	SNMP	7
3.2.2	ReMoS	8
3.3	Sondagem por Pacotes	9
3.3.1	Sondagem Ativa e Passiva	9
3.3.2	Atraso	10
3.3.3	Dispersão de Pacotes	10
3.3.4	Pacotes de Tamanhos Variados	11
3.3.5	Roteamento e Topologia	11
4	Atividades Realizadas	12
5	Conclusão	12

1 Introdução

Atualmente, nota-se um aumento no interesse [6] por grades computacionais que são, de forma simplificada, um conjunto de recursos computacionais (aglomerados de PCs e super-computadores) interconectados. A possibilidade de tal infra-estrutura nos leva a ampliar os horizontes para novos tipos de aplicações de alto desempenho que utilizem recursos da

grade de forma eficiente. Para isso, é necessário possibilitar a programação paralela no ambiente distribuído de uma grade computacional.

O projeto InteGrade [1], que está sendo desenvolvido no Instituto de Matemática e Estatística da USP, e no qual este trabalho está inserido, visa construir essa infra-estrutura genérica de middleware que permitirá a criação de uma grade computacional que integrará os recursos computacionais disponíveis em máquinas ociosas em um ambiente de execução unificado.

Para que essa infra-estrutura fornecida pelo InteGrade seja apropriada para programação paralela é necessário que a integração dos recursos computacionais disponíveis leve em conta o ambiente de rede disponível, de forma a minimizar o efeito da troca de mensagens realizadas no desempenho das aplicações paralelas. É necessário, portanto, que o InteGrade seja uma aplicação que consiga obter, de forma dinâmica, informações sobre a rede em que se encontra.

O restante deste trabalho é dividido como segue. A seção 2 explicita os objetivos deste trabalho. A seção 3 descreve os trabalhos relacionados estudados durante esta iniciação científica, enquanto que as atividades realizadas são descritas na seção 4. As conclusões são discutidas na seção 5.

2 Objetivo

O objetivo deste trabalho é pesquisar formas de obtenção do estado da rede de forma dinâmica. Queremos poder estimar as características da rede onde se encontram os recursos computacionais que serão integrados pelo InteGrade. Para tanto, é necessário desenvolver um método de inferência dos recursos da rede de forma não intrusiva, ou seja, que consuma pouca banda e que não necessite de alterações de baixo nível (alterações no sistema operacional, por exemplo) nos nós que compõem a grade.

3 Trabalhos Relacionados

Há três categorias de sistemas relacionados a este trabalho. Em um âmbito mais geral, todos os sistemas de grade computacional conhecidos estão relacionados com o projeto InteGrade e, consequentemente, com este trabalho. No âmbito de sondagem de rede, podemos

classificar os sistemas existentes em dois tipos: sistemas especializados, ou seja, sistemas que dependem de características específicas de rede em que se encontram e sistemas de sondagem de rede por pacotes, isto é, que não necessitam de nenhum conhecimento sobre o tipo de rede em que se encontram.

Os sistemas de grade computacionais estudados serão descritos na seção 3.1, os sistemas especializados de sondagem estudados serão descritos em 3.2 e os sistemas de sondagem por pacotes estudados serão descritos na seção 3.3.

3.1 Sistemas de Grade Computacionais

O padrão Grade [8] define uma grade computacional como um *middleware* que integra recursos distribuídos de modo a criar um ambiente de execução unificado para as aplicações. Uma grade computacional permite o compartilhamento de recursos de *hardware* e *software* em sistemas distribuídos e heterogêneos, viabilizando, assim, a execução de aplicações que fazem uso intensivo de computação.

Diversos sistemas de grade computacional estão sendo desenvolvidos. Como exemplos, podemos citar os sistemas Globus, Legion, Condor, BOINC e os brasileiros OurGrid, *Brazilian National Computation Grid*, GridRio e InteGrade, entre tantos outros projetos. Dois sistemas de grade computacional foram estudados neste trabalho e serão descritos a seguir: Legion e OurGrid.

3.1.1 Legion

Legion [5] é um sistema de grade computacional que foi desenvolvido pela Universidade de Virginia. Um dos pioneiros em computação em grade, o desenvolvimento do Legion começou em 1993. Em 2001, seus idealizadores fundaram a Avaki, uma empresa que atualmente desenvolve e comercializa sistemas que utilizam a tecnologia do Legion.

O Legion foi desenvolvido utilizando-se o paradigma de orientação a objetos. Em Legion, todos os elementos da Grade são representados por objetos, sejam eles dados ou objetos reais, tais como microscópios, telescópios e outros equipamentos. Alguns dos objetivos da arquitetura são os seguintes:

- autonomia para diferentes domínios: cada domínio administrativo, isto é, cada conjunto de recursos computacionais aglomerados em uma única unidade administrativa,

pode especificar como seus recursos podem ser utilizados. Legion não impõe nenhuma política aos usuários;

- núcleo extensível: a arquitetura de Legion permite que partes do sistema sejam trocadas ou aprimoradas conforme as necessidades dos usuários;
- arquitetura escalável: o sistema é totalmente distribuído de modo a permitir grande escalabilidade;
- ambiente de computação homogêneo: a grade deve esconder as diferentes plataformas sobre as quais está executando;
- espaço de nomes único e persistente;
- aproveitamento de recursos heterogêneos: em algumas situações, pode ser conveniente executar uma aplicação em uma máquina mais adequada;
- suporte a múltiplas linguagens e interoperabilidade;
- tolerância a falhas;
- executar com poucos privilégios, ou seja, não deve exigir nenhum tipo de privilégio de super-usuário.

O Legion se destaca pela sua preocupação com o suporte à aplicações paralelas. O Legion possui uma implementação das bibliotecas MPI (*Message Passing Interface*) e PVM (*Parallel Virtual Machine*). Para utilizar um programa escrito em MPI ou PVM no Legion basta recompilá-lo utilizando as bibliotecas fornecidas pelo Legion. Isso permite que a migração da infra-estrutura antiga para a infra-estrutura do Legion seja praticamente instantânea.

Além das bibliotecas, Legion fornece suporte nativo a algumas linguagens de programação paralela. É possível utilizar as linguagens MPL (*Mentat Programming Language*, uma extensão de C++ para programação paralela), BFS (*Basic Fortran Support*) e Java.

Por fim, aplicações legadas que não utilizem nenhuma das bibliotecas ou linguagens acima podem ser encapsuladas dentro de objetos Legion. Basta o usuário registrar o programa legado com o comando *legion_register_program* e o sistema constrói um objeto Legion que encapsula o programa legado e, automaticamente, ele se torna elegível para ser executado pelo sistema. Isso garante que qualquer programa possa ser executado no ambiente do Legion. Se o programa realizar algum tipo de comunicação, basta que o programador escreva um adaptador para o programa legado que converta as chamadas a biblioteca de comunicação à chamadas a métodos de comunicação do Legion.

3.1.2 OurGrid

OurGrid [4] é um projeto de grade computacional desenvolvido em conjunto pela Universidade Federal de Campina Grande e Hewlett-Packard (HP). O objetivo do OurGrid é pesquisar e desenvolver soluções para uso e gerenciamento de grades computacionais. O projeto OurGrid é fundamentado no projeto MyGrid, desenvolvido também na Universidade Federal de Campina Grande, que propôs e desenvolveu uma implementação de um sistema de grade computacional.

O MyGrid é um sistema de grade computacional para aplicações do tipo *Bag-of-Tasks*, ou seja, para aplicações que sejam compostos por uma seqüência de tarefas independentes entre si. Um dos objetivos do projeto MyGrid é a simplicidade. Seus idealizadores acreditam que um sistema de grade que seja simples de usar é o que os desenvolvedores de aplicações de alto desempenho precisam hoje. Essa simplicidade se reflete tanto na arquitetura do sistema quanto em seu uso.

Para utilizar a infra-estrutura do MyGrid, o desenvolvedor necessita apenas informar quais as informações de entrada de cada tarefa e onde as informações de saída serão armazenadas. Não é necessário utilizar nenhuma biblioteca específica nem se preocupar com questões como o escalonamento das tarefas ou configuração do sistema de grade. Surpreendentemente, não é necessário instalar nenhum programa nos computadores que formarão a rede. É necessário apenas que o usuário tenha algum tipo de acesso remoto como, por exemplo, via SSH (*Secure SHell*), aos outros nós que compõe a grade.

A arquitetura do MyGrid também é muito simples. Há apenas um escalonador centralizado para todos os nós que compõem a grade. Toda tarefa a ser executada na grade deve ser enviada para este escalonador que, após analisar a carga de trabalho em todos os nós da grade, decidirá qual nó executará cada tarefa. Está prevista na arquitetura do MyGrid a capacidade de interoperabilidade com outros sistemas de grade computacional. Um protótipo de integração com o sistema Globus já está sendo desenvolvido pela equipe do MyGrid.

O sistema Legion permite a execução de programas paralelos que necessitem comunicação entre nós. O OurGrid só permite que problemas embarcadosamente paralelizáveis sejam executados, devido ao modelo de aplicações *Bag-of-Tasks* utilizado. Porém, o sistema Legion requer que o usuário especifique os nós em que as tarefas serão executadas, ao contrário do OurGrid, que possui um escalonador que define esses nós dinamicamente. O projeto InteGrade visa fornecer suporte para todos os tipos de programas paralelos,

atribuindo cada tarefa a um nó da grade de forma dinâmica. Com o uso de sistemas de sondagem de rede, o sistema de escalonamento de tarefas poderá distribuir as tarefas entre nós próximos, diminuindo os efeitos da degradação de desempenho decorrentes da comunicação entre nós.

3.2 Sistemas Especializados

Chamamos de sistemas de sondagem especializados aqueles que necessitam conhecer alguma característica específica da rede onde o sistema se encontra. Este trabalho analisou dois sistemas de sondagem especializados: SNMP e ReMoS.

3.2.1 SNMP

- O *Simple Network Management Protocol* [7] (SNMP), foi desenvolvido por volta de 1998 e, desde então, tornou-se o padrão de gerenciamento de redes “de facto” da indústria. O SNMP prima por sua simplicidade e flexibilidade. Implementar um agente SNMP em um produto requer muito pouco código e adicionar novas funcionalidades de gerenciamento de rede às existentes requer pouco esforço. Outra característica que tornou o SNMP popular é que a implementação da camada de gerenciamento é independente da implementação da camada de hardware, o que permite mesclar a implementação de múltiplos fornecedores.

SNMP contém dois tipos de elementos primários: gerentes (*managers*) e agentes (*agents*). Os gerentes são as interfaces pelas quais o administrador de rede realiza suas tarefas de gerenciamento. Agentes são entidades que fornecem a interface com o dispositivo que está sendo gerenciado. *Bridges*, *hubs* e roteadores são exemplos de dispositivos gerenciáveis pelo SNMP. Estes dispositivos contém o que SNMP denomina de “objetos gerenciáveis”. Os objetos gerenciáveis são características que estão relacionadas com a operação do dispositivo, como parâmetros de configuração, estatísticas de desempenho, *hardware*, etc. Esses objetos são agrupados em banco de dados virtual de informações sobre a rede, chamado MIB (*management information base*). SNMP permite que gerentes e agentes se comuniquem para permitir o acesso a esses objetos.

Um agente, tipicamente:

- armazena e obtém dados de gerenciamento definidos no MIB;
- notifica, de forma assíncrona, um gerente sobre a ocorrência de um evento;

- serve de proxy para um dispositivo de rede que não suporta SNMP.

Um gerente, tipicamente:

- requisita informações sobre os objetos disponibilizados por cada agente;
- são notificados, pelos agentes, da ocorrência de eventos;
- define variáveis de configuração dos agentes.

A comunicação entre agentes e gerentes SNMP não é orientada a conexão. Em outras palavras, não há conexão pré-estabelecida entre o agente e o gerente antes da transmissão de dados ser realizada. Em consequência disso, SNMP não oferece nenhuma garantia sobre o envio dos dados. A comunicação é realizada utilizando-se UDP (*User Datagram Protocol*) e IP (*Internet Protocol*).

A desvantagem de utilizar o SNMP como sistema de obtenção de recursos de rede por um sistema de grade computacional é que nem todos os dispositivos de redes possuem agem como agentes SNMP. Muitas vezes o dispositivo não provê suporte a SNMP ou o administrador do sistema desativou o suporte por motivos de segurança. Isso faz com que SNMP não seja uma opção genérica para sistemas de sondagem de rede.

3.2.2 ReMoS

O Remos [3] (*R**E**s**o**r**c**u**re* *M**o**n**i**t**o**r**i**ng* *S**y**s**t**e**m*) ou sistema de monitoramento de recursos) foi desenvolvido na Carnegie Mellon University e permite que aplicações obtenham informações relevantes sobre o ambiente de rede onde se encontram.

O Remos possui uma interface que disponibiliza as informações sobre o ambiente de rede de forma independente da tecnologia utilizada pela rede. Dessa forma, aplicações que utilizem o Remos podem se adaptar ao ambiente de execução ainda que este seja composto por redes que utilizem diferentes tecnologias.

As informações disponibilizadas pelo Remos são independentes da tecnologia utilizada. Isso significa que, apesar da aplicação utilizar informações coletadas pelo Remos, essas informações são independentes da tecnologia de rede utilizada. Dessa forma, a mesma aplicação poderá ser executada em uma rede que utilize outra tecnologia sem necessidade de mudanças no código-fonte da aplicação.

Outra característica interessante do Remos é sua capacidade de previsão. Os dados coletados durante a execução do Remos são armazenados para referência futura. Aplicações

que utilizam o Remos podem obter informações sobre um estado passado. Por exemplo, é possível obter informações sobre o estado da rede há três horas atrás. O sistema Remos possui uma heurística de predição de estado. Um programa pode pedir para o sistema prever como estará a rede daqui a dez minutos, por exemplo, e o Remos utilizará os dados coletados para realizar a previsão.

Há duas formas de obtenção de dados no Remos. A primeira é uma pesquisa de alto nível, denominada *flow-based queries*, que permite que consultas sobre largura de banda e latência sejam realizadas. A segunda forma de obtenção de dados são as pesquisas de baixo nível, denominada no Remos por *logical network topology*, que permitem que pesquisas sobre a topologia de rede sejam realizadas. No Remos, a topologia de rede é representada por um grafo onde os vértices são os nós da rede e as arestas representam os custos de comunicação entre os dois nós interligados pela aresta.

A implementação do Remos utiliza o protocolo de gerenciamento de rede SNMP, descrito na seção 3.2.1.

3.3 Sondagem por Pacotes

Os sistemas de sondagem de redes por pacotes (*packet-probing*) são aqueles que utilizam apenas troca de mensagens para inferir características da rede. Isso possibilita usar o mesmo sistema independentemente da tecnologia de rede utilizada, ao contrário dos sistemas especializados vistos na seção 3.2.

A seguir, apresentaremos diferentes técnicas de sondagem por pacotes, conforme classificadas por Schoonderwoerd [9].

3.3.1 Sondagem Ativa e Passiva

As várias técnicas usadas por sistemas de sondagem de rede podem ser separadas em duas categorias: sondagem ativa e sondagem passiva. Sondagem ativa significa que o sistema enviará,ativamente, pacotes de sondagem pela rede. Já os sistemas de sondagem passiva, monitoram o tráfego de rede, sem nenhum tipo de interferência.

Sistemas de sondagem ativa podem ser tanto de “apenas-envio” (*one-way*) ou “envio-e-resposta” (*two-way*). Programas como ping são do tipo de sondagem de apenas-envio. O ping envia uma mensagem e espera por uma resposta automática, utilizando ICMP e UDP

ou TCP. Sistemas de envio-e-resposta utilizam uma arquitetura cliente-servidor. O cliente envia um pacote de sondagem ao servidor, que executa alguns cálculos e responde ou apenas reenvia os dados recebidos ao cliente.

Sondagem passiva é considerado um método menos confiável do que sondagem ativa. Alguns pesquisadores afirmam que não é possível extrair nenhum dado útil do monitoramento da rede. Entretanto, há diversos esforços de utilização de sondagem passiva.

3.3.2 Atraso

O atraso de envio (*one-way delay*) é o tempo necessário para um pacote viajar de sua origem ao seu destino, utilizando um *timestamp* e relógios sincronizados. O tempo de envio e resposta (*round-trip time*) é medido enviando-se um pacote de tamanho pequeno a um servidor e contando o tempo necessário para que a resposta chegue.

3.3.3 Dispersão de Pacotes

Sistemas que utilizam dispersão de pacotes funcionam do seguinte modo. Utilizando-se pacotes do tipo UDP ou ICMP ECHO, seqüências de pacotes são enviados consecutivamente para o nó de destino, que reenvia os pacotes de volta. Ao passar por um link de menor capacidade, os pacotes serão enfileirados. Após este link, os pacotes terão um espaço entre eles devido ao enfileiramento. O espaço ou dispersão entre os pacotes é definido como o tempo entre a chegada do último bit do primeiro pacote e o tempo da chegada do último bit do segundo pacote. Após passar pelo link de menor capacidade, o tamanho da dispersão, teoricamente, não ficará maior. A dispersão média entre os pacotes da seqüência é medido e a capacidade \hat{C} , em bytes por segundo, é dado pela fórmula:

$$\hat{C} = \frac{P}{t_{dispersao}},$$

onde P é a média do tempo de dispersão da seqüência, em segundos e $t_{dipersao}$ é a dispersão média, em bytes por segundo. A capacidade \hat{C} é, então, usada como estimativa para a capacidade por link.

3.3.4 Pacotes de Tamanhos Variados

A técnica de pacotes de tamanhos variados foi utilizada pela primeira vez na ferramenta *pathchar* [10]. A idéia básica é que a ferramenta mande um pacote onde o campo TTL (*time-to-live*) do cabeçalho IP do pacote é definido com um valor específico. Cada dispositivo da camada três de rede (TCP/IP) irá decrementar o valor do TTL em um. Quando o valor de TTL atingir zero, o dispositivo enviará uma mensagem de erro do tipo ICMP TTL-exceeded para o remetente do pacote. Ao receber essa mensagem de erro, a ferramenta pode estimar o tempo de envio e resposta. O tempo de envio e resposta é medido várias vezes para diferentes tamanhos de pacote. Com uma certa dose de sorte, alguns pacotes de resposta chegarão de volta ao remetente sem serem enfileirados nenhuma vez por nenhum dispositivo de rede. Estes representarão os menores tempos de envio e resposta para determinado tamanho de pacote. Os tempos obtidos permitirão que seja estimada a largura de banda do link de menor capacidade.

3.3.5 Roteamento e Topologia

Esta metodologia é a mais antiga e utilizada de todas. Desde 1988 a ferramenta *traceroute*, criada por Van Jacobson, é utilizada para descobrir por quais nós um pacote passa para chegar a um nó destino em particular. Desta forma, uma pequena parte da topologia de rede se torna visível. Porém, nem todos os pacotes necessariamente seguem um mesmo caminho. Todos as ferramentas que realizam medição por link, isto é, medem as características de rede levando em consideração todo o caminho que os pacotes percorrem, utilizam a mesma técnica. Descobrem todos os dispositivos de rede nível três, enviam pacotes de rede com o campo TTL modificado e esperam os pacotes com as mensagens de erros chegarem.

Devido a diferenciação que os equipamentos de roteamento de rede fazem entre pacotes ICMP e IP, os resultados produzidos por esta técnica não são confiáveis. Entretanto, nota-se que, atualmente, a utilização de outras técnicas de sondagem em conjunto com técnicas de roteamento e topologia estão se tornando uma tendência.

Pode-se notar que os sistemas de grade computacionais tradicionais não consideram as características de rede como um recurso do sistema de grade. Apenas assume-se que a rede existe, e que todos os nós são alcançáveis a partir de qualquer outro, ou seja, o grafo que representa a topologia e a comunicação entre os nós é completo. Nota-se, também, que sistemas de sondagem especializados produzem resultados mais precisos do que os de

sondagem por pacotes, mas necessitam de informações sobre a tecnologia de rede utilizada, o que nem sempre está disponível em ambientes heterogêneos como a Internet. Por fim, temos que sistemas de sondagem por pacotes produzem estimativas das características do ambiente de rede, mas não modelam bem equipamentos de rede que não operam na camada três de rede (TCP/IP).

4 Atividades Realizadas

As atividades realizadas, em ordem cronológica, foram:

- Estudo do protocolo de gerenciamento de rede SNMP;
- Estudo do sistema ReMoS de obtenção dinâmica do estado de rede;
- Familiarização com as diferentes linhas de pesquisa em grades computacionais [6];
- Análise do sistema de grade computacional OurGrid;
- Estudo do sistema de grade Legion, com enfoque no gerenciamento de recursos de rede;
- Familiarização com o conceito de sondagem por pacotes [11] e pesquisa de ferramentas que implementassem esse conceito;
- Estudo das ferramentas *netperf* [12], *pathchar* [10], *igi* [13], *nettimer* [14].

5 Conclusão

Neste estudo, analisamos como alguns sistemas de grade computacional tratam as limitações de desempenho que as redes oferecem. Pesquisamos, também, diversos métodos de sondagem de rede que permitissem que os recursos computacionais disponíveis para sistemas conscientes da rede pudessem obter informações sobre o ambiente de rede de forma dinâmica.

Os sistemas de grade tradicionais não consideram as características de rede como, por exemplo, largura de banda disponível, um recurso do sistema de grade computacional. Informações desta natureza são fundamentais para aplicações paralelas de alto desempenho. No futuro, algoritmos de escalonamento de tarefas [2] farão uso destas informações para minimizar a degradação de desempenho causada pela comunicação feita entre nós da grade.

Entre os projetos de grade computacional que tratarão as características de rede como recursos, podemos citar o projeto Globus e o InteGrade.

A pesquisa sobre métodos de sondagem de rede nos mostrou que sistemas especializados normalmente produzem resultados mais precisos. Porém, é necessário que o sistema conheça com antecedência qual o tipo de tecnologia de rede utilizada entre os nós que participam da rede. Essa abordagem não é adequada para grades computacionais, onde o ambiente de rede computacional é heterogêneo e dinâmico, uma vez que qualquer computador interligado a internet pode começar a participar da grade ou sair dela.

Os métodos de sondagem por pacotes, em sua maioria, não modelam corretamente os diferentes equipamentos de rede nem a utilização da rede por outros aplicativos. Equipamentos que operam apenas no nível dois da camada de rede, como alguns roteadores, não são percebidos por métodos que utilizam apenas pacotes IP. Métodos que utilizam ICMP detectam esses equipamentos, porém mensagens ICMP são tratados com menor prioridade do que mensagens IP e, portanto, levam a resultados menos precisos. Certas especificidades do sistema operacional e do dispositivo de rede como, por exemplo, o modo como os pacotes são enfileirados no dispositivo de rede se a conexão física estiver sendo usada, deveriam ser considerados por estes métodos para produzir um resultado mais preciso. Mas isso tornaria inviável a criação de um sistema genérico que seja independente da tecnologia utilizada.

No ambiente de grade computacional, a escolha do método de sondagem de pacote deve ser cuidadosa. O método deve ser rápido e não intrusivo, ou seja, o método não pode degradar o desempenho da comunicação de rede das aplicações que estão sendo executadas nos nós da grade. O sistema deve assegurar, também, que um mesmo nó não participe de duas sondagens em paralelo, pois isso resultaria em resultados enviesados.

Referências

- [1] InteGrade: Rumo a um Sistema de Computação em Grade para Aproveitamento de Recursos Ociosos em Máquinas compartilhadas. Andrei Goldchleger, Fabio Kon, Alfredo Goldman Vel Lejbman, Marcelo Finger e Siang Wun Song. Relatório Técnico MAC-IME-USP, outubro 2002.
- [2] A model for parallel job scheduling on dynamical computer grids. Alfredo Goldman e Carlos Queiroz. ACM/IFIP/USENIX International Workshop on Middleware for Grid Computing, Rio de Janeiro, junho 2003.

- [3] ReMoS: A resource monitoring system for network aware applications. DEWITT, T., GROSS, T., LOWEKAMP, B., MILLER, N., STEENKISTE, P., SUBHLOK, J., AND SUTHERLAND, D. Tech. Rep. CMU-CS-97-194, School of Computer Science, Carnegie Mellon University, dezembro 1997.
- [4] Open Grid: A User-Centric Approach for Grid Computing. Walfredo Cirne and Keith Marzullo, Proceedings of the 13th Symposium on Computer Architecture and High Performance Computing, setembro 2001.
- [5] The Core Legion Object Model. M. J. Lewis e A. Grimshaw, Proceedings of the Fifth IEEE International Symposium on High Performance Distributed Computing, agosto 1996.
- [6] The grid forum <http://www.gridforum.org/>.
- [7] Simple Network Management Protocol <http://www.snmplink.org/>.
- [8] Grade: um Padrão Arquitetural. Andrei Goldchleger, Márcio Rodrigo de Freitas Carneiro e Fabio Kon. Relatório Técnico MAC-IME-USP, março 2003.
- [9] Network Performance Measurement Tools - a Comprehensive Comparison. Rody Schoonderwoerd. Tese de mestrado, Vrije Universiteit, Amsterdam, novembro 2002.
- [10] Using pathchar to Estimate Link Characteristics. Allen Downey. In Proceedings of the ACM SIGCOMM, 1999.
- [11] Packet Probing for Network-aware Scientific Applications in Cluster Environments, Sam Storie e Masha Sosonkina. HeteroPar'03.
- [12] Nettest: An Tool to Measure the Maximum Burst Size, Available Bandwidth and Achievable Throughput. Guojun Jin e Brian Tierney. Disponível em <http://www-didc.lbl.gov/NCS/>.
- [13] Evaluation and Characterization of Available Bandwidth Probing Techniques. Ningning Hu e Peter Steenkiste. In the IEEE JSAC Special Issue in Internet and WWW Measurement, Mapping, and Modeling, Vol. 21(6), agosto 2003.
- [14] Nettimer: A Tool for Measuring Bottleneck Link Bandwidth. Kevin Lai e Mary Baker. Proceedings of the USENIX Symposium on Internet Technologies and Systems, março 2001.