

Monitoramento de Clusters e Grids Computacionais Utilizando o Ganglia

Autores

Jose Luis Zem

Samuel Henrique Bucke Brito

1. Introdução

Os sistemas de computação de alto desempenho e de alta disponibilidade fazem uso de *hardwares* específicos e caros, os quais implementam técnicas de disponibilidade e confiabilidade através de componentes gerenciadores de redundantes que contém a lógica para que os mesmos possam se recuperar de possíveis falhas [1]. À medida que a tecnologia se desenvolve, os supercomputadores tornam-se cada vez mais potentes [2] e, ao mesmo tempo, os preços também crescentes, acabam por inviabilizar o uso dos mesmos.

Com o desenvolvimento das tecnologias de interconectividade e interoperabilidade das redes de computadores e de comunicação, são propostas arquiteturas de organização de computadores que utilizam o poder e recursos computacionais de várias máquinas interligadas. Tais arquiteturas propõem a interconexão de computadores atuando em conjunto e somando seus recursos, comportando-se como um ambiente único, equivalente a um supercomputador.

1.1 Clusters de Computadores

Clusters de Computadores podem ser definidos como sendo um conjunto composto por dois ou mais computadores interconectados através de alguma tecnologia de rede de comunicação, trabalhando juntos para fins comuns [3].

A idéia consiste em unir o poder computacional de todos os nós deste agrupamento de computadores para oferecer a computação de alto desempenho e de alta disponibilidade, criando um ambiente computacional único.

Assim, os *clusters* necessitam de técnicas de conectividade e comunicação entre os processos executados nos diferentes nós que os diferenciam do modelo cliente-servidor, uma vez que a finalidade dos *clusters* vai além da troca de solicitações e serviços entre clientes e servidores, sugerindo um relacionamento mais colaborativo.

1.1.1 Contextualização para *Clusters* de Computadores

Atualmente existe um interesse crescente em relação aos *clusters* como consequência de fatores resultantes do desenvolvimento da tecnologia. Esse crescimento ocorre em conjunto com o surgimento de novas tecnologias e estudos de técnicas de computação que necessitam da infra-estrutura de computação presente nos *clusters*. Dentre os fatores responsáveis por esse interesse destacam-se desde a difusão da computação de alto desempenho e de alta disponibilidade até a padronização de modelos e técnicas de interconexão de redes.

A infra-estrutura baseada em *clusters* é necessária para a execução de aplicações paralelas, sendo que o paralelismo é uma maneira de se conseguir melhorar o desempenho e superar o gargalo da velocidade de processamento.

1.1.2 Arquitetura de *Clusters* de Computadores

A arquitetura de um *cluster* representa um tipo de sistema distribuído idealizado para a realização de processamento paralelo por um conjunto de máquinas interconectadas, através de alguma tecnologia de rede e protocolo de comunicação.

Nesta arquitetura, tem-se a abstração de um ambiente computacional único equivalente à união de todos os nós individuais. Esta abstração fica sob responsabilidade de alguns *softwares* específicos denominados *middleware*, executados transparentemente nos sistemas operacionais.

Cada nó pode possuir arquiteturas diferentes de *hardware*, desde que a tecnologia de interconexão das máquinas seja compatível e haja a interoperabilidade entre os protocolos de comunicação e sistemas operacionais.

Podem ser observados os seguintes elementos presentes no modelo proposto em [4] para a arquitetura de um *cluster*: *Backbone*, *Interface de Rede*, *Protocolo de Comunicação*, *Nó ou Máquina*, *Middleware*, *Ambientes de Programação Paralela*, *Aplicativos Paralelos* e *Seqüenciais*.

1.1.3 Categorias de *Clusters* de Computadores

Fatores externos e necessidades diferentes podem interferir na maneira como serão os arranjos dos computadores baseados em *clusters*; estas necessidades específicas podem influenciar na organização e funcionalidades do *cluster*.

Podem ser encontrados diversos tipos de *clusters*, porém os que se destacam são os de [1]:

- Alto Desempenho.

- Alta Disponibilidade.
- Balanceamento de Carga.

1.2 Grid Computing

Esses ambientes normalmente dizem respeito a um grande número de nós conectados e sugerem a conexão de *clusters* geograficamente dispersos pelo mundo, através da infra-estrutura da Internet. Essa idéia de conexão para uso da computação sobreposta deu origem a um novo campo na computação distribuída, denominado *metacomputação*.

O termo metacomputador é definido como sendo um supercomputador virtual, composto dinamicamente por recursos geograficamente distribuídos e interconectados através de enlaces de alta velocidade, porém entende-se hoje, que *grid computing* é o termo mais adequado para denotar este sistema, embora ainda haja divergências [5].

O termo *grid computing* (computação em grade) surgiu como uma proposta para denotar um sistema distribuído de computação que pudesse oferecer serviços, que são recursos computacionais, de acordo com a real necessidade de seus usuários e de maneira transparente [6, 7].

A idéia deste modelo reside em coordenar os recursos computacionais do maior número possível de computadores de maneira descentralizada usando padrões, interfaces e protocolos abertos para oferecer alto desempenho com qualidade de serviços sob demanda ao usuário do *grid* e de maneira transparente.

1.2.1 Contextualização para Grid Computing

Pelo fato de tratar-se de um assunto relativamente recente, ainda não existe uma padronização dos conceitos relacionados aos *grids* e encontram-se diferentes soluções de *software* que buscam essa interconexão maciça.

Algumas literaturas apresentam a idéia de um esforço coletivo mundial para construção de um *grid* global em que haja interconexão de máquinas espalhadas por todas as partes do mundo e com gerenciamento descentralizado, visando à disponibilidade contínua de serviços e poder absoluto de processamento. Porém, esta é uma idéia utópica e que tem como objetivo, apenas demonstrar o quão poderoso poderia ser este ambiente se não existissem tantas divergências políticas e culturais [6].

1.2.2 Arquitetura de Grid Computing

A arquitetura do *grid* é baseada em um modelo de sistema distribuído que virtualiza os recursos computacionais de uma infra-estrutura de comunicações complexa composta por vários computadores, através do uso de padrões abertos que viabilizam a interoperabilidade de sistemas heterogêneos [8], sendo que seus nós podem ser de qualquer tipo, desde simples estações até complexos ambientes

computacionais.

Como não existe um padrão internacionalmente definido para esta arquitetura apresenta-se em [9, 7 e 10] um modelo dividido em camadas e elaborado a partir de arquiteturas propostas. O modelo em questão é composto pelas camadas *Conectividade*, *Middleware do Sistema* e *do Usuário e Aplicação*.

O Ganglia é um sistema de monitoramento distribuído desenvolvido para ser executado em ambientes de computação de alto desempenho [11], nos quais se enquadram os *clusters* e *grids*.

Ele consiste em um *software open-source*, que teve sua origem na Universidade da Califórnia, *Campus* de Berkeley. Baseia-se em um protocolo que faz uso da técnica de difusão *multicast* para comunicação entre os nós e utiliza uma estrutura de árvores para representar as conexões ponto-a-ponto [12]. Utiliza tecnologias como o XML, o XDR e o RRDtool; é um sistema portátil para uma variedade de sistemas operacionais, além de diversos processadores.

Originalmente utilizado para monitorar *clusters* universitários interligados em ambientes de *grids*, é considerado um sistema confiável, escalável, leve e robusto e atualmente está em uso em mais de 500 ambientes de *clusters* e *grids* computacionais espalhados pelo mundo [12].

1.3 Ganglia

1.3.1 Desafios do Monitoramento Distribuído

São muitas as dificuldades na implementação de sistemas para o monitoramento de ambientes distribuídos, em função da complexidade destes mesmos ambientes, sendo a maior delas a de permitir o monitoramento do estado global do sistema à medida que estes crescem e aumentam a heterogeneidade dos componentes [12].

Isso acontece porque quanto maior o número de nós, maior será o tráfego na rede e o número de atividades de E/S, o que refletirá no aumento da probabilidade de falhas ocorrerem.

Com isso, as falhas passam a ser comuns, tornando necessário um sistema de monitoramento capaz de identificar e recuperar-se, sem comprometer as atividades que estejam sendo executadas naquele momento.

Outros desafios relevantes, além da escalabilidade, são a robustez, o gerenciamento, a portabilidade e o tráfego gerado pelo sistema [12].

1.3.2 Arquitetura do Ganglia

O Ganglia é composto por dois *daemons* (*gmond* e *gmetad*), um programa de linha de comando (*gmetric*), um programa de estado (*gstat*) e por bibliotecas instaladas na máquina cliente [12].

O *gmond* (*ganglia monitoring daemon*) deve ser executado em todos os nós e é responsável por responder às solicitações dos clientes enviando os dados monitorados e pela comunicação entre os múltiplos *daemons* *gmond* assegurando que todos os outros nós estejam sempre atualizados em relação ao estado global do ambiente. Já o *gmetad* (*ganglia meta daemon*) viabiliza o agrupamento de *clusters* em ambientes de *grid*, implementando uma árvore hierarquizada.

O sistema possui por padrão algumas métricas, porém permite o uso do *gmetric*, que possibilita ao administrador inserir suas próprias métricas para que o sistema realize o monitoramento desejado.

2. Objetivos

Este artigo apresenta o resultado de um estudo sobre a computação de alto desempenho e de alta disponibilidade, através do levantamento bibliográfico sobre sistemas distribuídos organizados em ambientes de *clusters* de computadores e *grid computing*, propondo a utilização de uma ferramenta *open-source*, no caso o *Ganglia*, para auxiliar na tarefa de gerenciamento e monitoramento destes ambientes computacionais distribuídos.

3. Desenvolvimento

Foi construída uma infra-estrutura distribuída com computadores arranjados em *cluster* para a execução dos procedimentos de instalação, configuração e teste da ferramenta de monitoramento. A infra-estrutura contou com 5 computadores sendo que 4 atuaram como nós de computação e 1 como nó monitorador.

Optou-se por utilizar um nó monitorador conectado ao *cluster*, sem compartilhamento de recursos e responsável por solicitar o estado do sistema diretamente aos nós participantes e exibir as saídas do monitoramento em um *front-end*.

O nó monitorador foi configurado com os serviços de *cgi gateway*, *daemons gmond* e *gmetad*, o aplicativo *rrdtool* e com o servidor *http* com suporte à linguagem *php* e os demais nós do *cluster* foram simplesmente instalados e configurados com o *daemon gmond*.

4. Resultados

Após a instalação e configuração do Ganglia foram realizados os procedimentos para inicializar os serviços do sistema monitorador e em seguida foram executados os processos *gmond* nos demais nós do ambiente distribuído. A figura 1 apresenta os resultados obtidos ao acessar-se o resumo do *cluster* monitorado.

O objetivo principal dos testes foi o de analisar qual seria a reação do sistema monitorador ao deparar-se com uma falha em um dos nós do *cluster*. Para simular a falha, optou-se por encerrar a execução do *daemon gmond* no nó denominado *nodo2*. A figura 2 traz a saída obtida referente ao acesso detalhado do *nodo2*, ainda antes do processo *gmond* ser encerrado.

Pode ser observado pela figura 2 que o *nodo2* encontrava-se em perfeita atividade no *cluster* e neste momento o processo *gmond*, que estava sendo executado no nó em questão foi finalizado. O sistema imediatamente reconheceu que o nó não estava mais em atividade e seu estado foi automaticamente atualizado (figura 3).

5. Considerações Finais

Os objetivos de se realizar o levantamento teórico e implementar o sistema de monitoramento foram atingidos plenamente, porém torna-se interessante realizar novos experimentos com uma quantidade maior de nós arranjados como um *cluster* de computadores, bem como buscar a construção de um *grid* para a comprovação do funcionamento do ambiente.

Referências Bibliográficas

- [1] Ferreira, Luis; Berstls, Viktors; Armstrong, Jonathan et al. *Introduction to Grid Computing with Globus*. IBM. Setembro/2003.

- [2] IBM©. *IBM Blue Gene Project*. (01.01.2006).

- [3] Baker, Mark et al. *Cluster Computing White Paper*. University of Portsmouth.UK. 2000. Acesso em Março/2005 em .

- [4] Baker, Mark; Buyya, Rajkumar. *Cluster Computing at a Glance, High Performance Cluster Computing: Architectures and Systems*. Vol. 2. New Jersey/Estados Unidos. P. 3-47. /1999.

- [5] Foster, Ian; Kesselman, Carl. *Globus: A Metacomputing Infrastructure Toolkit*. International Journal of Supercomputer Applications and High Performance Computing. 1997.

- [6] Ferreira, Luis; Lucchese, Fabiano; Yasuda, Tomoari et al. *Grid Computing in Research and Education*. IBM. Abril/2005.

- [7] Chetty, Madhu; Buyya, Rajkumar. *Weaving Computational Grids: How Analogous Are they with Electrical Grids?* IEEE Computer Society. Piscataway/Estados Unidos. P. 61-71. Agosto/2002.

- [8] Maldonado, Martin F. *Grid computing in higher education: Trends, values and offerings*. Demmand

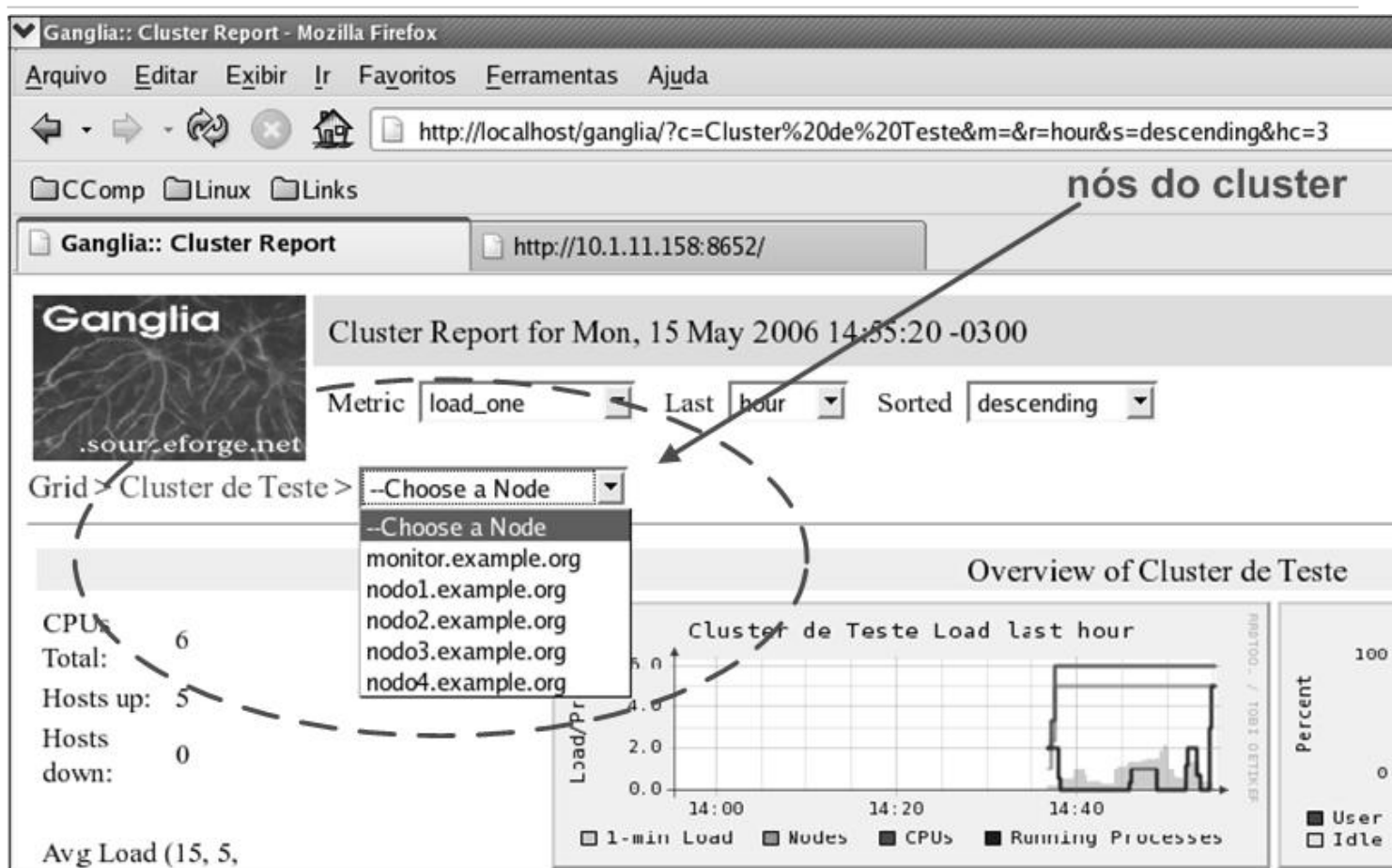
[9] Baker, Mark; Buyya, Rajkumar; Laforenza, Domenico. *Grids and Grid technologies for wide-area distributed computing*. Software – Practice and Experience. John Wiley & Sons Ltd., 2002. 30p.

[10] Casanova, Henri. *Distributed Computing Research Issues in Grid Computing*. ACM SIGACT News Distributed Computing Column 8. California, Julho/2002. 21p.


[11] Ganglia Project. *Ganglia Monitoring System*". (01.01.2006).

[12] Massie, Mathew L.; Chun, Brent N.; Culler, David E. *The ganglia distributed monitoring system: design, implementation, and experience*. Berkeley/Estados Unidos. P. 817-840. Junho/2004.

Anexos



This host is up and running.



nodo2

Time and String Metrics


boottime	Mon, 15 May 2006 11:00:18 -0300
gexec	OFF
gmond_started	Mon, 15 May 2006 11:38:12 -0300
last_reported	0 days, 0:00:11
machine_type	x86
os_name	Linux
os_release	2.4.28
uptime	0 days, 4:04:21



Host Report for Mon, 15 May 2006 15:08:25 -0300

Last

This host is down.



nodo2

Time and String Metrics

Constant Metrics

Gmetrics