Monitoração de *clusters* com a ferramenta Ganglia: avaliação e adaptação

Marcelo Veiga Neves, Tiago Scheid, Andrea Schwertner Charão

[1]Laboratório de Sistemas de Computação - LSC
Curso de Ciência da Computação – Universidade Federal de Santa Maria - UFSM
Informática/CT - UFSM Campus - 97105-900, Santa Maria, RS

{veiga,scheid,andrea}@inf.ufsm.br

Resumo. *Neste artigo apresenta-se uma avaliação de Ganglia, que é uma das principais ferramentas de Software Livre para monitoração de* clusters *de computadores. Descreve-se também algumas adaptações efetuadas nesta ferramenta, tornando-a capaz de monitorar programas paralelos e permitindo a sua integração com outras ferramentas de monitoração. Estas adaptações ampliam as possibilidades de utilização de Ganglia, contribuindo para sua valorização junto à comunidade de Software Livre aplicado ao processamento de alto desempenho.*

1. Introdução

O processamento paralelo vem se tornando mais popular em função da demanda por alto desempenho, exigido por diversas áreas da ciência e indústria. Dentre as arquiteturas paralelas da atualidade, os *clusters* de computadores destacam-se por sua vantajosa relação custo-desempenho. Basicamente, um *cluster* é um conjunto de computadores (nós) independentes interligados, trabalhando de forma integrada como se fossem um recurso computacional único. Estas arquiteturas geralmente utilizam componentes (processadores, adaptadores de rede, etc.) produzidos em larga escala, permitindo obter alto desempenho a um custo relativamente baixo.

A utilização eficiente de um *cluster* geralmente necessita de alguma forma de monitoração de seus recursos, através da coleta e apresentação de dados sobre o estado do *cluster* ao longo do tempo. Existem atualmente diversas ferramentas para monitoração de *clusters*, muitas delas distribuídas sob licenças de Software Livre. Este artigo apresenta uma avaliação de Ganglia[Massie et al., 2004], que é uma das mais populares ferramentas livres para monitoração de *clusters*. Descreve-se também algumas adaptações efetuadas a fim de aumentar as funcionalidades de Ganglia e integrá-lo com outras ferramentas.

2. Ferramentas de Monitoração

Dentre algumas ferramentas para monitoração de *clusters* disponíveis atualmente, pode-se destacar Parmon, SCMS, RVision, PCP e Ganglia. O restante desta seção apresenta brevemente as quatro primeiras ferramentas, enquanto Ganglia é apresentado em maior profundidade na seção seguinte.

Parmon [Buyya, 2000] é uma ferramenta proprietária, disponível para Solaris e GNU/Linux RedHat 8.0 ou superior. Esta ferramenta permite monitorar de forma integrada os recursos dos vários nós que compõem um *cluster* (CPU, memória, rede, discos, processos, etc.). Sua arquitetura é composta por processos servidores (*parmond*),

que coletam informações em cada nó do *cluster*, e por um ou mais processos clientes que consultam e centralizam os dados monitorados, apresentando-os em uma interface gráfica.

SCMS[Uthayopas and Rungsawang, 1999] é uma ferramenta de código aberto disponível sob licença baseada em BSD. A arquitetura de SCMS é composta por um módulo monitor e um módulo centralizador, o qual mantém um histórico e responde a requisições de clientes. SCMS oferece uma interface de programação para que diferentes programas façam uso dos dados monitorados.

A ferramenta RVision[Ferreto et al., 2002] é distribuída sob licença GPL. Sua arquitetura é composta por processos monitores, que executam em cada nó do *cluster*, e por um processo que centraliza os dados monitorados. RVision não possui um módulo de apresentação dos dados coletados, mas fornece uma interface para comunicação com processos clientes.

PCP[Goodwin, 2005] é composto por um conjunto de ferramentas e uma biblioteca, distribuídos respectivamente sob as licenças GPL e LGPL. A arquitetura de PCP é composta por um ou vários coletores e um centralizador, localizados na mesma máquina. A visualização dos dados se dá através de um cliente implementado com auxílio de uma interface de programação.

3. Ferramenta Ganglia

Ganglia possui uma arquitetura distribuída que permite monitorar sistemas como *clusters* de larga escala e grades computacionais compostas por federações de *clusters*. Mais recentemente, Ganglia vem sendo usado em um sistema de escala planetária, o PlanetLab [Peterson et al., 2002]. Ganglia é um Software Livre distribuído sob a licença BSD, disponível para vários sistemas operacionais (incluindo GNU/Linux e FreeBSD) e várias arquiteturas de hardware.

A implementação de Ganglia consiste em dois tipos de processos *daemons* (*gmond* e *gmetad*), um programa de linha de comando (*gmetric*) e uma biblioteca para implementação de clientes. A monitoração de um único *cluster* é realizada por *gmond* (*Ganglia Monitoring Daemon*), sendo que este deve estar presente em todos os nós. O processo *gmond* responde a requisições de clientes retornando uma representação em XML dos dados coletados. O processo *gmetad* (*Ganglia Meta Daemon*) é responsável por monitorar uma federação de *clusters*. Uma árvore de conexões entre vários *daemons gmetad* permite agregar as informações de vários *clusters*. O programa *gmetric*, por sua vez, permite estender as métricas monitoradas por Ganglia.

O processo *gmond* faz uso de um protocolo de escuta/anúncio baseado em *multicast* para manter o estado do *cluster*. Este processo monitora dois tipos de métricas: as pré-definidas e as definidas pelo usuário. O primeiro tipo é formado pelas métricas coletadas pelo próprio sistema, como por exemplo porcentagem de uso de CPU, carga média, uso de memória, rede, etc. Já as métricas definidas pelo usuário são informações arbitrárias coletadas por programas externos e incorporadas em Ganglia através de uma interface específica.

Por motivo de desempenho, *gmond* utiliza limiares (*thresholds*) altos, assim uma métrica coletada só é enviada pela rede quando sofrer uma mudança significativa. Além disso, *gmond* utiliza intervalos de coleta e publicação aleatórios para evitar sincronizações entre o envio das métricas pelos nós.

Ganglia usa RRDtool (*Round Robin Database*) para armazenar e visualizar os

dados monitorados. RRDtool é um banco de dados compacto e de tamanho constante, comumente utilizado para armazenar séries temporais de dados. RRDtool gera gráficos (métrica *versus* tempo) para diferentes granularidades de tempo (de minutos até anos). Estes gráficos são usados por Ganglia e apresentados ao usuário através de uma interface Web em PHP (figura 1).

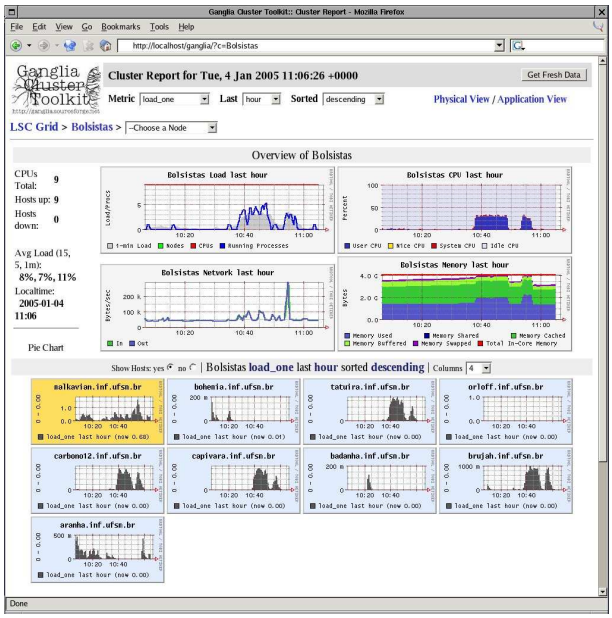

Figura 1: Interface Web de Ganglia.

Além disso, é importante mencionar que Ganglia está sendo incorporado em algumas distribuições de software para *clusters*, como ROCKS[Hoffman, 2005] e OSCAR[des Ligneris et al., 2003].

4. Avaliação de Ganglia

A ferramenta Ganglia está sendo utilizada no Laboratório de Sistemas de Computação da Universidade Federal de Santa Maria há aproximadamente um ano, para auxiliar no gerenciamento dos recursos computacionais deste grupo de pesquisa. A escolha desta ferramenta foi motivada por sua facilidade de instalação e utilização, capacidade de apresentação dos dados de forma simples e significativa, e por ser uma das ferramentas mais utilizadas.

O grupo LSC possui um *cluster* composto por 12 máquinas com bi-processadoras (Pentium III a 1GHz), compartilhado entre usuários finais e pesquisadores/desenvolvedores de ferramentas para processamento paralelo. Além disso, o laboratório possui uma rede local com diversos computadores pessoais que também são esporadicamente dedicados ao processamento paralelo.

Neste período de utilização de Ganglia, foi possível verificar algumas qualidades e limitações desta ferramenta. Uma importante qualidade de Ganglia é sua baixa intrusividade, devido a intervalos variados para coleta e publicação de métricas. Além disso, a ferramenta implementa a descoberta automática de nós que são adicionados e/ou removidos do *cluster*, o que facilita o gerenciamento dos recursos. Outra característica a ser ressaltada é que Ganglia resiste a possíveis falhas em nós do sistema, uma vez que cada nó possui o estado completo do *cluster* (redundância de dados). Vale igualmente mencionar a extensibilidade de Ganglia, caracterizada principalmente pela facilidade de definição de novas métricas de monitoramento através do programa *gmetric*.

Dentre as limitações detectadas, pode-se mencionar a incapacidade de Ganglia em monitorar a execução de processos. Esta característica, oferecida por exemplo em Parmon, permitiria coletar dados sobre a execução de programas paralelos no *cluster*. Outra limitação diz respeito à falta de interoperabilidade entre Ganglia e outras ferramentas que coletam dados em sistemas distribuídos. Por fim, pode-se citar a existência de somente uma interface Web para visualização das métricas monitoradas, o que limita as formas de apresentação dos dados.

5. Adaptação da Ferramenta

Aproveitando a extensibilidade de Ganglia e sua disponibilização como Software Livre, foram implementadas algumas adaptações nesta ferramenta a fim de contornar parte das limitações descritas na seção anterior. Num primeiro momento, incorporou-se a Ganglia a habilidade de monitorar uma dada aplicação ao longo de sua execução, através da coleta de informações sobre os processos que a compõem. Num segundo momento, implementou-se em Ganglia a capacidade de aproveitar os dados coletados por outras ferramentas de monitoração. Estas adaptações são descritas em maior detalhe nas seções que seguem, e encontram-se disponíveis em `http://www.inf.ufsm.br/lsc/ganglia`.

5.1. Monitoração de Processos com Ganglia

Primeiramente, foram implementadas rotinas para coleta de informações relativas a um dado processo, como a quantidade de recursos utilizados. Essas rotinas foram usadas na construção de um programa monitor que é lançado em todos os nós do *cluster*, juntamente com a aplicação paralela. Cada monitor localiza os processos da aplicação no nó local e acompanha sua execução, coletando métricas e enviando-as diretamente ao *multicast* de Ganglia, através do programa *gmetric*.

Na monitoração de programas paralelos, o intervalo de coleta de dados é um ponto crítico, uma vez que a dinamicidade das aplicações requer um acompanhamento com uma pequena granularidade de tempo. Para tornar a visualização mais significativa, foi necessário eliminar a aleatoriedade e reduzir o intervalo de publicação de Ganglia. Para manter a escalabilidade, que é uma das principais características da ferramenta, optou-se por reduzir apenas os intervalos de publicação das métricas definidas pelo usuário, já que o intervalo reduzido só é necessário enquanto a aplicação estiver em execução.

Quando uma nova métrica é publicada no *multicast*, o centralizador *gmetad* cria uma base de dados para armazenar as novas informações de monitoração. A visualização dessas informações é feita através de gráficos criados com RRDtool. Para facilitar a visualização foi implementada uma extensão da interface Web de Ganglia, agrupando as informações de uma mesma aplicação e permitindo a seleção do intervalo de tempo a ser visualizado.

Esta adaptação de Ganglia foi utilizada para monitorar uma aplicação de computação científica que faz parte do pacote PETSc[Balay et al., 2001]. Para execução da aplicação, utilizou-se 4 computadores (Intel Pentium 4 a 2.40 GHz) da rede do laboratório de pesquisa, nos quais foi instalada a versão adaptada de Ganglia. Estes computadores foram dedicados exclusivamente para a execução da aplicação, de modo semelhante a um *cluster*.

Os gráficos da figura 2 apresentam informações sobre uma execução da aplicação em 2 nós da rede, ao longo de aproximadamente 3 minutos. Nestes gráficos, a métrica monitorada é o uso da CPU em cada nó. No primeiro gráfico, nota-se um alto percentual

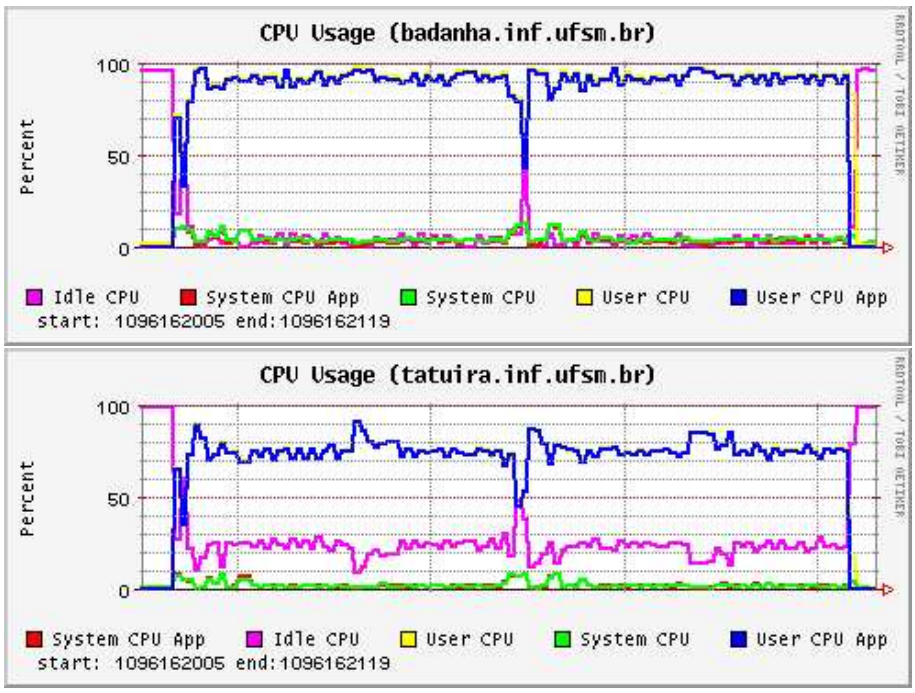

Figura 2: Monitoração do uso de CPU por uma aplicação.

de utilização da CPU (User CPU App), enquanto que no segundo observa-se uma subutilização de recursos, caracterizada por uma maior ociosidade da CPU (Idle CPU). Este comportamento evidencia um desequilíbrio de carga nesta execução da aplicação. É importante ressaltar que a monitoração dos tempos de CPU alocados à aplicação não seria possível com Ganglia sem a extensão implementada.

5.2. Interoperabilidade com outras Ferramentas de Monitoração

Como mencionado anteriormente, Ganglia permite monitorar federações de *clusters* de forma hierárquica. No entanto, Ganglia exige que os *clusters* tenham um processo *gmond* instalado em todos os seus nós. Quando se trata de federações formadas por *clusters* em diferentes domínios administrativos, isso pode tornar-se um problema, já que cada *cluster* pode ter uma ferramenta diferente instalada. Existem, também, *clusters* gerenciáveis via SNMP (Simple Network Management Protocol), que provê o mesmo tipo de informações monitoradas por *gmond*. Nesse caso não se faz necessária a coleta de informações em todos os nós, mas apenas um mecanismo de centralização e armazenamento das informações de monitoração (*gmetad*).

Neste contexto, incorporou-se à ferramenta Ganglia a capacidade de interoperar com outras ferramentas de monitoração (SCMS, Parmon e PCP), além de permitir a coleta de informações via protocolo SNMP. Após um estudo das ferramentas escolhidas, foi implementada uma biblioteca capaz de ler as informações coletadas por cada uma delas. Esta biblioteca foi utilizada para implementação de uma nova versão do programa *gmond*, capaz de obter uma cópia do estado do *cluster* com as ferramentas citadas.

A obtenção das informações de SCMS, PCP e SNMP foi implementada utilizando suas respectivas bibliotecas para implementação de clientes. Como Parmon é uma ferramenta proprietária e o protocolo de comunicação entre os monitores e o cliente não é documentado, foi necessária a utilização de técnicas de engenharia reversa, como a análise dos executáveis *parmond* e a decompilação das classes Java do cliente, para descobrir o protocolo de comunicação de Parmon. A etapa final foi a implementação de rotinas que se conectam diretamente ao *parmond* e recebem as informações monitoradas.

6. Conclusão

A ferramenta Ganglia é uma das principais ferramentas de Software Livre disponíveis para a monitoração de *clusters*. Neste artigo apresentou-se uma avaliação desta ferramenta, baseada na comparação de Ganglia com outras ferramentas de monitoração e na experiência de vários meses de utilização de Ganglia no gerenciamento de um *cluster* pertencente a um laboratório de pesquisa. A fim de aumentar as funcionalidades desta ferramenta, incorporou-se a Ganglia a capacidade de monitorar uma dada aplicação paralela ao longo de sua execução. Além disso, implementou-se uma nova versão do programa coletor de dados de Ganglia, de modo a permitir a obtenção de dados coletados por outras ferramentas de monitoramento que eventualmente estejam instaladas em diferentes *clusters* de uma mesma federação. Com este trabalho, contribui-se para a ampliação do domínio de aplicações de Ganglia e, ao mesmo tempo, para a valorização desta ferramenta de Software Livre.

Referências

Balay, S., Buschelman, K., Gropp, W. D., Kaushik, D., Knepley, M. G., McInnes, L. C., Smith, B. F., and Zhang, H. (2001). PETSc Web page. http://www.mcs.anl.gov/petsc.

Buyya, R. (2000). PARMON: a portable and scalable monitoring system for clusters. *Software Practice and Experience*, 30(7):723–739.

des Ligneris, B., Scott, S. L., Naughton, T., and Gorsuch, N. (2003). Open Source Cluster Application Resources (OSCAR) : design, implementation and interest for the [computer] scientific community. In *To appear, Proceeding of* 17^{th} *Annual International Symposium on High Performance Computing Systems and Applications (HPCS 2003)*, Sherbrooke, Canada.

Ferreto, T. C., de Rose, C. A. F., and de Rose, L. (2002). Rvision: An open and high configurable tool for cluster monitoring. *2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'02)*, page 75.

Goodwin, M. (2005). Performance co-pilot web page. http://oss.sgi.com/projects/pcp/.

Hoffman, F. M. (2005). The Rocks Cluster Distribution. http://www.rocksclusters.org/.

Massie, M., Chun, B., and Culler, D. (2004). The ganglia distributed monitoring system: Design, implementation, and experience. Technical report, University of California, Berkeley Technical Report.

Peterson, L., Culler, D., Anderson, T., and Roscoe, T. (2002). A blueprint for introducing disruptive technology into the Internet. In *Proceedings of the 1st Workshop on Hot Topics in Networks (HotNets-I)*. PlanetLab.

Uthayopas, P. and Rungsawang, A. (1999). SCMS: An extensible cluster management tool for beowulf cluster. In *Proceedings of Supercomputing'99 (CD-ROM)*, Portland, OR. ACM SIGARCH and IEEE. Department of Computer Engineering, Kasetsart University.