Uma Proposta de Escalonamento Descentralizado de Tarefas para Computação em Grade

Lucas Alberto Souza Santos[1] **, Patrícia Kayser Vargas**[2,3] **, Cláudio F. R. Geyer**[1]

[1]Instituto de Informática, UFRGS – Porto Alegre, RS, Brasil

[2]UniLaSalle – Canoas, RS, Brasil

[3]COPPE/Sistemas, UFRJ – Rio de Janeiro, RJ, Brasil

kayser@cos.ufrj.br, {lassantos,geyer}@inf.ufrgs.br

Resumo. *Neste artigo é proposto um modelo de escalonamento de tarefas em ambiente de grade. Este modelo segue uma estrutura descentralizada baseada na tecnologia das redes Par-a-Par (P2P). As redes P2P possuem as vantajosas características de tolerância a falhas e escalabilidade. Com o progressivo aumento do compartilhamento de recursos geograficamente distribuídos, estas características se tornarão fundamentais para o sucesso da Computação em Grade.*

O modelo proposto será integrado ao projeto ISAM na forma de um framework de escalonamento de aplicações. O sistema ISAM é um ambiente de Computação Pervasiva que engloba características da Computação em Grade. O modelo proposto será avaliado primeiramente por simulação e depois prototipado para avaliação experimental.

1. Introdução

A Computação em Grade é cada vez mais utilizada para a execução de aplicações que demandam elevados custos computacionais. Atualmente, os maiores usuários da tecnologia de grades computacionais são os pesquisadores da área de BioInformática e HEP(*High Energy Physics*). Há uma grande demanda nestas áreas por desempenho computacional e compartilhamento de dados. A tendência atual é o aumento da utilização das grades computacionais. Quando o número de domínios administrativos aumentar, exigirá soluções de escalonamento cada vez mais sofisticadas. Estas soluções devem promover uma maior escalabilidade do sistema, mantendo boa performance das aplicações que executam, além de utilizar mecanismos de tolerância a falhas, e suportar políticas locais de controle do uso dos recursos.

2. Sistemas de Gerenciamento de Recursos em Grade

Nota-se nos últimos anos um crescente desenvolvimento de grades computacionais e seus respectivos middlewares: Globus [Foster et al. 2001], Condor/Condor-G [THAIN et al. 2003], Legion [GRIMSHAW et al. 1999], OurGrid [ANDRADE et al. 2003], ISAM [Yamin et al. 2003]. Muitas das propostas de gerenciamento de recursos e escalonamento de tarefas em grades formadas por domínios administrativos autônomos seguem uma organização hierárquica (forma de árvore) ou centralizada. Abaixo estão as características das principais propostas de gerenciamento de recursos em grade que utilizam um modelo de organização centralizado ou hierárquico:

O sistema **Globus** provê uma infraestrutura que permite que aplicações vejam recursos distribuídos heterogêneos como uma única máquina virtual. O projeto Globus é um esforço de pesquisa multi-institucional que busca permitir a construção de grades computacionais. O Globus oferece serviços de informação através de uma rede de diretórios hierárquica baseada em **LDAP**, esta rede é chamada **Metacomputing Directory Services (MDS)**. O MDS é formado por um conjunto de GRIS (Grid Resource Information Service) indexados por GIIS (Grid Index Information). Os recursos atualizam suas informações no GRIS periodicamente. Ferramentas de alto-nível como resources brokers e meta-escalonadores realizam buscas através de consultas ao MDS usando protocolos LDAP. O namespace do MDS é organizado hierarquicamente em forma de árvore. A versão atual do Globus Toolkit (GT4), utiliza uma nova versão do MDS. MDS4 possui funcionalidades similares mas utiliza protocolos baseados em XML ao invés do protocolo LDAP, e possui uma série de aprimoramentos.

O modelo de escalonamento do sistema Globus é descentralizado. O Globus é um toolkit para grid que não fornece suporte nativo à políticas de escalonamento, mas ele permite que resource brokers externos adicionem esta capacidade. Os serviços Globus são utilizados em uma variedade de sistemas de gerenciamento de recursos: Nimrod/G, NetSolve, GrADS, AppLeS e Condor-G.

O **Condor** é um sistema de gerenciamento de recursos, que tem o objetivo de prover uma plataforma computacional de alta-vazão, desenvolvido pela Universidade de Wisconsin em Madison nos EUA. O Condor realiza a descoberta de recursos ociosos em uma rede e a alocação destes recursos para execução de tarefas. A função principal do Condor é utilizar máquinas distribuídas que de outra forma estariam ociosas, promovendo a máxima utilização dos recursos disponíveis. O Condor é formado por um conjunto de diferentes daemons. Um cluster de estações de trabalho chamado de Condor pool é gerenciado pelo sistema. O sistema Condor possui uma arquitetura de escalonamento centralizada. O sistema Condor possui uma funcionalidade chamada flocking que permite que usuários possam utilizar recursos compartilhados de múltiplas Condor pools distribuídas.

3. Motivação

Um escalonador para grade depende diretamente da sistema de informação, módulo que possibilita a descoberta de recursos e monitoração. O Toolkit Globus e o sistema Condor, embora amplamente utilizado nas grades atuais, não apresentam as seguintes características fundamentais para suportar o crescente uso da computação em grade:

- Elevada escalabilidade;
- Tolerância a falhas e ataques;
- Auto-adaptação e dinamicidade;
- Completa autonomia dos centros regionais sobre seus recursos.

O modelo estruturado em árvore do sistema MDS do Globus é escalável e possui boa eficiência, todavia os nós mais altos da hierarquia são pontos críticos de falha, tornando o sistema suscetível a falhas e ataques. Qualquer falha no sistema de informação dos recursos afeta o escalonamento da grade.

A alternativa proposta neste trabalho é uma estrutura de escalonamento descentralizado colaborativo, onde domínios administrativos autônomos compartilham re-

cursos com outros domínios vizinhos, criando um sistema de compartilhamento, não-hierárquico, dinâmico, organizado em rede **Par-a-Par** *(Peer-to-Peer)* **P2P**. Este modelo de escalonamento será integrado ao ambiente em grade do sistema ISAM.

O sistema **ISAM (Infra-estrutura de Suporte às Aplicações Móveis Distribuídas)** [Yamin et al. 2002], em desenvolvimento no Instituto de Informática da UFRGS, é um middleware direcionado ao gerenciamento de recursos em redes heterogêneas, suportando a execução de aplicações distribuídas baseadas em componentes. A arquitetura do ambiente ISAM é organizada na forma de células autônomas cooperativas. Cada célula possui um escalonador local e políticas próprias de uso de seus recursos. Desta forma, um modelo de escalonamento de tarefas descentralizado cooperativo para a plataforma ISAM é fundamental para que o sistema alcance níveis elevados de escalabilidade, balanceamento de carga e tolerância a falhas.

4. Trabalhos Relacionados

Existem algumas propostas de modelo para escalonamento descentralizado não-hierárquico em grades. Em [Shan et al. 2003], os autores afirmam que uma arquitetura descentralizada em P2P possibilitou melhor escalabilidade e tolerância a falhas, quando comparada com uma arquitetura centralizada. O projeto **OurGrid** [ANDRADE et al. 2003] utiliza uma rede de favores estruturada em P2P para compartilhamento de recursos de forma justa entre participantes de uma grade. O modelo de aplicações do sistema OurGrid é do tipo Bag-of-Task. O OurGrid possui um protótipo implementado com a tecnologia JXTA [Gong 2001]. O sistema **Triana Grid** [Taylor et al. 2003] utiliza um modelo de grade descentralizado em P2P. O Triana Grid utiliza as tecnologia JXTA e OGSA [Foster et al. 2002]. O sistema **LEAF** (*Learnig Agent based FIPA toolkit*) [Lynden and Rana 2002] é um sistema de grade descentralizado que utiliza explicitamente o modelo de agentes como pares de uma rede P2P.

5. Modelo Proposto

O modelo de grade proposto neste trabalho objetiva criar uma comunidade de centros regionais de computação como pares de uma rede lógica (*overlay network*) P2P. Os centros (domínios) que participam da rede P2P procuram compartilhar recursos computacionais, ou seja, utilizam recursos remotos e fornecem recursos computacionais para usuários de outros domínios. O sistema possui suporte para que os donos dos recursos possam estabelecer políticas de acesso a seus recursos, assim eles têm controle total sobre sua participação na grade computacional. Cada domínio administrativo da grade P2P é formado por uma célula ISAM, que é composta de uma máquina base e outros nodos computacionais. A máquina base é a entidade responsável pelo gerenciamento de todos os recursos da célula, onde os serviços principais executam, e onde executará o serviço que manterá a rede de escalonamento P2P funcionando.

Desta forma, a arquitetura da rede P2P formada pelas células ISAM se aproxima do modelo **Super-Peer** [Yang and Garcia-Molina 2002] de rede P2P (Figura 1). Este modelo de rede, é considerado um modelo híbrido de rede par-a-par, pois existem nodos da rede que se comportam como entidades centrais (base da célula) para alguns outros nodos (recursos da célula). Para facilitar a compreensão do modelo, podemos abstrair o modelo super-peer, encapsulando todos os recursos de uma célula ISAM, em uma única entidade nodo da rede P2P, sem perda informações.

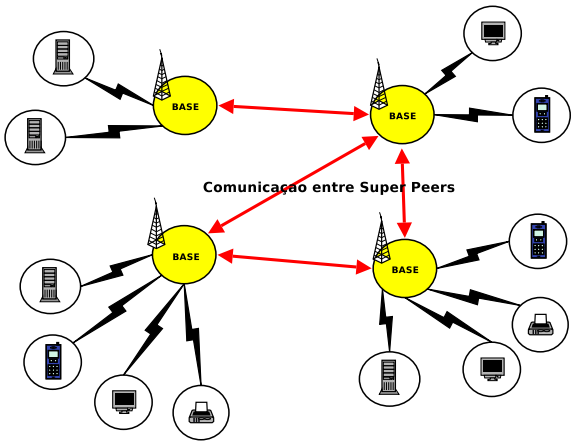

Figura 1. Modelo de rede super-peer formado pelas células ISAM

Os usuários deverão submeter aplicações organizadas em forma de DAG(Directed Acyclic Graph) a um dos pares da rede e o serviço de escalonamento P2P servirá de *Grid Broker* para comunicação com outros *grid brokers* de células vizinhas. Os recursos de um nodo da rede (célula ISAM) poderão ser acessados por usuários locais da célula, ou por usuários de outras células através da comunicação entre Grid Brokers. O serviço de Grid Broker é composto por um escalonador de grade com uma fila de tarefas globais, um escalonador local com uma fila de tarefas locais, um repositório de políticas e uma lista de nodos vizinhos (Figura 2).

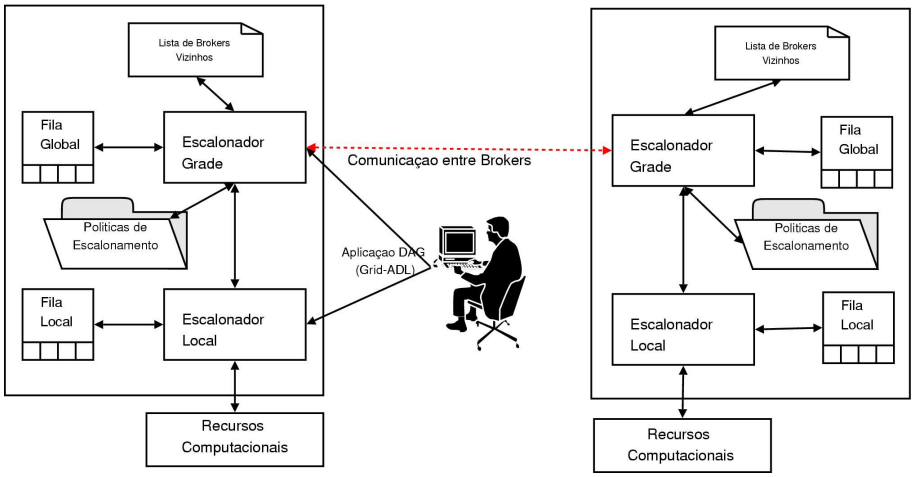

Figura 2. Arquitetura do componente Grid Broker

Os componentes do serviço de broker possuem as seguintes funções:

Escalonador Grade Sua função é escalonar aplicações descritas na linguagem Grid-ADL nos recursos da grade. Este componente utiliza protocolos P2P para manter a rede em funcionamento e gerenciar a lista de células vizinhas na rede lógica par-a-par. É responsável por verificar o estado das tarefas das aplicações, e tomar as decisões adequadas para que as aplicações terminem com sucesso, mesmo em caso de falhas. Para escalonar, este componente compara o estado dos recursos da célula com as políticas de escalonamento. Através de um algoritmo de escalonamento, decide qual tarefa da fila de tarefas globais pode executar nos recursos

de sua célula local. As tarefas escalonadas são transferidas para a fila de tarefas locais, e não poderão mais migrar para outra célula.

Fila Global Nesta fila estão as tarefas globais da grade, pertencente a uma dada aplicação (DAG) disparada em algum ponto da grade. Este tipo de tarefa pode migrar de um centro administrativo a outro, de célula em célula, até que seja colocada em uma fila de escalonamento local.

Escalonador Local Sua função é escalonar as tarefas da fila local nos recursos disponíveis da célula. Este componente realiza a monitoração do estado dos recursos e a contabiliza a utilização destes. Um algoritmo de escalonamento é responsável pela seleção da tarefa a ser executada e escolha do recurso onde será destinada a tarefa. As decisões do escalonamento são diretamente influênciadas pelas políticas de escalonamento definidas pelos gerentes dos recursos da célula. O usuário local (não-remoto) da célula tem a opção de disparar uma aplicação Grid-ADL diretamente neste escalonador. Desta forma, as tarefas desta aplicação passarão diretamente à fila de tarefas locais.

Fila Local Nesta fila estão as tarefas locais da célula, pertencente a uma dada aplicação (DAG) disparada em algum ponto da grade. Este tipo de tarefa não pode mais migrar a outro centro administrativo, será executada nos recursos da célula local.

Políticas de Escalonamento As políticas de escalonamento são propriedades que controlam quais recursos são escalonados dentre os recursos que atendem as requisições das tarefas. Estas propriedades definem permissões de **quem** pode acessar, **quando** acessar e **como** podem ser acessados os recursos conectados ao domínio administrativo. As políticas são divididas em dois grupos: As *Políticas Globais* são definidas pelo gerente do domínio administrativo, estas políticas são respectivas ao conjunto formado por todos os recursos da célula. As *Políticas Locais* estão relacionadas a recursos individuais da célula, como um computador ou um banco de dados. Estas são definidas pelo dono do recurso.

6. Conclusão e Trabalhos Futuros

O modelo atende os requisitos de um Escalonador para Grades. O Grid Broker é a entidade responsável por manter a rede P2P, executar as políticas definidas pelos donos dos recursos e escalonar as aplicações. A proposta será avaliada inicialmente através de simulação com o simulador MONARC 2 [monarc2 2005] implementado na linguagem Java. As principais classes do sistema MONARC serão estendidas para tornar possível o escalonamento em rede P2P. O modelo de aplicação utilizado no framework será baseado no framework GRAND [Vargas et al. 2004], que define aplicações descritas através de um DAG. As aplicações do modelo GRAND são constituídas por tarefas com ou sem dependências de arquivo entre si. O protocolo P2P utilizado no framework JXTA [Gong 2001] está sendo analisado e possivelmente será adotada uma extensão deste protocolo para atender os requisitos do modelo proposto.

As informações obtidas na simulação serão utilizadas para implementar um protótipo do modelo proposto. O protótipo será desenvolvido a partir dos componentes do middleware EXEHDA incluído no sistema ISAM utilizando a linguagem Java. A validação do protótipo será feita através da análise do desempenho de algumas aplicações escolhidas de forma a abranger os tipos de carga (leve, baixa, alta)(computacional, armazenamento) a que estão sujeitos os ambientes de computação em grade.

Referências

ANDRADE, N., CIRNE, W., BRASILEIRO, F. V., and ROISENBERG, P. (2003). OurGrid: An approach to easily assemble grids with equitable resource sharing. In *Proceedings of the 9th Workshop on Job Scheduling Strategies for Parallel Processing*.

Foster, I., Kesselman, C., Nick, J. M., and Tuecke, S. (2002). The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. http://www.globus.org/research/papers/ogsa.pdf.

Foster, I., Kesselman, C., and Tuecke, S. (2001). The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of Supercomputer Applications*, 15(3).

Gong, L. (2001). Jxta: A network programming environment. *IEEE Internet Computing*, 5(3):88–95.

GRIMSHAW, A. S., FERRARI, A., KNABE, F., and HUMPHREY, M. (1999). Wide-area computing: Resource sharing on a large scale. *IEEE Computer*, 32(5):29–36.

Lynden, S. and Rana, O. F. (2002). Coordinated learning to support resource management in computational grids. In *P2P '02: Proceedings of the Second International Conference on Peer-to-Peer Computing*, page 81, Washington, DC, USA. IEEE Computer Society.

monarc2 (2005). Models of networked analysis at regional centres for lhc experiments.

Shan, H., Oliker, L., and Biswas, R. (2003). Job superscheduler architecture and performance in computational grid environments. In *Proceedings of the Supercomputing 2003*.

Taylor, I., Shields, M., Wang, I., and Philp, R. (2003). Distributed p2p computing within triana: A galaxy visualization test case. In *IPDPS '03: Proceedings of the 17th International Symposium on Parallel and Distributed Processing*, page 16.1, Washington, DC, USA. IEEE Computer Society.

THAIN, D., TANNENBAUM, T., and LIVNY, M. (2003). Condor and the Grid. In BERMAN, F., FOX, G., and HEY, T., editors, *Grid Computing: Making The Global Infrastructure a Reality*. John Wiley.

Vargas, P. K., Dutra, I. d. C., and Geyer, C. F. (2004). Gerenciamento hierárquico de aplicações em ambientes de computação em grade. In *Escola Regional de Alto Desempenho (ERAD 2004)*, Pelotas, RS.

Yamin, A., Augustin, I., Barbosa, J., and Geyer, C. F. (2002). ISAM: a pervasive view in distributed mobile computing. In *Proceedings of the IFIP TC6 / WG6.2 & WG6.7 Conference on Network Control and Engineering for QoS, Security and Mobility (NETCON 2002)*, pages 431–436.

Yamin, A., Augustin, I., Barbosa, J., Silva, L. C. d., Real, R. A., Cavalheiro, G., and Geyer, C. F. (2003). Towards merging context-aware, mobile and grid computing. *International Journal of High Performance Computing Applications*, 17(2):191–203.

Yang, B. and Garcia-Molina, H. (2002). Designing a super-peer network. Technical Report 2002-13, Stanford University.