OAR: Estudos para Gerenciamento de Recursos em Ambientes Distribuídos

Carlos Vinícius Rasch Alves[1], **Gerson Geraldo H. Cavalheiro**[1]

{cvralves, gerson.cavalheiro}@inf.ufpel.edu.br

[1]Universidade Federal de Pelotas (UFPEL)
Computação - CDTec

Resumo. *Atualmente um problema constante da computação é a demanda cada vez maior por processamento. Muitas vezes é necessário mais que um computador para executar uma tarefa, sendo necessário o uso de uma agregação de computadores de uma forma dedicada para a execução de aplicaçõeses específicas. O presente trabalho apresenta um estudo para o uso do escalonador OAR para controle de uma grade montada sobre um ambiente virtualizado para execução de tarefas específicas.*

1. Introdução

Desde o surgimento dos Sistemas Distribuídos, as tarefas podem ser executadas em máquinas distintas e de forma paralela. Deste fato, se apresentou a necessidade de um mecanismo para gerenciar, dentre as máquinas disponíveis, aquelas que melhor atenderiam a demanda para uma parte específica da aplicação. Tornava-se, desta forma, evidente o uso de escalonadores de processos de mais alto nível, absorvendo a responsabilidade de distribuir a carga entre diferentes máquinas ligadas em uma rede. Essas políticas de escalonamento determinam quando uma aplicação seria executada, como e onde.

Os sistemas de gerenciamento para *clusters* contém mecanismos de descoberta, seleção e monitoração de recursos. Alguns destes sistemas para controle de *clusters* sofreram adaptações para serem utilizados em grades computacionais, onde a função de escolher os nós para execução é ainda mais difícil, visto a existência de uma heterogeneidade e dinamicidade presente nestes ambientes computacionais. Fica clara que as tarefas de escalonamento de processos é de grande valia para o bom desempenho e para o aumento da utilização dos recursos de um sistema computacional distribuído.

O presente trabalho tem por objetivo descrever um estudo realizado em ambiente virtualizado em torno do escalonador de recursos OAR, na qual foram realizados testes e estudos sobre suas funcionalidades em ambiente distribuído. O artigo esta dividido na seguinte forma: A Seção 2 apresenta a terminologia estudada. Na seção 3 é descrita a metodologia e o ambiente criado para os estudos. Por fim, a seção 4 descreve a conclusão e os trabalhos futuros.

2. Terminologia

Nesta seção serão apresentados os conceitos básicos que serão utilizados ao longo deste trabalho. Dentre os conceitos básicos, serão apresentadas definições de *clusters*, computação em grade, escalonador de tarefas em ambientes distribuídos.

2.1. *Clusters*

Clusters em termos de arquiteturas computacionais pode ser entendido como uma agregação de computadores de uma forma dedicada para a execução de aplicações específicas. Um *cluster* dedicado é projetado para rodar exclusivamente aplicações paralelas [Aliaga 2011].

2.2. Computação em Grade

A ideia de se utilizar computadores independentes e amplamente dispersos como plataforma de execução de aplicações paralelos. Consiste em compartilhar de forma coordenada e dinâmica recursos por organizações virtuais entre várias instituições. Cada organização virtual é um conjunto de indivíduos ou instituições que fornecem recursos executores e consumidores os quais definem claramente o que é compartilhado e sob que condições o compartilhamento é possível, fazendo uso dos recursos de forma coordenada e controlada [Kon and Goldman 2008].

O compartilhamento coordenado é feito através de um *middleware*, que é um pacote de software que faz a interface entre o usuário e o ambiente computacional [Reis 2008].

2.3. Escalonador de Tarefas

Para um melhor entendimento do escalonador de tarefas faz-se necessário termos o conceito de escalonamento, muitas vezes realizado por atribuições de tarefas aos elementos de processamento, recursos, um possível objetivo é que, essa atribuição seja efetuada de forma eficiente para minimizar o tempo das aplicações.

O principal objetivo é maximizar a utilização dos recursos computacionais disponíveis minimizando os custos relativos à comunicação. Significa minimizar o tempo de término das aplicações (*makespan*).

Dentre as funções do escalonador, encontram-se: recebimento de demandas de tarefas por recursos e a atribuição de recursos a essas tarefas realizando a alocação do que é buscado com o que é oferecido pela infra-estrutura.

Para uma boa qualidade do escalonamento, deve-se ter o conhecimento da funcionalidade de emparelhamento, sendo este realizado com o intuito de maximizar a qualidade de serviço estabelecido pelos usuários da grade. A qualidade de serviço pode ser dada pela minimização do tempo de espera pelos resultados de uma tarefa. Para atingir essa qualidade de serviço, existem políticas de escalonamento que ditam como, quando e onde determinada aplicação deve ser executada.

3. Metodologia

O ambiente de testes foi projetado afim de entender o funcionamento, a forma de instalação e o como as tarefas são escalonadas com a ferramenta OAR, que é um escalonador de recursos para *clusters* de grande porte. Para isso foi montado um cenário com máquinas virtuais para fazer a simulação de máquinas distribuídas. Para um propósito de melhor entendimento do escalonamento, ao invés de simples máquinas, foram criados *clusters* para distribuição interna das tarefas, todo o conjunto formando uma grade computacional.

O OAR trabalha de forma simples utilizando benfícios da linguagem SQL, ferramentas de alto nível como a linguagem Perl, o banco de dados MySQL e a ferramenta TakTuk que opera como uma interface para seus serviços.

A arquitetura do servidor OAR é composta por dois componentes, sendo eles o banco de dados relacional e os *scripts* Perl, como ilustra a figura 1.

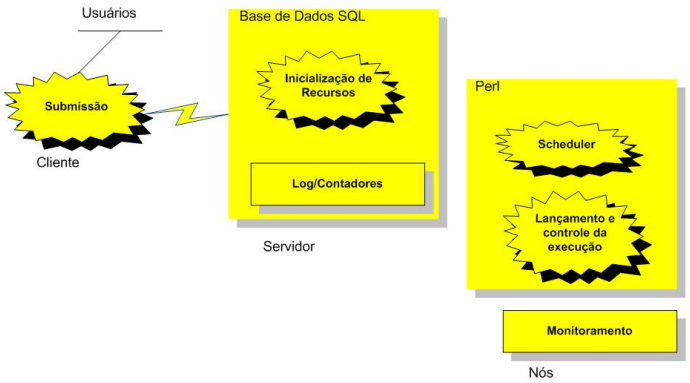

Figura 1. Arquitetura OAR - [Aliaga 2011]

A distribuição de trabalho e o monitoramento é realizado pela ferramenta TakTuk[1], esta uma ferramenta para a implantação de execução remota que utiliza um algoritmo adaptativo se espalhando e configurando uma rede. Utiliza um algoritmo de roubo de trabalho reativo que mistura paralelização local e distribuição de trabalho [SCHNORR 2005].

A arquitetura do OAR é baseada sobre um nível mais abstrato que minimiza a complexidade de concepção de seu software. Sua arquitetura interna é constituída de dois componentes principais: uma ferramenta genérica e escalável para administração do *cluster* escrita na linguagem de programação Perl e um banco de dados MySQL, como única forma de compartilhar informação. No banco de dados são armazenados todos os dados internos sobre as aplicações e recursos, o acesso é unicamente por comunicação entre módulos.

A outra parte do sistema, servidor, é composta por um conjunto de módulos independentes implementados com *scripts* Perl. Cada módulo é responsável por tarefas específicas, como por exemplo, iniciar e controlar a execução de aplicações.

O monitoramento das tarefas é tratado pela ferramenta TakTuk, que permite a execução paralela de comandos em grandes *clusters*. Sua vantagem é ser uma ferramenta altamente paralelizada e distribuída. No OAR o TakTuk é utilizado para realizar tarefas administrativas nos nós dos *clusters* através de serviços de execução remota fazendo esta tarefa com troca de informações com o OAR através do banco de dados.

O módulo de escalonamento possui funcionalidades importantes tais como prioriade sobre tarefas, reserva de recursos e *backfilling*, que permite uma aplicação submetida após uma outra seja escalonada se não for atrasar a execução das aplicações há mais tempo na fila. O algoritmo consiste no FCFS (*First Come First Served*).

[1]TakTuk Web site: `http://taktuk.gforge.inria.fr`

Na implementação do cenário foram utilizadas duas máquinas reais com processadores i7 Quad-core e 16 GB-RAM DDR3 como hospedeiras das estruturas virtualizadas. No total foram criadas 12 máquinas virtuais, cada máquina virtual, criada com a ferramenta VMWare[2], teve como sistema operacional o Linux, com a distribuição Debian[3]. A Figura 2 ilustra o ambiente de testes na forma como foi configurado.

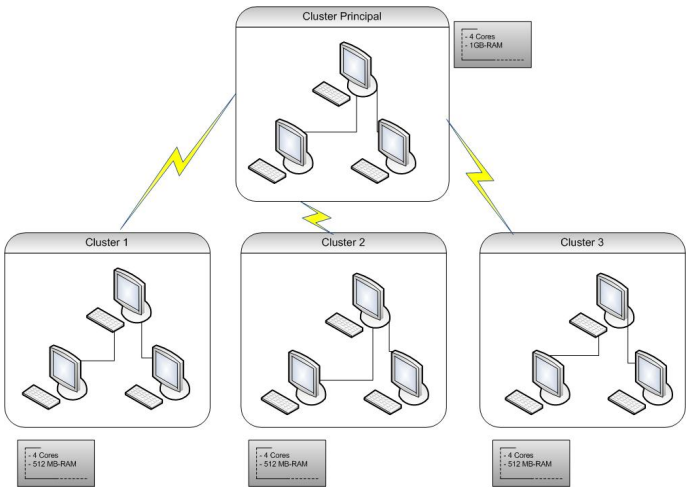

Figura 2. Ambiente de Estudos

Neste ambiente, foram testados com sucesso as funcionalidades presentes, tais como execução de tarefas interativas ou de lote, emparelhamento de tarefas/recursos, suporte para multi-escalonadores, multi-filas com prioridade e notificações de *checkpoints*.

4. Conclusões

Neste trabalho, sendo um primeiro contato com a ferramenta OAR ficaram claras as vantagens deste escalonador, sua simplicidade e modularidade. Com os testes o OAR mostrou um escalonamento modular e que pode ser adaptado às necessidades do administrador da grade.

Como trabalhos futuros pretende-se testar o OAR em um ambiente real com máquinas reais distribuídas.

Referências

Aliaga, A. H. M. (2011). *Estudo comparativo de técnicas de escalonamento de tarefas dependentes para grades computacionais*. PhD thesis, Universidade de São Paulo.

Kon, F. and Goldman, A. (2008). Grades computacionais: Conceitos fundamentais e casos concretos. *Proceedings of the Tomasz Kowaltowski e Karin Breitman (Org.) Atualizações em Informática*, pages 55–104.

Reis, V. Q. (2008). Gerenciamento de recursos em ambientes distribuídos: uma visão do escalonamento de processos.

SCHNORR, L. M. (2005). Monitoramento de aplicações grid: Uma análise entre ambientes de programação e ferramentas de monitoramento.

[2]VMware Web site: `http://www.vmware.com`
[3]Debian Web site: `http://www.debian.org`