

Um modelo de servidor HTTP com instanciação dinâmica de *proxies* em uma ambiente de grade computacional

Dorival Luiz Fransozi ¹, Cristiano Cachapuz e Lima ², Adenauer Corrêa Yamin ³, Luciano Cavalleiro da Silva ⁴, Cláudio Fernando Resin Geyer ⁵

^{1,2} URCAMP - Curso de Informática
Bagé, RS

¹ dorivalfransozi@yahoo.com.br,

² cristiano@urcamp.tche.br

Universidade Federal de Pelotas
Pelotas, RS

³ adenauer@ufpel.tche.br

^{4,5} UFRGS - Instituto de Informática
Porto Alegre, RS

^{4,5} {lucc,geyer}@inf.ufrgs.br

Introdução

Este trabalho apresenta uma solução voltada para o domínio de servidores Web que explora uma infra-estrutura de grade com vistas a reduzir o tempo de resposta no tratamento das requisições HTTP direcionadas ao servidor. Para isso, emprega-se um mecanismo de alocação dinâmica de *proxies* e redirecionamento de requisições HTTP para melhorar a característica do acesso ao servidor Web. A alocação dinâmica de *proxies* emprega informação proveniente de um mecanismo de monitoração para classificação dos nodos e tomada de decisão na alocação dos *proxies*. Esse ambiente é baseado no modelo master/*proxies* onde o servidor master é o responsável: por prover o conteúdo Web, pelo primeiro atendimento ao cliente, pela escolha de qual nodo tem melhores condições de atender as requisições e, se necessário, pela instanciação de um servidor *proxy* para prover o conteúdo a esse cliente.

Processamento em Grade na Plataforma ISAM

Segundo Foster [FOS 02], uma grade computacional é uma infra-estrutura de hardware e software que provê acesso seguro, consistente, penetrante, barato e expressiva capacidade computacional final. As pesquisas em grade de computação preocupam-se com o compartilhamento coordenado de recursos e com a resolução de problemas em organizações virtuais dinâmicas e multi-institucionais.

Os recursos compartilhados na grade podem ser dados, softwares e hardwares de qualquer tipo. Porém cada provedor de recurso terá controle total sobre o que está sendo compartilhado, quem e quando pode acessar esses recursos. Esses provedores são chamados de Organização Virtual (VO) [FOS 02].

No contexto do projeto ISAM (<http://www.inf.ufrgs.br/~isam>) encontra-se em desenvolvimento a proposta de um *middleware*, o EXEHDA (<http://www.inf.ufrgs.br/~exehda>) que incorpora características da execução distribuída

em ambientes de grade na perspectiva de suportar um ambiente de computação pervasiva.

Exehda é, segundo Yamin [YAM 04], “um *middleware* direcionado às aplicações distribuídas, móveis e conscientes do contexto da Computação Pervasiva”. A computação pervasiva é um novo paradigma computacional, no qual seu foco está no acesso do usuário ao ambiente computacional a partir de qualquer lugar, podendo estar em movimento ou não e a qualquer tempo.

O Exehda é um subprojeto do projeto ISAM – Infra-estrutura de Suporte às aplicações Móveis Distribuídas, desenvolvido pela UFRGS desde 2001. O foco do ISAM é computação pervasiva e adaptação ao contexto, direcionado para redes de grandes áreas, com grande heterogeneidade entre os diversos componentes do ambiente.

Ainda segundo Yamin [YAM 04], “na computação pervasiva, as aplicações devem ter a propriedade de adaptabilidade ao contexto”. Essa adaptação é fundamental para equipamentos móveis pois eles devem ter esse suporte gerenciado por um ambiente de execução, que proporcione acesso seguro em qualquer lugar e tempo, dentro de um contexto oscilante, devido principalmente a heterogeneidade de recursos e da rede.

No ISAM o contexto é definido como “toda informação relevante para a aplicação e que pode ser obtida por esta”. Essas informações podem ser a carga computacional, espaço disponível de memória, fila de processos. Esse conjunto de informações representa o estado atual dos componentes que fazem parte do ambiente de execução. Yamin [YAM 04] destaca que “O modelo de contexto para cada aplicação é a descrição da dependência entre o estado do contexto, e sua interpretação tanto pela aplicação, como pelo *middleware* que gerencia sua execução”.

Servidor Web Adaptativo - SWA

A perspectiva de disponibilização de um ambiente de computação pervasiva enunciado pelo projeto ISAM (abordado na seção anterior) motivou o desenvolvimento de uma solução, baseada no modelo *master/proxies*, chamada de Servidor Web Adaptativo – SWA (figura 1), que possa aproveitar melhor os recursos dos componentes do ambiente de uma grade computacional para melhorar a característica de acesso ao servidor.

No SWA, um dos nodos usados será definido como master. Esse nodo é responsável por prover o conteúdo *Web*, através de um servidor HTTP.

Ele também será o primeiro contato do cliente externo. Nesse primeiro contato, o servidor deverá percorrer uma tabela de nodos existentes no sistema, classificando-os conforme seu estado atual. Essa tabela é mantida somente o servidor master e atualizada sempre que um nodo mudar seu estado atual.

Esse estado atual será definido através da carga de trabalho do nodo, do número de clientes que o nodo está atendendo e da localização do nodo em relação ao cliente. Quanto maior o estado, menores condições o nodo tem de atender no momento.

Após determinado qual nodo possui melhores condições para atender o cliente, o servidor irá instanciar um serviço (que poderá ser *HTTP* ou *proxy*, dependendo do nodo) no nodo selecionado para atender o cliente em questão.

Com isso o atendimento ao cliente não fica comprometido e a carga dos nodos é dinâmica, pois a medida que cresce a demanda dos clientes, o ambiente tende a se adaptar para suprir a necessidade no momento.

Características do servidor HTTP:

- Prover o conteúdo *Web*, através de um serviço HTTP;
- Atender e instanciar serviços HTTP ou *proxy* aos clientes novos;
- Gerenciar o ambiente, mantendo informações sobre o estado de cada nodo que faz parte do contexto.

Características do servidor *Proxy*:

- Prover o conteúdo *Web* através de um serviço *proxy*, baseado no conteúdo do servidor master;
- Manter o servidor master informado sobre seu estado atual quando houver mudanças;

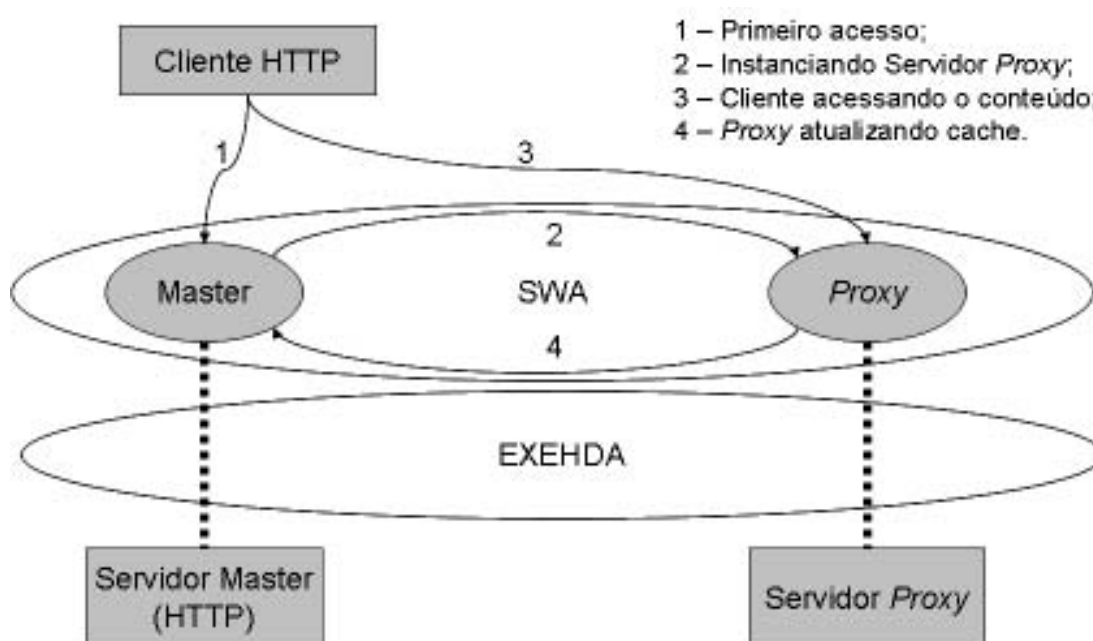


Figura 1 - Ambiente SWA

Aspectos da Implementação: o SWA foi projetado de forma a tirar proveito da infra-estrutura provida pelo EXEHDA, o middleware da plataforma ISAM. Nesse sentido, emprega as facilidades de instalação dinâmica de código e instanciamento remoto de objetos providas pelo EXEHDA para a instanciamento dos Servidores HTTP e *Proxy*. Com isso, o nodo não precisa necessariamente ter o serviço operando para fazer parte do ambiente, permitindo que os recursos sejam usados somente quando o servidor (HTTP ou *Proxy*) for instanciado. Além disso, é possível também finalizar o serviço quando o número de acessos e carga do sistema diminuir, liberando recursos do nodo.

O SWA está sendo implementado em linguagem Java, fazendo chamadas a API de serviços do EXEHDA. A infra-estrutura de hardware atualmente empregada nos testes consiste de um conjunto de servidores Linux.

Conclusões

Conclui-se que a criação de uma ferramenta para uso em um ambiente de grade computacional pode proporcionar uma melhoria no atendimento a clientes principalmente em sítios de comércio eletrônico onde a demora no atendimento pode resultar em diminuição no faturamento.

Também conclui-se que a troca de informações entre diversos nodos que compõem um ambiente de grade computacional melhoram o desempenho do conjunto, aproveitando de forma balanceada os recursos existentes.

Outro fator importante que pode ser destacado é o fato que de há uma forte tendência no uso de ambientes de grade computacional com objetivo obter alto poder de processamento somando os recursos dos diversos componentes do ambiente.

Referências

- [FOS 02] FOSTER, Ian. **What is the Grid? A Three Point Checklist**. 2002. Disponível em <<http://www-fp.mcs.anl.gov/~foster/Articles/WhatIsTheGrid.pdf>>. Acesso em: 12 jun 2004.
- [SIL 00] SILBERSCHATZ, Abraham; GALVIN, Peter Baer. **Sistemas Operacionais: conceitos**. 5. ed. São Paulo: Prentice-Hall, 2000.
- [YAM 04] YAMIN, Adenauer Corrêa. **Arquitetura para um Ambiente de Grade Computacional Direcionado às Aplicações Distribuídas, Móveis e Conscientes do Contexto da Computação Pervasiva**. 2004. 194 f. Tese (doutorado) – Universidade Federal do Rio Grande do Sul, Curso de Pós-Graduação em Computação.