

Otimizando a transferência de aplicações em ambientes de aproveitamento de ciclos ociosos de CPU utilizando níveis de *cache**

Felipe Z. Grazziotin, César A. F. De Rose

CPAD - PUCRS/HP

Av. Ipiranga, 6681, Telefone: 3320-3558 ramal 4463, Fax: 3320-3758
felipe@cpad.pucrs.br, derose@inf.pucrs.br

Introdução

Em ambientes de aproveitamento de ciclos ociosos de CPU de máquinas não dedicadas, não basta ter somente como objetivo maximizar o tempo que se estará executando aplicações, é necessário também minimizar os efeitos contrários ao usuário que o compartilhamento de recursos venha eventualmente trazer. Entre estes efeitos contrários podemos citar sobrecarga de componentes utilizados pelo usuário ou compartilhados na rede. Por esta razão, otimizações devem ser estudadas para criar economia de tempo e melhorar o desempenho do sistema.

O desperdício de poder computacional que ocorre em laboratórios de universidades, como o da Faculdade de Informática da PUCRS, é um dos fatores motivadores para a criação de projetos como o vCLUSTER, desenvolvido desde 2002 pelo CPAD¹. O vCLUSTER permite a utilização dos computadores que estão ociosos em certos períodos do dia como nós de um agregado (*cluster*, em inglês). Ao contrário de agregados tradicionais, que são compostos por máquinas dedicadas, os computadores utilizados no projeto são máquinas compartilhadas que passam a pertencer ao agregado à medida que tornam-se ociosos, formando o chamado *cluster virtual*.

No entanto, ao mesmo tempo em que não se pode monopolizar os recursos da rede onde se encontra o vCLUSTER, deve-se buscar otimizar a utilização deste agregado. Com isso, algumas soluções tiveram de ser criadas conforme problemas foram surgindo. Destacam-se entre os principais pontos para otimização o uso excessivo da rede e a sobrecarga de componentes do sistema, ocorrendo principalmente durante as requisições de usuários para a execução de aplicações nos nós do *cluster virtual*.

Neste trabalho será analisado o sistema de *cache* que foi desenvolvido para contornar os problemas de busca dos aplicativos dos usuários para a execução dos mesmos. Serão abordados também os três níveis hierárquicos desta *cache*, da forma que estão atualmente implementados.

*Este trabalho foi desenvolvido em colaboração com a HP Brasil P&D

¹Centro de Pesquisa em Alto Desempenho - <http://www.cpad.pucrs.br>

Definição do Problema

Apesar de comumente o sistema possuir um número relativamente pequeno de nós virtuais funcionando simultaneamente [PEZ 04], deve-se lembrar que a rede onde se encontram não é dedicada, mas compartilhada com outros computadores que não pertencem ao agregado virtual.

Em um primeiro momento o compartilhamento da rede parece ser um problema de menor importância, porém o agregado utiliza o modelo mestre-escravo [QUI 04]. Neste modelo o mestre representa um gargalo no sistema, respondendo a vários escravos ao mesmo tempo, causando atraso em todo o processo de funcionamento.

Por Que Utilizar *Cache*?

Sistemas de *cache* são muito utilizados na Internet (e.g. *Squid*²) para evitar o tráfego excessivo de dados, pois a banda de comunicação geralmente é escassa. No caso específico de agregados virtuais, tem-se ainda mais vantagens do que apenas a economia na utilização da rede. Economiza-se ainda o processamento do mestre, que não necessitará enviar aplicações para cada um de seus escravos.

Quando se utiliza *cache*, a rede também tem benefícios. Evita-se o uso desenfreado de recursos, poupando estes, que já são compartilhados. Controlando as frequentes requisições de arquivos ao mestre, aumenta-se a capacidade de um mestre controlar vários escravos.

Níveis de *Cache*

A seguir, serão descritos cada um dos três níveis de *cache* que compõem o VCLUSTER.

Nível 1

O primeiro nível que os nós verificam ao serem requisitados para iniciar uma aplicação é o seu próprio repositório, onde encontram os aplicativos mais recentemente utilizados por aquela máquina. Este repositório localiza-se no disco local, evitando qualquer conexão desnecessária com a rede.

Ao fazer a procura pela aplicação, faz-se uma verificação pelo CRC32 [ISO 84] do mesmo, para ter certeza que a aplicação sendo iniciada é a mesma que está presente no disco. Em caso contrário, retorna-se um erro de execução.

Se a aplicação não existir no disco local, o sistema do VCLUSTER começará a busca entre os nós vizinhos, definido no Nível 2 de *cache*.

²Squid Web Proxy Cache - <http://www.squid-cache.org/>

Nível 2

Quando o repositório local falha, o nó requisita ao mestre a lista de seus vizinhos ativos, ou seja, máquinas que também participam do vCLUSTER.

A transferência da lista é menos onerosa ao sistema do que a transferência da aplicações, o que justifica a adoção deste método. Após obter a lista, o escravo contata um vizinho para obter a aplicação desejada. Caso o vizinho também não tenha a aplicação com o CRC32 esperado, o escravo continua a procurar em um outro vizinho. Esse procedimento termina quando o escravo obter com sucesso a aplicação com o CRC32 esperado ou quando já tiver percorrido toda a lista de vizinhos.

Se não for possível conseguir o aplicativo necessário nos vizinhos, o escravo se comunica com o mestre para obter a aplicação, definido no Nível 3 da *cache*

Nível 3

Nos casos em que o escravo não consegue a aplicação requisitada, por exemplo quando a aplicação acaba de ser carregada no ambiente, o escravo necessitará pedir para seu mestre lhe enviar o aplicativo. Procura-se evitar ao máximo este terceiro nível por ser o mais oneroso para os componentes do sistema.

Não existe caso algum que o mestre não tenha o arquivo requisitado, pois a aplicação não inicia sem a existência do mesmo. A Figura 1 ilustra o funcionamento dos três níveis, conforme previamente discutido.

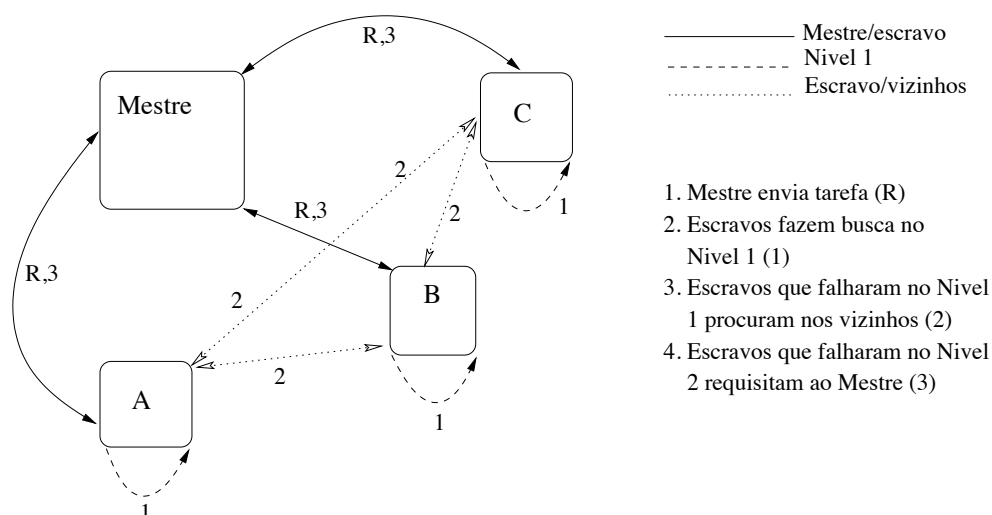


Figura 1: funcionamento do vCLUSTER

Considerações Finais

Muitos projetos de computação distribuída, como o SETI@Home³, suportam sistemas elementares de *proxy* e *cache*, para evitar que máquinas *networkless*⁴ necessitem de comunicação constante com um servidor distante. Sistemas como estes comprovam a eficácia da implementação da *cache* para suprir as necessidades de economia de recursos, sejam eles compartilhados ou não.

A implementação do *cache* dentro do projeto vCLUSTER tem como finalidade aumentar o desempenho de um agregado virtual e não dedicado, mas a base do sistema de distribuição de aplicativos de forma descentralizada pode ser implementada em praticamente qualquer tipo de agregado.

Referências

- [DER 02] C. De Rose, F. Blanco, N. Maillard, K. Saikoski, R. Novaes, O. Richard, and B. Richard. **The Virtual Cluster**: a Dynamic Environment for Exploitation of Idle Network Resources. SBAC-PAD, Vitória-ES-Brazil, p.141–148, Out. 2002.
- [PEZ 04] G. P. Pezzi, N. Maillard, C. A. F. De Rose, and K. Saikoski. **Anais da Escola Regional de Alto Desempenho**. [S.l.]: Instituto de Informática da UFRGS, 2004.
- [QUI 04] M. Quinn. **Parallel Programming**: Manager/Worker Paradigm. New York: McGraw-Hill, 2004. p.218-219.
- [ISO 84] International Organization for Standardization, “Information Processing Systems–Data Communication High-Level Data Link Control Procedure–Frame Structure”, **ISO 3309**, October 1984, 3rd Edition.

³Search for Extraterrestrial Intelligence at Home - <http://setiathome.ssl.berkeley.edu/>

⁴Máquinas sem conexão à uma rede, ou sem conexão permanente