vCluster *vs.* vCluster[2] *

Rodrigo D. Cassali, Felipe M. Franciosi, César A. F. De Rose

CPAD - PUCRS/HP
Av. Ipiranga, 6681 Telefone: 3320-3558 ramal 4463, Fax: 3320-3758
{cassali, ozzy}@cpad.pucrs.br, derose@inf.pucrs.br

Introdução

O CPAD[1] visa facilitar o desenvolvimento de aplicações voltadas a problemas que necessitam de arquiteturas especiais para serem solucionados. Com base em um estudo realizado no laboratório de programação da Faculdade de Informática da PUCRS, foi constatado um alto índice de ociosidade em diversos períodos do dia [PEZ 04]. Logo, foi iniciado um projeto para utilização de boa parte deste poder computacional desperdiçado através da formação de um "agregado virtual" [2].

O principal objetivo é implementar uma arquitetura na qual o usuário possa utilizar os computadores pessoais ociosos da mesma forma que utiliza os "agregados convencionais" [3]. Para pôr em prática estas idéias, o CPAD criou um protótipo chamado vCluster [DER 03].

Na ocasião da concepção do projeto, a intenção limitava-se em conseguir distribuir e processar tarefas, desconsiderando fatores como sobrecarga no servidor principal. Em outras palavras, não foi considerada a escalabilidade [ZOM 96] da arquitetura de forma a criar ambientes com um número maior de escravos e múltiplos mestres.

Para que estes objetivos pudessem ser alcançados, foi necessário elaborar um ambiente com maior escalabilidade através de políticas de gerência mais adequadas, como a introdução de um sistema de procura dos nós escravos. Estas características envolvem a possibilidade de incluir mais mestres com a capacidade de atuar sobre um número maior de escravos, evitando assim a sobrecarga das máquinas, no que diz respeito à gerência de alocação.

Arquitetura Atual: vCluster

Para uma aplicação ser executada no ambiente do vCluster, é necessário que a mesma tenha sido implementada utilizando o modelo de distribuição de tarefas chamado de "saco de trabalho" [WIL 99]. Este modelo trabalha com tarefas independentes, fornecidas somente pelo mestre. O resultado é devolvido ao mestre, sendo ele o único fornecedor de trabalho e receptor dos resultados, não havendo comunicação entre os escravos.

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

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

[2] Conjunto de máquinas não dedicadas.

[3] Conjunto de máquinas dedicadas.

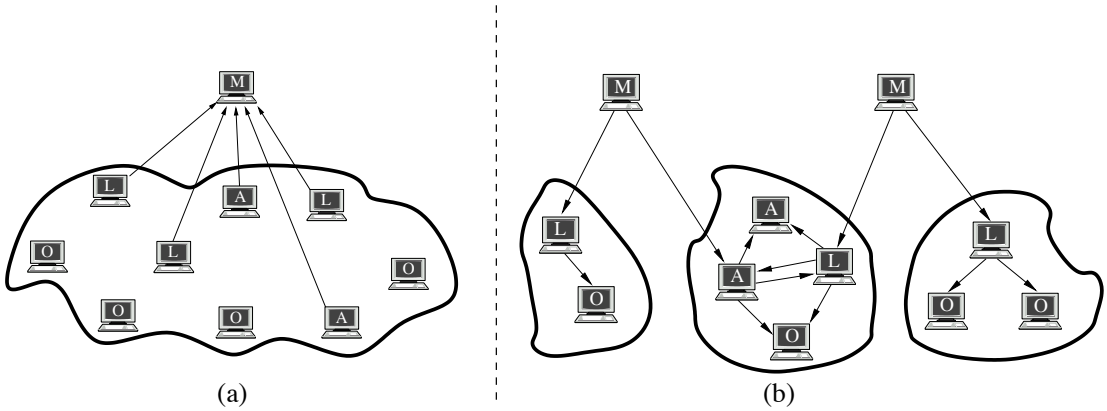

Figura 1: Arquitetura do vCluster (a) e vCluster^2 (b).

Neste ambiente virtual existem dois tipos de nós, o mestre e o escravo. Na Figura 1 o mestre está representado com a letra "M" no monitor.

Já os nós escravos, que servem unicamente para o processamento de tarefas, podem apresentar três estados distintos:

ocupado: este nó está fora do ambiente do vCluster por estar sendo utilizado por algum usuário ou porque o tempo de inatividade ainda não é suficiente para sua troca de contexto. Na Figura 1 os computadores neste estado estão representados com um "O" no monitor;

livre: nó que está participando do ambiente do vCluster, porém sem trabalho para realizar. Na Figura 1 os computadores neste estado estão representados com um "L" no monitor;

alocado: nó que está participando do ambiente vCluster, com trabalho para realizar. Passou pelos passos anteriormente mencionados, está alocado e executando uma tarefa. Na Figura 1 os computadores neste estado estão representados com um "A" no monitor.

Nova Arquitetura: vCluster^2

No que diz respeito à capacidade (número de tarefas sendo executadas) e escalabilidade (número de escravos no ambiente) da arquitetura, observou-se que seria inviável sustentar um modelo onde as máquinas escravas necessitassem conectar a todos os mestres (pontos de entrada) que pudessem estar presentes no ambiente. Para contornar este problema, foi adotado um outro comportamento para os nós escravos. No novo modelo, eles não comunicam a nenhum componente do sistema quando entram no modo vCluster^2, apenas aguardam que um mestre o encontre (conforme ilustrado na Figura 1(b)).

Do ponto de vista conceitual, vale mencionar que, devido à nova técnica utilizada para alocar recursos, tornou-se desnecessário manter uma estrutura lógica dos escravos. Isto simplificou a estrutura do gerenciador de recursos, facilitando o desenvolvimento de um modelo mais poderoso e capaz de encontrar máquinas em um número superior de redes.

Estação de Desenvolvimento

Na arquitetura atual, o desenvolvedor possui acesso ao módulo responsável pela distribuição de tarefas aos nós escravos (que é executado na máquina mestre). Esse modelo, além de apresentar falhas de segurança por este motivo, dificulta a execução de mais de uma aplicação simultaneamente, visto que este módulo trabalha como um serviço e possui uma porta padrão para aguardar conexões dos escravos. Se os desenvolvedores de diferentes aplicações não trocarem as configurações das portas padrão de forma que elas fiquem diferentes, haverá conflito no momento da execução desta parte do sistema.

O novo modelo é composto por uma aplicação gerenciadora do vCluster^2, que faz o envio dos binários e dos trabalhos da máquina do usuário até o computador mestre. Também é necessário respeitar uma organização de diretórios para ser utilizada na criação da aplicação. Este módulo encontra-se apenas na área do desenvolvedor, não sendo responsável pelas principais funcionalidades do vCluster^2.

Mestre

Neste novo modelo foram definidos serviços que executam constantemente no servidor mestre. Estes novos módulos agregam funcionalidades como a alocação de escravos e a distribuição de tarefas, visando delegar de uma forma mais consistente as responsabilidades que realmente cabem ao servidor mestre.

Conforme pode ser analisado na Figura 1(b), são os mestres que fazem a comunicação com os escravos. Desta forma, outros mestres, conhecendo pelo menos uma rede de escravos, podem ser agregados ao ambiente.

Na Figura 1(a) pode-se analisar que, na versão atual, são todos os escravos, que estão no estado “A” ou “L”, que estabelecem uma conexão com o mestre. No novo modelo, o mestre pode alcançar os escravos de uma mesma rede conectando-se em apenas um deles. Assim sendo, conversa com somente um escravo para alocar recursos naquela rede.

Escravo

A organização das máquinas escravas também mudou consideravelmente em relação ao modelo atual. O serviço gerenciador de recursos permanece aguardando requisições, sejam elas sobre o seu estado ou sobre alocação. Quando o escravo é alocado, é criado um arquivo para a tarefa contendo informações sobre o mestre, a aplicação e o trabalho a ser realizado. O serviço transferidor de trabalho, ao observar a criação deste arquivo, lê seu conteúdo, busca o executável da aplicação e o trabalho a ser executado.

No caso de um escravo previamente alocado ser contactado para alocação, ele funcionará como um representante do mestre e irá enviar um pacote *broadcast*[4] para sua rede. As máquinas escravas livres irão responder, possibilitando que ele encaminhe o pedido de alocação e comunique ao mestre qual escravo foi alocado.

Considerações Finais

Neste trabalho foi abordada uma nova modelagem para o sistema de aproveitamento de ciclos ociosos desenvolvido pelo CPAD. Para tal, foi realizado uma avaliação do modelo original, considerando os requisitos estabelecidos na ocasião. Em seguida, foi apresentada uma nova modelagem para suprir as novas necessidades.

O novo modelo criado busca estabelecer um novo conceito sobre a organização de mestres e escravos em sistemas distribuídos. Neste conceito, é o mestre quem busca os escravos, ao contrário dos modelos atuais, onde os escravos procuram o mestre para receber trabalho.

Acredita-se que este conceito possibilita que mais mestres possam ser associados a uma mesma rede de escravos sem causar sobrecarga no controle e gerência de recursos. Esta mudança de paradigma, além de buscar o aumento da escabilidade do ambiente, elimina a necessidade da existência de uma estrutura lógica de escravos.

Atualmente existe um protótipo desta nova arquitetura em fase de testes. Após realizados os experimentos e através dos pareceres dos usuários, novas características poderão ser incorporadas. Também serão investigadas algumas funcionalidades, como escalabilidade, tolerância a falhas e escalonamento de tarefas na nova arquitetura.

Referências

[DER 03] C. De Rose, F. Blanco, N. Maillard, K. Saioski, R. Novaes, O. Richard, and B. Richard. **The Virtual Cluster**: a Dynamic Enviroment for Exploitation of Idle Network Resources. SBAC–PAD, Vitória-ES-Brasil, p.141–148, Out. 2002.

[ZOM 96] Albert Y. Zomaya. **Parallel & Distributed Computing Handbook**. Editora McGraw-Hill. Primeira Edição. 1996.

[PEZ 04] G. P. Pezzi, N. Maillard, C. A. F. De Rose, and K. Saikoski. **Anais da Escola Regional de Alto Desempenho**. Pelotas. Instituto de Informática da UFRGS, 2004.

[WIL 99] B. Wilkinson and M. Allen. **Parallel Programming**. Editora Printice Hall. Primeira Edição. 1999.

[4]Enviado à rede e recebido por todos os membros da mesma.