



CLUSTER – POSSIBILIDADES DE EFICIÊNCIA E SEGURANÇA

-Revisão Bibliográfica-

JOSÉ VANDERLEI DA SILVA¹

¹ Departamento de Informática, Universidade Estadual de Maringá (UEM). Av. Colombo, 5790, Zona 07, Bl. 20, Sala 14^a, CEP: 87.020-900 – Maringá – PR – Brasil. E-mail: vander@intermam.com.br.

RESUMO

Este artigo apresenta a principal idéia de um Cluster de computadores, que é a de ser uma máquina com grande poder computacional, conectando computadores pessoais, usando preferencialmente software gratuito, a um custo muito menor do que os supercomputadores, por meio de estudos e algumas aplicabilidades, propondo uma área comum de troca de informações entre cada nó do Cluster.

Palavras Chaves: Cluster; paralelismo; aplicações; troca.

CLUSTER – EFFICIENCY AND SECURITY POSSIBILITIES

ABSTRACT

This paper shows the main idea of a computer Cluster which is a machine with a great computational power, connecting personal computers, using rather free software, with less cost than supercomputers, by means of studies and some applicability, proposing a common area of exchanging information among each Cluster node.

Word keys: Cluster; parallelism; applications; exchange.

INTRODUÇÃO

A utilização de algoritmos cada vez mais complexos, juntamente com o aumento do tamanho do conjunto de dados a serem processados, levou à exigência de respostas cada vez mais rápidas no mundo computacional. Este foi um dos fatores que vieram a motivar a necessidade do processamento paralelo, que pode ser entendido como a divisão de grandes tarefas em tarefas menores que serão executadas simultaneamente em um ou mais processadores. Dessa maneira, foram criadas máquinas com grande poder de processamento, chamadas de supercomputadores. Estes, pela sua própria definição, são máquinas de grande custo que podem ser encontradas no mercado, devido ao emprego das avançadas tecnologias disponíveis.

Para manter um computador dentro da definição de supercomputador é preciso um investimento considerável em investigação e desenvolvimento, mas, isso só é possível para grandes companhias sólidas. É precisamente este alto investimento que faz com que os supercomputadores sejam onerosos e estejam

fora do alcance da grande maioria das pessoas e das empresas. Projetos com altos requerimentos de ciclos de CPU, usualmente solicitam tempo em supercomputadores e caso sejam realizados em máquinas lentas, pode-se levar semanas e até mesmo meses para atingir o resultado desejado.

Neste artigo serão enfocados alguns conceitos sobre Cluster de computadores, objetivando a compreensão do leitor no assunto. Outros aspectos discutidos ao longo desta reflexão serão as aplicações reais que se beneficiam de Cluster de computadores, que a muito deixaram de ser somente estudos e/ou pesquisas. Também se procurou deixar claro que as pesquisas neste segmento da computação são um universo ainda pouco explorado e vêm sofrendo mudanças e adaptações constantes.

CLUSTER – UMA IDÉIA QUE CRESCE

Atualmente os avanços na arquitetura dos microprocessadores tem sido significativos, no entanto, não podemos depender constantemente de processadores

mais rápidos para obter maior eficiência. Existem limites físicos, como a velocidade da luz que, eventualmente, irão desacelerar e reduzir os avanços que presenciamos ano após ano no tempo que dura um ciclo de CPU.

As dificuldades em melhorar a eficácia de um processador, a convergência em eficiência entre microcomputadores, os supercomputadores tradicionais e o relativo baixo custo dos microprocessadores, têm permitido o desenvolvimento de computadores paralelos, viáveis comercialmente com dezenas, centenas e até milhares de microprocessadores. Um computador paralelo é um conjunto de processadores capaz de cooperar na solução de um problema. Esta definição inclui supercomputadores com centenas de processadores, máquinas com múltiplos processadores, redes de estações de trabalho (*Network of Workstations*) e redes de computadores pessoais (Cluster de computadores pessoais).

Quando agrupamos máquinas para solucionar um dado problema, a este agrupamento denominamos Cluster, e cada um dos computadores é chamado de nó. Um Cluster de computadores procura atender alguns princípios básicos como:

Escalabilidade: possibilidade de acrescentar ou remover componentes no Cluster, sem interromper a disponibilidade dos serviços ativos.

Transparência: capacidade de se apresentar como um sistema único.

Confiabilidade: deve garantir a realização da tarefa a qual foi designado.

Gerenciamento e Manutenção: mecanismos que permitem gerenciar de forma simples a complexidade de configurações e manutenção.

Há ainda outros conceitos que estão diretamente relacionados com Clusters de computadores, entre eles destacamos aqui:

Paralelismo: possibilidade da divisão da tarefa em partes que podem ser executadas de forma independente, a fim de obter

resultados mais rapidamente. Sua implementação pode ser em nível de hardware, por meio do uso de múltiplas unidades funcionais, como *pipeline* e cache. Esse tipo de paralelismo é basicamente transparente. No entanto, conhecendo-o, podemos tirar o máximo de desempenho em nosso código. Já na implementação em nível de software, a tarefa é dividida em subpartes que serão distribuídas e executadas em processadores diferentes.

Otimização: na escrita ou na hora de gerar um código deve-se levar em conta as características da máquina, sendo que o número de instruções e bifurcações devem ser o menor possível.

Tanto o paralelismo quanto a otimização buscam soluções diferentes para diminuir o tempo na execução de um programa. Ao tratarmos do paralelismo, assumimos que temos múltiplos processadores disponíveis. Existem dois paradigmas fundamentais de programação que estão baseados na visão de processos da memória. Por exemplo: quando a memória é vista por todos os processos como um só bloco e qualquer processo têm acesso a qualquer região dessa memória, temos uma memória compartilhada. Neste caso, os processos compartilham dados que residem na memória. Porém, a mesma é chamada distribuída quando os processadores possuem memórias privadas, não acessíveis a outros processadores. Nesse caso, a comunicação entre os processos acontece por meio de troca de mensagens, sendo hoje mais usadas as bibliotecas *Parallel Virtual Machine* (PVM) e a *Message Passing Interface* (MPI).

Dependência: ocorre quando uma parte do código não pode prosseguir sem os resultados de outros fragmentos deste.

Sincronização: colocamos lado a lado dois ou mais processos ou subtarefas. Quando um processo vai executar um código dependente de códigos calculados por outro processo, executa uma instrução de sincronização que o faz esperar pelos resultados necessários, diminuindo seu trabalho.

Latência: tempo transcorrido entre o momento em que se dá a solicitação da transferência de dado e o momento que a transferência efetivamente começa. Deve-se principalmente a inicialização de dispositivos e a preparação dos dados. Dá-se quando ocorre um acesso à memória, aos discos, e à rede.

Granularidade: tem relação com a quantidade de trabalho que se pode efetuar antes de ser necessário certo nível de sincronização, devido às dependências entre as subtarefas. Se o montante de trabalho é considerável, dizemos que a granularidade é grossa, no entanto, se for pequeno chamamos de granularidade fina. Clusters do tipo Beowulf tem um maior desempenho com granularidade grossa. Caso a granularidade seja fina, o tempo comunicação/sincronização faz com que a solução para paralelismo se torne ineficiente.

Uma maneira de realizar a carga do sistema é aumentando o número de processos executados concomitantemente. Vários sistemas computacionais não executam apenas um único processo, mas sim um balanceamento, consistido de uma lista destes. Isto significa que, em um sistema utilizado para executar múltiplos processos independentes simultaneamente, não existe a necessidade de que todos os processadores do sistema sejam hábeis a se comunicarem com todos os outros processos (HOGANSON, 1999).

Desde as primeiras pesquisas a respeito de Clusters, podemos considerar como marco histórico o ano de 1994 quando, Donald Becker e sua equipe na NASA conseguiram conectar vários computadores pessoais, mediante um software especial, criando um sistema que foi chamado de Beowulf, com uma eficiência comparada aos supercomputadores e que se transformou em um modelo para construção de Clusters de computadores (DÍAZ, 2002). Portanto, um Cluster Beowulf é uma rede de computadores conectados, onde todas as máquinas estão dedicadas à solução de um único problema. A idéia é conseguir processamento paralelo e, com isso, reduzir o tempo necessário para executar cálculos e/ou simulações complexas, dividindo as tarefas entre as máquinas (nó) da rede (MEREDITH, 2003).

O Sistema Operacional mais popular para um Cluster de computadores é o Linux, onde as principais razões para sua utilização podem ser resumidas no fato deste ser um software livre e com o código fonte aberto para ser usado em pesquisa e desenvolvimento (UTHAYOPAS, 1998).

O projeto OpenMosix também é de código aberto, surgindo como extensão do proprietário *Multicomputer Operating System Unix* (MOSIX) (ROCHA, 2004). Este software possui um núcleo cuja principal característica é proporcionar um ambiente para que estações e servidores trabalhem cooperativamente como se fossem um único sistema (BARAK, 1999). Ele foi desenvolvido pelos alunos do professor Ammon Barak na Universidade de Hebrew em Jerusalém. Assim, o OpenMosix é uma extensão do kernel para sistemas paralelos fazendo com que este(s) possua(m) uma imagem uniforme, onde a principal meta é a de distribuir a carga do sistema dinamicamente, otimizando a alocação de recursos existentes.

Em termos de Cluster podemos dizer que um Cluster Beowulf é classificado como “dedicado”, pois todos os seus nós são utilizados exclusivamente para computação paralela e um Cluster OpenMosix é considerado “não-dedicado” sendo que nele as aplicações são executadas baseadas na ociosidade das estações de trabalho.

Um Cluster de computadores pessoais tem um desempenho equivalente aos supercomputadores, porém seu custo é muito menor, além de possuir inúmeras outras vantagens, tais como:

Facilidade de Montagem: atualmente existe grande facilidade para construção de Cluster, não exigindo um alto grau de qualificação.

Manutenção e Disponibilidade: os elementos que compõem um Cluster estão disponíveis no mercado.

Hospedagem: os Clusters requerem hospedagem simples solicitando principalmente um sistema elétrico adequado. Em muitos casos não há sequer necessidade de se possuir sistemas de ar condicionado.

Modernização e Expansão: os Clusters estão compostos por elementos disponíveis de múltiplos fabricantes e, devido à compatibilidade que estes conseguem manter com as diferentes gerações de uma mesma família de componentes, é muito simples modernizá-los. Atualizar os Clusters com CPUs mais potentes pode ser tão simples como tirar uma CPU da placa mãe e instalar outra, ou talvez, substituir a placa mãe e a CPU conservando o resto dos componentes como memória, placas de vídeo, entre outros. Expandir a capacidade de memória e de armazenamento em disco não requer muito investimento, dado o baixo custo destes componentes. Unir CPUs, implica agregar PCs de fácil aquisição. No entanto, (HOGANSON, 1999) o retorno obtido de cada processador adicionado ao Cluster diminui com o acréscimo de novos processadores.

Uma importante diferença entre computadores organizados em Clusters e supercomputadores com memória compartilhada é a consideração de que há um custo maior na comunicação nos sistemas baseados em Cluster.

Um fator preponderante para este aumento de custo da comunicação é a latência da inicialização de uma mensagem, devido ao requisito de que os dados devem trafegar não no barramento interno de um supercomputador, mas em uma rede de comunicações; o que geralmente ocorre em velocidades menores e, por exemplo, em uma rede ethernet, ainda estarem sujeitos a colisões e retransmissões (PITANGA, 2003).

Com o progresso crescente em pesquisas sobre Cluster, algumas universidades começaram a ofertar cursos relacionados a processamento paralelo e distribuído. Existe ainda dificuldade na seleção de material para este tipo de estudo, no entanto, sugestões são apresentadas para compor os conteúdos de seus currículos. Assim, nestes casos, a grade curricular poderá estar focada em alguns tópicos, incluindo: arquitetura de sistemas, ambientes e linguagens de programação, a construção de algoritmos e aplicações. Atualmente, algumas experiências importantes estão sendo realizadas com os cursos de Cluster de computadores em universidades como a *University of Arkansas*, *Monash University*, *University of Southern California* (USC) e *Lewis & Clark College* (APON, 2003).

PESQUISAS E APLICAÇÕES EM CLUSTERS

Os Clusters computacionais estão cada vez mais presentes em aplicações científicas, principalmente com os processadores de alto desempenho, a baixa latência/aumento da largura de banda de redes além de software e ferramentas de desenvolvimento que facilitam a configuração e a manutenção do Cluster (BACKER, 2005). Segundo Pitanga (2003), cientificamente, a utilização de Cluster de alto desempenho está fortemente relacionada à execução de aplicações paralelas complexas, tais como: simulações aeroespaciais, projetos mecânicos de automóveis, aeronaves, processamento de imagens, realidade virtual, cálculos, seqüenciamento de DNA, entre outras. Já no campo comercial, seu uso está voltado para implementação de servidores de alta disponibilidade, como os requeridos em sistemas bancários, telefonia, bolsas de valores, sistemas *Web*, servidores de aplicações e de conteúdo, servidores de correio eletrônico, etc.

Os Clusters são ideais como servidores de arquivos, portanto, também para bases de dados. Eles permitem distribuir tanto a consulta como os próprios dados entre diferentes processadores e diferentes unidades de disco respectivamente. Isso permite acelerar consideravelmente as respostas do sistema (DÍAZ, 2002).

Pesquisadores da Universidade Federal de Santa Catarina desenvolvem um Projeto de Cluster denominado *PrevMeso*, com o objetivo de aumentar a precisão da previsão do tempo, apontando os lugares onde irão ocorrer os fenômenos naturais. Para tanto, segundo a equipe responsável, é necessário criar uma atmosfera cibernética, onde é representado todo o meio ambiente, incluindo as cidades, florestas, solo, montanhas, oceanos, etc. Esta atmosfera cibernética usa um modelo regional de previsão do tempo, desenvolvido pela Universidade de Oklahoma, chamado *Advanced Regional Prediction System* (ARPS). Assim, levando-se em conta as condições de temperatura, pressão, umidade do ar, direção e velocidade do vento, radiação solar, presença de nuvens, condições do solo e a temperatura do mar, pode-se prever o estado futuro da atmosfera. Neste projeto, um outro problema normalmente enfrentado é o custo de aquisição e de manutenção dos supercomputadores, ou computadores de alto desempenho. Neste

sentido, o projeto construiu um agrupamento ou Cluster de 9 microcomputadores que custa uma ínfima parte das máquinas tradicionais possuindo desempenho semelhante. O ambiente de processamento de alto desempenho do projeto PrevMeso é composto por 16 computadores pessoais com a seguinte configuração: CPU Intel 2.4 GHz , 256 Mb de memória RAM - DDR 333 MHz (512M no Servidor) ,placa de rede Fast-Ethernet. O servidor possui dois discos SCSI 40 Gb enquanto os nós utilizam a tecnologia *diskless*¹ juntamente com o recurso PXE² para efetuar o boot remoto. Os 16 nós obtiveram 32 GFlops de velocidade de pico e 16 GFlops de média (HOGANSON, 1999).

No artigo de Gwo (2000), onde foram explicados os testes realizados com implementações de algoritmos genéticos em um Cluster Beowulf, foi apresentado como um ponto fraco no desempenho, o momento em que as CPUs mas rápidas ficavam ociosas enquanto aguardavam o término do processamento das mais lentas. Isso ocorria devido ao fato da rotina de gerenciamento não realizar um discernimento entre as performances individuais dos componentes do Cluster.

BUSCANDO E MELHORANDO SOLUÇÕES

Os sistemas operacionais estão em constante evolução e atualização, procurando garantir transparência e segurança aos usuários e suas aplicações. Com a possibilidade de trabalhar de forma paralela e distribuída, (como nos Clusters que fazem uso do sistema operacional e exigem deste um grau maior de complexidade), ocorrem investimentos em novas pesquisas para garantir também ao ambiente (onde um grupo de computadores realiza tarefas) o máximo de desempenho e segurança.

Uma das necessidades, que nos parece clara, é a implementação de ferramentas de monitoração e de informações para tomadas de decisões, com base em cada nó registrado no Cluster.

De acordo com o artigo de PETHICK (2003), no OpenMosix, quando um nó está

sobrecarregado, o sistema percorre toda a rede procurando um nó que possua disponibilidade de recursos em termos de memória e CPU, deslocando então a atividade ou parte dela para ser executada. Evidentemente, para que isso ocorra, há um algoritmo que mede a velocidade do nó, carga da CPU e a memória disponível. Essa busca pela disponibilidade de máquinas demanda tempo que poderia ser usado para computação útil.

Imagine agora um repositório de informações sobre cada nó, e estas sendo usadas de acordo com as operações a serem efetuadas. Com este repositório, teríamos um registro de acesso que tornaria simples a criação de ferramentas, monitoração e conseqüentemente as tomadas de decisões.

Nossa proposta é exatamente a de criarmos uma área centralizada de armazenamento de informações, ou seja, o “*Status*” de cada nó que faça parte do Cluster. Dessa forma, toda a informação necessária sobre os nós estaria centralizada numa única fonte de informações.

Criação da Área de Troca: na montagem do Cluster, seriam definidos os nós que iriam receber essas informações. Propomos então, que essa área esteja centralizada no Cluster, procurando deixar todas as máquinas a uma mesma distância desses nós, garantindo assim que todos eles terão distâncias médias equivalentes a essa base de informações.

Armazenamento e Disponibilidade: os nós que forem incumbidos de guardar as informações terão um único ponto (endereço) de entrada, procurando garantir dessa maneira a replicação transparente dos dados e a manutenção de um bom nível de segurança, tolerância às falhas e alta disponibilidade de informações.

Acesso: o acesso deverá ser restrito a eleição do nó, que teria o privilégio de leitura e gravação nesta base comum de informação. Uma vez que o nó seja conhecido, os demais teriam um endereço para onde enviar suas informações, através do uso de um pequeno algoritmo que enviaria todas essas informações.

¹ *Diskless* – permite acessar (fazer boot) no sistema sem que a máquina tenha disco.

² *Preboot Execution Environment (PXE)* – uma forma de ROM de partida inteligente embutida em algumas placas de rede ou placa mãe.

Utilização das Informações: esta área de troca terá como função o auxílio na resolução de problemas, dentre os quais estão os balanceamentos de carga. Para tanto, será feito o uso das estatísticas das informações. Em nossa proposta, os algoritmos para o paralelismo seriam definidos nessa área de troca e esta decidiria a melhor opção para executar a tarefa em paralelo, no momento que fosse disparada uma tarefa a ser executada. No entanto, não ficaríamos restritos a esse tipo de abordagem. Suponhamos um Cluster tipo Beowulf, com 10 máquinas mais o nó mestre (Figura 1).

Nessa situação o nó mestre, ao disparar uma tarefa, daria a possibilidade de seguir duas abordagens diferentes no que diz respeito à paralelização de uma dada tarefa: 1) Essa decisão do paralelismo poderia estar centralizada no mestre, que ao olhar para a área de informações iria disparar o algoritmo de paralelismo baseado nas informações sobre cada nó; ou 2) O mestre simplesmente dispara em direção a área de dados (centralizador de informações) a tarefa a ser processada e nesta área seria tomada a decisão (a melhor decisão) com base em estatística dos dados sobre cada nó.

Consideramos que existe uma vantagem da segunda abordagem em relação à primeira quanto ao tempo utilizado para disparar a tarefa. Na primeira, o mestre deve buscar informações e decidir o paralelismo, já na segunda, o nó mestre somente se preocuparia em disparar a tarefa.

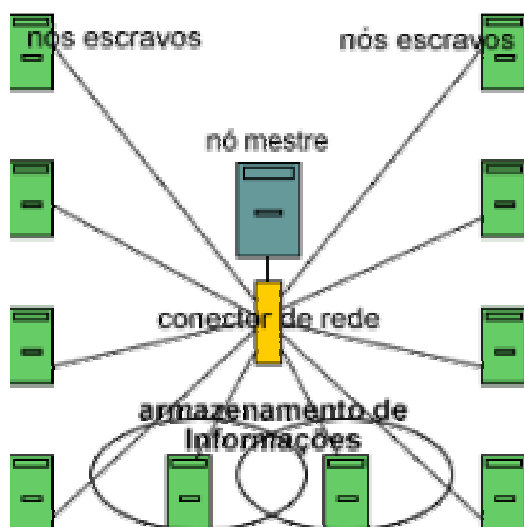


Figura 1. Modelo de um Cluster.

CONCLUSÃO

Temos cada vez mais aplicações baseadas em Cluster rodando. Existe investimento em pesquisas na programação paralela para garantir: alta disponibilidade dos serviços, redução dos riscos em erros operacionais, balanceamento de carga e segurança, entre outros. Pelo que se percebe existem muitas falhas a serem sanadas devido à complexidade em manter controle sobre várias máquinas, sem ter um custo muito grande em termos de tempo e alocação de recursos para o próprio controle.

A nossa proposta em manter uma área de troca de informações (para um gerenciamento e tomada de decisões sobre qualquer tipo de rotina e algoritmos que o sistema em cluster precise) seria de extrema importância para reduzir as falhas e aumentar o desempenho do sistema como um todo.

O que se espera é que os investimentos na área de ensino e pesquisa venham contribuir para que, além da popularização de Clusters computacionais (que já são realidade) tenhamos também a garantia de usar um sistema computacional baseado em Cluster que trabalhe de forma segura e transparente, resolvendo problemas complexos utilizando o mínimo de tempo.

REFERÊNCIAS

APON, A.; BUYYA, R.; JIN, H.; et al. Cluster computing in the classroom: topics, guidelines, and experiences. In: INTERNATIONAL SYMPOSIUM ON CLUSTER COMPUTING AND THE GRID, 3st, 2003, Tokyo. **Anais...** Tokyo: IEEE, 2003. p. 476.

BACKER, V. Para que previsão quantitativa de chuva? Disponível em: <http://150.162.19.150/conheca_projeto.htm>. Acesso em: 20 ago. 2005.

BARAK, A.; LA'ADAN, O.; SHILOH A. Scalable cluster computing with MOSIX for linux. Disponível em: <<http://scholar.google.com/scholar?hl=pt-BR&lr=&q=cache:xrjy0IMFIHoJ:www.mosix.org/mosultp/mosix4linux.pdf+BARAK,+A.%3B+LA%27ADAN,+O.%3B+SHILOH+A.+Scalable+cluster+computing+with+MOSIX+for+linux.+Institute+of+Computer+Science+The+Hebr>>. Acesso em: 20 ago. 2005.

DÍAZ, G.; HOEGER, H.; NUÑEZ, L.A. Clusters de PCs. Universidad de Los Andes. Julho 2002.

Revista Eletrônica - Saber ULA,
<http://www.saber.ula.br/index.html> , acesso 18 ago
 2005.

GWO, J.P.; HOFFMAN, F.M.; HARGROVE, W.W.
 Mechanistic-based genetic algorithm search on a
 beowulf cluster of Linux PCs. In: PROCEEDINGS
 OF THE HIGH PERFORMANCE COMPUTING,
 2000, Washington. **Conference...** Washington: DC,
 2000.

HOGANSON, E. K. Workload Execution Strategies
 and Parallel Speedup on Slustered Computers.
 Computers, IEEE Transactions on, Volume:
 48, Issue: 11, On page(s): 1173-1182,1999. ISSN:
 0018-9340.

MEREDITH, M.; CARRIGAN, T.; BROCKMAN, J.; et
 al. **Exploring beowulf cluster**. Carlinville:
 Blackburn College, 2003.

PETHICK, M.; LIDDLE, M.; WERSTEIN, P.; et al.
 Parallelization of a backpropagation neural network
 on a cluster computer. Disponível em:
 <<http://www.actapress.com/PaperInfo.aspx?PaperID=13847>>. Acesso em: 20 ago. 2005.

PITANGA, M. **Computação em cluster: o estado
 da arte**. Rio de Janeiro: Brasport, 2003.

ROCHA, L.L. Cluster visão geral Universidade
 Católica de Brasília. Julho – 2004. Disponível,
http://scholar.google.com/scholar?hl=pt-BR&lr=&q=cache:K9vyjGc9iwgJ:www.cesmic.ucb.br/Documentacao/ArtigosTutoriais/cluster_vg.pdf+Cluster+vis%C3%A3o+geral. Acesso em: 1o ago.
 2005.

UTHAYOPAS, P.; ANGSKUN, T.; MANEESILP, J.
 Building a Parallel Computer from Cheap PCs:
 SMILE Cluster Experiences. Disponível em:
 <<http://citeseer.ist.psu.edu/uthayopas98building.html>>. Acesso em: 20 ago. 2005.



Recebido 29 nov. 2005
 Aceito 20 mai. 2006