INFORMAÇÕES GERAIS DO TRABALHO

Título do Trabalho: Implantação de um Cluster Computacional *Beowulf*

Autor (es): Samuel Terra Vieira
Vinicius Duarte Batista

Orientador (es):
Diego Mello da Silva
Everthon Valadão dos Santos
Rafael Vinicius Tayette
Reginaldo Gonçalves

Palavras-chave: *Beowulf, Cluster, High Performance Computing, MPI*

Campus: Formiga – MG

Área do Conhecimento (CNPq): Ciências Exatas e da Terra / Ciência da Computação / Sistemas de Computação / Sistemas Distribuídos.

RESUMO

Este trabalho trata de algumas das atividades desenvolvidas no projeto de pesquisa "*Cluster Computacional para Implementação de Aplicações de Computação Científica de Alto Desempenho em Ciências e Engenharias*". É apresentado um estudo sobre a computação de alto desempenho (HPC) em um grupo de computadores heterogêneos conectados por uma rede local. Foi projetado e implantado um *cluster beowulf* no IFMG *Campus* Formiga, utilizando 15 desktops ociosos com um total de 36 processadores, no qual foram conduzidos testes de escalabilidade para ilustrar o ganho de desempenho (*speed-up*) na execução de um algoritmo distribuído. Foi observado um significativo ganho de tempo (69,8% ±7,9%) quando o algoritmo é distribuído entre 2 a 16 processadores (4 computadores quad-core), porém a partir de 17 processadores (5 ou mais computadores) os custos relacionados com a comunicação entre os computadores e gerenciamento do cluster ultrapassa o benefício de ganho de tempo. Tal problema pode ser minimizado com o investimento em uma rede de dados de baixa latência. Como neste trabalho são abordados tópicos essenciais à implantação e uso de um *cluster beowulf*, espera-se que auxilie estudantes interessados em computação de alto desempenho, incentivando-os a desenvolver seu próprio *cluster* computacional de baixo custo.

INTRODUÇÃO

Em problemas atuais de ciências e engenharia, observa-se uma crescente necessidade de uso de recursos computacionais com grande poder de processamento, para resolução de experimentos complexos, massivamente paralelizados ou de larga escala (BACELLAR, 2010). Para que tal poderio computacional seja alcançado e atenda à demanda, existe a opção de se utilizar uma estação de trabalho de alto desempenho. Também conhecida como *workstation*, consiste em um computador robusto, dedicado e fortemente acoplado, ou seja, é um único sistema computacional com vários processadores, compartilhando uma mesma memória principal e sendo controlado por um único sistema operacional. Porém, uma séria desvantagem das *workstations* é seu custo elevado e, por isso, sua utilização atende um público limitado (HENNESSY, 2012). Uma alternativa mais acessível e prática, fracamente acoplada mas que também consegue atender a altas demandas de processamento, com desempenho comparável ou superior a uma *workstation*, é a utilização de um *cluster* computacional (DE VASCONCELOS, 2009). Neste trabalho, iremos abordar uma arquitetura de cluster chamada *beowulf*, construída a partir de quaisquer computadores ociosos, de hardware heterogêneo, que estejam à disposição da instituição (TONIDANDEL, 2008).

Um *cluster* computacional utiliza no mínimo dois mas, tipicamente, muito mais computadores sendo um deles o “mestre” e os demais “escravos”, que irão processar as tarefas por ele designadas. Tal processamento é realizado de forma distribuída em diversos computadores e seus processadores porém, dá ao usuário a impressão de ser conduzido em um sistema com um único computador. Observe que, no caso de um único sistema computacional (ex.: uma *workstation*), as tarefas do problema podem ser divididas via programação *multithreading* (várias linhas de execução paralelas ou concorrentes) que se beneficiam do barramento comum entre a memória principal e o(s) processador(es) (MAIA, 1998). Já no caso dos *clusters*, como é composto de vários computadores, é necessário utilizar um mecanismo de passagem de mensagens para dividir as tarefas do problema, do mestre aos escravos, o que naturalmente incorre em latências da comunicação em rede. Vale observar que um *cluster* não apresentará melhoria de desempenho linear (ex.: quatro computadores não realizarão a tarefa em exato ¼ do tempo), pois haverá um custo computacional (*overhead*) inerente à troca de informações entre o mestre e seus escravos.

Tais *clusters* computacionais, tipo de sistema computacional de alto desempenho, ganham cada vez mais espaço e se tornam mais populares devido à frequente busca pelo melhor custo/benefício computacional. Também, um outro motivo para a utilização de *clusters* computacionais pode ser a necessidade de tolerância a falhas, bem como uma alta disponibilidade de serviços, questões que um cluster também consegue atender (LEAL & FILHO, 2012).

METODOLOGIA

O ponto forte de se construir um *cluster beowulf* é a utilização de máquinas ociosas, geralmente obsoletas e com configurações distintas (STERLING, 2002). Neste quesito, ganham destaque e comumente são adotados *desktops*, que outrora foram utilizados individualmente. Levando em consideração que um

cluster computacional é um conjunto de computadores operando de forma distribuída, pode-se notar que há recursos que vão influenciar no desempenho desses computadores, tipicamente heterogêneos. Podemos citar componentes tais como processador (CPU), memória RAM, interface de rede, placa mãe e disco rígido. No entanto, o *cluster* irá se apresentar para o usuário como um único sistema. Com isso, o poder de processamento individual de cada máquina irá se tornar uma pequena parte do *cluster* em si, que por sua vez se apresenta como uma máquina robusta e com um grande poder de processamento mas, na realidade, composta de várias computadores com seus componentes pouco potentes. Assim, um *cluster beowulf* por si só nada mais é que um aglomerado de computadores independentes (na maioria das vezes obsoletos) e que estão interligados em rede, no qual há um mestre que faz a divisão e atribuição de tarefas entre seus escravos.

No desenvolvimento do *cluster* foram empregados computadores outrora utilizados nos laboratórios de informática do IFMG *Campus* Formiga. Após vários anos de uso, tais máquinas foram sendo substituídas, principalmente pelo fato de apresentarem frequentes "travamentos" e baixo desempenho no uso do sistema operacional *Microsoft Windows*. Pode-se observar como um ponto positivo o fato das máquinas apresentarem desempenho insuficiente apenas no uso do MS Windows, não impossibilitando seu aproveitamento em outros sistemas operacionais. Optamos por utilizar o ABC GNU/Linux, uma distribuição linux *open cluster* especializada para **A**utomatização de um **B**eowulf **C**luster (SINGH et al., 2012).

Na arquitetura deste sistema, todas as máquinas realizam um *Network Boot* (inicialização pela rede) onde obtém seu sistema operacional e configurações pela rede local (LAN). A exceção é o próprio computador mestre, onde já deve ter sido previamente instalado o ABC GNU/Linux e estar sempre ligado para receber e gerenciar os computadores escravos. Para utilizar inicialização pela rede, a opção PXE (*network boot*) deve ser habilitada na BIOS de cada máquina que será utilizada como nó escravo. A grande vantagem de se realizar o *network boot* é a facilidade em serem adicionados novos computadores escravos no *cluster*: basta ativar o PXE e configurar o *network boot* com maior prioridade (*boot order*), conectar a interface de rede do novo computador no mesmo concentrador que o mestre do cluster *beowulf* e reiniciar o computador.

Após a instalação, configuração e implantação do *cluster beowulf* no Laboratório de Arquitetura e Redes (L.A.R.) do IFMG Campus Formiga, foram realizados testes preliminares para mensurar o *speed-up*, ou seja, o aumento de desempenho entre dois sistemas executando uma mesma tarefa. A configuração atual do cluster beowulf do IFMG *Campus* Formiga contém 15 computadores totalizando 36 processadores. Foi comparando o tempo de execução de uma tarefa no *cluster* com vários computadores e em um único computador. O algoritmo utilizado foi o programa `/usr/share/doc/lam4-dev/examples/main/pi/cpi.c` da biblioteca LAM (*Local Area Multicomputer*), uma implementação *open-source* do padrão *Message Passing Interface* (MPI), que realiza o cálculo do valor aproximado de π (pi) pela integração da função $f(x) = 4 / (1 + x^2)$, com 100 mil iterações (BURNS et al., 1996; GROPP et al., 1999).

O computador mestre foi instruído a executar o algoritmo com o parâmetro (`mpiexec.openmpi -n`) da quantidade de processadores variando de 1 a 36 (GABRIEL et al., 2004). Para cada configuração dessas, foram conduzidas 5 repetições e calculada a média (e desvio padrão) do tempo gasto na execução do algoritmo (duração), métrica utilizada para mensurar o *speed-up*. A configuração de hardware e software do *cluster* é apresentada abaixo:

- Hardware: 15 computadores (1 mestre e 14 escravos)
  - 05 computadores (1 deles é o mestre): CPU: AMD Phenom(tm) 9650 Quad-Core (2300 MHz, 64 bits); RAM: 4x1 GB DIMM 800 MHz
  - 10 computadores: CPU: AMD Athlon(tm) 5400B Dual-Core (3200 MHz, 64 bits); RAM: 4x1 GB DIMM 667 MHz
  - OBS.: todos os 15 computadores (1 mestre e 14 escravos) possuem em comum:
    - NIC: 100 Mbit/s, Realtek RTL8111/8168B PCI Express Gigabit Ethernet controller
    - HD: 160 GB SAMSUNG HD161HJ (SATA), 142 GiB EXT3, 6 GiB linux-swap
    - MB: M3A78 ASUSTeK Computer INC.
- Software:
  - Sistema Operacional: ABC_GNULinux_1.0rev2_x86
  - Linux Kernel: 2.6.28-16-generic
  - Bibliotecas: OpenMPI v1.3-2, LAM4-dev v7.1.2-1.5; GCC v4.3.3-1
- Rede:
  - Concentrador: switch hub Encore 24 portas 10/100 Mbits/s ENH924-AUT
  - Cabos UTP Cat-5e

RESULTADOS E DISCUSSÕES

A Figura 1 exibe um gráfico do *speed-up* observado, onde o eixo das ordenadas (duração, em segundos) é a métrica do tempo total gasto para executar todos os cálculos de forma distribuída, enquanto o eixo das abscissas (nº de processadores) informa o parâmetro da quantidade de CPUs de computadores escravos utilizados na execução do algoritmo. No gráfico abaixo, o ponto de inflexão da curva indica a quantidade de processadores em que se tem o maior ganho de desempenho no tempo gasto para a realização da tarefa (69,8% ±7,9%). Note que quando é utilizada uma quantidade superior a 5 processadores, o custo do gerenciamento e comunicação entre os computadores do *cluster* começa comprometer o benefício de tempo na distribuição da execução do algoritmo. Já com 17 processadores, a duração da tarefa supera a original.

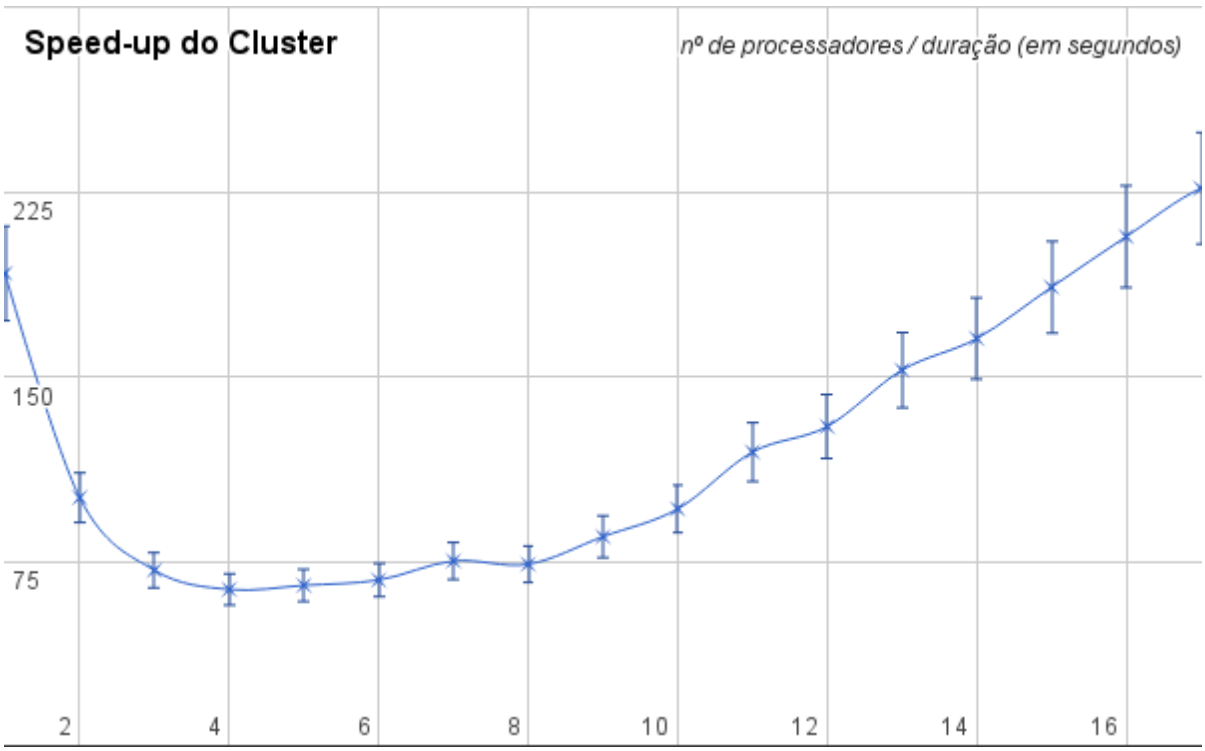

Figura 1 - *Speed-up* observado no *cluster* para o programa cpi.c

Na Figura 2, o eixo das abscissas é apresentado em escala logarítmica ($\log_2$). Ela exibe o gráfico anterior com uma visão diferente, complementada com uma linha representando o limite teórico de *speed-up*, caso não houvesse custo computacional adicional (*overhead*) inerente à troca de informações entre o mestre e seus escravos, ao gerenciamento do *cluster* e às latências de rede. Vale ressaltar que quando utiliza-se mais computadores (e processadores) na execução do algoritmo, o desempenho observado nunca será inferior a um ganho linear de escala, portanto, o desempenho de um *cluster* sempre será superlinear. Portanto, o que busca-se num *cluster* é uma maior aproximação do *speed-up* apresentado em relação ao limite teórico linear.

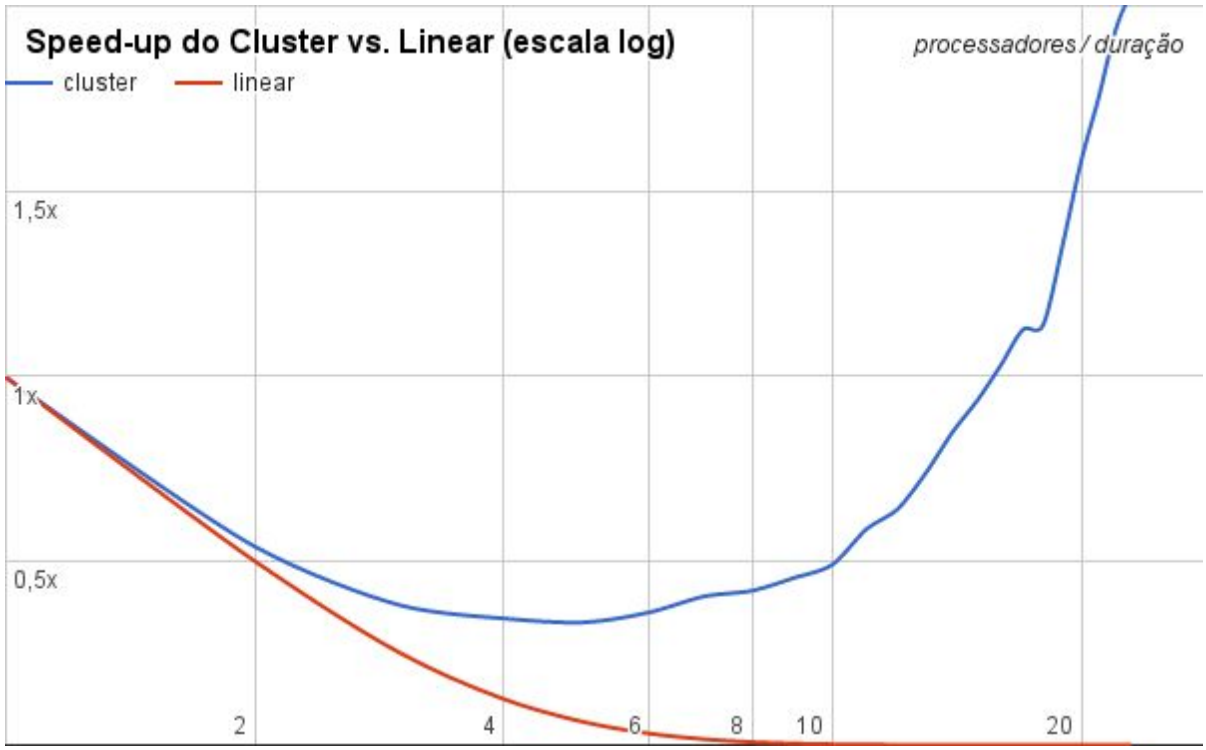

Figura 2 - *Speed-up* do *cluster* (superlinear) vs linear

Observe que, para determinar o desempenho de um computador, outras métricas podem ser utilizadas. Por exemplo, a quantidade de operações de ponto flutuante que o sistema computacional pode realizar por segundo, mais conhecido como FLOP/S, do original *Floating point Operations Per Second* (ROCHA, 2007). Independente da métrica escolhida, seja ela *speed-up* (ganho de tempo) ou flop/s (taxa de cálculos), para o *cluster* apresentar um desempenho favorável na execução distribuída de tarefas, alguns requisitos importantes devem ser levados em consideração. Dentre eles, a escolha do sistema operacional, as bibliotecas de paralelização e ferramentas de gerenciamento, a tecnologia das interfaces de rede e o(s) equipamento(s) concentrador(es), dentre outros, podem influenciar no desempenho apresentado pelo cluster. Atualmente, o fator limitador do desempenho do *cluster beowulf* implantado no *campus* não são os computadores em si, mas o desempenho da rede que propicia a comunicação entre eles. O concentrador da rede utilizado é um switch Fast Ethernet 100BASE-T (IEEE 802.3u), de propósito geral, apresentando vazão máxima de 100 Mbit/s, RTT (atraso bidirecional) de 100-140 µseg e latência (unidirecional) de 5-7 µseg.

CONCLUSÕES

Quando um projeto requer alto desempenho e alta disponibilidade, o maior desafio enfrentado é a limitação de orçamento. Uma solução prática, que na maioria dos casos irá atender às limitações de orçamento (e ainda assim alcançar o desempenho desejado) é a utilização de um *cluster* computacional. Uma arquitetura popular que permite a utilização de hardware heterogêneo é o *cluster beowulf*, que apresenta um baixo custo desde o planejamento até a sua implantação, se comparado a soluções comerciais como estações de trabalho de alto desempenho (*workstations*).

Uma outra vantagem da utilização de um *cluster beowulf* é o vasto campo de aplicações que se beneficiariam dele, visto que consiste em um *cluster* flexível, heterogêneo e *open source*. Caso algum estudante ou pesquisador queira aprofundar seus conhecimentos no assunto, facilmente encontrará amplo material na literatura. Existem algumas distribuições GNU/Linux especializadas em *clusters beowulf*, prontas para serem instaladas em *clusters* computacionais "caseiros", tornando assim sua implantação mais ágil. Este trabalho adotou a distribuição ABC GNU/Linux, que automatiza a implantação do cluster, bem como fornece uma interface web ("ganglia") para monitoração dos recursos utilizados (MASSIE, 2004).

É importante observar que com o mecanismo PXE de *network boot*, não é necessário realizar qualquer formatação ou alteração no sistema operacional dos computadores escravos, podendo assim adicionar ao *cluster beowulf* computadores que ainda estejam em uso institucional, mas no momento, ociosos. Assim, cada instituição que possua dezenas de computadores disponíveis em seus laboratórios de informática (ociosos em determinados horários e dias da semana), pode utilizá-los quando necessário como se fossem um único computador de altíssimo desempenho, ou seja, cada instituição pode ter seu supercomputador com custo praticamente zero. Para tal, basta que seja mantido ligado e conectado em rede um computador dedicado, devidamente instalado com uma distribuição linux *open cluster* (ex.: ABC GNU/Linux), para atuar como mestre do *cluster beowulf* e gerenciar os computadores escravos.

Nos experimentos, observamos que não basta adicionar computadores para melhorar o desempenho do *cluster*, então, como trabalhos futuros, pretendemos melhorar a infraestrutura de comunicação substituindo o concentrador de rede por um que apresente uma menor latência de comunicação. Os 15 computadores atualmente utilizados no *cluster* já possuem interface Gigabit Ethernet 1000BASE-T (IEEE 802.3ab), que garante latências de 1-12 µseg (7 vezes menor que a atual). Também, será levantada a viabilidade de um investimento de médio custo, atualizando as interfaces de rede dos computadores e adquirindo um concentrador mais robusto, de propósito específico para a computação de alto desempenho. Por exemplo, a tecnologia InfiniBand ofereceria latências de 0,5-5 µseg, ou seja, 14 vezes menor que a atual (Infiniband Trade Association, 2000). Por outro lado, o padrão 10 Gigabit Ethernet 1000BASE-T (IEEE 802.3an) suporta uma vazão de até 10 Gbit/s (100 vezes maior que a atual) com latências de 2 - 4 µseg (3,5 vezes menor que a atual).

REFERÊNCIAS BIBLIOGRÁFICAS

BACELLAR, Hilário Viana. **Cluster: Computação de Alto Desempenho**. Campinas: Instituto de Computação, Universidade Estadual de Campinas, 2010.

BURNS, Greg; DAOUD, Raja; VAIGL, James. **LAM: An open cluster environment for MPI**. In: Proceedings of supercomputing symposium. 1994. p. 379-386

DE VASCONCELOS, Luiz Eduardo Guarino et al. **Proposta de um Laboratório de Alto Desempenho de Cluster Beowulf** para Instituições de Ensino. Relatório Técnico: Faculdade de Tecnologia de Guaratinguetá, 2009.

GABRIEL, Edgar et al. **Open MPI: Goals, concept, and design of a next generation MPI implementation**. In: European Parallel Virtual Machine/Message Passing Interface Users' Group Meeting. Springer Berlin Heidelberg, 2004. p. 97-104.

GROPP, William; LUSK, Ewing; SKJELLUM, Anthony. **Using MPI: portable parallel programming with the message-passing interface**. MIT press, 1999.

HENNESSY, John L.; PATTERSON, David A. **Computer architecture: a quantitative approach**. Elsevier, 2012.

INFINIBAND TRADE ASSOCIATION et al. **InfiniBand Architecture Specification: Release 1.0**. InfiniBand Trade Association, 2000.

LEAL, Liliam Barroso; FILHO, Francisco Xavier de Vasconcelos. **Uma Abordagem para Alta Demanda de Processamento Utilizando Cluster de Beowulf**. Ponto de Presença da Rede Nacional de Ensino e Pesquisa no Piauí, Piauí, 2012.

MAIA, Luiz Paulo. **Multithread**. Monografia - Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, 1998.

MASSIE, Matthew L.; CHUN, Brent N.; CULLER, David E. The ganglia distributed monitoring system: design, implementation, and experience. Parallel Computing, v. 30, n. 7, p. 817-840, 2004.

ROCHA, Ricardo. **Programação Paralela e Distribuída - Métricas de Desempenho**. Notas de aula: DCC-FCUP, Universidade do Porto, 2007.

SINGH, A.; CHAUHAN, Rajesh; THAKUR, Balvir Singh. **Beowulf Cluster: Cost Effective Solution for E-Governance**. C. Unnithan, & B. Fraunholz, Towards e-Governance in The Cloud: Frameworks, Technologies and Best Practices, p. 18-22, 2012.

STERLING, Thomas Lawrence. **Beowulf cluster computing with Linux**. MIT press, p. 369 - 371, 2002.

TONIDANDEL, D. A. V. **Manual de montagem de um cluster Beowulf sob a plataforma GNU/Linux**. 2008. 89 f. Monografia (Graduação em Engenharia de Controle e Automação) – Escola de Minas, Universidade Federal de Ouro Preto, Ouro Preto, MG, 2008.