

**LIGA DE ENSINO DO RIO GRANDE DO NORTE
FACULDADE NATALENSE PARA O DESENVOLVIMENTO DO RN**

Kleber Tadeu de Santana

UM ESTUDO SOBRE A TECNOLOGIA DE COMPUTAÇÃO EM CLUSTER

UM ESTUDO SOBRE A TECNOLOGIA DE COMPUTAÇÃO EM CLUSTER

Monografia apresentada ao curso de
Especialização em Redes de Computadores da
Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte como requisito para
obtenção do título de Especialista em Redes de
Computadores

Orientador

Abelardo Pereira da Rocha Neto, Msc

NATAL/RN

2007

KLEBER TADEU DE SANTANA

UM ESTUDO SOBRE A TECNOLOGIA DE COMPUTAÇÃO EM CLUSTER

UM ESTUDO SOBRE A TECNOLOGIA DE COMPUTAÇÃO EM CLUSTER

Monografia apresentada ao curso de Especialização em Redes de Computadores da Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte como requisito para obtenção do título de Especialista em Redes de Computadores

Orientador:

Aluizio Ferreira da Rocha Neto, Msc.

NATAL/RN

2007

RB- 30.283

FARN

BIBLIOTECA

Nº Reg M000748

14 / 12 / 07

004.72

5223e

2007

Catálogo na Publicação – Biblioteca da FARN
Setor de Processos Técnicos

Santana, Kleber Tadeu de

Um estudo sobre a tecnologia de computação em Cluster / Kleber Tadeu de Santana. – Natal: FARN, 2007.

74 f.

Orientador: Aluizio Ferreira da Rocha Neto (M. Sc.)

Monografia (Especialização em Redes de Computadores) – Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte.

1. Redes de Computadores – Monografia. 2. Computação Cluster – Monografia. 3. Supercomputadores – Monografia. I. Título.

RN/FARN/BC

CDU 004.72

KLEBER TADEU DE SANTANA

UM ESTUDO SOBRE A TECNOLOGIA DE COMPUTAÇÃO EM CLUSTER

Monografia apresentada ao curso de Especialização em Redes de Computadores da Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte e aprovada em 01 de junho de 2007, pela banca examinadora composta pelos seguintes membros:

Aluízio Ferreira da Rocha Neto, Msc.
Orientador

Gleydson Azevedo Ferreira de Lima, Msc.
Membro Examinador

AGRADECIMENTOS

Ao meu Orientador Professor Alcirio Pereira de Souza Neto, pelo incentivo e proeza no auxílio às atividades e discussões sobre o andamento e elaboração desta Monografia de Conclusão de Curso, bem como ao longo do curso, nas diversas atividades por ele ministradas.

Aos demais idealizadores e colaboradores do Curso de Pós-Graduação em Redes de Computadores da FARN - Faculdade Novecento para o Desenvolvimento do Rio-Grande do Norte.

A todos os professores e seus convidados pela dedicação e entusiasmo demonstrado ao longo do curso.

Aos colegas de classe pela espontaneidade e alegria na troca de informações e experiências técnicas, bem como a demonstração de companheirismo e solidariedade.

Aos meus pais, irmão, minha esposa Isabele Santiago, meu filho Kleber Junior que, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa da minha vida.

E, finalmente, a DEUS pela oportunidade e pelo privilégio que me foram dados em compartilhar minha experiência e, ao frequentar este curso, perceber e sentir pela relevância de temas que não fazem parte, em profundidade, dos conhecimentos técnicos aprendidos.

A meu filho primogênito, Kleber Jr., que nos presenteou com o seu nascimento em plena elaboração desse trabalho.

À minha esposa, Isabele, pelo apoio constante e compreensão.

Aos meus pais, como resultado de meu esforço e também do apoio que deles recebi em todas as etapas da minha vida.

AGRADECIMENTOS

Ao meu Orientador Professor Aluizio Ferreira da Rocha Neto, pelo incentivo e presteza no auxílio às atividades e discussões sobre o andamento e normatização desta Monografia de Conclusão de Curso, bem como ao longo do curso, nas diversas atividades por ele ministradas.

Aos demais idealizadores e colaboradores do Curso de Pós-Graduação em Redes de Computadores da FARN – Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte.

A todos os professores e seus convidados pela dedicação e entusiasmo demonstrado ao longo do curso.

Aos colegas de classe pela espontaneidade e alegria na troca de informações e experiências técnicas, numa admirável demonstração de companheirismo e solidariedade.

Aos meus pais, irmão, minha esposa Isabele Santana, meu filho Kleber Júnior. que, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida.

E, finalmente, a DEUS pela oportunidade e pelo privilégio que me foram dados em compartilhar tamanha experiência e, ao frequentar este curso, perceber e atentar para a relevância de temas que não faziam parte, em profundidade, dos conhecimentos técnicos acumulados.

RESUMO

A necessidade de se obter elevada capacidade de processamento em diversas aplicações foi uma motivação para o desenvolvimento de supercomputadores. Porém, alguns problemas, incluindo custos, levaram os cientistas a projetar e implementar uma tecnologia alternativa, para a necessidade de alto desempenho: a Tecnologia de Computação em Cluster. Quando se menciona tecnologia de cluster de computadores é inevitável a associação com centros de pesquisa realizando grandes quantidades de computação intensiva. Contudo, com a redução considerável dos custos dos computadores convencionais e a sua utilização em larga escala, presencia-se a inserção dessa tecnologia também em ambientes corporativos, principalmente em pequenas e médias empresas. Os clusters são utilizados para diversos propósitos, cada um deles requerendo características distintas. A utilização de um agrupamento de máquinas como solução para computação de alto desempenho, ou de alta disponibilidade ou ainda de balanceamento de carga é uma realidade presente em qualquer rede de computadores. Esse trabalho se propõe a promover um estudo teórico e dirigido, bem como implementar uma visão geral sobre a Tecnologia de Cluster, favorecendo a ambientação de administradores de rede e outros profissionais de informática no campo da Computação em Cluster, em particular na tecnologia de alta disponibilidade.

Keywords: Supercomputers, Cluster, High Performance, High Availability, Load Balancing, Cluster Management.

Palavras-chave: Supercomputadores. Cluster. Alto Desempenho. Alta Disponibilidade. Balanceamento de Carga. Gerenciamento de Cluster.

LISTA DE ABSTRACTS

The necessity to obtain high processing capacity in several applications was a motivation to develop supercomputers. However, some problems, including costs, induced scientists to design and implement an alternative technology, due to a high performance necessity: Cluster Computing Technology. When cluster computing technology is mentioned it is inevitable to associate research centers there by accomplishing great quantities of intensive data processing. Although, with the a considerable reduction of the costs of the conventional computer and its large scale use, an insertion of this technology is also made within corporative environments, mainly in small and medium companies. Clusters are used for many purposes, each one of them requiring distinct kinds of characteristics. The use of a machinery gathering as a solution for a high performance, or high availability or load balancing is a reality in any computer network. This work proposes to promote a theoretical and directed study, as well as implementing a general view about the Cluster Computing, favoring the adaptation of net administrators and other professionals in the Cluster's processing, especially in the high availability technology, load balancing and high performance processing.

Key-words: Supercomputers. Cluster. High Performance. High Availability. Load Balancing. Cluster Management.

FIGURA 10: UMA VISÃO DO MODULEWARE	31
FIGURA 12: ARQUITETURA MIMD	34
FIGURA 13: ARQUITETURA SIND	35
FIGURA 14: ARQUITETURA MIMD	35
FIGURA 15: CLUSTER DE ALTA DISPONIBILIDADE ASSIMÉTRICO	35
FIGURA 16: CLUSTER DE ALTA DISPONIBILIDADE SIMÉTRICO	35
FIGURA 17: COMPARTILHAMENTO TOTAL	36
FIGURA 18: SEM COMPARTILHAMENTO	36
FIGURA 19: ESPELHAMENTO DE DADOS	37
FIGURA 20: APLICAÇÃO EXATA	39
FIGURA 21: APLICAÇÃO EXATA - ALTAMENTE DISPONÍVEL	39
FIGURA 22: CLUSTER HEARTBEAT PADRÃO	41

LISTA DE FIGURAS

FIGURA 1: O ÁBACO HOJE: UM CAIXEIRO DE LOJA EM BEIJING, CHINA, USANDO O ÁBACO.....	19
FIGURA 2: MARK I - 1943. OCUPAVA 120 M3	20
FIGURA 3: MODELO VON NEUMANN - VISÃO SIMPLIFICADA.....	21
FIGURA 4: PARALELIZANDO AS TAREFAS	21
FIGURA 5: CLUSTER BEOWULF	22
FIGURA 6: CLUSTER ALTA DISPONIBILIDADE – ARQUITETURA CLÁSSICA.....	27
FIGURA 7: CLUSTER BALANCEAMENTO DE CARGA – EXEMPLO CONCEITUAL.....	29
FIGURA 8: CLUSTER BEOWULF – TOPOLOGIA BÁSICA.....	31
FIGURA 9: CLUSTER BEOWULF – ASSOCIADO À SOLUÇÃO DE STORAGE.....	31
FIGURA 10: UMA VISÃO DO MIDDLEWARE.....	33
FIGURA 11: ARQUITETURA SISD	34
FIGURA 12: ARQUITETURA MISD	34
FIGURA 13: ARQUITETURA SIMD	35
FIGURA 14: ARQUITETURA MIMD.....	35
FIGURA 15: CLUSTER DE ALTA DISPONIBILIDADE ASSIMÉTRICO	55
FIGURA 16: CLUSTER DE ALTA DISPONIBILIDADE SIMÉTRICO	55
FIGURA 17: COMPARTILHAMENTO TOTAL	56
FIGURA 18: SEM COMPARTILHAMENTO	56
FIGURA 19: ESPELHAMENTO DE DADOS.....	57
FIGURA 20: APLICAÇÃO EXATA	59
FIGURA 21: APLICAÇÃO EXATA – ALTAMENTE DISPONÍVEL	59
FIGURA 22: CLUSTER HEARTBEAT PADRÃO	61

LISTA DE TABELAS

TABELA 1: NÍVEIS DE DISPONIBILIDADE27

TABELA 2: ESQUEMA DE PARTICIONAMENTO60

CI	<i>Cluster Infra-structure</i>
CMS	<i>Cluster Management Software</i>
COTS	<i>Commodity off the Shelf</i>
CPU	<i>Central United Processor</i>
CSMA/CD	<i>Carrier Sense Multiple Access with Collision Detection</i>
DHCP	<i>Dynamic Host Configuration Protocol</i>
DNS	<i>Data Named Server</i>
DRDB	<i>Distributed Replicated Block Device</i>
FAI	<i>Fully Automatic Installation</i>
FTP	<i>File Transport Protocol</i>
GFS	<i>Global File System</i>
GIGAFLOP	<i>Unidade de Medida de Processamento</i>
GNOME	<i>GNU Network Object Model Environment</i>
G4U	<i>Ghost For Unix</i>
HA	<i>High Availability</i>
HD	<i>Hard Disk</i>
HPC	<i>High Performance Cluster</i>
HTTP	<i>HyperText Transfer Protocol</i>
IP	<i>Internet Protocol</i>
IPAT	<i>IP Address Takeover</i>
LB	<i>Load Balance</i>
MIMD	<i>Multiple Instruction Multiple Data</i>
MISD	<i>Multiple Instruction Single Data</i>
MPP	<i>Massively Parallel Processor</i>
MPI	<i>Message Passing Interface</i>
NFS	<i>Network File System</i>
NIS	<i>Network Information Service</i>
NOW	<i>Network of Workstations</i>
NUMA	<i>Non-Uniform Memory Access</i>

LISTA DE ABREVIATURAS

AFS	<i>Andrew File System</i>
CCS	<i>Computing Center Software</i>
CI	<i>Cluster Infra-structure</i>
CMS	<i>Cluster Management Software</i>
COTS	<i>Commodity off the Shelf</i>
CPU	<i>Central United Processor</i>
CSMA/CD	<i>Carrier Sense Multiple Access with Collision Detection</i>
DHCP	<i>Dynamic Host Configuration Protocol</i>
DNS	<i>Data Named Server</i>
DRDB	<i>Distributed Replicated Block Device</i>
FAI	<i>Fully Automatic Installation</i>
FTP	<i>File Transport Protocol</i>
GFS	<i>Global File System</i>
GIGAFLOP	<i>Unidade de Medida de Processamento</i>
GNOME	<i>GNU Network Object Model Environment</i>
G4U	<i>Ghost For Unix</i>
HA	<i>High Availability</i>
HD	<i>Hard Disk</i>
HPC	<i>High Performance Cluster</i>
HTTP	<i>HyperText Transfer Protocol</i>
IP	<i>Internet Protocol</i>
IPAT	<i>IP Address Takeover</i>
LB	<i>Load Balance</i>
MIMD	<i>Multiple Instruction Multiple Data</i>
MISD	<i>Multiple Instruction Single Data</i>
MPP	<i>Massively Parallel Processor</i>
MPI	<i>Message Passing Interface</i>
NFS	<i>Network File System</i>
NIS	<i>Network Information Service</i>
NOW	<i>Network of Workstations</i>
NUMA	<i>Non-Uniform Memory Access</i>

OPEN SOURCE	<i>Código Livre</i>	
OSCAR	<i>Open Source Cluster Application Resources</i>	
PBS	<i>Portable Batch System</i>	
PC	<i>Personal Computer</i>	12
PVM	<i>Parallel Virtual Machine</i>	12
PVFS	<i>Parallel Virtual File System</i>	13
PXE	<i>Pre-boot eXecution Environment</i>	13
RAID	<i>Redundant Array of Inexpensive Disks</i>	17
RAM	<i>Random Acces Memory</i>	21
RPM	<i>Red Hat Package Manegement</i>	23
SCMS	<i>Scalable Cluster Management System</i>	24
SIMD	<i>Single Instruction Multiple Data</i>	27
SISD	<i>Single Instruction Single Data</i>	29
SMP	<i>Symmetric Multi-Processing</i>	32
SPOF	<i>Single Point Of Failure</i>	33
SSH	<i>Secure Shell</i>	36
SSI	<i>Single System Image</i>	37
TCP	<i>Transmission Control Protocol</i>	39
TFLOPS	<i>TeraFlops</i>	42
TI	<i>Tecnologia da Informação</i>	45
UMA	<i>Uniform Memory Access</i>	46
xCAT	<i>Extreme Cluster Administration</i>	47

5.1.1 Considerações Finais	51
5.1.2 Escolha da Arquitetura do hardware	52
5.1.3 Escolha da Rede de Interconexão	52
5.1.4 Escolha do Sistema Operacional	52
5.1.5 Escolha do Software de Comunicação	53
5.1.6 Escolha do Software de Gerenciamento	53
5.2 CONSTRUINDO UM AMBIENTE DE ALTA DISPONIBILIDADE	54
5.2.1 Alta Disponibilidade na Prática	54
5.3 O PROJETO - APLICAÇÃO EXATA	59
5.3.1 Preparando as Máquinas	62
5.3.2 Instalando e Configurando o Hamilton	63
5.3.3 Instalando e Configurando o Mono	65
6 CONCLUSÕES	72
REFERÊNCIAS	73

SUMÁRIO

1 INTRODUÇÃO	12
1.1 OBJETIVOS	12
1.1.1 Objetivo Geral	12
1.1.2 Objetivos Específicos	13
1.2 ORGANIZAÇÃO DO TRABALHO	13
2 REFERENCIAL TEÓRICO	15
2.1 HISTÓRIA E EVOLUÇÃO DOS (SUPER) COMPUTADORES	17
2.2 FUNDAMENTOS DE CLUSTER	21
3 TECNOLOGIA DE CLUSTERS	25
3.1 TIPOS DE CLUSTER	25
3.1.1 Alta Disponibilidade	25
3.1.2 Balanceamento de Carga	27
3.1.3 Alto Desempenho	29
3.2 ASPECTOS RELEVANTES	32
3.2.1 Imagem Única do Sistema - SSI	32
3.2.2 Arquitetura do Hardware	33
3.2.3 Tecnologia de Redes para Cluster	36
3.2.4 Programação Paralela e Distribuída	37
3.2.5 Sistema de Arquivos Distribuídos	39
4 GERENCIAMENTO DE CLUSTERS	42
4.1 INSTALAÇÃO E CONFIGURAÇÃO	42
4.2 ESCALONAMENTO DE JOBS E RECURSOS	45
4.3 MONITORAÇÃO	46
4.4 POLÍTICA DE SEGURANÇA E BACKUP	47
5 CONSTRUÇÃO DE CLUSTERS	51
5.1 CONSIDERAÇÕES NO PROJETO	51
5.1.1 Considerações Físicas	51
5.1.2 Escolha da Arquitetura do hardware	52
5.1.3 Escolha da Rede de Interconexão	52
5.1.4 Escolha do Sistema Operacional	52
5.1.5 Escolha do Software de Comunicação	53
5.1.6 Escolha do Software de Gerenciamento	53
5.2 CONSTRUINDO UM AMBIENTE DE ALTA DISPONIBILIDADE	54
5.2.1 Alta Disponibilidade na Prática	54
5.3 O PROJETO – APLICAÇÃO EXATA	59
5.3.1 Preparando as Máquinas	62
5.3.2 Instalando e Configurando o Heartbeat	63
5.3.3 Instalando e Configurando o Mon	65
6 CONCLUSÕES	72
REFERÊNCIAS	73

1 INTRODUÇÃO

A importância que os computadores possuem nos dias atuais vem se tornando ponto crítico em nossas atividades diárias. As evoluções tecnológicas vividas por nossa sociedade nos últimos anos têm evidenciado o valor da informação e provocado uma utilização crescente de computadores, em suas mais diversas aplicações.

Algumas aplicações são, de fato, extremamente dependentes de recursos computacionais, como por exemplo, a renderização de imagens 3D complexas e softwares de modelagem e simulações específicas em aplicações científicas. Havia um tempo em que se pensava que aplicações como essas só poderiam ser concluídas mais rapidamente, caso os novos hardwares fossem adquiridos, com processadores mais velozes e maior quantidade de memória RAM. É evidente que essa é uma solução, no entanto, dependendo da aplicação utilizada, ela não é a única e tampouco a mais vantajosa.

Com o surgimento da computação distribuída, a utilização de recursos de processamento, memória e armazenamento de computadores distintos aplicados na execução de um objetivo comum tornou-se possível, dando início a uma nova realidade que possui como principais benefícios a possibilidade real de variação do fator escalabilidade e uma maior viabilidade do ponto de vista econômico.

Atualmente existem várias aplicações fantásticas que fazem uso desse tipo de tecnologia de maneira extremamente eficiente. Uma delas é a Tecnologia de Computação em Cluster, que permite a utilização de hardware anteriormente considerado obsoleto, mas que quando utilizado dessa forma, ainda possui uma sobrevida considerável. Para entender mais sobre o surgimento dos clusters, precisamos conhecer um pouco da história e evolução dos (super) computadores e das redes de computadores.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

O objetivo da monografia é promover um estudo teórico e dirigido, bem como implementar uma visão geral sobre a tecnologia de Cluster. Esse trabalho é pautado no desenvolvimento de alguns conceitos de forma mais didática do que propriamente uma linguagem científica, pois é adotada uma postura de introduzir fatores importantes para sua compreensão e implementação na prática.

1.1.2 Objetivos Específicos

Um dos objetivos específicos desta pesquisa é apresentar, ao longo do estudo, a terminologia existente na área de computação em cluster. O leitor precisa dominar a terminologia para entender o texto.

Capacitar o profissional na compreensão das diversas aplicações da tecnologia de computação em cluster, compreendendo melhor os vários aspectos de sua utilização, a fim de identificar qual tipo de plataforma é aplicada para cada necessidade específica.

Introduzir o conhecimento necessário ao entendimento da programação paralela em sistemas em cluster. Entender que o processamento paralelo consiste em dividir uma tarefa em suas partes independentes e na execução de cada uma destas partes em diferentes processadores.

Demonstrar ferramentas para a administração dos serviços e o gerenciamento dos recursos de uma rede de computadores em cluster. Nesta pesquisa, são apresentadas algumas ferramentas que facilitam a instalação de sistemas operacionais e programas para o gerenciamento e monitoração de todo o sistema.

Criar um guia de construção, através de estudo de caso, para uma plataforma específica de cluster, considerando a importância de demonstrar e conduzir a aprendizagem de forma fácil e didática. O leitor vai poder instalar e configurar uma rede de computadores em cluster.

1.2 ORGANIZAÇÃO DO TRABALHO

Esta monografia está assim organizada.

O capítulo 2 apresenta o referencial teórico, onde é descrita uma visão geral da tecnologia de computação em cluster, ressaltando tanto a sua história e evolução, quanto os seus fundamentos. A importância de se ter uma noção ampla do assunto, favorece o entendimento contextual ao longo do trabalho.

O capítulo 3 apresenta as categorias básicas da tecnologia de cluster, bem como conceitos relevantes ainda desconhecidos para a maioria dos administradores de rede. Dentro das categorias básicas, é apresentada a funcionalidade de cada uma das configurações existentes. Quanto aos aspectos relevantes, são abordados conceitos específicos e característicos dessa tecnologia.

O capítulo 4 apresenta as várias formas e ferramentas utilizadas para o gerenciamento de um cluster. Instalar, configurar, monitorar e manter uma rede em cluster ainda são grandes

desafios a serem conquistados pelos profissionais de Tecnologia da Informação. O grande objetivo desse capítulo é auxiliar esse profissional na escolha das ferramentas certas para a necessidade específica.

O capítulo 5 apresenta uma metodologia aplicada para a construção de cluster e apresenta um projeto desenvolvido e implementado em nível de produção de uma plataforma clustering. Apresentar na prática o resultado de um estudo teórico foi o foco principal desse capítulo, assim como atender em nível de produção uma demanda computacional existente.

O capítulo 6 apresenta as conclusões da monografia e recomendações de trabalhos futuros.

2 REFERENCIAL TEÓRICO

Segundo Pitanga (2004), a tecnologia de computação em cluster é muitas vezes empregada para solucionar tanto o problema de necessidade de maior poder computacional quanto em outras áreas nas organizações. Conceitualmente, na totalidade das obras e artigos estudados, o cluster foi definido como um sistema composto de dois ou mais computadores (denominados nodos) que trabalham em conjunto, executando tarefas ou aplicações de maneira que os usuários que utilizam o sistema tenham a impressão de que ele é único (computador virtual). Esse conceito é conhecido como transparência do sistema.

As características fundamentais para o uso desta plataforma são: elevação da confiança, da distribuição de carga e de desempenho. Com base nessas características, identificam-se três tipos principais de cluster:

- Alta Disponibilidade – Consiste de um conjunto de duas ou mais máquinas com serviços “redundantes” e interligadas entre si. O motivo da existência de uma conexão exclusiva entre as máquinas redundantes é a freqüente monitoração dos serviços fornecidos pela máquina mestre realizada pelas máquinas escravas, através de mensagens do tipo “keep alive”, por exemplo. Se uma das máquinas escravas ou secundárias, detectar que o serviço provido pela máquina mestre não está operacional por algum motivo, esta assume o controle do serviço no lugar da máquina primária.
- Balanceamento de Carga – Esse tipo de cluster tem como finalidade distribuir requisições de serviço originadas de máquinas clientes entre um conjunto de servidores idênticos da melhor forma possível, considerando como métrica a disponibilidade de recursos destes. À medida que a demanda de acesso ao serviço aumenta, basta que o administrador adicione novos nós ao cluster de servidores.
- Alto Desempenho – A finalidade desse tipo de cluster é distribuir processos entre as máquinas que compõem o cluster. A métrica utilizada para a eleição da máquina ideal é a disponibilidade de recursos de memória e CPU de cada um dos nós no momento em que o processo deve ser exportado. Desse modo, máquinas ociosas que estejam executando processos considerados simples são fortes candidatas a receberem processos remotos de outro nós do cluster.

A versatilidade da utilização de um cluster de computadores aparece como uma forma alternativa, para suprir a demanda computacional, que até então somente supercomputadores

seriam capazes de resolver. Conhecer a tecnologia é, portanto, um desafio para os administradores de rede.

Conforme exposto por Pitanga (2003), o estudo do cluster deve ser feito através de uma linguagem fácil, didática no estilo “faça você mesmo”, onde devem ser demonstradas a escolha do hardware, a ligação dos computadores, a configuração e instalação do sistema operacional e demais configurações do Sistema.

Bookman (2003), em uma abordagem mais prática, fornece informações sobre clusters paralelos e do tipo Beowulf, e aprofunda-se em clusters de alta disponibilidade e equilíbrio de carga. Proporciona também instruções para o leitor desenvolver seus próprios aplicativos distribuídos, incorporando os melhores procedimentos e abordagem e oferecendo métodos testados passo a passo para levar um cluster Linux ao nível de produção.

Seguindo uma linha de pesquisa mais aprofundada, Pitanga (2003) apresenta conceitos avançados na construção de clusters com plataforma de computação de alto desempenho com GNU/Linux. Um de seus ensinamentos demonstra todos os passos necessários para implementação de um supercomputador de alta performance do tipo Beowulf e openMosix, um cluster israelense que se faz transparente tanto para aplicações seriais quanto paralelas.

Outro autor contemporâneo também foi fundamental na construção desse trabalho. Kopper (2005) elaborou um guia prático para construir e instalar cluster empresariais para aplicações críticas, usando hardware simples e software livre. Incluiu informação sobre como desenvolver um servidor extremamente potente, usando a ferramenta Heartbeat.

Ao longo da pesquisa notou-se que, como toda rede de computadores, o cluster precisa ter sua operacionalidade estável para poder atuar em conformidade com o desejado. Para isso, o gerenciamento de clusters envolve diversos fatores, desde a instalação do sistema operacional, até a definição de ferramentas para a configuração, manutenção, monitoramento e escalonamento de tarefas.

O desafio de exercer atividades especializadas em diferentes áreas de atuação é uma constante na atualidade. Estar sempre à mercê das solicitações dos clientes e empresários quanto ao suporte técnico, muitas vezes, leva o administrador de rede a necessitar de uma fonte de consulta objetiva e que promova um grau de conhecimento satisfatório para exercer adequadamente as atividades.

Diante desses fatos, pergunta-se qual seria a importância de realizar um estudo sobre a tecnologia de computação em cluster, ressaltando os principais conceitos e terminologia utilizada, bem como verificando na prática o seu funcionamento? A fim de responder essa pergunta, essa pesquisa buscou referenciais teóricos que abrangessem conteúdos com pontos

diversificados de abordagem, com a finalidade de elencar e explorar os principais itens julgados essenciais para o seu estudo.

2.1 HISTÓRIA E EVOLUÇÃO DOS (SUPER) COMPUTADORES

Existem vários registros históricos do surgimento de máquinas e dispositivos que eram utilizados para facilitar o cálculo e operações matemáticas. Esses registros datam de há aproximadamente 2.000 a.C., onde um aparelho muito simples, formado por uma placa de argila, auxiliava nos cálculos. Esse aparelho era chamado de **ÁBACO**. Na figura 1, pode-se observar que o ábaco ainda é muito usado em alguns países do mundo, por exemplo a China.



Figura 1: O Ábaco Hoje: Um caixeiro de loja em Beijing, China, usando o ábaco
Fonte: Wouda (1997)

Do ábaco, utilizado no antigo Egito e China, até os modernos supercomputadores, a evolução da informática precisou esperar a evolução de outras tecnologias. O desenvolvimento de ferramentas capazes de criar engrenagens e peças mecânicas foi necessário para a criação das primeiras máquinas de calcular.

O maior desenvolvimento tecnológico se deu a partir da Segunda Guerra Mundial, compreendido no período de 1939 a 1945, pois, houve a necessidade de se processar um grande volume de dados para a realização de cálculos balísticos, de decodificar mensagens, do país inimigo para a obtenção de informações importantes. A Marinha americana, em conjunto com a Universidade de Harvard, desenvolveu o computador Harvard Mark I, projetado pelo professor Howard Aiken, com base no calcular analítico de Babbage. A figura 2 mostra a imagem do Mark I, um dos grandes marcos da era da computação.

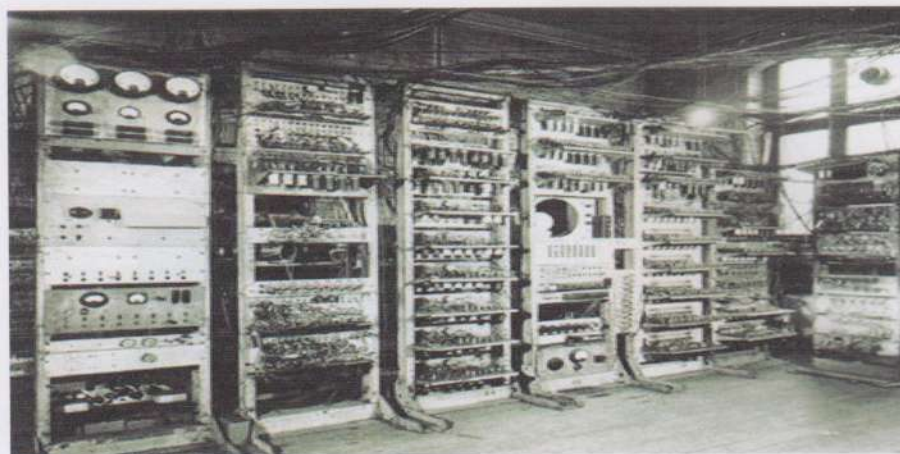


Figura 2: Mark I - 1943. Ocupava 120 m³
Fonte: Librandi (2005)

Simultaneamente e em segredo, o exército americano desenvolvia um projeto semelhante, chefiado pelos engenheiros J. Presper Eckert e John Maunchy, cujo resultado foi o primeiro computador a válvulas, o *Electronic Numeric Integrator And Calculator* (ENIAC), capaz de fazer quinhentas multiplicações por segundo. Apesar da alta velocidade para a época, era extremamente difícil mudar as instruções contidas dentro do computador, já que a programação era feita por meio de válvulas e fios que eram trocados de posição de acordo com o que se desejava.

No ENIAC, o programa era feito rearranjando a fiação em um painel. Nesse ponto John Von Neumann, um matemático húngaro, de origem judaica, naturalizado americano, propôs a idéia que transformou os calculadores eletrônicos em “cérebros eletrônicos”: modelar a arquitetura do computador segundo o sistema nervoso central. Para isso, eles teriam que ter três características:

- Codificar as instruções de forma possível de ser armazenada na memória do computador. Von Neumann sugeriu que fossem usados uns e zeros.
- Armazenar as instruções na memória, bem como toda e qualquer informação necessária a execução da tarefa.
- Quando processar o programa, busca as instruções diretamente na memória, ao invés de lerem um novo cartão perfurado a cada passo.

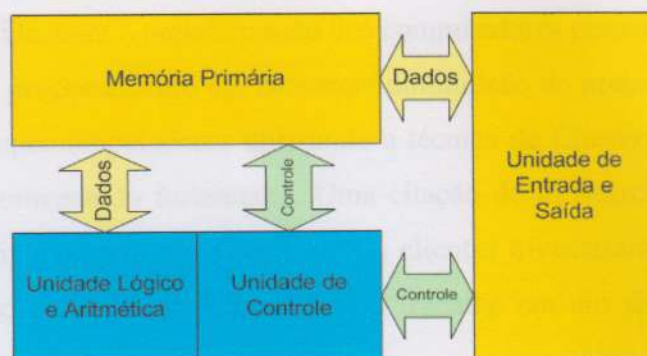


Figura 3: Modelo Von Neumann - Visão simplificada
Fonte: Geraldo (2006)

Este é o conceito de programa armazenado, cujas principais vantagens são: rapidez, versatilidade e automodificação. Assim, o computador programável que conhecemos hoje, onde o programa e os dados estão armazenados na memória ficou conhecido como Arquitetura de Von Neumann, conforme pode ser observado na figura 3. Assim estava criado o conceito do computador moderno.

Enquanto as operações eram mais simples, funcionava muito bem. A partir do momento em que usuários de computadores notaram que todo o trabalho computacional não poderia ser realizado em tempo hábil ou de forma confiável por um simples computador, surgiu a idéia de dividir o trabalho em processos separados: Execução de tarefas em paralelo. A figura 4 demonstra como se processa essa paralelização.

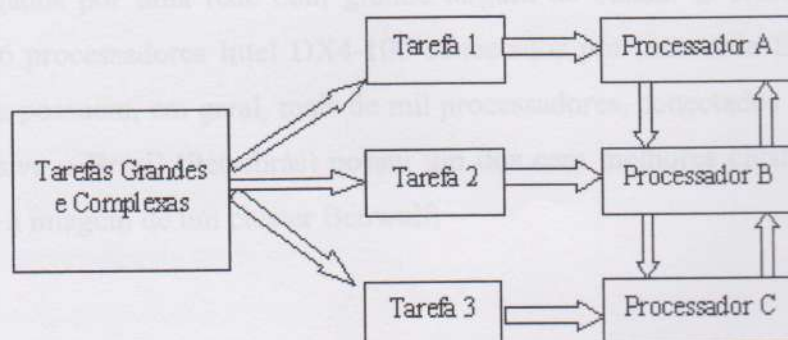


Figura 4: Paralelizando as tarefas
Fonte: Pitanga (2004)

Para Pitanga (2004), é impossível falar de supercomputadores de processamento paralelo sem citar o pioneirismo das máquinas fabricadas por Seymour Cray, que já usava esse conceito de processamento vetorial. Apesar da evolução desse tipo de tecnologia verificou-se que construir máquinas paralelas para uso genérico era coisa muito complicada, difícil e cara. Várias empresas especializadas na construção dessas máquinas foram à falência.

Em contrapartida, com a popularização dos computadores pessoais (PC's), do aumento da sua capacidade de processamento e a constante diminuição do preço destes equipamentos, a popularização de supercomputadores utilizando a técnica de *Clustering* (Agrupamento) de computadores vem aumentando fortemente. Uma citação de In Search of Clusters de Greg Pfister, ajuda a definir a origem dos Clusters: "Os clientes inventaram os cluster a partir do momento em que não conseguiam colocar todo o trabalho em um simples computador, ou precisavam de um backup.

Entretanto, a base formal do Cluster computacional como meio de executar tarefas em paralelo foi criada por Gene Amdahl da IBM, que em 1967 publicou o artigo "*Validity the Single Processor Approach to Achieving Largescale Computing Capabilities*". Neste, Amdahl apresenta a conhecida Lei de Amdahl que descreve matematicamente o ganho de velocidade resultante da paralelização de uma tarefa serial.

O primeiro cluster comercial foi o Attached Resource Computer NETWORK (ARCNET), porém a idéia de cluster só obteve sucesso comercial em 1984 quando a Digital Equipment Corporation (DEC) lançou o VAXcluster para o sistema operacional VAX/VMS. Com o invento do Parallel Virtual Machine (PVM) e com a popularização dos Personal Computer (PC's), foi possível construir supercomputadores a partir de PC's normais.

Em 1993, a NASA iniciou um projeto para a criação de supercomputadores a partir de PC's comuns. Um cluster desse tipo, chamado Beowulf, era formado por computadores comuns interligados por uma rede com grande largura de banda. O cluster da NASA era composto de 16 processadores Intel DX4-100 conectados por uma rede Ethernet. Hoje, os maiores clusters possuem, em geral, mais de mil processadores, conectados por redes Gigabit Ethernet, inclusive o Brasil (Petrobrás) possui um dos cem melhores clusters do mundo. A figura 5 mostra a imagem de um cluster Beowulf.



Figura 5: Cluster Beowulf
Fonte: Pitanga (2004)

2.2 FUNDAMENTOS DE CLUSTER

Sistemas de processamento paralelo vêm se tornando cada vez mais populares em função da demanda por processamento, exigido pelas diversas áreas da ciência. Uma forma de propiciar alto poder computacional é a utilização de Máquinas Paralelas, que utilizam mais de um processador para executar uma mesma tarefa e, assim, conseguir um desempenho melhor.

Os tipos mais comuns de Máquinas Paralelas são: supercomputadores e os clusters de computadores. Os Supercomputadores são máquinas desenvolvidas com tecnologias proprietárias e, geralmente, possuem finalidades pré-definidas que acabam gerando desvantagens (PITANGA, 2004), como:

- Utilização de softwares proprietários e caros;
- Alto custo de manutenção;
- Total dependência de fornecedores;
- Dificuldade de atualização.

O uso de Clusters tem sido uma saída para substituir os supercomputadores, pois podem oferecer desempenho semelhante e com um custo mais baixo. Algumas das vantagens dos clusters sobre os supercomputadores dedicados são (PITANGA, 2004):

- Escalabilidade: é possível aumentar o desempenho do mesmo adicionando ou trocando os microcomputadores que compõem o cluster;
- Tolerância à falhas: o cluster mantém o funcionamento mesmo com a paralisação de alguns nós.
- Baixo custo: utilizam recursos de fácil acesso e de uso comum;

- **Independência de fornecedores:** principalmente por utilizar microcomputadores comuns (inclusive de plataformas heterogêneas) e sistemas operacionais de livre distribuição, ou seja, não estão presos a uma tecnologia específica.

Com a adoção de um cluster, as empresas passaram a dispor de recursos computacionais equivalentes aos de um supercomputador de milhões de dólares com custos incomparavelmente menores.

Atualmente, diversas instituições científicas possuem clusters com um porte e poder computacional razoável, com número de computadores variando desde dois até milhares de nós. Todos sendo usados de forma muito diversificada. Estas aplicações incluem videoconferência, ambientes de trabalhos cooperativo, diagnósticos em medicina, base de dados paralelos utilizados para suporte a decisões e gráficos avançados em realidade virtual.

Seguem abaixo alguns exemplos de áreas onde há a utilização de cluster:

- **Servidores de Internet** – o grande crescimento de utilização de internet revelou fragilidades nos sites muito visitados. Um cluster pode distribuir a carga e aumentar a capacidade de resposta;
- **Segurança** – a grande capacidade que o processamento paralelo oferece beneficiará qualquer processo para identificação, quebra na segurança (criptografia) e verificação de possíveis soluções;
- **Base de Dados** – pesquisas intensivas (cujo tempo de resposta seja pequeno) em banco de dados podem demorar muito tempo em um sistema comum. A utilização de um cluster pode reduzir esse tempo significativamente;
- **Computação Gráfica** – é muito comum o tempo de processamento ser uma limitação para a evolução da qualidade dos projetos. A utilização de um cluster pode diminuir o tempo de renderização de imagens durante a elaboração de um filme, por exemplo;
- **Análise de elementos finitos** – cálculos de barragens, pontes, navios, aviões, grandes edifícios, veículos espaciais;
- **Aplicações em sensoriamento remoto** – análise de imagens de satélite para a obtenção de informações sobre a agricultura, florestas, geologia, etc;
- **Inteligência artificial e automação** – processamento de imagens, reconhecimento de padrões, visão por computador, reconhecimento de voz, etc;

- **Exploração Sísmica** – empregado especialmente pelas companhias petrolíferas (Petrobrás) para determinação de local de poços de petróleo;
- **Previsão do tempo** – um processo demorado e com pouca precisão quando gerado através de computadores seqüenciais;

Muitas dessas aplicações já fazem parte do dia-a-dia de trabalho de diferentes empresas no mercado, seja de pequeno, médio ou grande porte, de forma que o administrador de rede deve estar preparado para fornecer o suporte técnico necessário para manter a operação do cluster estável e conseqüentemente manter as aplicações “rodando”. Conhecer os principais componentes básicos de um cluster de computadores é um dos pontos mais importantes:

- **Nó ou node** – Um cluster pode ser composto de simples PCs, poderosas estações de trabalho SMPs e até mesmo supercomputadores vetoriais. Cada PC que faz parte do cluster chama-se nó do cluster;
- **Sistema Operacional** – Entre os sistemas operacionais mais adequados para clusters de computadores estão as versões do UNIX: Linux, FreeBSD ou NetBSD. O Windows 2003 Server, também pode administrar um cluster, porém o melhor desempenho é em versões do UNIX;
- **Rede Local** – O hardware de uma rede local tem total influência na performance do cluster, a placa de rede, o switch, o cabeamento. Segue como exemplos de rede local: Gigabit, Myrinet e SCI;
- **Protocolos** – Conjunto de regras e procedimentos que oferece meios de efetuar uma comunicação de dados rápida e confiável entre nós do cluster, sendo o protocolo TCP/IP o mais utilizado entre diversos clusters. Citamos também o Active e Fast Messages, para grandes plataformas de clusters com redes/switchs especiais;
- **Cluster Middleware** – É a camada responsável por oferecer uma ilusão de uma imagem de um sistema único (SSI) e de disponibilidade de uma coleção de computadores independentes, mas conectados;
- **Ferramentas de comunicação** – São as bibliotecas para passagem de mensagens e ferramentas, tais como compiladores, PVM e MPI;

Apesar das aplicações de cluster serem diversas, sempre apresentarão no todo ou em parte, em sua formação básica, esses componentes. Duas variáveis que influenciam para o aumento de performance e ganho no custo de um cluster é o uso de hardware genérico “de

prateleira” – COTS, combinado com o software livre. Isso tem influenciado e muito no uso dessa tecnologia.

O que podemos notar, de uma maneira geral, é que a crescente utilização de arquiteturas em clusters tem causado um forte impacto no mercado de supercomputadores. A verdade é que “A utilização de um agrupamento de máquinas como solução para computação de alto desempenho, alta disponibilidade e balanceamento de carga é uma realidade e não se pode virar os olhos quanto a isso”. (PITANGA, 2003, p.2.).

Essa forma, para atender a todas essas necessidades, a tecnologia de cluster possui configurações diferentes, tanto em relação ao hardware quanto à configuração do sistema.

Podemos dividir os clusters em três categorias básicas: Alta Disponibilidade (HA – High Availability), Balanceamento de Carga (LB – Load Balance) e Alto Desempenho de Computação (HPC – High Performance Computing).

3.1.1 Alta Disponibilidade

Para que se entenda o conceito de Alta Disponibilidade, é necessário perceber que a Alta Disponibilidade não é apenas um atributo de uma aplicação ou de um sistema, mas sim uma característica de um sistema computacional. Clusters de alta disponibilidade são projetados para garantir a continuidade de serviços prestados por um sistema computacional, replicando serviços e aplicativos, através de redundância de hardware e software.

Isto é feito através de nós redundantes que são instalados em cada um dos nós do cluster. Normalmente estes clusters possuem dois nós, que é o mínimo necessário para fornecer redundância. Entretanto existem clusters com um número de nós dependendo do nível de confiabilidade desejada. Encontramos casos que na configuração de aplicações que executam tarefas críticas, como por exemplo no desenvolvimento de sistemas de controle de voo, o objetivo principal dessa tecnologia é a alta performance.

A figura 6 ilustra a configuração típica de um sistema de alta disponibilidade dual-nó. Como se pode observar, não existe um único ponto único de falha, ou seja, não há um único ponto de falha (SPOF). A redundância tanto de servidores, quanto de equipamentos ativos, bem como de fontes de energia e unidades de armazenamento a base de RAID, são soluções que tornam esta arquitetura confiável.

3 TECNOLOGIA DE CLUSTERS

3.1 TIPOS DE CLUSTER

Quanto mais os computadores ganham espaço nas empresas, nos escritórios e na vida das pessoas em geral, mais se houve falar em “Cluster”. Por simples e bons motivos: nenhum usuário quer que sua máquina pare, de repente, de funcionar; ou que o serviço de alguma aplicação fique muito lento; ou ainda, que o processamento de algum cálculo científico tenha seu desempenho insatisfatório e fique processando horas e horas sem fim.

Dessa forma, para atender a todas essas necessidades, a tecnologia de cluster possui configurações diferentes, tanto em relação ao hardware quanto à configuração do sistema. Podemos dividir os clusters em três categorias básicas: Alta Disponibilidade (HA – *High Availability*), Balanceamento de Carga (LB – *Load Balance*) e Alto Desempenho de Computação (HPC – *High Performance Computing*).

3.1.1 Alta Disponibilidade

Para que se entenda o conceito de Alta Disponibilidade, faz-se necessário perceber que a Alta Disponibilidade não é apenas um produto ou uma aplicação que se instale, mas sim uma característica de um sistema computacional. Clusters de alta disponibilidade são construídos para garantir a continuidade de serviços prestados por um sistema computacional, replicando serviços e servidores, através de redundância de hardware e software.

Isto é feito através de nós redundantes que são usados em caso de falha de componentes. Comumente estes clusters possuem dois nós, que é o mínimo necessário para prover redundância. Entretanto existem clusters com até dezenas de nós, dependendo do nível de confiabilidade desejada. Encontramos esse tipo de configuração em aplicações que executam tarefas críticas, onde perdas no desempenho ou na capacidade de processamento são normalmente aceitáveis; o objetivo principal dessa tecnologia é não parar.

A figura 6 ilustra a configuração típica de um sistema de alta disponibilidade dual-node. Como se pode observar, não existe um único ponto nesta arquitetura que, ao falhar, implique a indisponibilidade de outro ponto qualquer (SPOF). A redundância tanto de servidores, quanto de equipamentos ativos, bem como de fontes de energia e unidades de armazenamento a base de RAID, são soluções que torna essa arquitetura confiável.

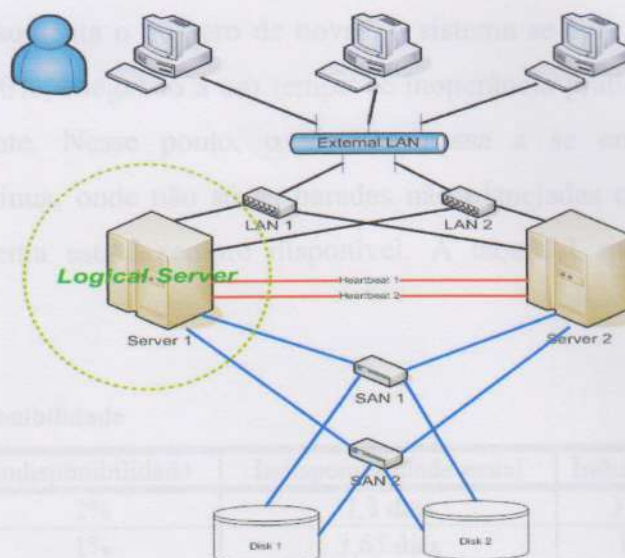


Figura 6: Cluster Alta Disponibilidade – Arquitetura clássica
Fonte: Sommer (2004)

Na alta disponibilidade, o SPOF deve ser eliminado. Para tanto, a alta disponibilidade pode ser alcançada de várias formas e em vários níveis. O espectro de soluções possíveis tem dois extremos: os casos em que apenas um hardware especializado é usado para criação de redundância, e os casos que são totalmente implementados por software com equipamento padrão, sem modificações. Embora a primeira ofereça maior segurança, mas é muito mais cara, o que tem levado esta última a uma maior popularidade.

Na área de software, em que o termo alta disponibilidade está realmente sendo utilizado, o objetivo é obter a continuidade de serviços sem o uso de um hardware especial e caro. Por serem mais baratas, as técnicas de software são muito mais comuns e mais amplamente utilizadas.

De modo geral, todo sistema possui uma disponibilidade característica. Dispositivos que não possuam nenhum mecanismo especial, por hardware ou por software, que vise de alguma forma mascarar possíveis falhas, possuem uma disponibilidade caracterizada como Disponibilidade Básica. Apesar de tratar-se de dados empíricos, são aceitos como senso comum na literatura da área como tendo disponibilidade variando de 99% até 99.9%.

Com a adição de mecanismos especializados em detectar, recuperar e mascarar falhas, pode-se aumentar a disponibilidade de um sistema de forma que ele se enquadre na classe de Alta Disponibilidade. Nesta classe, em geral, os dispositivos apresentam uma disponibilidade variando de 99,9% até 99,99999%.

Conforme se aumenta o número de noes, o sistema se aproxima cada vez mais da disponibilidade de 100%, chegando a um tempo de inoperância praticamente desprezível ou até mesmo inexistente. Nesse ponto, o sistema passa a se enquadrar na classe de Disponibilidade Contínua, onde não só as paradas não planejadas como as planejadas são mascaradas e o sistema estará sempre disponível. A tabela 1 mostra alguns níveis de disponibilidade.

Tabela 1: Níveis de Disponibilidade

Disponibilidade	Indisponibilidade	Indisponibilidade anual	Indisponibilidade semanal
98%	2%	7,3 dias	3 horas, 22 minutos
99%	1%	3,65 dias	1 hora, 41 minutos
99,8%	0,2%	17 horas, 30 minutos	20 minutos, 10 segundos
99,9%	0,1%	8 horas, 45 minutos	10 minutos, 5 segundos
99,99%	0,01%	52,5 minutos	1 minuto
99,999%	0,001%	5,25 minutos	6 segundos
99,9999%	0,0001%	31,5 segundos	0,6 segundos

Fonte: Simões (2003, P. 12)

Para se conseguir níveis elevados de disponibilidade, o uso de ferramentas específicas é necessário. Basicamente são baseadas nos seguintes conceitos: IP Takeover, Espelhamento local ou via rede e DNS Rotativo. Especificamente para Linux, existem alguns projetos em andamento, dos quais podemos citar: Heartbeat, Eddie, DRDB, Fake, Failover.

3.1.2 Balanceamento de Carga

Esses clusters operam tendo toda carga de trabalho chegando a um ou mais “*front-ends*” que distribuem a carga entre os demais nós (denominados “*back-end servers*”). Este tipo de cluster é utilizado para prover alta-performance, entretanto ele comumente possui recursos que adicionam características de clusters de alta-disponibilidade. Normalmente, este tipo de cluster é conhecido como “*Server Farm*”.

Essa tecnologia de balanceamento de carga entre servidores faz parte de uma solução abrangente, devido a uma explosiva e crescente utilização da rede e da internet. Basicamente ela provê um aumento na capacidade da rede, melhorando seu desempenho.

A figura 7 sugere a existência de um mecanismo responsável pelo balanceamento (director). Na verdade, ele pode existir sob várias formas, dependendo do(s) serviço(s) que se pretende balancear. Este director serve também de interface entre o cluster de servidores e os

clientes dos serviços - tudo o que os clientes conhecem é o endereço “semi-público” deste servidor. Quando efetuam uma requisição de serviço, o director direciona para um servidor que estiver disponível e com condições de processar a solicitação.

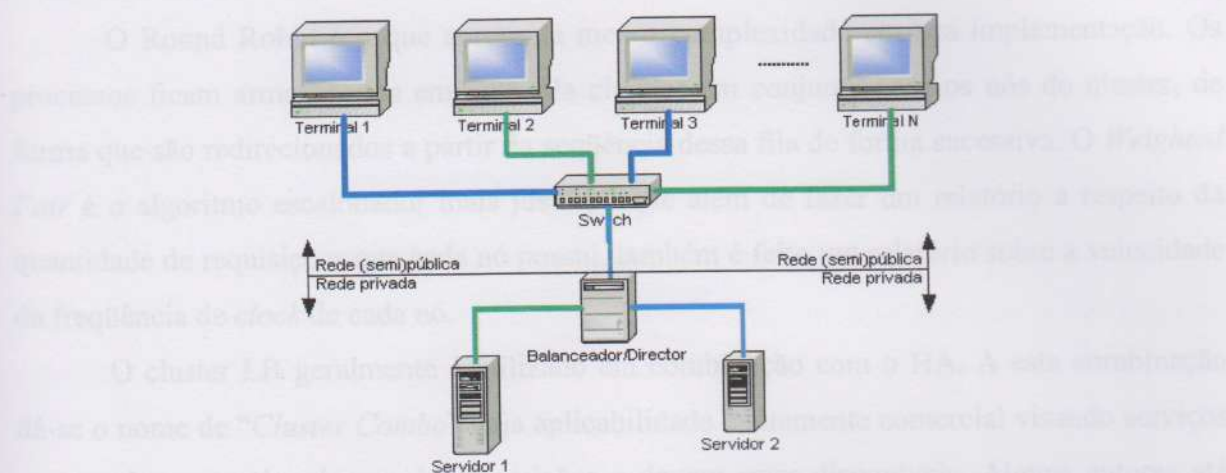


Figura 7: Cluster Balanceamento de Carga – Exemplo conceitual
 Fonte: Under-linux.org (Microtik, 2007)

Embora alguns pensem que o balanceamento de carga seja um simples redirecionamento do tráfego dos clientes para outros servidores, verificamos que essa concepção é errada. Para implementação correta, o equipamento que fará o balanceamento precisa ter algumas características específicas, tais como: verificação permanente de comunicação, checagem dos servidores e redundância. Todos esses itens são necessários para que suporte a escalabilidade do volume de tráfego das redes para que não venham a se tornar gargalo ou ponto único de falha (SPOF).

Devemos ainda salientar alguns pontos para uma implementação de um ambiente de sucesso com balanceamento de carga nos servidores:

- O algoritmo utilizado deve ser levado em consideração na necessidade de um cliente requisitar o endereço virtual. Todo o processo de escolha e resposta do servidor deve ocorrer de forma transparente e imperceptível para o usuário.
- Criar um método usado para checar se os servidores estão vivos e funcionando, vital para que a comunicação não seja redirecionada para um servidor que acabou de ter uma falha.
- Um método usado para se ter certeza de que um cliente acesse o mesmo servidor quando for o caso.

Para essa implementação, algoritmos de escalonamento de processos se fazem necessários. Os mais utilizados são: *Least Connection*, *Round Robin* e *Weighted Fair*. O primeiro realiza a distribuição de processos de acordo com o número de requisições que estão em domínio de cada nó do cluster. Então o nó que possuir o menor número de requisições será escolhido.

O Round Robin é o que apresenta menor complexidade em sua implementação. Os processos ficam armazenados em uma fila circular em conjunto com os nós do cluster, de forma que são redirecionados a partir da sequência dessa fila de forma sucessiva. O *Weighted Fair* é o algoritmo escalonador mais justo porque além de fazer um relatório a respeito da quantidade de requisições que cada nó possui, também é feito um relatório sobre a velocidade da frequência de *clock* de cada nó.

O cluster LB geralmente é utilizado em combinação com o HA. A esta combinação dá-se o nome de "*Cluster Combo*" cuja aplicabilidade é altamente comercial visando serviços que recebam grande número de requisições e devam estar disponíveis. Alguns autores até definem essa tecnologia como sendo uma nova especificação de tipo de tecnologia de cluster. Como exemplo clássico para balanceamento de carga tem o projeto: Linux Virtual Server, Piranha, Turbo Cluster, dentre outros.

3.1.3 Alto Desempenho

Clusters de alto desempenho são construídos com objetivo de aumentar a velocidade de tarefas dividindo-as entre inúmeros nós. Este tipo de cluster é mais comumente utilizado para aplicações científicas. Muitas aplicações que rodam em HPC foram desenvolvidas especialmente para tirar proveito do paralelismo oferecido pelo cluster.

Cientificamente, os clusters de computação de alto desempenho têm se sobressaído devido a sua grande capacidade de processamento. O tipo de cluster em questão é uma forma de paralelismo de alto nível do processador, sendo capaz de dividir grandes e complexas tarefas em simples e pequenas, diminuindo, de forma significativa o tempo de processamento, aumentando a quantidade de tarefas a serem processadas pelo agrupamento.

Quando se fala em processamento de alto desempenho, que pode ser traduzido em processamento paralelo e processamento distribuído, muitas pessoas pensam em grandes máquinas dedicadas (supercomputadores), que custam milhões de dólares, difíceis de serem operados e com salas superprotegidas. Mas hoje em dia, devido aos clusters, os custos foram

reduzidos e existem máquinas muito rápidas, o que viabiliza o uso do processamento de alto desempenho na solução de problemas em diversas áreas.

O cluster HPC é um tipo de sistema para processamento paralelo ou distribuído que consiste de uma coleção de computadores interconectados, trabalhando juntos como um recurso de computação simples e integrado. O Beowulf é uma das mais populares implementações de HPC. É um modelo criado para obter um sistema de alta performance com baixo custo. Roda em Sistema Operacional Linux e utiliza software livre para explorar paralelismo.

O cluster Beowulf é dividido em um nó controlador denominado front-end, também conhecido como nó mestre, cuja função é controlar o cluster, monitorando e distribuindo as tarefas, em alguns casos atuando como servidor de arquivos e executando o elo entre os usuários e o cluster. Os demais nós são conhecidos como clientes ou back-ends, ou simplesmente nós escravos, e são exclusivamente dedicados para processamento das tarefas enviadas pelo nó controlador.

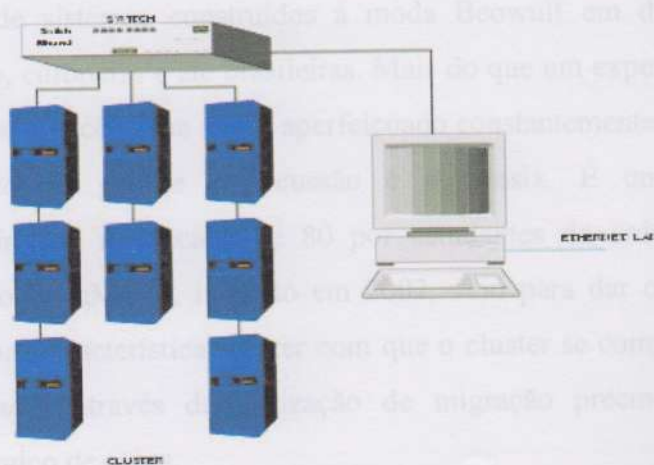


Figura 8: Cluster Beowulf – Topologia básica.
Fonte: Cluster DSEE (2007)

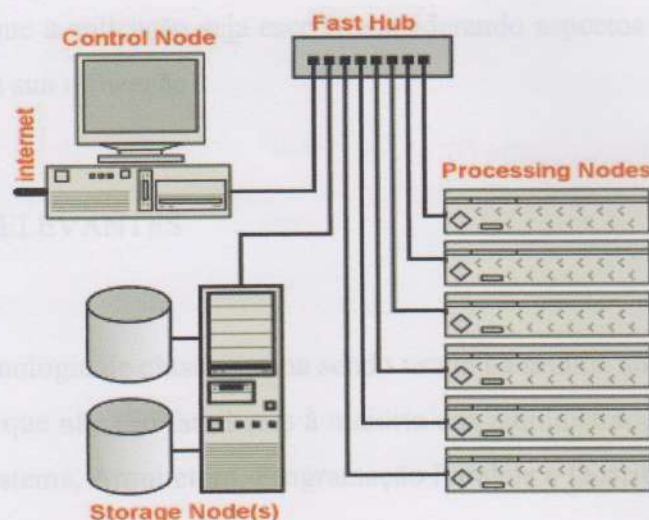


Figura 9: Cluster Beowulf – associado à solução de storage.
Fonte: Growing Up (2007)

O Beowulf é um projeto bem sucedido. A opção feita por seus criadores de usar hardware popular e software aberto tornou-o fácil de se replicar e modificar, a prova disso é a grande quantidade de sistemas construídos à moda Beowulf em diversas universidades, empresas americanas, européias e até brasileiras. Mais do que um experimento, foi obtido um sistema de uso prático que continua sendo aperfeiçoado constantemente.

Outro projeto de grande repercussão é o Mosix. É um sistema distribuído desenvolvido inicialmente na década de 80 por estudantes da universidade de Hebrew, Jerusalém. O projeto openMosix, iniciado em 2002, veio para dar continuidade ao Mosix original. Sua principal característica é fazer com que o cluster se comporte como um grande único supercomputador através da utilização de migração preemptiva de processos e balanceamento dinâmico de carga.

A implementação da migração preemptiva faz com que um processo em um nó escasso de recursos seja transportado para outro nó que tenha maior disponibilidade, melhorando, assim, o desempenho do sistema. Uma das qualidades do openMosix é ser transparente para o usuário. Quando um aplicativo é iniciado, o openMosix envia os processos para um melhor computador da rede e monitora sua execução. Dessa forma, ele escolhe qual o melhor computador que poderá executar tal processo em um determinado momento. Assim, o usuário não vê nada, o cluster faz tudo transparentemente.

O openMosix diferencia-se de outros tipos de cluster, por não exigir mudanças no código fonte das aplicações nele utilizadas. Outros tipos de cluster populares, como o Beowulf, por exemplo, utilizam bibliotecas específicas (PVM e MPI) que realizam essa

tarefa, mas exigem que a aplicação seja escrita considerando aspectos de paralelismo, o que, de certo modo, limita sua utilização.

3.2 ASPECTOS RELEVANTES

Embora a tecnologia de cluster venha sendo usada há muitos anos em muitos sistemas, há alguns conceitos, que não são familiares à maioria dos administradores de rede e usuários. Imagem Única do Sistema, Arquitetura, Programação Paralela e Distribuída são alguns desses conceitos.

3.2.1 Imagem Única do Sistema - SSI

Quando se utiliza um cluster de computadores, uma questão importante que deve ser considerada é a de que o usuário do sistema computacional não tenha qualquer preocupação em saber como estão organizados os componentes que compõem e disponibilizam o serviço. Para conseguir essa ilusão deve-se empregar uma técnica conhecida como Imagem Única do Sistema ou SSI.

Os clusters, atualmente, disponibilizam um serviço limitado de imagem única. No futuro, aumentarão a escalabilidade e diversas melhorias serão acrescentadas no que diz respeito a todos os conceitos de SSI. O SSI é considerado uma camada *middleware*, que reside entre o sistema operacional e nível usuário.

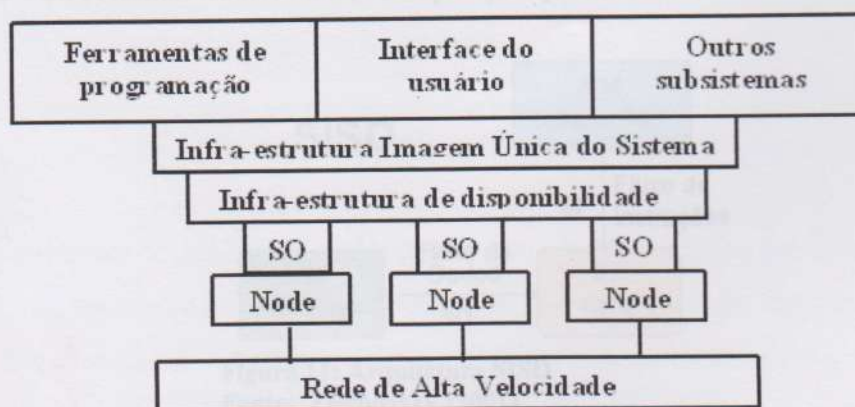


Figura 10: Uma visão do middleware
Fonte: Pitanga (2004)

Como pode ser observado na figura 10, o *middleware* consiste essencialmente em duas subcamadas de infra-estrutura de software: são elas: a infra-estrutura de imagem única (SSI) e a infra-estrutura de disponibilidade. A primeira está presente em todos os sistemas operacionais para oferecer uma unificação dos recursos. Já a segunda disponibiliza serviços de cluster como checkpoint, *failover* automático, recuperação e tolerância à falhas para todos os nós do cluster. Seguem abaixo, alguns benefícios que a propriedade SSI fornece na sua correta implementação em um cluster:

- Os serviços podem ser solicitados a partir de qualquer nó;
- A localização dos dispositivos físicos pode ser totalmente transparente ao usuário, tais como o armazenamento de dados;
- Alta disponibilidade em seus serviços;
- Quanto à concorrência, os usuários não devem notar que existem outras pessoas no sistema.

3.2.2 Arquitetura do Hardware

A arquitetura de um hardware é a base para se conhecer o funcionamento de uma tecnologia. Michel Flynn, em 1972, caracterizou diversos modelos de arquiteturas de computadores. Basicamente esses modelos se dividiam em quatro classes de computadores, são elas:

- Arquitetura SISD: É a identificação mais simples, onde o equipamento é considerado sequencial, pois só executa uma instrução por vez. São as máquinas de Von Neuman tradicionais, sem qualquer paralelismo.

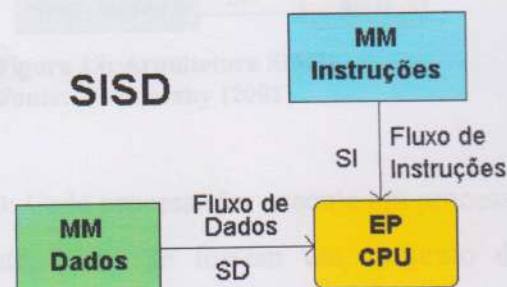


Figura 11: Arquitetura SISD
Fonte: Zelenovsky (2001)

- Arquitetura MISD: Este tipo de classificação não possui representante e até mesmo o Flynn duvidou que algum dia isso possa existir. Seriam máquinas que executam várias instruções ao mesmo tempo sobre um único dado.

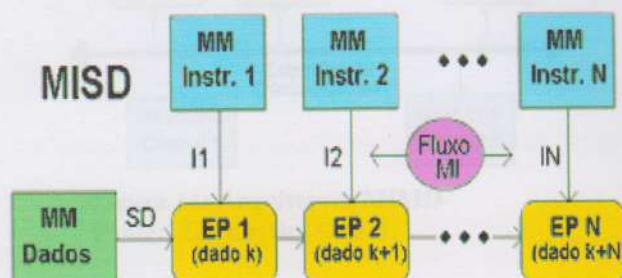


Figura 12: Arquitetura MISD
Fonte: Zelenovsky (2001)

- Arquitetura SIMD: Essa arquitetura é o equivalente ao paralelismo de dados, onde uma simples instrução é executada paralelamente utilizando vários dados de forma síncrona. Esse contexto se encaixa dentro dos processadores vetoriais Cray 1 e CM-2, por exemplo. Nessa característica o hardware é muito mais específico.

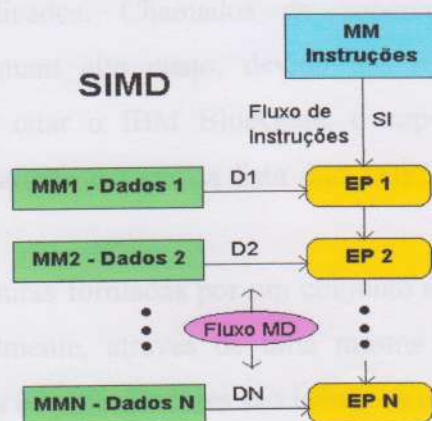


Figura 13: Arquitetura SIMD
Fonte: Zelenovsky (2001)

- Arquitetura MIMD: Cada processador executa um processo em paralelo, agindo de forma independente, como se fossem um conjunto de máquinas SISD. São exemplos dessa categoria servidores com múltiplos processadores, sistemas MPP e os clusters de computadores.



Figura 14: Arquitetura MIMD

Fonte: Zelenovsky (2001)

Embora a classificação de Flynn seja conveniente numa primeira instância, é bastante imprecisa para classificar as variedades de multiprocessadores existentes. Como se pode notar, a categoria MIMD engloba grande variedade de máquinas, de forma específica as de processamento paralelo.

Considerando a capacidade de execução de aplicações paralelas, podemos classificar as arquiteturas da seguinte forma:

- **MPP:** Esse tipo de arquitetura possui processadores altamente poderosos e links de comunicação dedicados. Chamados de supercomputadores ou arquiteturas dedicadas, apresentam alto custo, devido aos recursos que oferecem. Como exemplo, pode-se citar o IBM BlueGene, o supercomputador mais rápido na atualidade, que compõe o topo da lista dos TOP500, com a capacidade de 280 Tflops.
- **SMP:** São arquiteturas formadas por um conjunto de processadores iguais, que se comunicam, geralmente, através de uma mesma memória. O termo simétrico significa que todos os processadores são idênticos em termos de arquitetura interna e poder de processamento. O Xeon ou Opteron são exemplos mais comuns dessa arquitetura.
- **NOW:** Correspondem a um conjunto de estações de trabalho interligadas através de uma rede local (LAN), podendo ser usadas como plataforma de execução de aplicações distribuídas. O fundamento da comunicação entre as estações é o uso de troca de mensagens. Estações Sun interligadas por rede Ethernet são exemplos desse tipo de arquitetura.

Nesse contexto, um cluster pode ser caracterizado como uma plataforma alternativa, aliando o poder e a velocidade de processamento das arquiteturas dedicadas (MPPs) com a disponibilidade de recursos (hardware e softwares baratos) das redes de estações.

Considerando o tipo de sistema, a arquitetura do cluster pode apresentar as seguintes características:

- **Arquiteturas homogêneas:** Os nodos que compõem esse cluster possuem a mesma arquitetura e sistema operacional. Estão se tornando um padrão na área de alto desempenho por serem mais simples de operar.
- **Arquiteturas heterogêneas:** Para esse caso, o cluster possui processadores diferentes e, possivelmente, diferentes sistemas operacionais.

Considerando o tipo de nodo utilizado no cluster, podemos dividir em dois tipos:

- **Arquitetura simétrica:** Os nodos são todos homogêneos, possuindo a mesma velocidade e capacidade de processamento, além de possuir a mesma quantidade de recurso computacional. A análise de desempenho é mais realista com esse tipo de cluster.
- **Arquitetura assimétrica:** Possuem nodos diferentes. Podem até possuir nodos homogêneos, mas com diferentes velocidades e capacidades de processamento. Esse tipo de arquitetura dificulta a análise de desempenho.

3.2.3 Tecnologia de Redes para Cluster

A grande vantagem evidenciada pelos clusters é o seu baixo custo, aliado a grandes avanços na tecnologia empregada nas redes de interconexão. Isso os torna uma opção muito competitiva para o mercado, principalmente de pequenas e médias empresas. Para garantir a qualidade da transmissão nas redes de interconexão, algumas métricas foram definidas, relacionadas à transferência de dados. São elas:

- **Packing time (tempo de empacotamento):** É o tempo necessário para disponibilizar os dados para o envio. Incluem codificação e armazenamento.
- **Latência:** É considerado o tempo que decorre desde o envio da mensagem até a sua disponibilização no lado do receptor.
- **Bandwidth (Largura de banda):** Indica a maior taxa de transferência de rede, normalmente medida em Mbytes/second.

Considerando essas métricas de análise, vários padrões de interconexão estão sendo utilizados para conectar os nós dessas máquinas. Fast Ethernet, Gigabit, Myrinet e SCI são alguns exemplos. A seguir veremos as características de cada tecnologia:

- Fast Ethernet: Atualmente é a tecnologia de rede mais barata e permite largura de banda de 100 Mbps. É uma rede não determinística, baseada no protocolo CSMA/CD. Foi inicialmente proposta e projetada para oferecer uma evolução tranqüila a já saturada tecnologia Ethernet. Esta sendo substituída gradativamente pela Gigabit.
- Gigabit: Surgiu da necessidade criada pelo aumento de largura de bandas nas “pontas” das redes, juntamente com a demanda das aplicações atuais. Esse padrão fornece uma largura de banda mínima de 1 Gbps. Essa tecnologia é uma tendência de liderança no campo dos clusters, pois apresenta confiabilidade, escalabilidade, ferramentas de gerenciamento (snmp) e custo relativamente acessível.
- Myrinet: Foi originalmente projetada para ser utilizada em sistemas multicomputadores (MPPs e NOWs). Atualmente a Myrinet vem sendo usada em máquinas baseadas em clusters. Fundamenta-se na comunicação por troca de mensagens, com tecnologia diferenciada, mas que provê baixa latência, protocolo próprio de alta velocidade, grande largura de banda. Ainda é uma tecnologia cara.
- SCI: Essa tecnologia favorece o uso de programação com memória compartilhada ao invés de troca de mensagens. Um cluster utilizando a tecnologia SCI torna-se um sistema de coerência global compartilhada, semelhante às máquinas SMP. Além do desempenho, a possibilidade de programação com memória compartilhada gera um enorme atrativo.

Como o uso dos clusters de computadores cresce a cada dia, a rede se tornou um ponto fundamental em pesquisas, onde se busca o melhor método de conexão entre os nós e reduzir os gargalos de comunicação, através de maiores larguras de banda e protocolos de baixa latência.

3.2.4 Programação Paralela e Distribuída

A programação paralela era visto como uma sub-área da computação. Porém, nos últimos anos, o paralelismo tem se tornado o tema central na programação. Conceitualmente, o processamento paralelo consiste em dividir uma tarefa em suas partes independentes e na execução de cada uma destas partes simultaneamente.

As aplicações podem ser executadas paralelamente em um mesmo processador, em uma máquina multiprocessada ou em um grupo de máquinas interligadas que se comportam como uma só máquina (clusters). A programação paralela é muitas vezes confundida com a programação distribuída. Esta última consiste em executar aplicações cooperantes em máquinas diferentes, podendo ser interligadas por intranet ou internet. Diferenciam-se de acordo com os seguintes itens:

- **Acoplamento:** Sistemas paralelos são fortemente acoplados. Normalmente compartilham hardware ou se comunicam através de um barramento de alta velocidade. Sistemas distribuídos são fracamente acoplados.
- **Previsibilidade:** O comportamento de sistemas paralelos é mais previsível. O de sistemas distribuídos é mais imprevisível.
- **Influência do tempo:** Os sistemas distribuídos são bastante influenciados pelo tempo de comunicação pela rede. Em sistemas paralelos, o tempo de troca de mensagens pode ser desconsiderado.
- **Controle:** Em geral, em sistemas paralelos se tem o controle de todos os recursos computacionais; já os sistemas distribuídos tendem a empregar recursos de terceiros.

Além disso, a diferença entre programação paralela e distribuída está no objetivo que se tem quando se usa a aplicação ou sistema. Embora possam ser semelhantes, segundo Pitanga (2004, P.21), “o programa paralelo é definido como uma coleção de elementos de processamento que cooperam e comunicam para resolver problemas de uma maneira bem mais rápida”. Portanto, poderíamos dizer que os sistemas paralelos são um tipo específico de sistema distribuído.

Ambientes de memória compartilhada, ou seja, ambientes com a mesma memória para vários processadores, utilizam um paralelismo explícito, já que se podem valer desta memória para ativar os processos paralelos. Já os Ambientes de memória distribuída, onde cada processador possui sua própria memória, devem utilizar um modelo chamado troca de mensagens para comunicação e sincronização das tarefas. O ambiente de *clustering* é um deles. Há dois principais padrões para troca de mensagem em um ambiente de cluster:

- **PVM:** é o conjunto integrado de bibliotecas e de ferramentas de software, cuja finalidade é emular um sistema computacional concorrente, heterogêneo, flexível e de propósito geral. Este é um pacote de software que permite que uma rede heterogênea de computadores de todos os tipos seja programada como se fosse

apenas uma única "Máquina Paralela Virtual". A programação é baseada no paradigma de troca de mensagens.

- MPI: é uma biblioteca de subrotinas de comunicação, desenvolvidas em linguagem C e que são utilizadas no desenvolvimento de programas para serem executados em mais de um processador, simultaneamente (Programas Paralelos). Usa o paradigma de programação paralela por troca de mensagens e pode ser usado em clusters ou em redes de estações de trabalho.

Muitas são as razões para se adotar o MPI ao invés de PVM. Vejamos algumas delas:

- O MPI possui diversas implementações;
- Gerencia eficientemente os buffers das mensagens;
- Pode ser usado de forma eficiente tanto para arquiteturas MPP quanto em sistemas em cluster;
- Possui portabilidade real;

Muitos são os artigos e obras relacionadas ao estudo dessas bibliotecas para desenvolvimento de programação paralela. É de extrema importância que o programador conheça essas "linguagens" para poder aproveitar o máximo o poder dessa tecnologia.

3.2.5 Sistema de Arquivos Distribuídos

O sistema de arquivos é uma parte importantíssima dos sistemas operacionais, pois ele fornece uma visão abstrata dos dados persistentes, além de ser responsável pelo acesso a arquivos e de sua organização geral. Um Sistema de Arquivos Distribuído (SAD) tem o objetivo de fornecer os mesmos serviços e recursos de um sistema de arquivos convencional, com a diferença pode ser acessado de qualquer máquina dentro de uma rede (acesso remoto).

Em geral, existem sistemas de arquivos distribuídos para todos os tipos de aplicações e problemas que se deseje resolver. Alguns mais tradicionais outros mais inovadores e criados para resolver problemas específicos. Juntamente com os clusters, os sistemas de arquivos distribuídos estão se tornando mais comuns. Quando aliamos as duas áreas, temos um ramo do estudo denominado *Cluster Filesystem*.

A tecnologia de *Cluster Filesystem* é complementar aos sistemas de arquivos distribuídos. Na verdade, qualquer nó do cluster pode exportar o *filesystem*, que pode ser visto pela rede usando algum protocolo que assim o possibilite. Mas qual o sistema de arquivos usar num sistema distribuído? Para resolver esse tipo de dúvida, costuma-se levantar qual o tipo de aplicação que rodará nesse sistema, e a partir disso descobrir o que é mais importante,

como tolerância à falhas, acesso concorrente, etc. Dentre as principais tecnologias de sistemas de arquivos distribuídos usados em cluster, temos:

- AFS: Projeto que começou na Universidade de Carnegie Mellon em 1983, com suporte provido pela IBM. Seu objetivo era projetar e implementar um sistema de arquivos distribuído para o ambiente acadêmico que permitiria compartilhar uma estrutura comum de diretórios entre milhares de clientes. A particularidade dessa tecnologia é que quando um arquivo do AFS é aberto, todo o arquivo é levado para o cliente, passando grande parte do trabalho dos servidores de arquivos aos seus clientes. Somente quando o arquivo é fechado, desde que modificado, é que retorna para o servidor.
- NFS: Sistema de arquivos distribuído desenvolvido inicialmente pela Sun e um dos mais usados em sistemas Unix. O objetivo principal do sistema NFS é permitir que um conjunto de estações de trabalho compartilhem um sistema de arquivos de uma forma comum e transparente. Normalmente ele é utilizado quando existem aplicações que não necessitam de uso intensivo de disco, pois as transações entre cliente e servidor consomem muitos recursos da rede. Uma excelente característica sua é a transparência do sistema de arquivos para o usuário final.
- PVFS: É um sistema de arquivos distribuído desenvolvido para prover alto desempenho e escalabilidade paralela para clusters de PCs linux. Tanto aplicações seriais quanto paralelas podem utilizar esse tipo de tecnologia para armazenamento e recuperação de arquivos de dados, que estão distribuídos por diversos servidores de I/O. O PVFS é um sistema de arquivos distribuído e paralelo que se preocupa em diminuir o gargalo provocado pelo tráfego de dados. Em comparações com o NFS, o PVFS ganhou em acesso a arquivos grandes, e ficou pouco atrás no acesso a arquivos pequenos.
- GFS: Sistema de arquivos implementado no sistema operacional Silicon Graphics IRIX, que pode ser acessado usando-se comandos e utilitários comuns do UNIX. A idéia principal é compartilhar fisicamente o sistema de armazenamento de dados. O objetivo do GFS é prover velocidade de banda e altíssima capacidade de armazenamento para aplicações multimídia, científica e visualização. Em geral o GFS é uma boa opção para aplicações que tratam de arquivos muito grandes (aplicações multimídia, por exemplo). Porém, é necessário usar hardware especial (que é caro) para que seu funcionamento seja conforme o esperado.

Existem muitos outros sistemas de arquivos distribuídos, mas todos basicamente provêm algumas características fundamentais em sua implementação, tais como: transparência quanto à localização, replicação e autenticação.

4.1 INSTALAÇÃO E CONFIGURAÇÃO

De um modo geral, clusters são formados por um número relativamente alto de computadores. A instalação e a configuração individual de cada máquina podem levar muito tempo. Porém, como as máquinas normalmente são formadas pelo mesmo sistema operacional, é possível se fazer uso de processos automáticos de instalação. A seguir serão apresentadas algumas delas, procurando descrever suas características.

- **Kickstart** é um sistema desenvolvido para RedHat Linux, e distribuidoras baseadas em sua estrutura (CentOS e WhiteBox, por exemplo). Permite colocar todas as opções que o usuário fará na instalação manual, como seleção da linguagem, partições, pacotes a serem instalados, etc, em um arquivo de configuração, eliminando toda interação com o usuário. Podem ser utilizados os métodos de instalação via NFS, HTTP ou FTP. No site: <http://www.redhat.com/docs/manuals/guides/RHEL-5-Manual/creating-guides>, encontrando um guia de utilização do tipo tutorial.
- **FAI** é um conjunto de scripts e arquivos de configuração para instalação automatizada de sistema Debian Linux em um agrupado com um grande número de nós. FAI é um método escalável, onde cada nó realiza a sua própria instalação a partir de um arquivo de configuração de um servidor. Para tanto, um nó cliente cria um sistema temporário, via rede ou disquete, que começa a instalação propriamente dita. Maiores detalhes sobre esta ferramenta podem ser encontradas no site: <http://www.informatica.ufrj.br/~fai/fai-guide.html>.
- **G4U** é um programa relativamente simples, baseado no sistema NetBSD. Automaticamente faz uso da utilização de um disquete ou CDROM para efetuar um

4 GERENCIAMENTO DE CLUSTERS

Segundo Charles Bookman (2003), os clusters exigem uma manutenção e um olhar atento, por parte do administrador, para funcionarem sempre em perfeitas condições de operação. De fato, a configuração e a manutenção deste tipo de arquitetura envolvem diversos fatores, sendo simplificada pela existência de ferramentas específicas para cada problema. Algumas ferramentas facilitam a instalação de sistemas operacionais e outras o gerenciamento e monitoração de todo o sistema. Esse estudo não esgotará plenamente os conhecimentos referentes ao assunto, mas que se justifica, pois servirá de guia para pesquisas mais detalhadas por parte do administrador.

4.1 INSTALAÇÃO E CONFIGURAÇÃO

De um modo geral, clusters são formados por um número relativamente alto de computadores. A instalação e a configuração individual de cada máquina podem levar muito tempo. Porém, como as máquinas normalmente são formadas pelo mesmo sistema operacional, é possível se fazer uso de processos automáticos de instalação. A seguir serão apresentadas algumas delas, procurando descrever suas características.

- **Kickstart:** é um sistema desenvolvido para RedHat Linux, e distribuições baseadas em sua estrutura (CentOS e WhiteBox, por exemplo). Permite colocar todas as seleções que o usuário faria na instalação manual, como seleção da linguagem, partições, pacotes a serem instalados, etc, em um arquivo de configuração, eliminando toda interação com o usuário. Podem ser utilizados os métodos de instalação via NFS, HTTP ou FTP. No site: <http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/sysadmin-guide/>, encontramos um guia de utilização do tipo tutorial.
- **FAI:** é um conjunto de scripts e arquivos de configuração para instalação automatizada de sistema Debian Linux em um agregado com um grande número de nós. FAI é um método escalável, onde cada nó realiza a sua própria instalação a partir de um arquivo de configuração de um servidor. Para tanto, um nó cliente carrega um sistema temporário, via rede ou disquete, que começa a instalação propriamente dita. Maiores detalhes sobre essa ferramenta podem ser encontradas no site: <http://www.informatik.uni-koeln.de/fai/fai-guide.html/>.
- **G4U:** é um programa relativamente simples, baseado no sistema NetBSD. Basicamente funciona na utilização de um disquete ou CDROM para efetuar um

boot, a partir daí disponibiliza duas funções principais. A primeira é enviar para um servidor FTP remoto a imagem comprimida do(s) disco(s) da máquina. A outra recupera a imagem que está no servidor, efetuando descompressão e escrevendo no disco local da máquina. A configuração da rede pode ser via DHCP. De modo particular, esse é um dos métodos mais utilizados em nossas práticas diárias, na Petrobrás, não só para realização de instalações nos nós dos clusters, mas pela praticidade do processo, utilizamos para recuperar imagem de máquinas em geral. Possui uma “desvantagem”: o nó fica inoperante no momento da transferência, diferentemente do que ocorre com o SystemImager. O site oficial do desenvolvedor é <http://www.feyrer.de/g4u/>.

- **Replicator**: outro recurso desenvolvido para sistemas Debian Linux, funcionando como um duplicador de instalação. Após a configuração inicial, os scripts criarão um bootdisk que reservam (re) instalam completamente o Debian. O pacote de instalação pode ser encontrada no site <http://sourceforge.net/projects/replicator/>.
- **ALICE**: é um sistema para SuSE Linux que permite instalar e configurar várias máquinas automaticamente com o mínimo possível de interação humana. Baseado em interfaces como syslinuxrc, YaST e suseconfig, além de instalar o sistema operacional, ALICE também pode criar grupos e usuários, ativar serviços, etc. O projeto pode ser verificado na página <http://www.suse.de/~fabian/alice/index.html>.
- **OSCAR**: é um ambiente para a instalação, configuração e gerenciamento de clusters. OSCAR apresenta, de forma integrada, os recursos considerados mais utilizados em cluster, disponibilizando a configuração automática de componentes, bem como a instalação eficiente do ambiente básico como sistema operacional e ferramentas de administração e operação. A página oficial do projeto é <http://oscar.openclustergroup.org/>.
- **SystemImager**: é um conjunto de scripts que simplificam os procedimentos para preparação da carga remota para instalação. Com ele, também é possível atualizar as imagens já distribuídas para os nós. As atualizações são rápidas porque somente as partes modificadas são mandadas ao cliente. O SystemImager também permite o armazenamento de várias imagens em um servidor, podendo estas serem associadas à nós específicos. O diferencial dessa ferramenta é que no momento da atualização, a estação estará funcionando normalmente, atualizando somente os

arquivos que modificaram no último momento. O site da ferramenta é <http://sourceforge.net/projects/systemimager/>.

- **Rembo Toolkit:** um inicializador remoto desenvolvido a partir da ferramenta BpBatch que utiliza o protocolo PXE e permite a execução de várias ações em tempo de boot, antes do sistema operacional ser iniciado. Assim, é possível desde particionar o disco rígido, como também criar uma imagem de um disco rígido clonando a estrutura de partições. Podem ser encontrados dados e aplicações para o Rembo Toolkit no site http://rembowiz.sourceforge.net/sysadm/install_in.htm.
- **Ka-deploy:** é uma ferramenta que faz parte do Ka Clustering Tools, que permite replicar uma máquina Linux muitas vezes ao mesmo tempo. A carga remota em clusters cria uma corrente de dados entre os nós, cada nó copia os dados para seu disco local e envia para o resto da corrente. No site http://sourceforge.net/project/showfiles.php?group_id=26218, encontramos maiores informações sobre a ferramenta.
- **SCMS:** é um sistema desenvolvido pela Universidade Kasetsart (Tailândia) que possui recursos úteis para configuração e atualização remota e permite a execução da mesma tarefa em vários nós ao mesmo tempo, bem como a instalação de pacotes RPMs em paralelo. Além disso, este aplicativo é composto de uma série de ferramentas e sistemas de suporte que auxilia o dia a dia de um administrador de supercomputadores baseados em cluster. Segundo Pitanga (2004), é uma das melhores ferramentas gráficas para gerenciamento de um cluster do tipo Beowulf. Esta ferramenta pode ser baixada do site <http://www.opensce.org>.
- **xCAT:** desenvolvido pela IBM, é um sistema que automatiza alguns processos de instalação e configuração. Ele permite ligar e desligar as máquinas remotamente e usar uma espécie de shell paralelo para executar o mesmo comando em vários nós. A página principal do projeto é <http://www.xcat.org/>, mas também podem ser verificadas informações mais detalhadas no site <http://www.alphaworks.ibm.com/tech/xcat>.

Existem muitas ferramentas para auxiliar o profissional na administração de um cluster. O importante é que o uso de uma ou de outra ferramenta será determinada pela necessidade do administrador no momento de uma atividade específica, seja de preventiva ou corretiva em um sistema clustering.

4.2 ESCALONAMENTO DE JOBS E RECURSOS

A tecnologia de cluster de computadores foi inicialmente aplicada para reduzir os custos do uso de supercomputadores em processamento de alto desempenho, com um benefício considerável. O seu funcionamento se baseia na característica de parecer um único sistema para o usuário. De fato, cada usuário que utiliza o sistema, submete algumas tarefas ou jobs ao cluster e deseja que seus processos sejam executados de forma transparente.

Embora o usuário não saiba, mas escalonar esses processos é uma tarefa bem difícil. E é pensando nisso que existem os Softwares de Gerenciamento de Recursos (CMS). A função desses tipos de softwares é de administrar e gerenciar tarefas submetidas para um cluster de computadores, usando técnicas de enfileiramento e escalonamento de tarefas baseadas em parâmetros. Abaixo descrevemos algumas delas:

- **PBS:** Desenvolvido inicialmente pela NASA e posteriormente apresentado em uma versão comercial, PBS apresenta suporte a tarefas tanto de um único sistema como de múltiplos sistemas. O sistema de escalonamento utilizado adiciona as tarefas primeiramente a uma fila, para que, posteriormente, o escalonador analise as tarefas. Além da versão comercial existe também uma versão de código aberta conhecida como openPBS. É um dos escalonadores de processos mais usados em cluster beowulf (Petrobrás). O site do projeto é <http://www.openpbs.org/>.
- **Condor:** é uma ferramenta que pode ser usada para gerenciar clusters. Para a execução de tarefas é necessário primeiramente definir os recursos necessários. A seguir as tarefas são armazenadas em uma lista de espera. Algumas das características de Condor são a submissão distribuída de tarefas, prioridades para usuários e tarefas, suporte a múltiplos modelos de tarefas, checkpointing e migração, suspensão de tarefa e posterior continuação, autenticação e autorização, entre outros. Tutorial, downloads e demais informações podem ser encontradas no site <http://www.cs.wisc.edu/condor/>.
- **CCS:** desenvolvido pelo Centro de Computação Paralela de Paderborn (Alemanha), o objetivo de CCS é gerenciar sistemas MPP e clusters. Para isso, a ferramenta permite o acesso aos recursos de forma exclusiva concorrentemente, processamento simultâneo do modo interativo e de fila, maximização do uso do sistema através do particionamento dinâmico e escalonamento, além de tolerância a falhas em acessos remotos. A arquitetura do CCS é modular, o que permite

integrar um grande número de sistemas. Algumas informações constam no site <http://sourceware.org/cluster/ccs/>.

- **ANTS:** é um escalonador de sistemas distribuídos que executa tarefas em uma rede. É distribuído sobre os termos GNU Public License e pode ser baixado de <http://unthought.net.antsd>.

Existem muitas outras soluções, sejam comerciais, sejam acadêmicas. Dentre elas podemos citar: Codine, Conect:Queue, Load Balancer, EASY, QBATCH e outras. Cada uma delas pode ser utilizada com a finalidade de controle de jobs em clusters de alto desempenho.

4.3 MONITORAÇÃO

Assim como a instalação e o escalonamento de tarefas, a monitoração de um aglomerado de máquinas (clusters) também é uma atividade importante para quem administra os clusters. A monitoração de sistemas consiste em coletar parâmetros de performance do sistema como utilização de CPU, utilização de memória, taxa de I/O, e apresentá-los de forma objetiva para o administrador. Atualmente existem diversas ferramentas que permitem verificar o estado de um ambiente de maneira simples e intuitiva. Como exemplos de aplicações temos Ganglia, SCMS, RVision, bWatch que serão vistas na sequência.

- **Ganglia:** é uma ferramenta de monitoração para clusters, desenvolvida de forma distribuída e escalável. Um módulo centralizador coleta e atualiza as informações, enquanto que cada nó mantém uma cópia do estado corrente do sistema. Os dados coletados podem ser visualizados graficamente através de uma interface Web. Com Ganglia é possível monitorar qualquer tipo de informação. Mais informações podem ser pesquisadas no site <http://ganglia.sourceforge.net/>.
- **SCMS:** tem como objetivo monitorar de forma simples, eficiente e robusta clusters de pequeno e médio porte através de uma arquitetura centralizada organizada num módulo de monitoração e num módulo de centralização, o qual armazena os dados monitorados e atende as requisições dos clientes. A ferramenta permite monitorar o uso de CPU, memória, rede e disco, além de fornecer informações úteis sobre a configuração dos nós do cluster. O site desse projeto é o <http://www.opensce.org>.
- **bWatch:** É um script desenvolvido por Jacek Radajewski. É utilizado para monitorar a carga e o uso da memória em todos os nós do cluster. Foi escrito exclusivamente para a plataforma GNU/Linux. Ferramenta leve e fácil de configurar. Foi desenvolvido em TCL/TK. Para gerenciar pequenos clusters de até

10 nodos, esta ferramenta é bastante eficaz. Essa ferramenta pode ser baixada de <http://sourceforge.net/projects/bwatch/>.

- **RVision:** é uma ferramenta de monitoração desenvolvida com o objetivo de ser adaptável a diferentes clusters, tendo uma arquitetura aberta e configurável. Para manter essas características, a ferramenta possui uma interface para a comunicação de clientes com o núcleo da ferramenta. A arquitetura de RVision é centralizada, sendo composta por um programa monitor e um programa centralizador. Maiores informações sobre o RVision podem ser obtidas em <http://www.cpad.pucrs.br/projetos/projinternos/rvision/index.html>.
- **Mon:** É uma ferramenta de gerenciamento destinada a monitorar aplicativos e responder a eles, com base em eventos. É totalmente escrito em perl e é destinado a monitorar a disponibilidade de rede e servidor. Esse pacote será utilizado em nossa construção de ambiente de alta disponibilidade, por se mostrar versátil ao se integrar com o funcionamento com outra ferramenta, o Heartbeat. Pesquisas mais detalhadas podem ser realizadas pelo site <http://www.kernel.org/software/mon/>.
- **Big-Brother / Hobbit:** Embora o mon seja uma ótima ferramenta, não é destinada a oferecer ao administrador uma rápida visão do ambiente. Para isso existem ferramentas de monitoramento gráfico, as quais podem ser implementadas. Big-brother é gratuito para uso não comercial. Já o hobbit é a versão openSource do Big-brother. O site oficial do Big-brother é o <http://bb4.com/>. Já o site do projeto do Hobbit é o <http://sourceforge.net/projects/hobbitmon/>.

De uma forma geral as ferramentas de gerenciamento tem evoluído no sentido de incluir diversas funcionalidades em um único componente, contemplando assim todas as necessidades exigidas na administração de clusters. Este é o caso das ferramentas ROCKS e OpenSCE, que apresentam diferentes funcionalidades ou buscam a integração entre vários softwares, disponibilizando em um único recurso a possibilidade de instalação e configuração de software, a monitoração e gerenciamento do estado do cluster, o balanceamento de carga, além de ferramentas que auxiliem no desenvolvimento de aplicações.

4.4 POLÍTICA DE SEGURANÇA E BACKUP

Segurança em computadores é um termo geral que abrange uma grande área da computação e processamento de dados. Setores que dependem de sistemas de computador e redes para conduzir transações diárias de negócios e acessar informações cruciais, encaram

seus dados como uma parte importante de seus recursos. Em alguns setores, tal como comércio eletrônico, a disponibilidade e confiabilidade de dados pode ser a diferença entre sucesso e fracasso. Esse conceito não seria diferente para a computação de cluster.

Na prática, o que normalmente se considera é o quanto vale a informação que está sendo processada, ou armazenada ou ainda transferida. Será que a falta de uma política de segurança não coloca em risco toda uma estratégia de operações de uma empresa? Foi com essa visão que a segurança em computadores, de forma particular em clusters, tornou-se uma despesa quantificável e justificável em todos os orçamentos de TI.

As Empresas que requerem integridade de dados e alta disponibilidade evocam as habilidades de seus administradores de sistemas, desenvolvedores e engenheiros para garantir confiabilidade de seus sistemas, serviços e informações 24 horas por dia, sete dias por semana. Cair nas mãos de usuários, processos ou ataques maldosos coordenados é uma ameaça direta ao sucesso da empresa.

Existem vários níveis de ataque. A própria engenharia social, atualmente, é um dos ataques mais utilizados. Devido a isso, ter uma política de segurança bem definida em seu local de trabalho é de vital importância.

Para isso, analisar a segurança em nível de camadas é uma ação que deve ser executada. Segundo Bookman (2003), existem três níveis de segurança: Nível de sistema e Nível de rede e Nível físico.

- **Segurança em nível de sistema:** É aquele efetuado diretamente no sistema operacional do sistema. Considerando o Linux como sistema operacional instalado no cluster, os arquivos de configuração `/etc/hosts.allow` e `/etc/hosts.deny` podem ser utilizados para garantir quem obtém o acesso a quais serviços. Para o caso de acesso remoto, usar sempre transmissão protegida (SSH) é outro exemplo. Fechar as portas desnecessárias no arquivo `/etc/inetd.conf`, instalar sistemas de detecção de intrusão e manter o sistema atualizado são outros exemplos.
- **Segurança em nível de rede:** Um dos itens mais importantes é bloquear serviços inseguros, tais como telnet, ftp e rsh. Só liberar acesso somente às portas absolutamente necessárias. Ferramentas de monitoramento, como Big-brother/Hobbit, podem ser ajustadas para notificar um administrador responsável, em determinados casos.
- **Segurança em nível físico:** O cluster deve ser mantido em áreas isoladas, de portas fechadas, inacessível ao público em geral. Pode ser considerada a remoção

dos drives de CDROM e de disquete, o que ajuda a evitar alguém de iniciar o sistema usando métodos alternativos.

O que deve estar na mente de todo administrador de uma plataforma em cluster é que, embora os clusters sejam caros, substituí-los é muito mais fácil do que substituir todos os dados que neles estão armazenados. Pensando nisso, selecionar a melhor estratégia de backup disponível também deve ser uma prática essencial. Com frequência, o orçamento é um fator decisivo para selecionar o método ou programa de backup apropriado para o ambiente específico.

Independente da análise de custo, e das ferramentas utilizadas, uma prática de recuperação de backup pode e deve ser definida. Dentre as principais práticas existentes, seguem abaixo alguns pontos importantes:

- **Desenvolver estratégias de backup e restauração e testá-las:** Com um bom plano, você poderá recuperar os dados com rapidez, caso sejam perdidos.
- **Treinar o pessoal adequado:** Em redes com segurança mínima e média, atribua direitos de backup a um usuário e direitos de restauração a outro usuário. Treine o pessoal com direitos de restauração para executar todas as tarefas de recuperação se o administrador não estiver disponível. Em uma rede de alta segurança, certifique-se de que somente os administradores podem restaurar arquivos.
- **Criar um conjunto de backup de recuperação automatizada do sistema:** Crie sempre um conjunto de backup de recuperação automatizada do sistema quando o sistema operacional for alterado; por exemplo, sempre que instalar um novo hardware e novos drivers ou aplicar um Service Pack, bem como novos “pacotes”. Com um conjunto de backup, fica mais fácil recuperar o sistema em caso de uma falha.
- **Criar um log de backup:** Opte sempre por criar um log de backup para cada backup e os imprima para referência. Mantenha um livro de logs para facilitar a localização de arquivos específicos. O log de backup é útil na restauração de dados.
- **Executar tentativas de restauração:** Faça uma tentativa de restauração periodicamente para verificar se o backup dos arquivos foi feito de forma correta. Uma tentativa de restauração pode descobrir problemas de hardware que não aparecem com as verificações.

- **Fazer backup do cluster de servidor de forma eficiente:** Para se restaurar o cluster de forma eficiente, em caso de falha no cluster inteiro ou em um nó individual, deve ser executado um backup de recuperação de sistema e recuperar o backup dos discos de cluster de cada nó defeituoso.

Os recursos e investimentos em segurança estão, obviamente, relacionados à importância da informação e ao risco que a mesma está exposta. No caso de rede de computadores, especificamente em cluster, a vulnerabilidade fica aumentada devido à fragilidade apresentada por cada um dos nossos “elos da corrente” ou nós do cluster. É preciso configurar a plataforma deixando-a com uma estrutura “sólida” e confiável.

3.1.1 Considerações Físicas

Ter em mente a localização física das máquinas é uma preocupação crucial no projeto. Salas apropriadas, sistemas de refrigeração, instalações elétricas adequadas, isolamento de ruído, gerenciamento de força interrupta e, além disso, se o espaço disponível é suficiente para acomodar as equipamentos, são alguns itens específicos.

O administrador deve verificar se o cluster que está implementado contém uma quantidade excessiva de componentes, para poder suportar o nível de energia e potência adequadas para a sua alimentação. Se já existir um centro de dados (CPD) é possível que já tenha uma estrutura no lugar para suportar alguns computadores extras.

5 CONSTRUÇÃO DE CLUSTERS

5.1 CONSIDERAÇÕES NO PROJETO

Considerando o conhecimento adquirido até o momento, a construção de um cluster em nível de produção é o objetivo seguinte para o administrador de rede solidificar seu aprendizado. Para se construir um cluster que ofereça os serviços de alguma das tecnologias estudadas e que seja de fácil utilização, alguns passos iniciais devem ser seguidos.

Na verdade, a escolha da tecnologia apropriada pode gerar um efeito de facilidade ou dificuldade na administração do cluster, podendo ocasionar vantagens ou desvantagens com relação à flexibilidade, desempenho, escalabilidade e a principal de todas, o retorno de investimento. Cabe ao administrador considerar quais aplicações desejará executar e o desempenho requerido para cada uma delas.

Considerações devem ser levantadas de forma a se perguntar qual tipo de cluster é o mais adequado às necessidades. Questionamentos do tipo: Para um servidor WEB, a solução deve incluir alta disponibilidade (HA) e se for o caso balanceamento de carga? Será preciso realmente um equipamento que possua um processador SMP para o aplicativo paralelo, ou processadores individuais podem ser utilizados?

Basicamente, para o início da construção de um cluster, deve-se fazer as seguintes considerações: Considerações físicas; Escolha da arquitetura do hardware; Escolha da rede de interconexão; Escolha do sistema operacional; Escolha do software de comunicação; Escolha do software de gerenciamento do sistema. Cada uma dessas considerações será visto na sequência..

5.1.1 Considerações Físicas

Ter em mente a localização física das máquinas é uma preocupação crucial ao projeto. Salas apropriadas, sistemas de refrigeração, instalações elétricas adequadas, sistemas de gerenciamento de força ininterrupta e, além disso, se o espaço disponível é suficiente para acomodar os equipamentos, são alguns itens específicos.

O administrador deve verificar se o cluster que está implementando contém uma quantidade excessiva de componentes, para poder empregar o nível de energia e potência adequadas para a sua alimentação. Se já existir um centro de dados (CPD), é possível que já tenha infra-estrutura no lugar para suportar alguns computadores extras.

5.1.2 Escolha da Arquitetura do hardware

A escolha do hardware é baseada praticamente no orçamento disponível e na necessidade do desempenho exigida. Embora isso pareça sem sentido, conseguir os melhores componentes pelo preço é extremamente importante quando o controle orçamentário é exigido. Normalmente o custo do hardware é negociável. Com relação à performance, dependerá exclusivamente do tempo de resposta desejado das aplicações.

Tradicionalmente, o software de cluster requer que uma única arquitetura seja usada no agrupamento de máquinas. Porém esse conceito tem modificado bastante devido à criação de uma camada intermediária de comunicação (*middleware*), conseqüentemente possibilitando o uso de um hardware mais heterogêneo.

O uso de equipamentos com tecnologia SMP, por exemplo, ou até mesmo CPUs do tipo Intel Xeon ou o AMD Opteron, com a opção da BIOS, que permite configuração do recurso de Hyper-Threading, deve ser corretamente analisada e aplicada, pois requer que a aplicação seja compatível com este recurso.

5.1.3 Escolha da Rede de Interconexão

O maior gargalo em qualquer configuração de cluster é a rede. Usar processadores mais rápidos e discos rígidos maiores em nada significará se a rede não favorecer a um desempenho de comunicação de baixa latência e melhor escalabilidade.

Como vimos anteriormente, as duas principais tecnologias de rede de alto desempenho envolvidas no mundo dos clusters são a Gigabit e o padrão Myrinet. Alguns clusters podem fazer uso de combinações das duas tecnologias, de forma que a primeira seria usada para monitoramento e montagem dos sistemas de arquivos, já a segunda seria usada exclusivamente para a troca de mensagens.

5.1.4 Escolha do Sistema Operacional

A idéia primordial para o uso de uma arquitetura de cluster para solucionar alguma necessidade em uma rede de computadores se apóia no fato que, normalmente, as empresas não tem condições para fazer um investimento de ordem mais cara. Algumas pesquisas mostram que o sistema operacional adotado em 90% da base instalada no mundo é o

GNU/Linux, sendo as três principais distribuições: RedHat, Suse e Mandrake. O Linux executa em quase todas as plataformas possíveis.

Na verdade, a escolha do Linux como sistema operacional para plataforma *Clustering* se dá devido às vantagens inerentes aos chamados “softwares de fonte aberta” ou *OpenSource*. Uma das principais e grandes vantagens dessa característica é a utilização da experiência de centenas, se não de milhares de programadores reformulando e aperfeiçoando o código, tornando mais estável e eficiente.

O Linux é um grande atrativo também pelo fato de ser bem aplicado em hardwares obsoletos, possibilitando que computadores mais antigos sejam trazidos de volta ao serviço. O kernel do Linux pode ser modificado em qualquer modo ou forma, dependendo da necessidade. Nenhum outro sistema operacional permite este nível de versatilidade. Essa característica de computação modular que torna o Linux perfeito para Clusters.

5.1.5 Escolha do Software de Comunicação

Na atualidade, existem duas bibliotecas utilizadas na computação paralela: PVM e MPI. A disponibilidade de softwares gratuitos, tais como Linux e compiladores do tipo gcc, favoreceram a utilização de software de comunicação também gratuito. Atualmente, o MPI foi padronizado e agora tomou a dianteira em programas portáteis e paralelos.

Segundo Pitanga (2003), existem diversas implementações de MPI e, portanto, deve-se pesquisar e avaliar cada uma delas de acordo com o contexto de suas aplicações. Há a MPICH, para ambientes Windows, a LAM/MPI, que também é muito popular. Há um MPI para o IBM, MPI para SGI e máquinas digitais. Há até rumores de MPI para Palm Pilot. Enfim, as bibliotecas de passagem, possibilitam aos programadores uma interface padronizada para escrever código portátil através de máquinas agrupadas e heterogêneas.

5.1.6 Escolha do Software de Gerenciamento

A escolha de um software de gerenciamento que seja eficaz é de vital importância ao Administrador da Rede, principalmente para os casos de clusters compostos por dezenas, centenas, ou até mesmo milhares de nós. Esse software pode ser composto por soluções OpenSource ou por soluções proprietárias. O uso de uma ou outra solução dependerá tanto da capacidade do administrador de assimilar o conhecimento necessário para uso da ferramenta, quanto da flexibilidade de ser executado em arquiteturas heterogêneas.

Embora esta etapa não seja de preocupação de muitos profissionais, um bom gerenciamento é essencial para o bom funcionamento do sistema. Essa prática está diretamente ligada ao uso de ferramentas de monitoramento, as quais mantêm um olhar atento em todos os nós e componentes do cluster.

Existem várias opções de softwares de monitoramento e gerenciamento, tais como o Hobbit, Netview, MRTG, etc. A verdade é que os clusters exigem uma manutenção específica, o que obriga o administrador a manter uma atenção especial neles.

5.2 CONSTRUINDO UM AMBIENTE DE ALTA DISPONIBILIDADE

5.2.1 Alta Disponibilidade na Prática

Como já foi descrito anteriormente, a Alta Disponibilidade é muito mais do que a instalação de alguns pacotes e aplicativos. É um conceito muito além, cuja implementação dependerá da percepção do administrador quanto à existência de SPOFs em sua estrutura. Existem várias formas de tentar contornar as falhas dos sistemas computacionais. Alguns exemplos são:

- Hardware especializado;
- Sistemas de tolerância de falhas;
- Alta Disponibilidade.

Essa última prevê a redundância total do sistema, atuando em caso de defeito, quando a máquina trava. Uma das formas para se alcançar esse objetivo é com a redundância de hardware, e controle via software. Normalmente é aplicada a solução de uma máquina como secundária da principal. Seguindo esse conceito, existem dois tipos básicos de clusters de alta disponibilidade, os simétricos e os assimétricos.

Clusters de alta disponibilidade assimétricos são os mais simples de se implementar. Eles se baseiam no conceito de servidores primário e secundário. Servidores que estão provendo algum tipo de serviço são considerados como primários. Servidores que aguardam a ocorrência de alguma falha no servidor primário são denominados servidores secundários. A detecção da falha ocorre através do monitoramento dos servidores primários por parte dos secundários. Esse tipo de operação pode ser observado na figura 15. O servidor primário fornece os serviços e “monta” a área de armazenamento. O servidor secundário monitora o funcionamento do primário, aguardando sua falha. No momento da falha, o secundário assume tanto os serviços quanto a área de armazenamento.

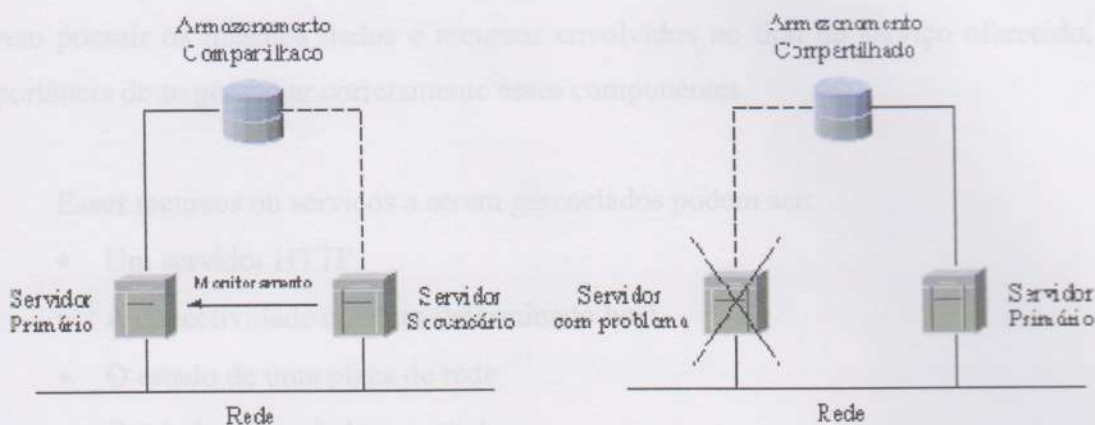


Figura 15: Cluster de Alta Disponibilidade Assimétrico
Fonte: Simões (2003)

A principal diferença entre clusters assimétricos e simétricos é que em um cluster simétrico não há a figura de um servidor inativo aguardando uma falha de outro, primário, para então servir a aplicação. Neste modelo, ambos servidores servem alguma aplicação. Assim, os termos primário e secundário deixam de ser usados para designar o servidor, passando a ser usado para a aplicação. É o que pode ser verificado na figura 16. Cada servidor é responsável em manter uma determinada aplicação (A e B). Ambos os servidores se monitoram simultaneamente. Quando ocorre a falha por parte de algum serviço, por exemplo a aplicação A, essa é assumida pelo outro servidor. O mesmo ocorrendo com relação a unidade de armazenamento.

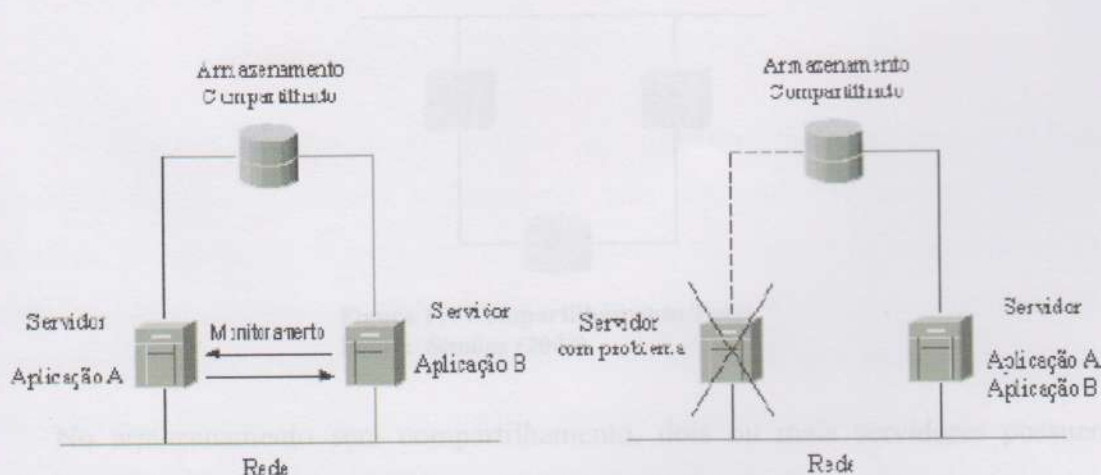


Figura 16: Cluster de Alta Disponibilidade Simétrico
Fonte: Simões (2003)

Para que uma máquina espelho assuma o lugar quando a principal falhar, ambas devem possuir os mesmos dados e recursos envolvidos no tipo de serviço oferecido, daí a importância de se gerenciar corretamente esses componentes.

Esses recursos ou serviços a serem gerenciados podem ser:

- Um servidor HTTP
- A conectividade com um determinado host
- O estado de uma placa de rede
- O estado de um hub ou switch
- Uma unidade de armazenamento

Vale ressaltar para o caso de unidade de armazenamento que, dependendo da escala, performance, funcionalidade e flexibilidade do cluster, são necessárias diferentes implementações de armazenamento. Os principais métodos de armazenamento são o compartilhamento total, sem compartilhamento e espelhamento de disco.

No compartilhamento total, todos os servidores acessam o mesmo meio de armazenamento. Este modelo apresenta potenciais problemas de corrupção de dados quando mais de um servidor está ativo. Esta implementação é bastante comum em clusters assimétricos, pois o servidor secundário só será ativado quando da falha do principal. Esse tipo de configuração pode ser observado na figura 17.

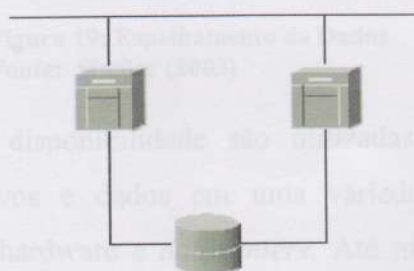


Figura 17: Compartilhamento Total
Fonte: Simões (2003)

No armazenamento sem compartilhamento, dois ou mais servidores possuem seu próprio meio de armazenamento. No evento de uma falha em um dos servidores, o servidor responsável pela substituição passa a ter acesso total ao meio de armazenamento original do nó defeituoso. Esta implementação é comum em clusters simétricos. A figura 18 mostra a montagem de uma unidade de armazenamento sem compartilhamento.

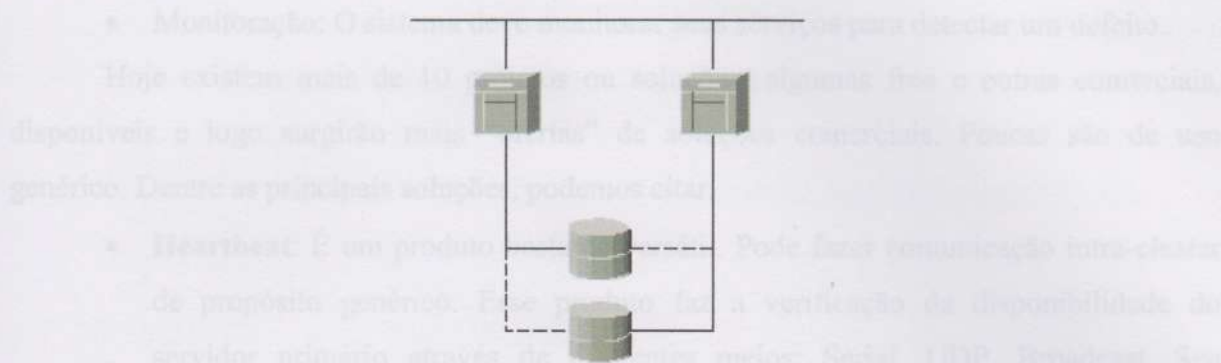


Figura 18: Sem Compartilhamento

Fonte: Simões (2003)

Em um ambiente de espelhamento de discos, não há nenhum tipo de compartilhamento entre os nós servidores. Através do uso de software, os dados são espelhados ou replicados de um servidor para o outro através da rede. Na figura 19 essa característica pode ser observada. Na realidade ocorre um espelhamento por parte dos dados dos dois servidores através de um software aplicativo.

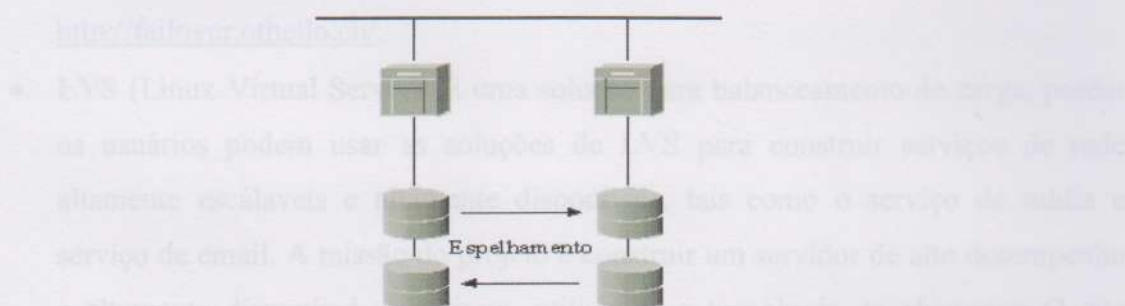


Figura 19: Espelhamento de Dados

Fonte: Simões (2003)

As soluções de alta disponibilidade são utilizadas para garantir o máximo de operacionalidade de aplicativos e dados em uma variedade de sistemas operacionais, aplicativos, componentes de hardware e *datacenters*. Até mesmo no ambiente de TI mais diverso, as várias soluções existentes podem proporcionar o máximo em disponibilidade de dados e de aplicativos. Porém, para se montar um ambiente que seja de Alta Disponibilidade, todas essas soluções devem fornecer, basicamente, as seguintes características:

- Hardware: Redundância de máquinas, conexão dedicada e alta velocidade.
- Instalação do Sistema: Os softwares devem ser independentes de distribuição.
- Integridade de dados: Os dados devem ser espelhados em tempo real para a completa disponibilidade do sistema em caso de defeito.
- Controle de serviços: Os serviços podem ser passados de uma máquina para outra.

- **Monitoração:** O sistema deve monitorar seus serviços para detectar um defeito.

Hoje existem mais de 10 projetos ou soluções, algumas free e outras comerciais, disponíveis e logo surgirão mais “ofertas” de soluções comerciais. Poucas são de uso genérico. Dentre as principais soluções, podemos citar:

- **Heartbeat:** É um produto bastante versátil. Pode fazer comunicação intra-cluster de propósito genérico. Esse produto faz a verificação da disponibilidade do servidor primário através de diferentes meios: Serial, UDP, Broadcast. Seu funcionamento é baseado no IP Takeover. Apesar de ser um produto que ainda tem muito a desenvolver, sua versão de produção já atende de forma satisfatória. No site <http://www.linux-ha.org/Heartbeat> podem ser encontrados os pacotes para instalação, alguns tutoriais para instalação e outras informações do projeto.
- **Failover:** Foi portado do Solaris para o Linux. Seu funcionamento é orientado para o serviço e não para o servidor. Também possui o conceito de masters e slaves. Seu desenvolvimento é baseado em TCL. Possui a funcionalidade de fazer o IPAT (IP Address Takeover) em interfaces virtuais. A página principal do projeto é <http://failover.othello.ch/>.
- **LVS (Linux Virtual Server):** É uma solução para balanceamento de carga, porém os usuários podem usar as soluções de LVS para construir serviços de rede altamente escaláveis e altamente disponíveis, tais como o serviço de mídia e serviço de email. A missão do projeto é construir um servidor de alto desempenho e altamente disponível para linux, utilizando a tecnologia de clustering. O site oficial do projeto é <http://www.linuxvirtualserver.org/>.
- **Eddieware:** O objetivo principal do projeto é a escalabilidade de servidores web. Consiste de um servidor altamente disponível e com balanceamento de carga no serviço HTTP, um servidor de DNS “aprimorado”, estrutura de servidor de duas camadas (Frontend e backend). Executa o IPAT e re-roteamento de tráfego após falha do servidor. Mais informações podem ser verificadas no site <http://www.eddieware.org/>.
- **RSF-1 (Resilient Server Facility):** Não é considerada uma solução especializada, mas de uso geral. Usa o Heartbeat para monitoramento de IP. Pode fazer failover manual ou automático. É multiplataforma. Mais informações podem ser verificadas no site <http://www.rsi.co.uk>.

- **Piranha:** É um produto para clustering da RedHat e inclui o código do LVS para o kernel, uma ferramenta gráfica para configuração do cluster e uma ferramenta de monitoração do cluster. Baixo custo de aquisição; a funcionalidade é parte da licença Red Hat Enterprise. <http://www.redhat.com/support/wpapers/piranha/>

5.3 O PROJETO – APLICAÇÃO EXATA

Considerando o estudo realizado até o momento, iniciar a montagem de um ambiente de Alta Disponibilidade na prática é a próxima etapa. A oportunidade surgiu quando da necessidade de garantir um funcionamento em alta disponibilidade de uma das aplicações estratégicas para a Petrobrás: **APLICAÇÃO EXATA**.

Dentre suas funcionalidades, o acompanhamento de perfuração de poços em tempo real é uma das principais. Esse aplicativo foi originalmente desenvolvido para funcionar em um servidor apache (HTTP), porém foi reinscrito em java para rodar em um servidor Jboss.

Inicialmente essa migração não representou um problema, mas ao longo da implementação, foram necessários vários testes com o script de inicialização do Jboss. Na figura abaixo, verificou-se a estrutura existente antes da implementação da Alta disponibilidade.

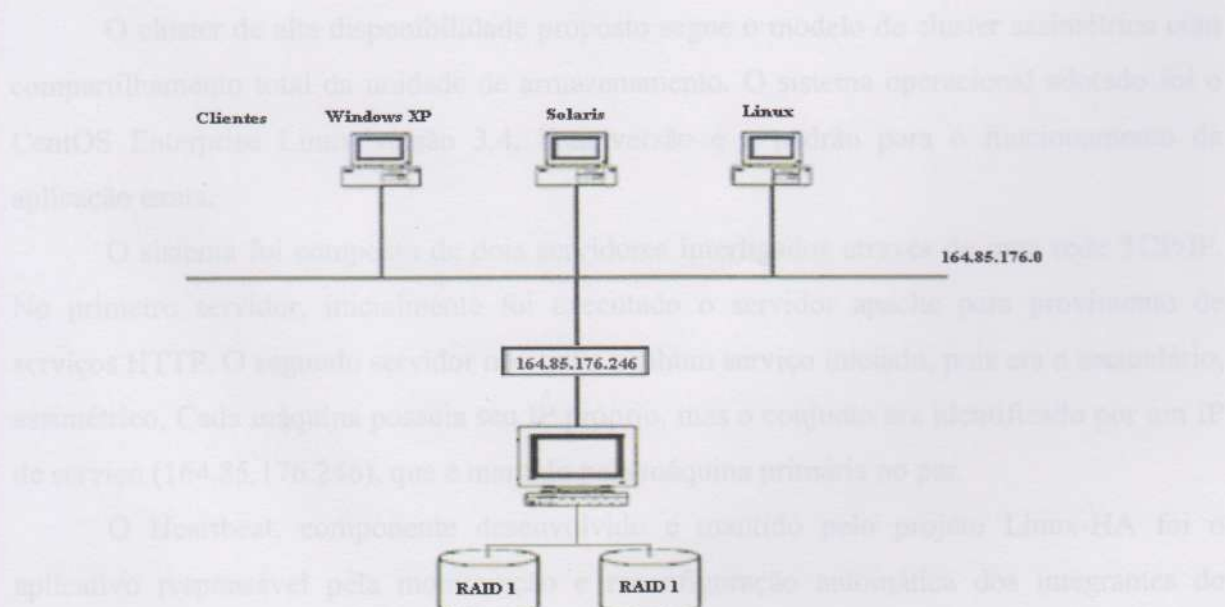


Figura 20: Aplicação Exata

Após análise da estrutura, verificou-se que para uma solução atender a nossa necessidade, teria que ser formada por “softwares livres”, ser eficiente, bem como de fácil gerenciamento, já que a implantação da arquitetura teria que ser rápida e objetiva. A solução proposta pode ser observada na figura abaixo.

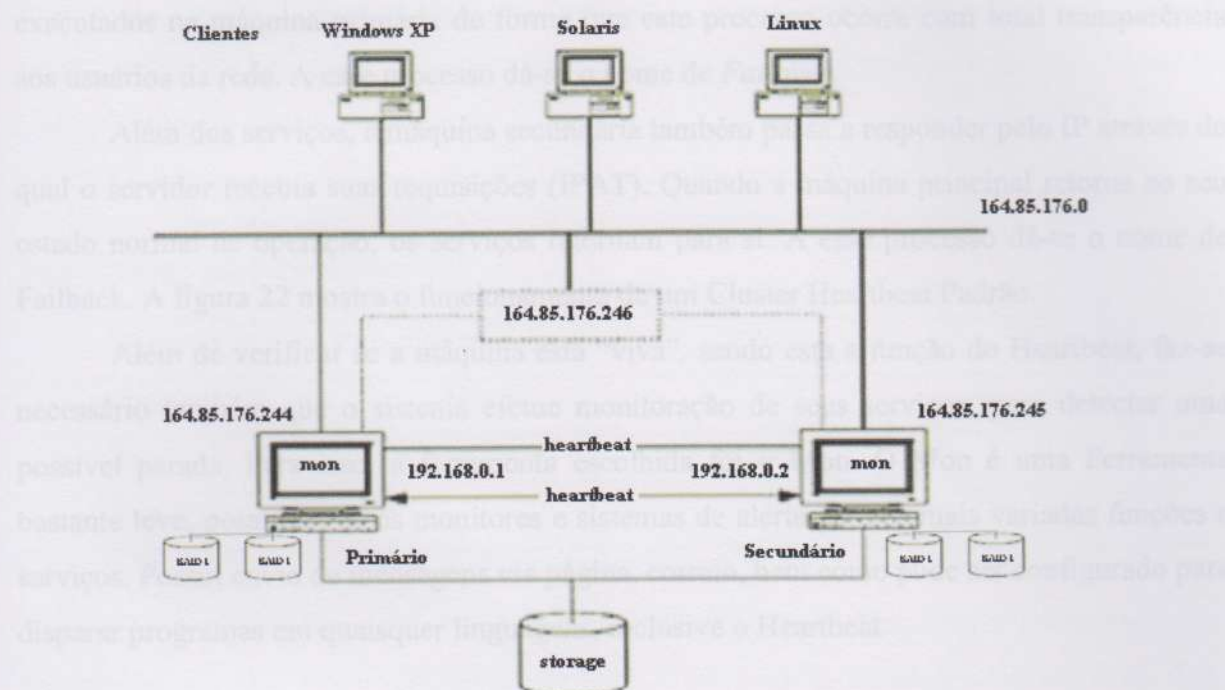


Figura 21: Aplicação Exata – Altamente Disponível

O cluster de alta disponibilidade proposto segue o modelo de cluster assimétrico com compartilhamento total da unidade de armazenamento. O sistema operacional adotado foi o CentOS Enterprise Linux versão 3.4. Esta versão é a padrão para o funcionamento da aplicação exata.

O sistema foi composto de dois servidores interligados através de uma rede TCP/IP. No primeiro servidor, inicialmente foi executado o servidor apache para provimento de serviços HTTP. O segundo servidor não tinha nenhum serviço iniciado, pois era o secundário, assimétrico. Cada máquina possuía seu IP próprio, mas o conjunto era identificado por um IP de serviço (164.85.176.246), que é mantido pela máquina primária no par.

O Heartbeat, componente desenvolvido e mantido pelo projeto Linux-HA foi o aplicativo responsável pela monitoração e reconfiguração automática dos integrantes do cluster na ocorrência de falhas. Seu funcionamento baseia-se no envio de sinais por parte dos membros do cluster.

Com base nas respostas obtidas, os membros conseguem determinar se os outros integrantes estão ou não funcionando. A esta troca de mensagens em intervalos específicos de

tempo com o intuito de verificar os estados de funcionamento dos integrantes do cluster dá-se o nome de heartbeat.

A partir do momento que essas pulsações não são mais detectadas, caracterizando-se a ocorrência de uma falha, a máquina secundária toma para si os serviços que estavam sendo executados na máquina primária de forma que este processo ocorre com total transparência aos usuários da rede. A esse processo dá-se o nome de *Failover*.

Além dos serviços, a máquina secundária também passa a responder pelo IP através do qual o servidor recebia suas requisições (IPAT). Quando a máquina principal retorna ao seu estado normal de operação, os serviços retornam para si. A esse processo dá-se o nome de *Failback*. A figura 22 mostra o funcionamento de um Cluster Heartbeat Padrão.

Além de verificar se a máquina está “viva”, sendo esta a função do Heartbeat, faz-se necessário também que o sistema efetue monitoração de seus serviços, para detectar uma possível parada. Para isso, a ferramenta escolhida foi o Mon. O Mon é uma Ferramenta bastante leve, possui diversos monitores e sistemas de alerta para as mais variadas funções e serviços. Possui envio de mensagens via página, correio, bem como pode ser configurado para disparar programas em quaisquer linguagens, inclusive o Heartbeat.



Figura 22: Cluster heartbeat Padrão

Fonte: acol (cluster linux, 2003)

Embora, em um primeiro momento, o uso do Mon possa ser dispensável, já que o Heartbeat já realiza a permuta do IP virtual entre as máquinas que compõem o cluster, verifica-se uma utilização bastante específica. Para os casos em que a máquina não perdeu seu sinal de máquina ativa (conectividade), porém o serviço (no caso o WEB) esteja inoperante, ou seja, o Mon seria o responsável pela monitoração do serviço na máquina primária e restauração na máquina secundária.

5.3.1 Preparando as Máquinas

O primeiro passo na implantação do *cluster* foi a instalação do sistema operacional. O sistema operacional instalado foi o CentOS Enterprise Linux versão 3.4. A instalação deu-se através de CDROM para a primeira máquina e como a segunda possuía as mesmas características de hardware, ou seja, cluster homogêneo, foi utilizada a ferramenta Ghost for Unix (G4U) para replicar a imagem.

O esquema de particionamento adotado é mostrado na Tabela 2.

Tabela 2: Esquema de Particionamento

Partição	Ponto de Montagem	Tipo	Tamanho
/dev/sda1	/boot	Ext3	100 M
/dev/sda2	/	Ext3	69484 M
/dev/sda3		swap	2096 M

Optou-se pelo uso de sistemas de arquivo Ext3 por este apresentar algumas vantagens em relação a outros tipos disponíveis. O Ext3 é um sistema de arquivos journaling, desta forma, os dados são escritos em disco de forma que o sistema de arquivos está sempre consistente. Assim, durante uma reinicialização causada por alguma falha, não há necessidade de uma checagem de todo o sistema de arquivo, contribuindo-se assim para o aumento da disponibilidade do sistema.

Durante a instalação, optou-se pelo método de instalação personalizada. Foram selecionados os pacotes básicos para o funcionamento de um servidor de rede, além dos pacotes de desenvolvimento, para compilação das ferramentas usadas. Ao final da instalação as estações foram configuradas para funcionamento em rede, com os respectivos IPs fixos.

Pelo que se pode observar na figura 21, as estações teriam no mínimo 2 duas interfaces de rede, com os respectivos IPs configurados: estação primária 164.85.176.244/eth0, 192.168.0.1/eth1; estação secundária 164.85.176.245/eth0, 192.168.0.2/eth1. As outras estações são consideradas clientes da estrutura do cluster.

As estações foram conectadas pela rede principal por um switch. Na conexão da rede privada, responsável pela verificação de máquina ativa, foi usado um “Cabo Cross”, interligando as interfaces de rede eth1 das duas estações. É por essa rede que o Heartbeat gerencia a operação das duas máquinas.

Vale salientar que também foi disponível para as duas estações o Storage/NFS, que serviu de repositório de dados com a configuração de compartilhamento total. Como o compartilhamento é total, a área compartilhada deve estar montada via automount (/etc/auto.master) ou montagem direta pelo arquivo /etc/fstab. Essa área deve conter todos os arquivos, diretórios e programas necessários ao perfeito funcionamento do cluster, nas duas máquinas, bem como o repositório da aplicação - Exata.

5.3.2 Instalando e Configurando o Heartbeat

O software Heartbeat pode ser encontrado em <http://linux-ha.org/download>. No momento a última versão é a 2.0.8-2. Os pacotes básicos para a instalação do Heartbeat são:

- heartbeat-pils-2.0.8-2
- heartbeat-stonith-2.0.8-2
- heartbeat-2.0.8-2

Após efetuar o download, foi necessário executar a instalação dos pacotes através do comando “rpm -ivh <pacote>”. Conforme abaixo:

```
# rpm -ivh heartbeat-2.0.8-2.el4.centos.i386.rpm
# rpm -ivh heartbeat-pils-2.0.8-2.el4.centos.i386.rpm
# rpm -ivh heartbeat-stonith-2.0.8-2.el4.centos.i386.rpm
```

Vale salientar que a instalação desses pacotes pode ocasionar algum conflito ou necessidade de instalação com algum outro pacote. Iremos considerar que o administrador de rede tenha conhecimentos suficientes como administrador do sistema para resolver tais ocorrências.

Com a instalação dos pacotes, foi criado um diretório /etc/ha.d, onde estão os arquivos de configuração. Do diretório de documentos (/usr/lib/share/heartbeat), foram copiados os arquivos ha.cf, haresources e authkeys para o diretório /etc/ha.d. Daqui por diante, foi utilizada uma configuração básica nesses arquivos para o funcionamento do nosso cluster. Primeiramente foi configurado o arquivo ha.cf. Abaixo segue uma configuração padrão:

```

# cd /etc/ha.d
# vi ha.cf
-----
debugfile /var/log/ha-debug
logfile /var/log/ha-log
keepalive 2 - Seta o tempo entre os heartbeat
deadtime 10 - Informa que o nodo será declarado indisponível
               após 10 segundos.
udpport 694 - Porta a ser usada na comunicação.
bcast eth1 - Interface a ser usada pelo heartbeat (broadcast)
auto_failback on - Opcional. Em caso de falha, quando o principal
                  retornar a rede, ele retomará os recursos.
node exatarncel - Obrigatório, hostname da máquina nodo conforme
                  descrito pelo comando "uname -s" (1º nó)
node exatarnce2 - Obrigatório, hostname da máquina nodo conforme
                  descrito pelo comando "uname -s" (2º nó)
-----

```

Uma vez configurado o arquivo ha.cf, foi salvo e fechado o arquivo. A seguir será verificada a configuração do arquivo haresources. Este arquivo especifica os serviços para o cluster e quem é o proprietário padrão. ***Deve-se atentar para uma dica importante: Este arquivo deve ser igual nos dois nodos.*** Para o nosso modelo inicial, foi considerado que o serviço configurado foi o apache (httpd). O endereço IP de serviço (IP virtual) para o cluster é obrigatório, e não deve ser configurado fora do arquivo haresources. Seguindo a configuração básica do sistema, o arquivo precisou apenas de uma linha:

```

# cd /etc/ha.d
# vi haresources
-----
exatarncel 164.85.176.246 httpd mon - Nome do servidor, IP virtual e
                                   serviços gerenciados pelo cluster
-----

```

O arquivo foi salvo e fechado. É importante ressaltar que httpd é o nome do script que inicia o serviço Web. Para o caso do mon, mais adiante será detalhado. O Heartbeat irá procurar por scripts com o mesmo nome nos seguintes lugares: /etc/ha.d/resource.d e /etc/rc.d/init.d.

O terceiro arquivo para configurar determina sua autenticação. Existem três métodos de autenticação: crc, md5 e sha1. Se o Heartbeat irá rodar sobre uma rede segura, com um cabo cruzado, como foi o caso do projeto implementado, pode ser usado o crc. Este é o método mais barato do ponto de vista de recursos. Se a rede é insegura, pode ser usado o md5. Porém para uma melhor autenticação sem se preocupar com recurso de CPU, pode ser usado sha1.

```
# cd /etc/ha.d
# vi authkeys
```

```
-----
auth 1
1 crc
-----
```

O arquivo foi salvo e fechado. Vale salientar que esses passos devem ser realizados nos dois nós integrantes do Cluster. Para finalizar foi iniciado o serviço heartbeat, cujo script deve estar em /etc/init.d, ou simplesmente efetuamos o comando usando a diretiva “service”.

```
# cd /etc/init.d
# ./heartbeat start      ou      service heartbeat start
```

Para se ter certeza que o serviço do Heartbeat está funcionando corretamente, no primeiro nó, deve aparecer uma saída para a interface virtual (eth0:0) quando efetuamos o comando “ifconfig -a”.

```
# ifconfig -a |grep eth0
-----
eth0      Protocol:Ethernet  Hardware Adress 00:E0:98:9E:AC:FA
eth0:0    Protocol:Ethernet  Hardware Adress 00:E0:98:9E:AC:FA
-----
```

5.3.3 Instalando e Configurando o Mon

O software escolhido para monitoração foi o Mon - *Service Monitoring Daemon*. Para a instalação e configuração do Mon, as estações devem ter o perl instalado e alguns módulos do perl. São eles:

- Time::Period
- Time::HiRes
- Convert::BER
- Mon::*

Eles podem ser baixados do site <http://www.kernel.org/software/mon/> . A instalação dos módulos é bastante simples e se seguiu conforme abaixo. Após o download, normalmente no formato tar.gz, descompactar todos os módulos. Para cada módulo, foram executados esses passos:

```
# gunzip <modulo>.gz
# tar -xvf <modulo>.tar
# cd <modulo>
# perl Makefile.pl
# make
# make test (se quiser ver se está tudo ok)
# make install
```

Efetuar o download da ferramenta mon (mon*.tar) do site. Ao descompactar, foi transferido o diretório da ferramenta para o do heartbeat, por questão de organização e facilidades no momento do gerenciamento.

```
# tar -xvf <mon*>.tar
# mv mon* mon
# mv mon /etc/ha.d
```

É importante ressaltar que o funcionamento do Mon é por monitores e alertas. Monitores verificam alguma coisa ou serviço, alertas executam ações quando o monitor detecta algum problema. Foi definido um diretório para o Mon. Foi escolhido o /etc/ha.d/mon/. Foi criado um arquivo mon.cf nesse diretório conforme abaixo.

```

# cd /etc/ha.d/mon
# vi mon.cf

-----

cfbasedir    = /etc/ha.d/mon/etc
alertdir     = /etc/ha.d/mon/alert.d
mondir       = /etc/ha.d/mon/mon.d
statedir     = /etc/ha.d/mon/state.d
logdir       = /var/log/
maxprocs     = 20
histlength   = 100
randstart    = 10s

authtype     = getpwnam

hostgroup web-fe 164.85.176.246
#hostgroup routers 164.85.176.254

watch web-fe
    service http
        interval 1m
        monitor http.monitor -p 8080 -u /portalAplicativos/Aplicativos
        allow_empty_group
        period wd {Mon-Sun}
            alert desativa-ha.alert -S "Serviço WEB do nó está fora!"
            alertevery 600s
            alertafter 2

#watch routers
#    service ping
#        interval 10s
#        monitor ping.monitor
#        allow_empty_group
#        period wd {Mon-Sun}
#            alert desativa-ha.alert -S "Interface de Rede Principal está \
#                                   fora" \
#            upalert mail.alert -S "Servidor Web está em backup!" \
#            alertevery 10s

```

Como pode ser verificado no arquivo, foi definido um script de alerta “desativa-ha.alert”. Esse script deve ser criado no diretório /etc/ha.d/mon/alert.d. Esse script será executado no momento que passado o tempo definido e o serviço monitoração não identificar que o serviço web esteja no ar. O conteúdo desse script deve ser conforme abaixo:

```

# cd /etc/ha.d/mon/alert.d
# vi desativa-ha.alert

-----

/etc/init.d/heartbeat stop

```

O arquivo foi salvo e fechado. Modificou-se a permissão do mesmo, tornando-o executável pelo comando:

```
# chmod 775 desativa-ha.alert
```

Para finalizar a configuração do mon, preparou-se os scripts de inicialização do serviço. Como exemplo, foi criado um arquivo mon-start, localizado no diretório /etc/ha.d/mon/.

```
# cd /etc/ha.d/mon/
# vi mon-start
-----
#!/bin/bash
MON_HOME=/etc/ha.d/mon

case "$1" in
    start)
        if [ -f $MON_HOME/mon.pid ]; then
            echo "mon already started"
            exit
        fi
        echo "Starting Mon"
        $MON_HOME/mon -c $MON_HOME/mon.cf -L $MON_HOME -P $MON_HOME/mon.pid &
        ;;
    stop)
        if [ -f $MON_HOME/mon.pid ]; then
            echo "Stopping Mon"
            kill -9 `cat $MON_HOME/mon.pid`
            rm -f $MON_HOME/mon.pid
        else
            echo "no server pid, server doesn't seem to run"
        fi
        ;;
    status)
        echo "doing good"
        ;;
    *)
        echo "Usage: $0 {start|stop|status|reload|restart}"
        exit 1
    esac
exit 0
-----
```

Vale ressaltar que o arquivo deve seguir ao padrão existente para scripts que funcionem pela inicialização do sistema operacional, bem como com a chamada pela diretiva "service". Na realidade, ele será inicializado pelo Heartbeat, e como o Heartbeat verifica os diretórios /etc/rc.d/init.d ou /etc/ha.d/resource pela existência dos scripts, devemos criar um link do mon para esses diretórios.

```
# cd /etc/ha.d/resource.d
# ln -s ../mon/mon-start mon
# ln -s /etc/rc.d/init.d/httpd httpd
```

Pronto, agora o Heartbeat sabe como inicializar e parar tanto o serviço web (httpd) quanto o serviço de monitoração (mon). Lembramos que esses passos foram realizados nos dois nós do cluster. Finalmente as duas estações foram reiniciadas. Inicialmente o servidor principal e quando estava completamente estável, reiniciamos o servidor secundário. Fizemos alguns testes e o sistema funcionou perfeitamente.

Toda a prática inicial foi executada com o serviço httpd, porém a versão final da aplicação exata foi desenvolvida para rodar em Jboss. Ao implementar a solução para o Jboss, encontrou-se uma operação indevida do cluster, pois não estava conseguindo “iniciar” nem “parar” o serviço. Após algumas experiências, conseguiu-se ajustar o sistema para que funcionasse perfeitamente. Segue abaixo a versão final do arquivo de inicialização do Jboss. Foi criada uma função para forçar a parada do serviço jboss. Verificamos que o sistema operacional não estava conseguindo efetuar a parada normalmente pela diretiva “service”. A partir dessa mudança, o sistema funcionou perfeitamente.

```
#!/bin/sh
#
# chkconfig: - 91 35
# description: Starts and stops the Samba smbd and nmbd daemons \
#               used to provide SMB network services.
#
# pidfile: /var/run/samba/smbd.pid
# pidfile: /var/run/samba/nmbd.pid
# config: /etc/samba/smb.conf
...
...
...

parajboss() {
    /bin/rm -f /tmp/jboss.log
    /bin/touch /tmp/jboss.log
    /bin/chmod 777 /tmp/jboss.log
    /bin/ps -ef | grep jboss | /bin/grep -v grep | /bin/grep -v start |
/bin/grep -v ResourceManager | /bin/awk '{print $2}' >/tmp/processos
    /bin/ps -ef | grep jboss | /bin/grep -v grep | /bin/grep -v start |
/bin/grep -v ResourceManager >>/tmp/jboss.log
    if [ -s /tmp/processos ]
    then
        /bin/cat /tmp/processos | while read pid
        do
            echo "matando processo $pid" >>/tmp/jboss.log
            /bin/kill -9 $pid >>/tmp/jboss.log
```

```

done
fi
}

start(){
    parajboss
    echo "CMD_START = $CMD_START" >>/tmp/jboss.log
    echo "JBoss_CONSOLE = ${JBoss_CONSOLE}" >>/tmp/jboss.log
    echo -n $"Iniciando Servidor de Aplicacoes JBoss: " >>/tmp/jboss.log
    cd $JBoss_HOME/bin >>/tmp/jboss.log
    if [ -z "$SUBIT" ]; then
        echo "SUBIT $SUBIT" >>/tmp/jboss.log
        #eval $CMD_START >${JBoss_CONSOLE} 2>&1 &
        eval $CMD_START >${JBoss_CONSOLE} 2>>/tmp/jboss.log
    >>/tmp/jboss.log&
    else
        echo "SUBIT $SUBIT" >>/tmp/jboss.log
        # $SUBIT "$CMD_START >${JBoss_CONSOLE} 2>&1 &"
        $SUBIT "$CMD_START >${JBoss_CONSOLE} 2>>/tmp/jboss.log
    >>/tmp/jboss.log &"
    fi
    RETVAL=$?
    echo
    # [ $RETVAL -eq 0 ]
    #     RETVAL=1
    return $RETVAL
    #     echo "RETVAL $RETVAL" >>/tmp/jboss.log
}

case "$1" in
start)
    start
    ;;
stop)
    parajboss
    echo "Jboss esta parado!"

    #if [ -z "$SUBIT" ]; then
    #     $CMD_STOP
    #else
    #     $SUBIT "$CMD_STOP"
    #fi
    ;;
restart)
    $0 stop
    $0 start
    ;;
*)
    echo "usage: $0 (start|stop|restart|help)"
esac

exit $?

```

A linha dos arquivos haresources dos dois nós foi modificada, de httpd para jboss, instalou-se o script do Jboss modificado nas duas estações, no diretório /etc/rc.d/init.d e as

estações foram “rebootadas”. Efetuou-se mais alguns testes e verificou-se que o sistema estava funcionando perfeitamente. Simulou-se por várias vezes a queda de operação do servidor primário, seja por falta de energia ou desligamento acidental, o servidor secundário, em todos as etapas, assumiu o serviço normalmente. Da mesma forma, simulou-se a parada do serviço no servidor principal. Após passado o tempo configurado para a monitoração, o servidor secundário novamente assumiu o serviço. Finalizada a etapa de testes, partiu-se para a implantação final.

Considerando o projeto implantado, pôde-se notar que a solução apresentada neste trabalho foi bastante vantajosa devido ao baixo custo, características mais próximas quando se trata de tecnologia de clustering. Esse custo reduzido foi proporcionado pela utilização de plataformas Intel-PC e ferramentas de código livre, tornando a solução proposta altamente competitiva em relação às soluções proprietárias vendidas por grandes empresas de infraestrutura.

Portanto, que, independente das especificidades das organizações, a tecnologia de computação em cluster pode servir-se em sua maioria de maneira como auxílio, instituições e centros de pesquisas que demandam por uma capacidade de computação intensiva bastante elevada e uma disponibilidade contínua. Clusters de computadores têm muitas aplicações e podem trazer grandes benefícios através da utilização correta dos recursos.

É desejável que o conteúdo desta etapa sirva como suporte a profissionais da área de redes, responsáveis pela implantação de clusters de alta disponibilidade, bem como alta performance e balanceamento de carga, além de estudantes em geral. Como trabalhos futuros, poderiam ser projetadas soluções de balanceamento de carga em conjunto com os servidores de alta disponibilidade implantados. Desta forma, conseguir-se-ia atingir um alto grau de disponibilidade, aliado a um baixo custo de investimento e escalabilidade.

6 CONCLUSÕES

A presente monografia permitiu conhecer a tecnologia de computação em cluster em suas diversas aplicações. Foram feitas inúmeras pesquisas em torno dessa tecnologia, análise de arquitetura, bem como análise de diversos softwares de gerenciamento e monitoração capazes de agir de forma automática na ocorrência de falha.

Desde o início pôde-se aprender conceitos relevantes. Através desse estudo, foi assimilado todo o funcionamento da estrutura de um cluster desde sua arquitetura, classificação e instalação, até seu funcionamento, gerenciamento e monitoramento, para enfim, testar e implementar uma aplicação prática sobre o assunto. Esses estudos foram acompanhados de uma investigação empírica, com a simulação de um ambiente utilizando as técnicas e ferramentas estudadas para um ambiente de produção em Alta Disponibilidade.

Considerando o projeto implantado, pôde-se notar que a solução apresentada neste trabalho foi particularmente vantajosa devido ao baixo custo, característica essa percebida quando do uso da tecnologia de clustering. Esse custo acessível foi proporcionado pela utilização de plataformas Intel-PC e ferramentas de código livre, tornando a solução proposta altamente competitiva em relação às soluções proprietárias vendidas por grandes empresas do ramo da computação.

Percebe-se que, independente das especificidades e do porte das organizações, a tecnologia de computação em cluster pode auxiliá-las em seu dia-a-dia da mesma forma como auxiliam instituições e centros de pesquisas que demandam por uma capacidade de computação intensiva bastante elevada e uma disponibilidade contínua. ***Clusters de computadores têm muitas aplicações e podem trazer grandes benefícios através da utilização correta dos mesmos.***

É desejável que o conteúdo deste estudo sirva como suporte a profissionais da área de redes, responsáveis pela implantação de clusters de alta disponibilidade, bem como alta performance e balanceamento de carga, além de estudantes em geral. Como trabalhos futuros, poderiam ser projetadas soluções de balanceamento de carga em conjunto com os servidores de alta disponibilidade implementados. Desta forma, conseguir-se-ia atingir um alto grau de disponibilidade, aliado a um ganho de desempenho e escalabilidade.

REFERÊNCIAS

- CLUSTER linux. Disponível em: <<http://www.acol.com.br/websolution/cluster/index.shtml>>. Acesso em: mar. 2007.
- ALTA Disponibilidade em Servidores Linux.. Disponível em: <http://labbi.uesc.br/labbi/aportilas/revista_do_linux/006/alta.html>. Acesso em: fev. 2007.
- ALECRIM, Emerson. **Cluster: principais conceitos**. Disponível em: <<http://www.infowester.com/cluster.php>>. Acesso em: mar. 2007.
- BALANCEAMENTO de carga. Disponível em: <http://pt.wikipedia.org/wiki/Balanceamento_de_carga>. Acesso em: fev. 2007.
- BEOWULF Project. Disponível em: <<http://www.beowulf.org>> Acesso em: mar. 2007.
- BOOKMAN, Charles. **Agrupamento de Computadores em Linux**. Rio de Janeiro. Ciência Moderna, 2003.
- CLUSTER DSEE. Disponível em: <<http://www.dsee.fee.unicamp.br/~asada/beowulf/descricao.html>>. Acesso em: abr. 2007.
- CONSTRUINDO um Cluster de OpenMosix com Slackware 11.0. Disponível em: <<http://www.slackwarezine.com.br>>. Acesso em fev. 2007.
- GROUWING Up – ways to make your linux server bigger. Disponível em: <<http://www.linux.org.mt/node/16>>. Acesso em: maio 2007.
- HEARTBEAT Project. Disponível em: <<http://www.linux-ha.org/Heartbeat>>. Acesso em: abr. 2007.
- KOOPER, Karl. **Linux Enterprise Cluster: Build a Highly Available Cluster With Commodity Hardware and Free Software**. No Starch Press, 2005.
- LIBRANDI, Rodrigo. **Hardware: Montagem de Micros**. Disponível em: <<http://www.juliobattisti.com.br/tutoriais/rodrigolibrandi/historiacomputacao001.asp>>. Acesso em: abr. 2007.
- MICROTIK. Disponível em: <<http://under-linux.org/tag/microtik>>. Acesso em: mar. 2007.
- MORIMOTO, Carlos E. **Brincando de Cluster**. Disponível em: <<http://www.guiadohardware.net/artigos/260>>. Acesso em: mar. 2007.
- NÓBREGA, Gustavo Franklin. **Alta Disponibilidade**. Disponível em: <<http://www.firewalls.com.br/files/altadispMysql.pdf>>. Acesso em: maio 2007.
- OLIVEIRA, CAF. **Cluster Beowulf: Uma Solução de Baixo Custo**. Disponível em: <<http://www.frb.br/ciente/Imprensa/Info/I.4.Oliveira,CAF.ClusterBeowulf.pdf>>. Acesso em: fev. 2007.

OPEN Mosix Project. Disponível em: <<http://www.openmosix.sourceforge.net>>. Acesso em: mar. 2007.

PITANGA, Marcos. **Computação em Cluster**: O estado da Arte da Computação. Rio de Janeiro: Brasport, 2003.

_____. **Construindo Super Computadores com Linux**. 2. ed. Rio de Janeiro: Brasport, 2004.

SIMÕES, Eduardo Issi. **Alta Disponibilidade em Servidores**. Disponível em: <http://www.eee.ufg.br/cepf/pff/2002/pf2002_04.pdf>. Acesso em: maio 2007.

SOMMER, Renato. **Computação em Cluster**. Disponível em: <www.inf.ufrgs.br/~asc/sodr/pdf/SommerSODR.pdf>. Acesso em: abr. 2007

SUPERCOMPUTADORES. Disponível em: <<http://pt.wikipedia.org/wiki/Supercomputador>>. Acesso em: fev. 2007.

TROCKI, Jim. **Mon Project**. Disponível em: <<http://www.kernel.org/software/mon/>>. Acesso em: maio 2007.

WOUDA, Peter. **Over de fotograaf**. Disponível em: <<http://www.ee.ryerson.ca:8080/~elf/abacus/portugues/intro.html>>. Acesso em: fev. 2007.

ZELENOVSKY, Ricardo. **Processadores Para o Próximo Milênio - Parte 2**. Disponível em: <<http://www.clubedohardware.com.br/printpage/993>>. Acesso em: maio 2007.

ZEM, José Luis. **Uso de Cluster de Computadores em Ambiente Corporativo**. Disponível em: <<http://www.unibratec.com.br/anaisdecongresso/diretorio/unimep+jlrevisado.doc>>. Acesso em: maio de 2007.