Universidade de Caxias do Sul
Centro de Ciências Exatas e Tecnologia
Departamento de Informática
Curso de Bacharelado em Ciência da Computação

Clustering com Software Livre para Implementação de Alta Disponibilidade

por
Mauricio Borne

Professor Dr. Alexandre Ribeiro
Orientador

Caxias do Sul, julho de 2001

Índice

Lista de Figuras

Resumo

Este trabalho apresenta soluções para o desenvolvimento de uma arquitetura de Cluster utilizando Software Livre. Estas soluções utilizam como base pacotes para o Linux como o Heartbeat, o Mon, o DRBD e o ReiserFS. Estes pacotes permitem compor clusters com recursos de Alta Disponibilidade.

As características de Alta Disponibilidade, como a detecção e o mascaramento de falhas, e de Clusters, como seus componentes e arquiteturas, são analisadas nas seções iniciais do trabalho e podem ser conseguidas em soluções de clusters baseadas na utilização de Software Livre.

Também são apresentadas as vantagens do uso do Software Livre, como a liberdade de aperfeiçoar e adaptar o software, para implementar um Cluster. São apresentadas soluções de cluster que utilizam o Linux, como a solução da Conectiva, o projeto Linux Virtual Server e o projeto Beowulf.

Abstract

This work presents solutions for the development of an architecture of Cluster using Free Software. These solutions use as a base, packages for the Linux as the Heartbeat, the DRBD, the Mon and the ReiserFS. These packages allow to compose clusters with features of High Availability.

The features of High Availability, as the detention and the disguise of feeds, and Clusters, as its components and architectures, are analyzed in this work in the initial sections and can be obtained in cluster solutions based in the use of Free Software.

Further one, the advantages of the use of Free Software are presented, as the freedom to improve and to customize software, to implement a Cluster. Cluster solutions using Linux are presented, as the solution of the Conectiva, the Linux Virtual Server Project and the Beowulf Project.

1. Introdução

Com a utilização em massa dos recursos de informática nas empresas e instituições, é necessário que as informações e aplicações fornecidas por esses sistemas baseados em computadores, estejam disponíveis vinte e quatro horas por dia nos sete dias da semana. Essas informações podem ser disponibilizadas através de uma rede de computadores local ou uma rede metropolitana, e através da Internet. Desta forma, os servidores de rede ou os servidores de *web* devem estar sempre prontos para atender solicitações de informações e serviços.

Existem várias técnicas e soluções para garantir a disponibilidade das informações em uma rede de computadores durante o maior tempo possível. A Alta Disponibilidade é a área responsável pelas características do sistema que garantem a disponibilidade das informações mesmo quando ocorrem problemas com esse sistema. A utilização de Clusters de computadores é uma técnica para implementar Alta Disponibilidade que, nos últimos anos, tem sido adotada nas situações que exigem disponibilidade total dos dados e acesso rápido a eles.

O conceito básico nos Clusters é utilizar vários servidores trabalhando juntos, acessando as mesmas informações, executando tarefas para atender com eficiência e rapidez às solicitações das estações clientes. E em caso de falha de um servidor, os outros servidores devem atender os clientes que estavam sendo servidos pela máquina que falhou. Esse processo deve ocorrer de forma transparente para esses clientes. Os Clusters utilizam características do *hardware* e do *software* para fornecer Alta Disponibilidade, e aqui entra um fator essencial: o custo.

Um Cluster de computadores pode ser implementado com a utilização de computadores simples, com *hardware* padrão, e com a utilização de Software Livre. Com isso, o custo para implementar um cluster com o objetivo de obter Alta Disponibilidade pode ser acessível. Muitas vezes, ao utilizar soluções de cluster comerciais disponíveis no mercado, o custo com o *software* pode superar os custos com o *hardware*. Porém, com o crescimento da popularidade do Software Livre, os custos com o *software* nesse tipo de solução são extremamente baixos.

Neste trabalho, são apresentados os conceitos e aspectos relacionados à Alta Disponibilidade, Clusters e Software Livre, e algumas soluções baseadas na utilização de Software Livre para implementação de clusters.

Este trabalho está organizado da seguinte forma. No capítulo 2, são apresentadas as principais definições de Alta Disponibilidade e os principais fatores relacionados ao assunto. O capítulo 3 mostra um estudo sobre Clusters onde são apresentados os principais componentes e a arquitetura típica de um cluster, e também as arquiteturas de cluster para Alta Disponibilidade. O Software Livre é abordado no capítulo 4, onde são apresentadas as suas características, é apresentado o Projeto GNU e as principais características do Linux. No capítulo 5 são mostradas algumas soluções de cluster baseadas em Linux e no capítulo 6 é apresentado um breve estudo de caso utilizando uma das soluções vistas no capítulo anterior.

2. Alta Disponibilidade

As informações armazenadas nos computadores são vitais para o funcionamento de empresas e instituições, por isso esses dados devem estar disponíveis durante o maior tempo possível. A Alta Disponibilidade tornou-se uma área muito importante devido ao atual crescimento da utilização dos computadores e da grande dependência que esse crescimento gerou. Alta Disponibilidade (*High Availability - HA*) é a área que trata das soluções para garantir o funcionamento contínuo e satisfatório de um sistema computacional no caso da ocorrência de algum problema. A Alta Disponibilidade é uma característica no sistema responsável pela detecção, correção e mascaramento de falhas que podem ocorrer nesse sistema, sendo que essas tarefas são realizadas de forma transparente aos usuários deste sistema.

Segundo *Rajkumar Buyya* do IEEETFC (IEEE Computer Society Task Force on Cluster Computing), Alta Disponibilidade refere-se à disponibilidade dos recursos de um sistema baseado em computadores no momento em que um componente do sistema venha a falhar. Esta falha pode ocorrer de várias maneiras e várias soluções podem ser aplicadas, como um *hardware* redundante (que é uma solução cara), ou soluções de *software* utilizando *hardware* com componentes simples [IEE 00].

O papel da Alta Disponibilidade é mascarar uma falha quando ela ocorre, a fim de manter o funcionamento do sistema. Isto é realizado através de mudanças internas neste sistema para manter os recursos disponíveis, de forma que os usuários externos não saibam dessas mudanças.

2.1 Conceitos relacionados com Alta Disponibilidade

Para uma melhor compreensão de tudo o que está envolvido com a Alta Disponibilidade, é necessário esclarecer alguns termos que são apresentados a seguir [RES 96]:

Sistema

O sistema engloba todos os componentes necessários em um ambiente onde o ponto central é a utilização de computadores para desenvolver determinadas tarefas. Os principais itens que fazem parte do sistema são:

- o *hardware* e *software* dos servidores;
- os dispositivos de armazenamento dos dados destes servidores;
- o *hardware* e o *software* das estações de trabalho;
- os equipamentos e os meios físicos de comunicação de dados;
- os equipamentos e os meios físicos da rede elétrica;
- os administradores

Estes são apenas alguns dos componentes de um sistema baseado em computadores. Este trabalho se concentra nas técnicas e soluções de Alta Disponibilidade especificamente para os servidores do sistema, mas isto não significa que outros componentes não sejam importantes. Pode-se dizer que os servidores são o centro de sistemas de computadores, pois, na maioria dos sistemas, são neles que estão armazenadas as informações e aplicações essenciais das empresas, e esses recursos precisam estar sempre disponíveis para o bom funcionamento dos negócios das mesmas.

Falhas

Uma falha no sistema é definida como um desvio do funcionamento correto do mesmo. Se o sistema não está fornecendo os serviços e recursos que ele tem a função de fornecer, algum comportamento anormal está ocorrendo, e isto é considerado uma falha. O sistema funciona com base em procedimentos, no *software* e no *hardware*. Uma falha pode ocorrer em qualquer um desses componentes. As falhas não são visíveis pelos usuários que interagem com o sistema, mas elas podem gerar erros que são percebidos pelos usuários.

Erros

Quando ocorre uma falha no sistema, esta geralmente gera um erro. Pode-se dizer que o erro é uma manifestação visível da falha. Um erro pode ser percebido pelas pessoas que interagem com o sistema, mas ele não identifica a causa da falha.

Defeitos

Os defeitos são considerados os efeitos, visíveis pelos usuários do sistema, dos desvios de comportamento que são causados pelas falhas e erros.

Failover

Quando ocorre uma falha em um recurso do sistema com Alta Disponibilidade, e o mesmo detecta essa falha e executa uma ação para que outro componente do sistema assuma a tarefa de fornecer este recurso de forma que os usuários não percebam que ocorreu algum problema, esse procedimento é chamado de *failover*. Geralmente, o sistema faz com que outro computador assuma esta tarefa.

Failback

É o procedimento contrário do *Failover*. O *Failback* ocorre quando um componente que estava com falhas, que agora não apresenta mais as mesmas, volta ao seu estado normal de funcionamento. O sistema, de forma transparente, deve executar esse processo.

Ponto Único de Falha (*Single Point of Failure - SPOF*)

São os pontos ou componentes do sistema que não possuem tratamento para garantir Alta Disponibilidade. Esses componentes podem ser de *hardware* ou *software*, e quando ocorre uma falha nestes componentes todo o sistema é comprometido.

Disponibilidade Básica

Um sistema com Disponibilidade Básica é projetado e implementado com os componentes considerados suficientes a nível de *hardware, software* e procedimentos para atender determinadas funções que foram requeridas. Neste sistema não é previsto nenhum tipo de solução para evitar a parada das atividades do mesmo por causa de algum problema, ou seja, mascarar ou evitar qualquer tipo de falha que possa ocorrer.

Disponibilidade Contínua

A Disponibilidade Contínua implica em um sistema sem paradas nos seus serviços. É um sistema que tem um alto nível de disponibilidade, utiliza todos recursos possíveis de *software* e *hardware* para fornecer tal disponibilidade. Alta Disponibilidade não implica em Disponibilidade Contínua.

2.2 A Alta Disponibilidade

Um sistema com Alta Disponibilidade é um sistema projetado e desenvolvido com componentes suficientes para satisfazer a execução de determinadas tarefas, e também com redundância suficiente em componentes de *software*, *hardware* e procedimentos, para garantir a disponibilidade em caso de ocorrência de falhas [RES 96].

A Alta Disponibilidade pode ser garantida através de diferentes formas. As soluções são baseadas em dois pontos distintos: *hardware* e *software*. Existem soluções que utilizam um *hardware* especializado para criação de redundância e soluções baseadas em *software* que utilizam equipamento padrão, sem modificações [AND 01].

As soluções baseadas em *hardware* oferecem um maior grau de disponibilidade e segurança, porém são soluções extremamente caras. Por isso, as soluções baseadas em *software* têm se tornado mais populares e os desenvolvedores de sistemas computacionais têm oferecido estas soluções em vários níveis. Uma boa solução é mesclar essas duas idéias para garantir Alta Disponibilidade por um preço acessível [AND 01].

Nas técnicas baseadas em *hardware*, redundância é a idéia básica. A proposta é garantir a disponibilidade utilizando servidores com componentes redundantes e com tolerância à falhas. Quando um componente falhar, deve haver outro que esteja devidamente preparado para assumir a sua função imediatamente e de forma transparente. Os componentes podem ser discos, processadores, placas de rede e fontes de alimentação de energia, entre outros. O RAID de discos é a técnica baseada em *hardware* mais utilizada hoje em dia. Essa técnica utiliza discos duplicados e espelhados para garantir a alta disponibilidade dos dados [AND 01].

Nas soluções baseadas em *software*, o principal objetivo é garantir a disponibilidade dos serviços sem a necessidade de utilizar um *hardware* especial. Devido a isso, a maioria das técnicas de *software* para alta disponibilidade baseiam-se na utilização de computadores com *hardware* padrão.

Uma boa solução para garantir a Alta Disponibilidade é a utilização de Clusters de servidores, onde são utilizadas soluções de *hardware* e *software* (Esse assunto será abordado no capítulo 3).

2.3 RAID de Discos

O RAID (“Redundant Array of Inexpensive Disks”) é o sistema baseado em *hardware* mais utilizado para garantir a Alta Disponibilidade dos dados em seus dispositivos de armazenamento. O objetivo do RAID é fazer com que os discos deixem de ser pontos únicos de falha. Conforme o nível utilizado, este sistema pode aumentar a performance ao acesso aos dados e garantir a disponibilidade dos mesmos. Existem cinco níveis de RAID:

2.3.1 Nível 0

Também chamado de “disk-striping”. Este método consiste em dividir os dados em blocos e então distribuir estes blocos entre os discos do array. Os dados são subdivididos em segmentos consecutivos que são escritos de forma seqüencial através de blocos formados pelos discos, conforme a figura 1 [TEI 00]. O RAID nível 0 oferece uma alta performance, porém não oferece nenhuma segurança em relação à disponibilidade dos dados. Se um disco falhar, todos dados ficam indisponíveis.

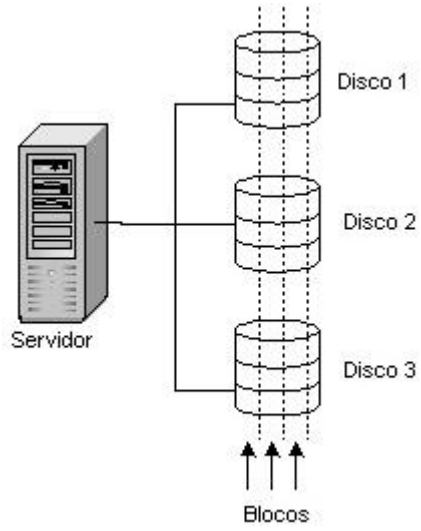

Figura 1: Raid Nível 0

2.3.2 Nível 1

Também conhecido como “espelhamento”. Aqui cada disco tem seu “espelho” sendo que toda vez que um dado é gravado no disco, o mesmo dado é

gravado em seu “espelho”. O Nível 1 é o mais caro dos RAIDs pois necessita o dobro de capacidade de armazenamento. Se ocorrer uma falha em um dos discos do espelhamento, as operações de leitura e gravação dos dados continuam somente no disco que restou até que seja reposto um outro disco, para que o espelhamento possa ser refeito [TEI 00]. Nenhum dado é perdido e o sistema continua em funcionamento em caso de falhas, garantindo a disponibilidade das informações.

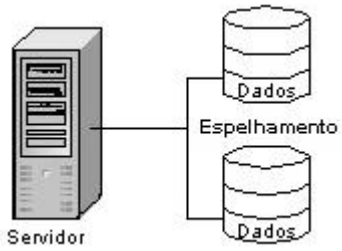

Figura 2: Raid Nível 1

2.3.3 Nível 2

Está técnica trabalha com redundância na forma de códigos de correção de erros (Error Correcting Code - ECC). Seu funcionamento é semelhante ao nível 0 mas ao invés de gravar bloco a bloco, os dados são gravados byte a byte ou até mesmo bit a bit. Este nível necessita de vários discos do array para armazenar os códigos de correção de erros. Em caso de falhas, o RAID 2 pode garantir a consistência dos dados buscando as informações nos códigos de controle de erros. As operações de gravação e leitura ocorrem em todas as unidades [TEI 00].

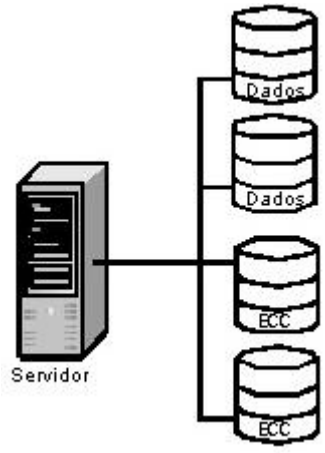

Figura 3: Raid Nível 2

2.3.4 Nível 3

É uma simplificação do Nível 2, pois utiliza somente um disco para armazenar os códigos de correção de erros. Sua grande vantagem é o fato de necessitarmos somente um disco a mais para os códigos de erro. As operações de gravação e leitura ocorrem em todas unidades [TEI 00].

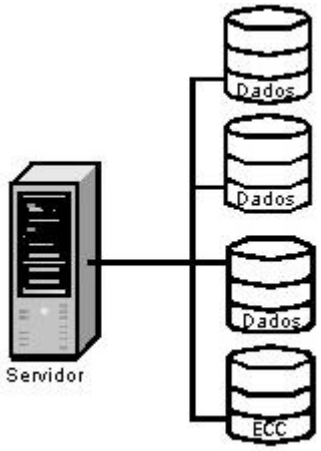

Figura 4: Raid Nível 3

2.3.5 Nível 4

No RAID 4, são utilizadas informações sobre paridade para a recuperação de dados em caso de falhas, essas informações ficam armazenadas em um disco dedicado. Para toda operação de gravação realizada nos discos, é necessário atualizar a unidade de paridade. A leitura pode ocorrer simultaneamente em todas os discos [TEI 00].

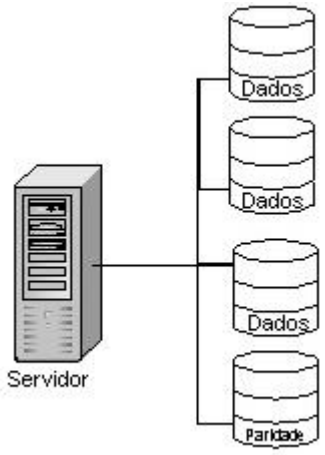

Figura 5: Raid Nível 4

2.3.6 Nível 5

É o nível mais popular do RAID. Utiliza o mesmo conceito de código de erros, mas ao invés de utilizar um ou vários discos somente para armazenar estes códigos, faz uso de todos os discos do sistema. Como todos códigos estão espalhados por todos discos, é possível acessar vários discos simultaneamente. Se um dos discos falhar, é possível recuperar seus dados através dos códigos de erro dos outros discos, e o acesso aos dados continua sem necessidade de interrupções [TEI 00].

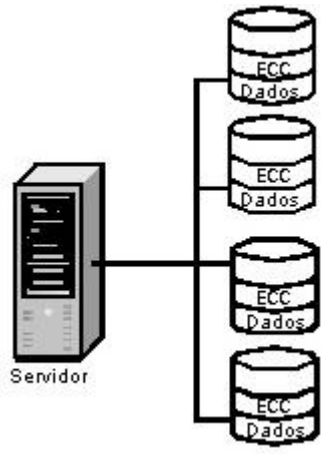

Figura 6: Raid Nível 5

2.4 Alta Disponibilidade com Clusters

Uma boa solução que vem sendo utilizada para garantir a Alta Disponibilidade em sistemas que rodam aplicações críticas é a utilização de Clusters. Um Cluster é um tipo de sistema de processamento paralelo ou distribuído, o qual consiste em uma coleção de computadores interconectados trabalhando como um só [BUY 99].

Os Clusters estão sendo utilizados em várias áreas, principalmente em sistemas com aplicações críticas. Uma grande parte dessas aplicações não são tolerantes a falhas no sistema, e em caso de uma falha isso pode resultar em perdas de serviços e dados. As aplicações requerem que o sistema no qual elas são executadas tenha o mínimo de paradas possíveis, o que torna a utilização de Clusters uma boa estratégia para garantir a Alta Disponibilidade [IEE 00].

No capítulo 3, os Clusters serão descritos mais detalhadamente. Serão apresentadas suas características, arquitetura básica e arquiteturas para Alta Disponibilidade.

3. Clusters

O propósito maior na criação dos Clusters foi o desejo de obter um sistema com a mais alta performance possível, e com essa idéia juntou-se o conceito de Alta Disponibilidade utilizando vários computadores trabalhando junto. Juntamente com os Clusters, surgiram na última década outros sistemas de computação paralela com o objetivo de obter um processamento de alta performance. Os fundamentos desses sistemas são baseados em como seus processadores, memória e dispositivos de conexão estão posicionados. Os sistemas mais comuns são [BUY 99]:

Processamento Paralelo Pesado (Massively Parallel Processor - MPP)

Um sistema MPP é geralmente um grande sistema de processamento paralelo com uma arquitetura que não “compartilha nada”. Esta arquitetura consiste de centenas de elementos de processamento (nodos), os quais são interconectados através de uma rede de alta velocidade. Cada nodo consiste geralmente de uma memória principal e de um ou mais processadores. Nodos especiais podem ter outros periféricos como discos ou dispositivos de backup. Cada nodo executa separadamente uma cópia do sistema operacional.

Multiprocessamento Simétrico (Symmetric Multiprocessors - SMP)

Um sistema SMP possui de 2 a 64 processadores e tem uma arquitetura que “compartilha tudo”. Neste sistema, todos processadores compartilham todos recursos globais disponíveis, como memória e sistemas de entrada/saída. Uma única cópia do sistema operacional é executada neste tipo de sistema.

Acesso a Memória Não-Uniforme com Cache Coerente (Cache-Coherent Nouniform Memory Access - CC-NUMA)

Um CC-NUMA é um sistema multiprocessado com uma arquitetura de acesso à memória não-uniforme com cache coerente. Como no SMP, neste sistema cada processador tem uma visão global de toda a memória. O nome deste sistema vem do número não-uniforme de acessos realizados nos pontos próximos e nos pontos remotos da memória.

Sistemas Distribuídos

Estes sistemas podem ser considerados redes convencionais de computadores independentes. Eles têm imagens de múltiplos sistemas, cada nodo podendo executar seu próprio sistema operacional, e as máquinas individuais em um sistema distribuído podem ser combinações de sistemas MPP, SMP, clusters e computadores individuais.

Pode-se dizer que um cluster é uma coleção de estações de trabalho ou PC's que estão inteconectados através de uma tecnologia de rede de computadores. Para computação paralela, um cluster consistirá de estações de trabalho ou PC's de alto desempenho interconectados por uma rede de alta velocidade [BUY 99].

Cada um dos computadores que são nodos de um cluster, pode ser um sistema simples ou multiprocessado com memória, recursos de entrada/saída e um sistema operacional [BUY 99].

Uma característica importante dos clusters é que cada nodo executa uma cópia separada do sistema operacional. Esta é uma das principais diferenças em relação a outros sistemas nos quais uma cópia do sistema operacional controla todos nodos ao mesmo tempo [BIR 96].

Outra característica dos Clusters é o que seus componentes podem ser simples, PC's e estações de trabalho com *hardware* padrão, utilizando esses componentes para montar uma arquitetura com custos baixos. São utilizados vários computadores com arquitetura simples ao invés de servidores com arquiteturas especiais e componentes com redundância a falhas [BIR 96].

Neste trabalho são estudados os Clusters implementados para garantir Alta Disponibilidade em um sistema de computadores. As três principais técnicas utilizadas em Clusters para garantir essa Alta Disponibilidade são [AND 01]:

Ip Takeover: quando um servidor ou um certo recurso tiver uma falha, um outro servidor assume o endereço IP da máquina com problemas.

Espelhamento via Rede: é um tipo de RAID 1 via rede, ou seja, um disco de um servidor é espelhado em outro servidor através da rede, garantindo dados atuais se o primeiro servidor falhar.

DNS Rotativo: nesta técnica é atribuído no servidor de nomes mais de um ip para um mesmo host, assim o servidor de nomes vai responder às requisições de forma circular. Esse servidor de nomes monitora se a máquina está operante antes de fornecer seu endereço.

Com a utilização de Cluster, todos os pontos onde podem ocorrer falhas em um sistema são observados, pois existe uma redundância completa. A falha nos computadores do Cluster pode ser de memória, de processador, de disco ou da fonte de alimentação, quando isso ocorrer os outros membros do cluster vão garantir a disponibilidade das informações e serviços sem que os usuários finais percebam que houve a falha. Em outras técnicas para garantir Alta Disponibilidade, como por exemplo à utilização de RAID, apenas um ponto passível de falha é observado (os discos).

3.1 Arquitetura Típica

Os computadores nodos de um Cluster podem estar em um único gabinete ou podem estar fisicamente separados e conectados via rede local. A seguir tem-se os principais componentes de uma arquitetura de Cluster [BUY 99]:

Múltiplos Computadores de Alta Performance: PC’s ou Estações de trabalho.

Sistema Operacional: sistemas que suportam multitarefas, multithreading e aplicações de rede.

Rede de alta velocidade: hub’s e switches que suportam redes de alta performance como FastEthernet, Gigabit Ethernet, ATM e Myrinet.

Placas de Rede: placas de rede que suportam redes de computadores de alta velocidade.

Serviços e protocolos de comunicação rápidos: serviços para transportar dados administrativos e dados de usuários.

Cluster *Middleware*: Imagem Única do Sistema (Single System Image - SSI) e Sistema com Infra-estrutura de Disponibilidade .

Ferramentas e Ambiente de Programação Paralela: compiladores, threads, máquinas virtuais paralelas e bibliotecas de troca de mensagens.

Aplicações: as aplicações em um cluster podem ser seqüenciais ou paralelas.

As placas de rede agem como um processador de comunicação e sua função é transmitir e receber pacotes de dados entre os nodos dos Clusters através de hub's ou switches.

O *software* de comunicação deve fornecer comunicação de dados rápida e confiável entre os nodos do Cluster e com o mundo externo.

Os nodos do Cluster podem trabalhar coletivamente, como um sistema de recursos integrados, ou podem funcionar como computadores individuais. O *cluster middleware* é responsável em criar uma ilusão de uma imagem de um sistema único (imagem única do sistema) e disponível [BUY 99].

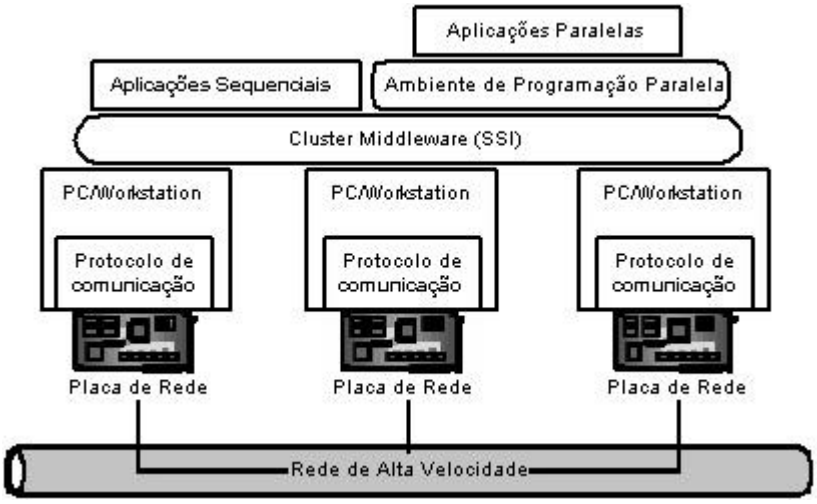

Figura 7: Arquitetura Típica do Cluster

3.2 Cluster Middleware [BUY 99]

O *Middleware* num Cluster é a camada que fica entre o nível do sistema operacional e o nível do ambiente do usuário. Ele é responsável em fazer que uma

coleção de computadores interligados seja vista como um único recurso, ou seja, imagem única do sistema (SSI).

O *Middleware* é composto essencialmente de duas subcamadas de infra-estruturas de *software*: a Infraestrutura SSI e a Infraestrutura de Disponibilidade do Sistema.

A infra-estrutura SSI agrupa os sistemas operacionais de todos os nodos a fim de fornecer um acesso único a um recurso. A Infraestrutura de Disponibilidade do Sistema é responsável por serviços como *failover* automático e recuperação de um defeito entre todos nodos do cluster.

3.2.1 Níveis da Imagem Única do Sistema

Uma SSI é a ilusão, criada pelo *hardware* e *software* do cluster, que uma coleção de recursos é vista como um único recurso poderoso. A camada de *Hardware*, o Kernel do Sistema Operacional e a camada de Aplicações e Subsistemas Específicos são responsáveis em criar a SSI e os serviços de disponibilidade do sistema.

No nível de *hardware*, Canais de Memória Digital e *hardware* com memória compartilhada distribuída (Distributed Shared Memory – DSM) fornecem SSI. Isto faz com que os usuários vejam o Cluster como um sistema de memória compartilhada.

Os sistemas operacionais do Cluster permitem uma execução eficiente de aplicações paralelas em um ambiente compartilhado com aplicações seqüenciais. O objetivo é unir os recursos em um cluster para fornecer melhor desempenho para esses dois tipos de aplicação. O sistema operacional deve identificar os recursos disponíveis no sistema (processadores, memória, rede e etc) e oferecer acesso global a eles. Os kernels dos sistemas operacionais que suportam SSI devem ter essas características sem que seja necessário adicionar componentes ou comandos. Uma SSI completa em um cluster permite que todos os recursos físicos e recursos do kernel sejam visíveis e acessíveis de todos os nodos do sistema. SSI completa pode ser chamada como “underware”, ou seja, SSI no nível de sistema operacional. Assim, o kernel de cada nodo apresenta a mesma visão de todas interfaces do kernel em todos nodos.

Aplicações e subsistemas podem suportar uma SSI quando os componentes são apresentadas ao usuários como uma única aplicação. O nível de aplicação numa SSI é o mais alto e de muita importância, pois é este que é visto pelos usuários finais. Uma ferramenta de administração do cluster oferece um ponto único de gerenciamento e

controle dos serviços da SSI. Esta ferramenta pode ser construída com uma interface para gerenciar e monitorar todo cluster ou para ver os nodos individualmente.

3.2.2 Metas do Middleware

A SSI traz vários benefícios para a arquitetura do Cluster, o principal é de esconder dos usuários a complexidade do sistema. Os usuários acessam os recursos e as aplicações, e não sabem onde eles estão localizados ou sendo executados. O suporte a SSI pode existir em diferentes níveis no sistema.

Um dos objetivos da camada do Middleware é garantir uma completa transparência para o usuário. Essa camada permite que o usuário trabalhe com o cluster facilmente e com eficiência sem conhecer a sua arquitetura. Outro objetivo do Middleware é ter boa escalabilidade. Os clusters podem ser expandidos com facilidade e melhorar a sua performance. Outra meta do Middleware é ter seus serviços com alta disponibilidade por todo tempo. Se um nodo ou serviço falhar, o recurso deve ser recuperado sem afetar as aplicações dos usuários.

3.2.3 Serviços Principais do Middleware

Serviços de Suporte a SSI:

- **Ponto de Entrada Único:** o usuário conecta no cluster como em um sistema simples e não precisa indicar qual nodo deve ser conectado.

- **Hierarquia de Arquivos Única**: o usuário vê um sistema de arquivos único, com arquivos e diretórios sob o mesmo diretório raiz.

- **Ponto de Gerenciamento e Controle Único:** o cluster inteiro pode ser monitorado de uma única janela usando uma única ferramenta gráfica.

- **Rede Virtual Única:** qualquer nodo pode acessar qualquer conexão de rede por todo o cluster mesmo se a rede não está fisicamente conectada em todos os nodos.

- **Espaço de Memória Único:** ilusão de memória compartilhada sobre memórias associadas com nodos do cluster.

- **Sistema de Gerenciamento de Jobs Único:** um usuário pode executar um trabalho de qualquer nodo usando um mecanismo transparente para executar.

- **Interface de Usuário Única:** o usuário pode utilizar o cluster através de uma única interface.

Suporte para Disponibilidade

- **Entrada / Saída Única:** permite que qualquer nodo execute operações de entrada/saída locais ou em discos e periféricos remotos.

- **Espaço de Processamento Único:** um processo tem único *id* (identificador) para todo o cluster.

- **Pontos de Verificação e Migração de Processos:** os pontos de verificação salvam o estado dos processos entre o seu processamento. Quando um nodo falha, os processos deste podem ser reiniciados em outro nodo. O serviço de migração de processos é utilizado para fazer balanceamento da carga entre os nodos do cluster.

3.3 Arquiteturas de Cluster para Alta Disponibilidade

Clusters para Alta disponibilidade podem ser implementados de várias maneiras. O número de nodos, o sistema de conexão com o dispositivo de armazenamento e entre os nodos, os serviços compartilhados, entre outros, são os fatores que determinam como será a arquitetura do cluster.

A seguir, temos as principais arquiteturas de clusters para Alta Disponibilidade [BUY 99].

3.3.1 Arquitetura Share-Nothing

Nesta arquitetura cada nodo do cluster tem sua própria memória e seu próprio dispositivo de armazenamento, ou seja, nada é compartilhado entre os nodos. Neste modelo, os nodos podem acessar dispositivos ou recursos comuns, esses recursos são alocados e gerenciados por um único sistema de cada vez, isto evita a complexidade de utilizar Gerenciadores de Bloqueio Distribuído (*Distributed Lock Managers - DLM*). Os DLMs controlam o acesso a recursos distribuídos.

Os dispositivos de armazenamento são espelhados entre os nodos e para diminuir a carga no tráfego na rede, pode-se utilizar uma segunda placa de rede ou outro dispositivo de conexão para separar a carga causada pelo espelhamento. Esta arquitetura elimina o sistema de armazenamento como ponto único de falha.

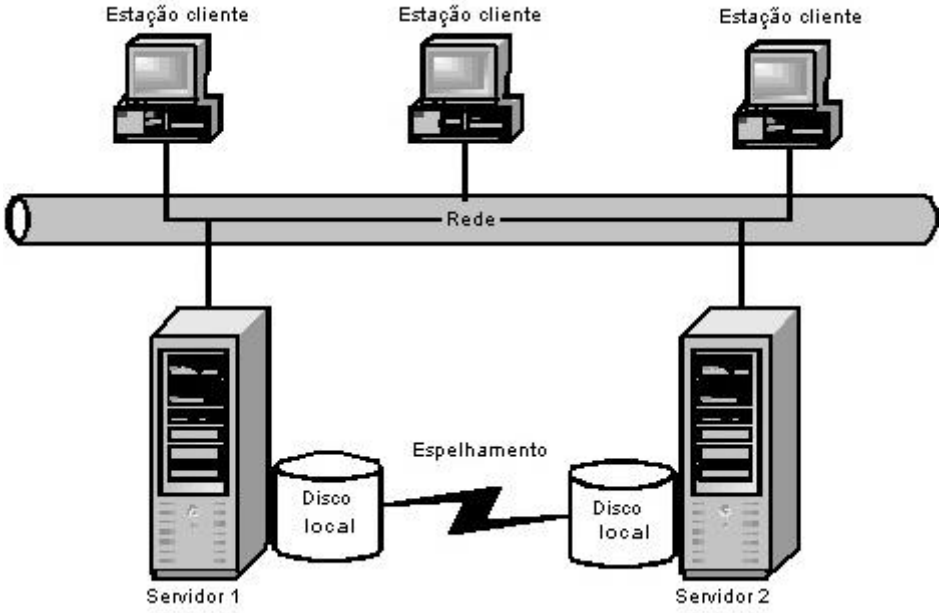

Figura 8: Arquitetura Share-Nothing

3.3.2 Arquitetura com Armazenamento Compartilhado (Shared-Storage)

Nesta abordagem, os computadores nodos do cluster compartilham o mesmo dispositivo de armazenamento externo. Estes discos externos geralmente são protegidos utilizando RAID.

Gerenciadores de Bloqueio Distribuído são utilizados nesta arquitetura para controlar os acesso aos discos.

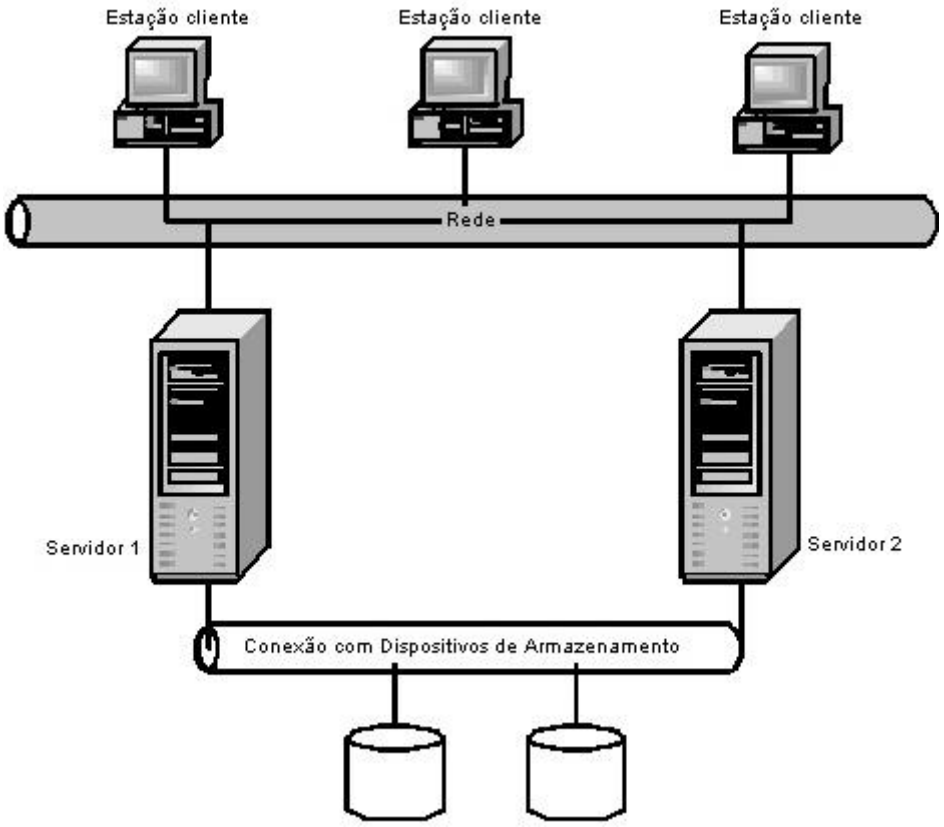

Figura 9: Arquitetura com Armazenamento Compartilhado

3.3.3 Arquitetura com Hot Backup

A idéia desta arquitetura é ter um servidor primário principal onde são executadas as aplicações críticas e ter um servidor secundário que é utilizado como backup quando o principal tiver uma falha. Este servidor secundário geralmente é um computador menos potente que o principal e ele fica em modo de espera, aguardando para entrar no ar quando for necessário.

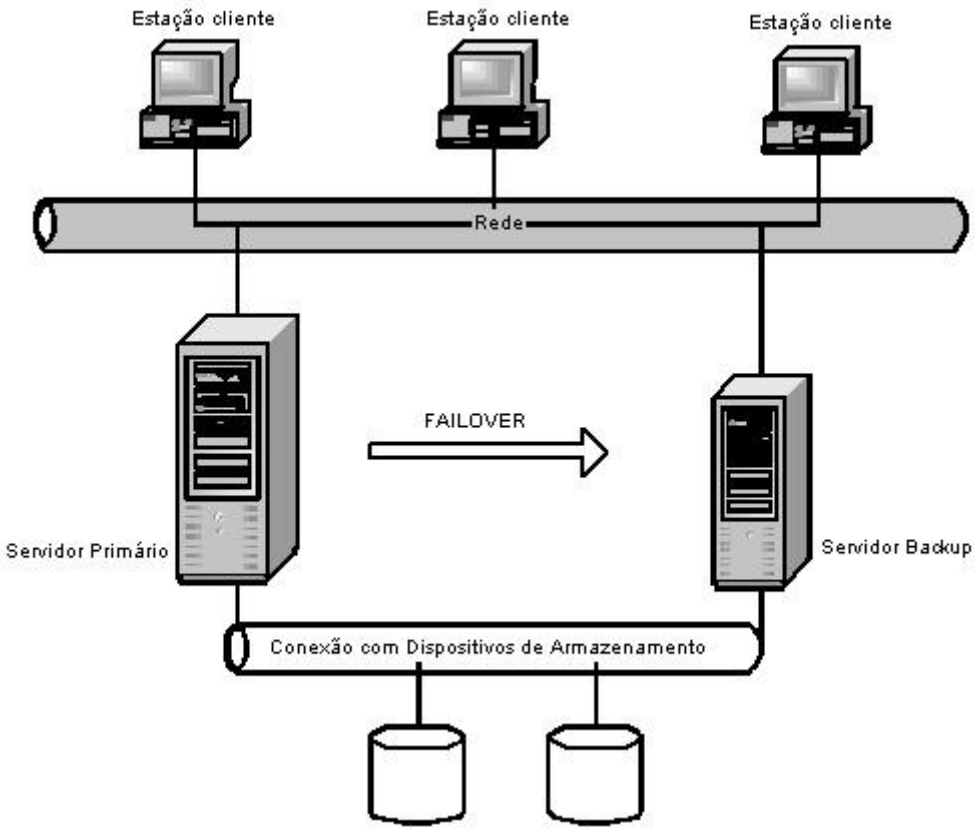

Figura 10: Arquitetura com Hot-Backup

3.3.4 Arquitetura N-Way

Neste método, todos servidores nodos do cluster ficam ativos e trabalhando, nenhum fica esperando um nodo falhar para começar a trabalhar. Esta solução proporciona *failover* bidirecional, que pode ser configurado de acordo com as aplicações e os servidores. Um servidor pode ser configurado para assumir as tarefas de outros dois servidores se esses falharem, e esses dois servidores podem ser configurados para assumirem os trabalhos do primeiro servidor se este falhar.

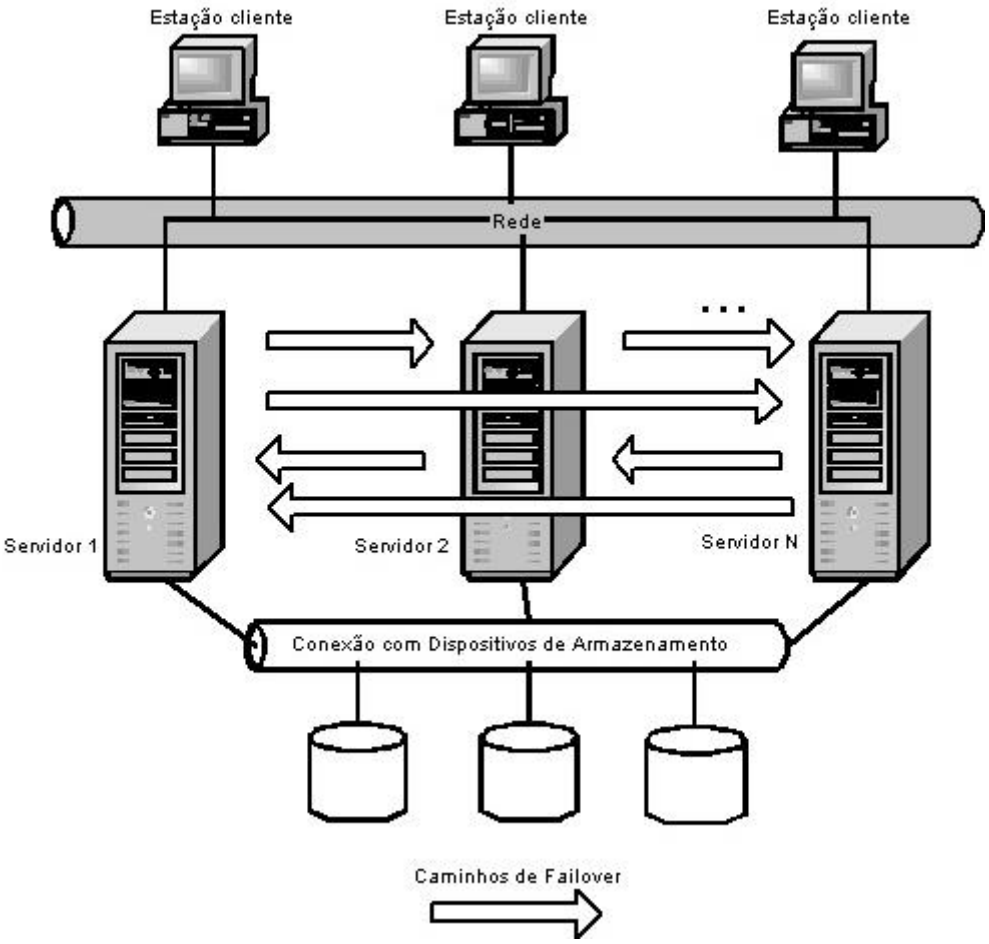

Figura 11: Arquitetura N-Way

3.3.5 Interconexões

Nesta arquitetura são considerados três tipos de interconexões. As interconexões de rede fornecem o meio de comunicação entre as estações clientes e o cluster. A interconexão de armazenamento é a tecnologia que faz a conexão entre os nodos do cluster e o dispositivo de armazenamento, sendo geralmente utilizada a tecnologia SCSI. As interconexões de monitoramento são um meio de comunicação adicional entre os nodos do cluster utilizado para comunicações com informações de controle do sistema. Esta conexão pode ser feita, por exemplo, através de um cabo serial entre os servidores.

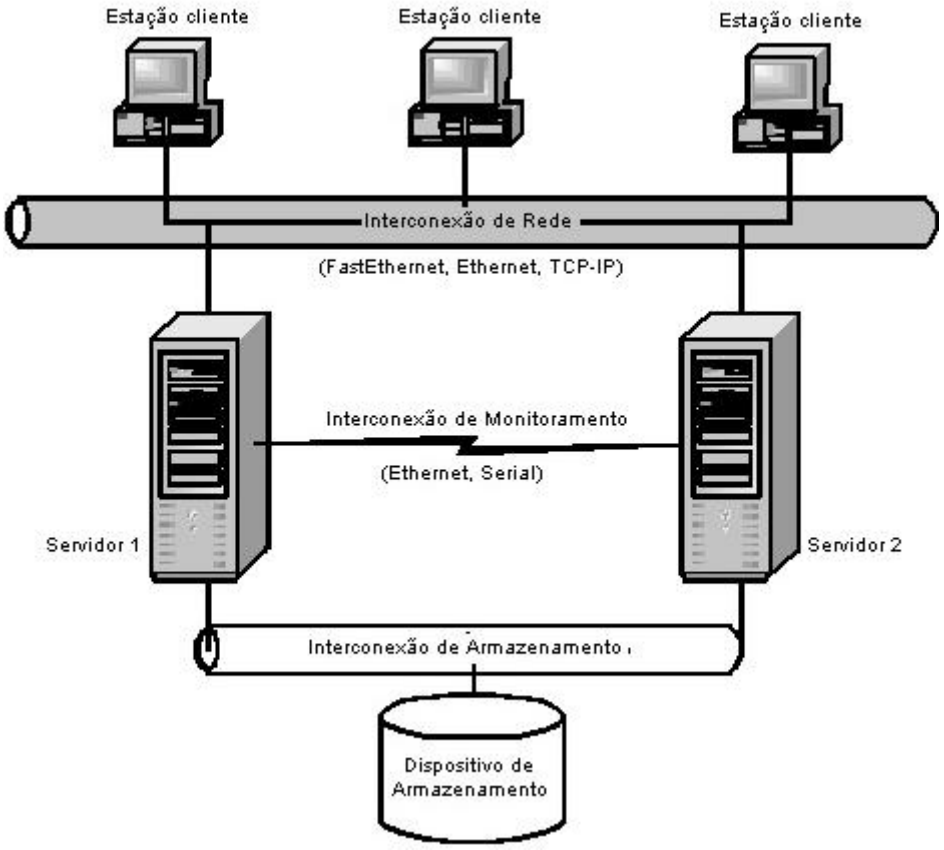

Figura 12: Interconexões

Nesta arquitetura, as tecnologias de interconexão são pontos essenciais. As tecnologias VIA (*Virtual Interface Architecture*), SAN (*Storage Area Network*) e Canal de Fibra são tecnologias recentes nesta área.

A VIA é uma especificação aberta promovida pela Intel, Compaq e Microsoft, que define uma interface para interconexão de alta performance de servidores e dispositivos de armazenamento em um cluster.

Outra arquitetura crescente é a SAN, que consiste em uma coleção de dispositivos de armazenamento que pode ser compartilhada pelos servidores da rede conectada através de hub`s ou switches usando a tecnologia de canal de fibra ou SCSI.

3.4 Exemplo de Produtos Comerciais para implementação de Clusters

A seguir serão apresentados dois exemplos de produtos comerciais para a implementação de Clusters, juntamente com suas principais características e funcionalidades.

3.4.1 Sun Cluster 3.0 [SUN 01]

O Sun Cluster 3.0 foi desenvolvido pelos laboratórios da Sun com a finalidade de ser uma solução poderosa para implementação de Clusters. O foco principal do Sun Cluster 3.0 é integrar ao núcleo do ambiente operacional Solaris características como: disponibilidade, escalabilidade, gerenciamento e facilidade de uso.

O SunPlex System é a chave para o Sun Cluster 3.0. Construído em volta do Sun Cluster, do ambiente operacional Solaris, servidores Sun e seus componentes, o SunPlex é projetado para gerenciar serviços e aplicações em ambientes acoplados.

O Sun Cluster é uma extensão do sistema operacional Solaris que capacita os serviços do núcleo do sistema, como os dispositivos, o serviço de arquivos e o serviço de comunicação, trabalharem em uma arquitetura com clusters mantendo compatibilidade total das aplicações existentes. O Sun Cluster fornece Alta Disponibilidade e Escalabilidade para as aplicações Solaris com uma comunicação contínua e disponibilidade dos dados.

Principais Benefícios

- Integração do Cluster com o sistema operacional Solaris: serviços do núcleo do sistema Solaris podem operar no cluster sem qualquer modificação e mantendo compatibilidade das aplicações.
- Disponibilidade Contínua de serviços do núcleo: serviços de arquivos e rede tem disponibilidade contínua. Outros nodos do cluster automaticamente assumem o serviço de um nodo que falha, mantendo os serviços para os usuários.
- Gerenciamento a Nível de Serviço: podem ser adicionados ao cluster mais domínios e sistemas, e são aumentadas a disponibilidade, a capacidade e a performance. Os níveis de serviços são mantidos no caso de qualquer problema.
- Gerenciamento Centralizado do Cluster: recursos, servidores e dispositivos de armazenamento são gerenciados por um único sistema, através de uma ferramenta gráfica em qualquer desktop, dentro do ambiente do cluster.

- Facilidade no Uso: tarefas complexas podem ser executadas facilmente, e os administradores do sistema podem gerenciar qualquer recurso do clusters de qualquer ponto da rede.

Principais Características

- Dispositivos Globais: o Sun Cluster 3.0 detecta automaticamente todos dispositivos de armazenamento do sistema, incluindo discos, unidades de fita e drives de cd-rom, e associa um nome global no dispositivo que será integrado ao ambiente do cluster. O sistema fornece uma visão única de todos dispositivos compartilhados no cluster e facilita na administração.
- Serviço de Arquivos Global: o Sun Cluster abstrai a localização dos dados dos serviços para que os dados não precisem ser associados ao servidor que recebe o serviço. O serviço de Arquivos Global trabalha entre o kernel e o sistema de arquivos do Unix, e é visto com o mesmo ponto de montagem em todo o sistema.
- Serviço de Rede Global: o serviço IP pode residir em qualquer lugar dentro do cluster. Assim pode existir uma relação de N para N entre os serviços de IP e endereços IP. Múltiplos endereços IP podem ser usados por um único serviço, ou múltiplos serviços podem utilizar um único endereço.
- Serviços Escaláveis: o Sun Cluster permite que uma aplicação, ou uma série de aplicações de um serviço, rode através de múltiplos domínios e sistemas. Múltiplos serviços são compartilhados no cluster e o administrador pode balancear a carga entre os servidores.
- Serviços de Failover Rápido: o Sun Cluster fornece uma rápida detecção de erros. Se um failover ocorrer, os clientes terão uma breve interrupção dos serviços. Isso minimiza o tempo de parada de serviços e garante alta disponibilidade das aplicações, arquivos e de serviços de rede.
- Suporte a Oito Nodos: o Sun Cluster 3.0 suporta até oito nodos em um único cluster.

3.4.2 Novell Cluster Services [NOV 01]

Os serviços Novell Cluster 1.01 para o Netware 5.1 é um sistema de cluster de servidores que garante alta disponibilidade e gerenciabilidade de recursos críticos da rede, como dados, aplicações e serviços. É um sistema multi-nodo do NDS (Novell Directory Services) que suporta *failover*, *failback* e administração de recursos individuais do cluster.

Este serviço de cluster da Novell garante alta disponibilidade rodando em servidores Intel. Os servidores devem rodar o sistema operacional Netware 5.1 e são suportados até 32 nodos no cluster. O Novell Cluster Services pode ser instalado de uma estação onde rode o *software* cliente da Novell, não é necessário ir fisicamente até os servidores da rede para instalar o cluster.

Principais Benefícios

- *Failover* Distribuído: aplicações e serviços podem ser distribuídos para múltiplos servidores ativos para evitar uma sobrecarga de um único nodo. Mesmo que mais de um nodo tenha uma falha, os usuários continuam acessando os recursos. Se um ou mais nodos falharem, qualquer nodo ativo pode assumir as tarefas destes.
- Reconexão Transparente do Cliente: os clientes são reconectados automaticamente e de forma transparente de um nodo que falhou para um nodo ativo. Os mapeamentos dos usuários são mantidos e os volumes são montados em outro servidor.
- Elimina a Interrupção de Serviços para Manutenção de Servidores: o serviço de cluster da Novell fornece acesso aos recursos da rede para os usuários, mesmo durante manutenções em servidores individuais do cluster. Os recursos podem ser migrados de um servidor para outro dentro do cluster e garantir a disponibilidade aos usuários durante uma manutenção.
- Adição Dinâmica de novos nodos: novos nodos são adicionados ao cluster sem a necessidade de parar os nodos já existentes para realizar ajustes na configuração.
- Migração de Direitos Automaticamente: quando um nodo assume os serviços de outro, os direitos de acesso do servidor que falhou são migrados para o servidor

que assumiu o serviço. Assim, os usuários continuam acessando os dados e serviços conforme os direitos de acesso.

- Previne Problemas na Montagem dos Volumes: os volumes adicionados ao cluster são vistos por todos servidores. Se um volume falhar na montagem, o serviço de cluster procura uma nova partição e tenta montar o volume novamente.
- *Failover* de Serviços de FTP e DHCP: fornece *failover* automático destes serviços.
- Ponto único para Administração: através do ConsoleOne, ferramenta de administração gráfica da Novell baseada em Java, todos os recursos, informações, protocolos e políticas do cluster podem ser gerenciadas.

3.5 Cluster com Software Livre

Existem soluções para implementação de Clusters para Alta Disponibilidade baseada no uso de Software Livre. Essas soluções podem ser muito mais baratas e podem ser tão eficientes quanto às soluções comerciais.

Será visto no capítulo 4 que Software Livre não é simplesmente uma questão de preço e sim uma questão de liberdades que esse software fornece a quem está utilizando.

As soluções baseadas em Software Livre estão voltadas na sua maioria para o uso do *kernel* do Linux. O Linux é a grande prova que o Software Livre pode ser eficiente e seguro. O Linux já é largamente utilizado em empresas e instituições como sistema operacional dos servidores, o que mostra que o Linux pode ser considerado um *software* eficiente.

4. Software Livre

Nesse capítulo são apresentadas as características do Software Livre. O Projeto GNU foi um dos principais responsáveis pela difusão desse tipo de *software*, juntamente com o sistema operacional Linux.

4.1 Projeto GNU [GNU 01]

O Projeto GNU começou a ser idealizado em 1983 através de *Richard Stallman*. O objetivo do projeto era desenvolver um sistema operacional completo e livre. Esse sistema, que foi chamado de sistema GNU, era muito similar ao Unix.

A idéia do Projeto GNU era trazer de volta o espírito de cooperação mútua que existia no início quando surgiram as descobertas da informática. Essa cooperação terminou devido ao surgimento dos *softwares* proprietários.

No início dos anos 70, *Richard Stallman* trabalhava em um grupo que utilizava Software Livre. As empresas de informática distribuíam *software* livremente e os programadores cooperavam uns com os outros.

Nos anos 80, os *softwares* eram proprietários, isto é, eles tinham donos que proibiam e impediam a cooperação entre os usuários. O projeto GNU surgiu a partir desta questão.

A partir disso, o projeto GNU teve seu início e o objetivo era projetar um sistema operacional livre, pois não era possível utilizar um computador sem sistema operacional e estes eram proprietários. Também era necessário incluir no sistema operacional várias ferramentas como editores, compiladores e muitas outras.

A equipe de pesquisadores do Projeto GNU decidiu que esse sistema operacional fosse similar ao Unix, pois sua estrutura era eficiente e portável, além de tornar mais fácil a mudança dos usuários do Unix para o GNU.

O sistema operacional livre foi atingido nos anos 90. A equipe do GNU havia desenvolvido ou encontrado vários componentes principais, menos o kernel. Então surgiu o Linux, um kernel livre desenvolvido por Linus Torvalds. Um sistema GNU baseado em Linux foi criado então através da combinação do kernel Linux e do quase completo sistema GNU.

O projeto GNU não está limitado a sistemas operacionais. O projeto pretende se estender a outros tipos de *software*, inclusive jogos e programas para usuários com pouca prática com computadores.

4.2 Software Livre [GNU 01]

A palavra “livre” está relacionada com liberdade e não com preço. Um Software Livre pode ser adquirido através de uma compra, porém uma vez o *software* adquirido ele pode ser utilizado e alterado sem restrições.

O Software Livre permite aos usuários executar, copiar, distribuir, estudar, modificar e aperfeiçoar o *software*. Basicamente são quatro tipos de liberdade para os usuários do *software*. A primeira delas é a liberdade do usuário em utilizar o *software* para qualquer finalidade.

Em segundo lugar tem-se a liberdade de estudar como o *software* funciona, e alterá-lo para fazer uma adaptação para necessidades específicas. Para isso é necessário acesso ao código fonte do programa e essa é uma condição necessária em um Software Livre.

A terceira liberdade básica é a liberdade de redistribuir cópias do *software* a fim de ajudar ao próximo sem qualquer tipo de restrição.

E por último, a liberdade de aperfeiçoar o *software* e disponibilizar esse *software* com suas melhorias para que toda a comunidade tenha benefícios. Novamente é necessário o acesso ao código-fonte para garantir essa liberdade.

Um *software* é dito “Livre” quando os usuários tem todas essas liberdades, ou seja, podem fazer essas coisas sem ter que pagar ou pedir permissão.

A fim de organizar a distribuição dos *softwares* livres algumas regras são utilizadas, porém essas regras não podem entrar em conflito com as liberdades básicas. Um exemplo é o “copyleft” que é a regra onde na redistribuição do *software* não podem ser adicionadas quaisquer restrições que possam tirar as liberdades vistas dos outros usuários. Esta regra tem o objetivo de proteger essas liberdades.

4.3 O Linux [BAR 01]

O Linux é um sistema operacional livre compatível com o sistema operacional Unix. O nome Linux vem da combinação do nome do seu criador, *Linus Torvalds*, com o nome Unix.

No início dos anos 90, a criação do Linux era um passatempo de *Linus Torvald* que era estudante de Ciência de Computação da Universidade de Helsinki na Finlândia. Linus interessou-se pelo sistema Minix e decidiu desenvolver um sistema operacional mais poderoso que esse. O Minix é uma espécie de clone do Unix, porém é grátis e com código fonte disponível. Ele foi desenvolvido para fins educacionais, para pessoas que queriam utilizar um sistema Unix-compatível e poder observar como ele funcionava por dentro.

Em 1991, Linus disponibilizou a versão 0.02 e continuou seu trabalho até 1994 quando liberou a versão 1.0. A versão atual do kernel do sistema é 2.4.

O Linux é um re-implementação das especificações POSIX, uma padronização da IEEE (Instituto de Engenharia Elétrica e Eletrônica) para sistemas operacionais. Por isso o Linux é muito semelhante ao Unix, mas não vem do mesmo lugar. O Linux está disponível tanto em binários, ou seja os seus executáveis prontos para utilizar, bem como em código fonte, para que possa ser alterado e compilado.

4.3.1 Principais Características do Linux

- Sistema Multitarefa Preemptiva Real: vários programas podem ser executados ao mesmo tempo independentemente ou não, tendo suas áreas de memória protegidas e evitando acessos de memória que comprometam o sistema.
- Sistema Multiusuário: vários usuários podem utilizar o mesmo sistema operacional numa mesma máquina, isso é útil em caso de máquinas em rede.
- Sistema com Multiprocessamento: pode-se ter mais de um processador na mesma máquina, o que dará ao sistema operacional melhor performance e eficiência.
- Leitura de Executáveis sob Demanda: o sistema traz do disco somente o que está sendo utilizado pelo programa para evitar sobrecarga.

- Recursos de Memória Virtual: quando os recursos de memória do computador estão saturados o sistema operacional utiliza espaço em disco como uma extensão da memória principal.
- Suporte a vários Sistemas de Arquivos: suporte para ler e gravar em vários tipos de sistemas de arquivos de sistemas operacionais diferentes.
- Suporte avançado ao protocolo TCP/IP e outros: o Linux tem um suporte eficiente ao protocolo TCP/IP, ele pode ser servidor de www, ftp, dns e outros sem nenhuma dificuldade.
- Código Fonte Disponível: o sistema operacional tem o seu código livre para qualquer programador estudar e modificar.
- Sistema MultiPlataforma: pode ser executado em várias arquiteturas diferentes, como PC’s Intel, Powerpc, Sun e outras.

5. Cluster com Linux

A implementação de Clusters para Alta Disponibilidade com o sistema operacional Linux ganhou força após o lançamento do kernel 2.0. O Linux tem muitas características que são necessárias para um *software* de Alta Disponibilidade. Neste capítulo, serão apresentadas algumas vantagens do uso do Linux para implementação de clusters e os pacotes utilizados nessas implementações. Também neste capítulo, são apresentadas soluções e projetos de clusters com a utilização do Linux.

5.1 Vantagens

As principais vantagens da implementação de Clusters utilizando Linux são o custo efetivo e facilidade na configuração. Junto a isso, são somadas as vantagens do Software Livre. Com a utilização de um Software Livre podem ser realizadas customizações no sistema a fim de melhorar a sua eficiência [LIN 01].

As soluções comerciais para implementação de Clusters têm na sua grande maioria um custo extremamente alto e na maioria delas é necessário à utilização de *hardware* especial. Isso torna inviável a implementação de uma solução para Alta Disponibilidade utilizando Clusters para a maioria das empresas. As soluções comerciais são eficientes e possuem características próprias, de acordo com o seu ambiente e seu sistema operacional. Essas soluções são fechadas, assim os usuários de soluções comerciais ficam limitados ao uso das funcionalidades oferecidas pelo fabricante. Não é possível fazer qualquer tipo de alteração ou adaptação do *software*, pois ele é proprietário. Já nas solução com Software Livre, alterações e adaptações podem ser realizadas livremente para satisfazer todas necessidades que possam surgir.

A maioria das soluções com Software Livre podem ser implementadas utilizando computadores convencionais. Dessa forma, tanto os custos com o *software* quanto os custos com o *hardware* são reduzidos. Com isso, essa alternativa se torna viável para a maioria das empresas.

As soluções de Clusters para Alta Disponibilidade que utilizam o Linux estão baseadas em quatro componentes: o Heartbeat, o DRBD, o Mon e o sistema de arquivos ReiserFs. São quatro pacotes desenvolvidos para o Linux onde cada um deles tem uma função específica dentro de uma solução de Cluster.

5.2 DRBD

O DRBD (*Distributed Replicated Block Device*) é o pacote responsável pela replicação dos dados dos discos. Ele é um *driver* de bloco para o kernel que cria um dispositivo de bloco virtual. Esse bloco virtual consiste de um disco real local e uma conexão de rede que na outra ponta terá outro *driver* DRBD trabalhando como secundário [TEI 00].

Esse driver é responsável de fazer o espelhamento dos discos dos nodos através da rede, é uma espécie de RAID 1 via rede. O *driver* DRBD pode ter dois estados: primário ou secundário [REI 01]. Todos dados que são gravados no dispositivo virtual de um host (drbd primário), são gravados no disco local e também enviados para o dispositivo virtual do outro host (drbd secundário) que fará a gravação dos mesmos dados no seu disco local. Isso mantêm os nodos ficam com os discos iguais.

Se ocorrer um falha no primeiro nodo, o heartbeat é encarregado de trocar o estado do driver DRBD do segundo nodo de secundário para primário e iniciar as aplicações nesse nodo [REI 01].

No capítulo 6, será visto um exemplo da utilização do DRBD e como ele pode ser configurado.

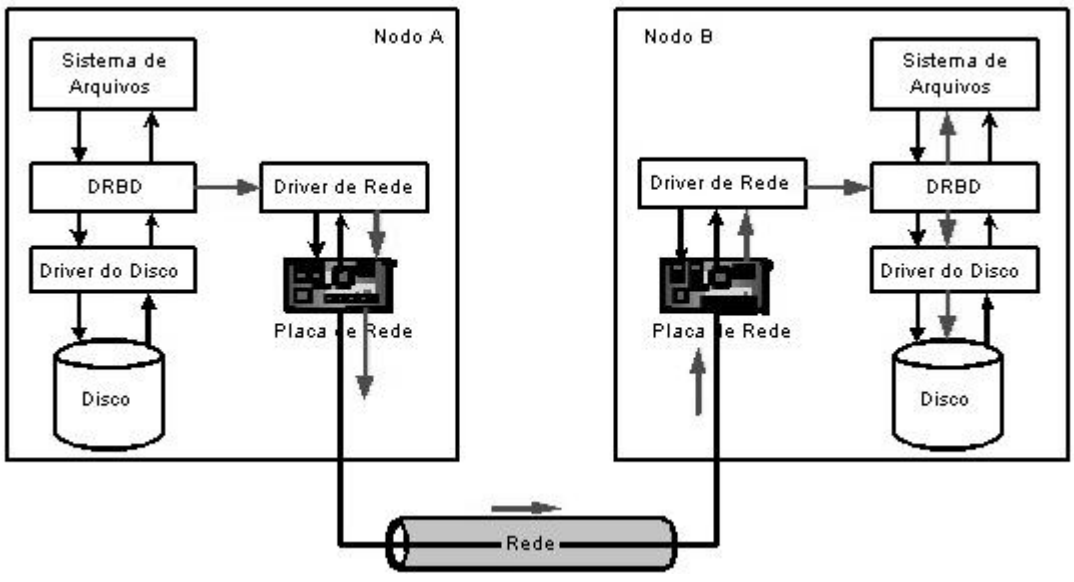

Figura 13: Funcionamento do DRBD

5.3 ReiserFS

Em um Cluster, o sistema de arquivos é um componente extremamente importante, ele deve ser acima de tudo consistente. Nas soluções baseadas em Linux, o sistema de arquivos utilizado em arquiteturas de Cluster para Alta Disponibilidade é o ReiserFs.

O ReiserFs garante maior desempenho e confiabilidade para servidores Linux. Esse sistema de arquivos utiliza uma variante orientada a objetos baseada em plugins de algoritmos clássicos de árvores balanceadas. Esses algoritmos são mais eficientes quando comparados aos sistemas de arquivos convencionais baseados em alocação de blocos [SIS 01].

A característica do ReiserFS que o aponta como o melhor sistema de arquivos para Clusters e que garante confiabilidade é o suporte a *journalling*. Com essa característica, as operações de disco são tratadas como operações atômicas, as alterações são antes registradas em disco. Se ocorrer um problema no sistema, as transações são recuperadas garantindo que os dados estão consistentes, e não é necessário a execução de verificador no sistema de arquivos. Para realizar o *jounalling*, são alocados trinta megabytes de espaço na partição com o sistema ReiserFs. Porém, pelo fato de não existir uma pré-alocação estática de um número de i-nodes e mapas de bits, todos blocos do disco ficam disponíveis. Isso faz com que o ReiserFs seja econômico na utilização do disco [SIS 01].

O ReiserFs também tem a característica de ser rápido devido aos seus algoritmos e árvores balanceadas. Isso o torna mais eficiente, por exemplo, quando existe no sistema um diretório com milhões de arquivos [SIS 01].

5.4 Heartbeat

O Heartbeat é responsável pela monitoração do Cluster. Ele tem a função de testar periodicamente os nodos e em caso de falhas, coordenar as ações de *failover* e *failback*. [TEI 00]

Soluções que utilizam reativação automática de serviços são baseadas no Heartbeat. Ele permite a execução de programas no processo de *failover* e *failback*, controlando os recursos desejados. [TEI 00]

Nos arquivos de configuração do Heartbeat são indicados os nodos do cluster e os recursos que serão monitorados. Também são indicados métodos e chaves de autenticação utilizados entre os nodos do cluster. No capítulo 6 será visto um exemplo de configuração do heartbeat.

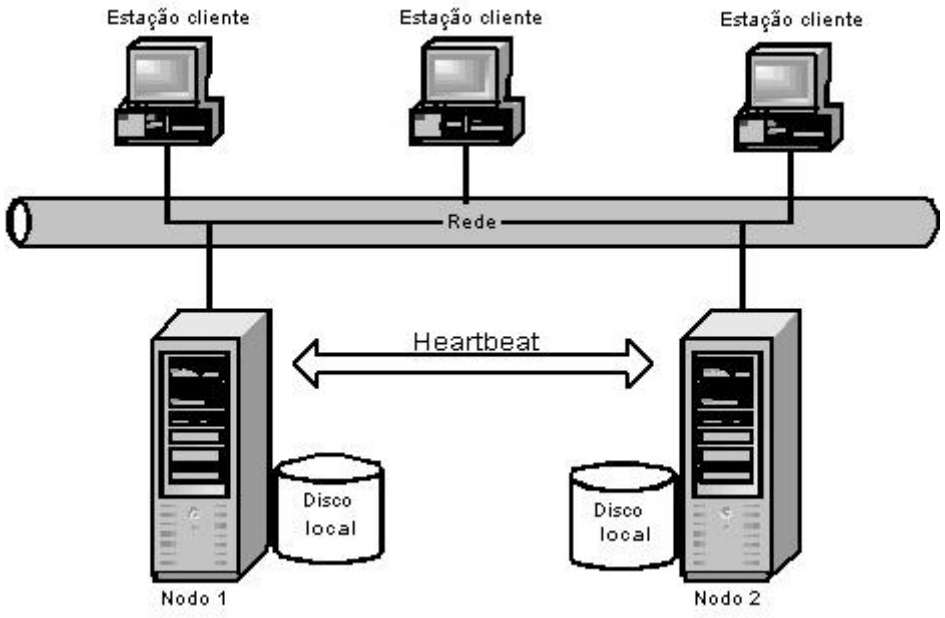

Figura 14: Heartbeat

5.5 Mon

O Mon é responsável pela monitoração dos serviços do sistema. Ele pode verificar centenas de máquinas e serviços de forma rápida [TEI 00]. O Mon monitora a disponibilidade dos recursos de rede, problemas nos servidores, condições do ambiente como temperatura ou outro fator [MAC 01].

Esse recurso de monitoração pode ser visto como duas tarefas diferentes: testar uma condição e disparar um alerta em caso de falha. O Mon foi desenvolvido para manter separadas essas tarefas, ele foi implementado como um “scheduler” que faz a monitoração e chama um certo alerta se essa verificação falhar [MAC 01].

Os alertas podem ser mensagens enviadas por correio eletrônico, para pagers ou telefones celulares, a fim de deixar os administradores do sistema bem informados [TEI 00].

5.6 Solução Conectiva Linux [TEI 00]

A distribuição do Linux da empresa brasileira Conectiva oferece uma solução para a implementação de um Cluster para Alta Disponibilidade.

A solução proposta pela Conectiva é uma solução simples e flexível. A base são os quatro pacotes estudados neste capítulo, portanto essa solução tem quatro blocos básicos: replicação de disco, monitoração dos nodos, monitoração de serviços e um sistema de arquivos robusto. Os quatro componentes podem ser utilizados em conjunto ou individualmente, assim pode-se criar soluções com *failover* e *failback* automáticos ou manuais, com ou sem replicação de disco e outras configurações.

Esta é uma solução idealizada para um cluster simples com apenas dois nodos. A configuração do sistema é simples, ela pode ser feita utilizando o linuxconf do ambiente gráfico do Linux da Conectiva ou pode ser configurado pelo modo texto através da edição dos arquivos de configuração dos pacotes.

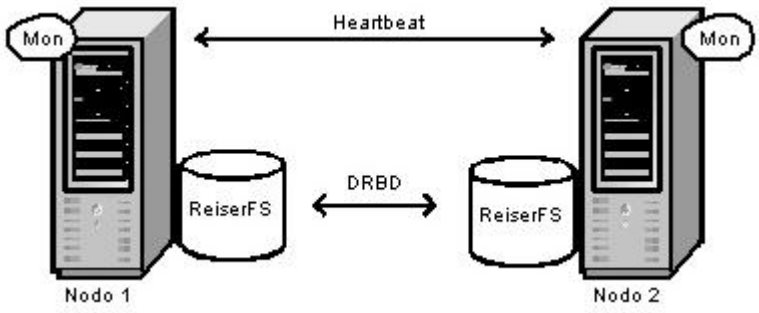

Figura 15: Solução da Conectiva

5.7 Linux Virtual Server [LVS 01]

O Linux Virtual Server é um projeto onde o objetivo principal é desenvolver um servidor virtual com Alta Disponibilidade e Alta Escalabilidade desenvolvido com um Cluster de servidores reais. Por ser um sistema de Cluster, a arquitetura é transparente para os usuários finais, eles tem a visão de um único servidor virtual. A idéia básica do projeto Linux Virtual Server é produzir um servidor Linux de alta performance e

disponibilidade baseada em cluster, o qual forneça uma boa escalabilidade e disponibilidade dos serviços. Este projeto propõem uma solução bem mais complexa que a solução da Conectiva, com a possibilidade de um número maior de nodos e mais recursos, como balanceamento de carga.

Os servidores podem estar interconectados por uma LAN de alta velocidade ou através de uma WAN. A interface de entrada dos servidores reais é um servidor que controla o balanceamento de carga. Esse servidor pode ser considerado como o gateway do Cluster, ele recebe as requisições e distribui para os diferentes servidores dentro do Cluster e faz com que os serviços paralelos pareçam um serviço virtual em um único endereço de IP.

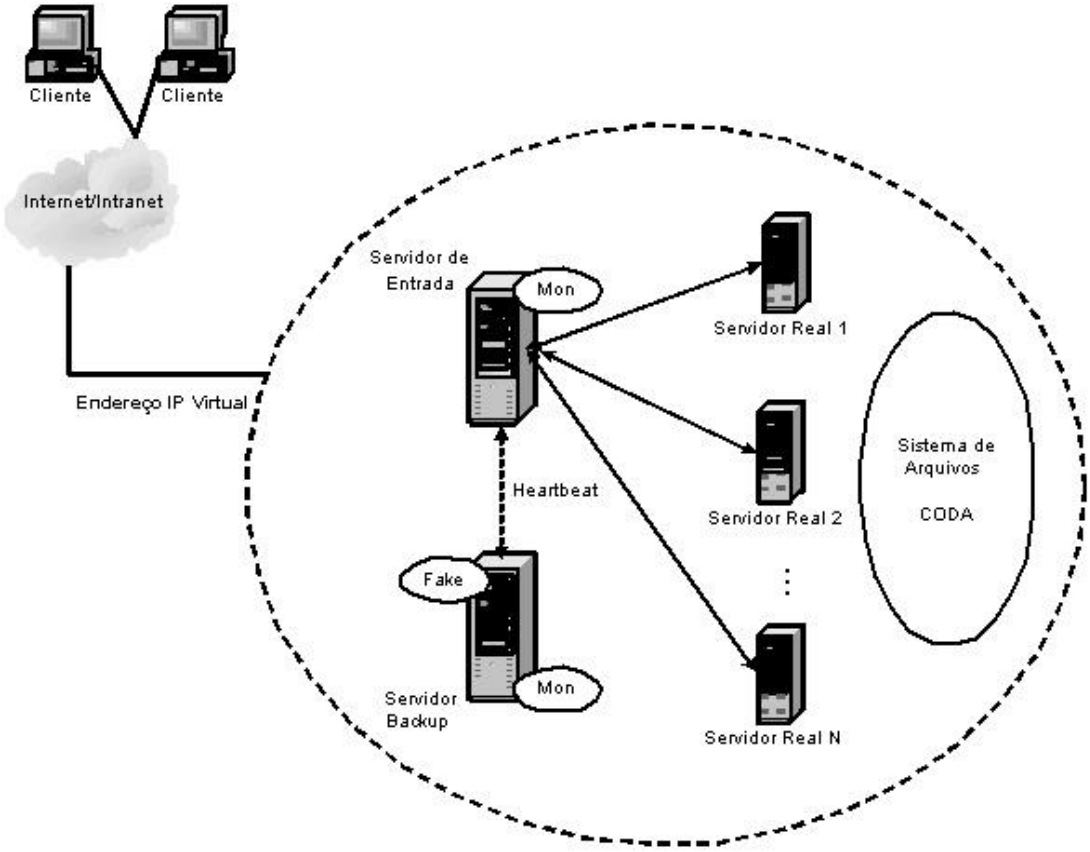

Figura 16: Arquitetura Típica de um Linux Virtual Server

A escalabilidade é garantida pois a adição ou a remoção de um nodo do Cluster pode ocorrer de forma transparente. A Alta Disponibilidade é garantida pela detecção de

um nodo que pode estar falhando, quando isso ocorre as requisições são enviadas para outro nodo em condições.

O servidor de entrada recebe todas as requisições direcionadas para um determinado endereço de IP. Ele repassa essas requisições para os servidores que estão "atrás" dele, essa distribuição pode ser feita através de algoritmos específicos ou de forma seqüencial. Quando um dos servidores do cluster falha, a requisição é repassada para outro.

Para garantir Alta Disponibilidade o Linux Virtual Server utiliza os pacotes Hearbeat e o Mon para fazer a monitoração dos nodos e serviços. No servidor que controla as requisições é utilizado um pacote chamado *Fake*, ele é responsável em fazer o *IP take over* se esse servidor falhar, ou seja, um servidor *backup* deve responder pelo endereço IP desse servidor que falhou.

Como sistema de arquivos é utilizado o Coda, ele é distribuído e tolerante a falhas. O Coda possui várias características essenciais em um sistema de arquivos distribuído, entre elas serviços de replicação, alta performance e segurança.

O Linux Virtual Server pode ter várias aplicações, porém a aplicação principal dessa arquitetura é de servidores de páginas web. Tanto servidores de web para Internet quanto para a Intranet.

5.8 Projeto Beowulf [BEO 01]

O projeto Beowulf é um gênero de cluster baseado em Software Livre no qual a finalidade principal é garantir alta performance para as aplicações do sistema. Esse projeto teve seu início em 1994 no CESDIS (*Center of Excellence in Space Data and information Sciences*) que é operado para a NASA pela USRA (*Universities Space Research Association*). O CESDIS é uma divisão da USRA que fica localizada no GSFC (*Goddard Space Flight Center*) em Greenbelt Maryland nos Estados Unidos.

O primeiro cluster Beowulf foi desenvolvido por *Thomas Sterling* e *Don Becker* que trabalhavam no CESDIS em um projeto chamado ESS (*Eatch and Space Sciences*). Esse primeiro Beowulf foi desenvolvido para trabalhar com problemas associados a grandes quantidades de dados que estavam freqüentemente envolvidos com as aplicações do projeto ESS. Eles construíram um cluster de computadores que consistia de 16 processadores DX4 conectados por rede ethernet de 10 Megabits por segundo. Eles chamaram essa "máquina" de Beowulf. A idéia de proporcionar sistemas baseados

em máquinas comuns para satisfazer necessidades computacionais específicas foi um grande sucesso. A máquina teve um grande destaque e o projeto se espalhou rapidamente para outros laboratórios da NASA e do mundo.

O Beowulf é uma arquitetura desenvolvida para aplicações com processamento paralelo massivo. O sistema geralmente consiste de um nodo ou servidor central e um ou mais nodos clientes, interconectados através de uma rede de comunicação. O *hardware* utilizado no sistema é comum, do tipo que pode ser comprado em qualquer loja. São utilizados computadores padrões, adaptadores de rede padrões e switches para interconectar os nodos. Não contém *hardware* especial em nenhum ponto do sistema e é facilmente reproduzível. Outra característica importante é a utilização de Software Livre, geralmente o sistema operacional Linux, utiliza as bibliotecas de passagem PVM (Parallel virtual machine) e MPI (Message Passing Interface). O nodo central controla todo o cluster e é servidor de arquivos para os nodos clientes. Também funciona como console do sistema e é a porta de saída para o mundo externo. Clusters Beowulf grandes podem ter mais de um nodo central e outros nodos dedicados a tarefas específicas como consoles e monitoração. Na maioria dos cluster Beowulf, os nodos clientes são considerados burros. Esses nodos são controlados pelo nodo central e não possuem discos. Em algumas implementações, os nodos clientes não possuem teclados, monitores e nem gabinete. As placas mãe são montadas em racks especiais para ocupar o mínimo de espaço. Cada nodo de um Beowulf pode ser considerado como um pacote de CPU e memória conectado ao cluster.

O Beowulf trabalha em cima do conceito de granularidade utilizado em programação paralela, que consiste no grau de divisão do dado de um problema entre as CPU’s que trabalharão na solução. Assim, a granularidade está diretamente relacionada com o tempo de processamento total e o tempo de comunicação. Quanto maior ela for, menor será o tempo de processamento. E quanto mais CPU’s participarem da tarefa, maior será o tempo de comunicação.

O objetivo principal do Projeto Beowulf é criar um “máquina” poderosa, capaz de realizar cálculos pesados, tendo um custo baixo utilizando *hardware* comum, que possa ser adquirido em qualquer fornecedor, e Software Livre.

6. Estudo de Caso

Neste breve estudo de caso, será apresentado o ambiente fictício de um escritório de advocacia juntamente com um problema enfrentado pelo mesmo e uma solução baseada no uso de um cluster com Software Livre.

O problema consiste na necessidade do escritório de fornecer aos seus clientes informações atualizadas sobres os seus processos em andamento. Essas informações sobre o andamento dos processos devem estar disponíveis durante as vinte e quatro horas do dia, pois seus clientes estão localizados em diferentes regiões e com fuso-horários diferentes. Os clientes também podem estar viajando e de qualquer ponto do mundo podem querer saber a situação atual de um determinado processo.

Esse problema pode ser facilmente resolvido utilizando recursos de informática disponíveis e por um custo baixo.

6.1 Ambiente Atual

O ambiente atual do escritório em relação a recursos de informática consiste em uma rede de computadores com doze estações, um servidor onde é executada uma aplicação que recebe e controla todas as informações sobre os processos, e um computador utilizado como firewall para separar a rede local da Internet. O acesso a Internet é feito por um link de 56 kbits por segundo de rádio através de uma antena instalada no topo do prédio, o escritório utiliza o serviço de um provedor e recebe na sua sala do prédio um cabo do tipo par trançado categoria 5 que é a comunicação com a Internet. O computador firewall é um computador com *hardware* simples, tem o sistema operacional Linux com um script que faz um filtro dos pacotes entre a Internet e a rede local. Esse computador possui duas placas de rede, na primeira placa está conectado o cabo proveniente da antena, ou seja, da Internet, e a segunda placa está conectada na rede local. As estações da rede estão configuradas para acessarem a Internet através desse firewall e não existe nenhum controle sobre esses acessos. Esse computador tem *hardware* padrão com um processador Pentium 133 megahertz, 32 megabytes de memória principal e um disco rígido de 1.2 gigabytes. Os computadores da rede estão interconectados através de um rede de 100 megabits (FastEthernet) utilizando cabos do tipo par trançado categoria 5 e um switch.

O proprietário do escritório não dispõem de grandes recursos financeiros para gastar na solução que tornaria disponíveis as informações para os seus clientes durante vinte e quatro horas por dia. Ele deseja aproveitar a estrutura atual e os recursos já existentes para o desenvolvimento dessa solução.

6.2 Solução Proposta

A solução proposta para satisfazer a necessidade do escritório em disponibilizar as informações aos seus clientes está baseada na utilização da sua estrutura atual de acesso a Internet. O escritório já possui um acesso contínuo a Internet, a disponibilização das informações dos processos através da Internet irá satisfazer por completo a necessidade do escritório. Os seus clientes poderão acessar via Internet a qualquer hora e de qualquer lugar as informações referentes ao andamento de seus processos. Essa solução propõem disponibilizar através de páginas web essas informações.

A solução utiliza como servidor web um cluster de servidores simples de dois nodos. Os dois computadores utilizados nesse cluster tem *hardware* padrão e o sistema operacional é o Conectiva Linux 6.0 da Distribuição Conectiva. É utilizado o serviço de httpd do linux como servidor para as páginas web e os pacotes Heartbeat, Mon, DRBD e ReiserFS para configurar o cluster. As informações dos processos são exportadas para um formato html através do servidor onde executa a aplicação que controla os processos.

O proprietário do escritório adquiriu dois computadores semi-novos para utilizar no cluster com as seguintes configurações: processadores pentium 166 megahertz, 64 megabytes de memória e 3.2 gigabytes de disco. Foi adquirido também uma placa de rede de 10/100 megabits para cada um desses dois computadores.

Nos dois computadores foi instalado o Linux Conectiva 6.0 com as mesmas configurações e com os seguintes hostnames: server1 e server2. Foi configurado e ativado o serviço de httpd nos dois computadores. Os diretórios dos arquivos html das páginas web serão direcionados para o diretório “/html” que ficará no sistema de arquivos “/dev/hda4” onde será configurado o DRBD.

Para configurar o cluster, o passo inicial é instalar nos dois computadores os pacotes necessários. Esse procedimento foi realizado com os seguintes comandos:

```
# rpm –ivh drbd*
# rpm –ivh reiserfs-utils*
# rpm –ivh heartbeat*
```

6.2.1 Configurando o DRBD e o ReiserFS

A configuração foi realizada em modo texto. Primeiramente é configurado o DRBD que é responsável pela replicação dos dados dos discos entre os nodos. As partições do DRBD têm tamanho igual nos dois nodos. A configuração é feita nos dois computadores no arquivo “drbd0” que fica no diretório “/etc/sysconfig/drbd”. O arquivo fica assim:

```
#nodo primário e secundário, no formato nome:[porta], 7788 é a porta
#padrão

MASTER_NODE="server1:7788"
SLAVE_NODE="server2:7788"

#endereços ip das interfaces se necessário. devem ficar vazios
#mas não comentados

MASTER_IF=""
SLAVE_IF=""
OPTIONS=""

#protoocolo de confirmação. pode ser A, B ou C

PROTOCOL="A"

#dispositivos do nodo primário e secundário (podem ser diferentes)

MASTER_DEVICE="/dev/nb0"
SLAVE_DEVICE="/dev/nb0"

#partições do driver drdb (podem ser diferentes mas com tamanho igual)

MASTER_PARTITION="/dev/hda4"
SLAVE_PARTITION="/dev/hda4"
```

Para inicializar o DRBD e verificar o seu estado no sistema são utilizados, respectivamente, os seguintes comandos:

```
# /etc/rc.d/init.d/drbd start

# /etc/rc.d/init.d/drbd status
```

As partições devem ser montados no dois nodos um de cada vez com seguinte comando:

```
# /etc/ha.d/resource.d/datadisk start
```

Após configurado o DRBD, é necessário criar o sistema de arquivos do tipo ReiserFS. Nos dois nodos, as partições devem ser iniciadas com o sistema ReiserFs através dos comandos:

```
# dd if=/dev/zero of=/dev/hda4
# mkreiserfs /dev/hda4
```

6.2.2 Configurando o Heartbeat

Depois de configurar o DRBD e o sistema de arquivos, é configurado o Heartbeat que faz a monitoração dos nodos.

Por primeiro é configurado o arquivo “haresources” que está localizado no diretório “/etc/ha.d”. Nesse arquivo vai uma lista dos recursos que são monitorados pelo cluster. O arquivo fica dessa forma nos dois nodos:

```
#nome do nodo, endereço de IP do nodo, recursos
server1 10.1.1.1 httpd datadisk
# e no server2 fica
# server2 10.1.1.2 httpd datadisk
```

Depois de configurar o arquivo “haresources”, deve ser configurado o arquivo “ha.cf” no diretório “/etc/ha.d”. Neste arquivo são configuradas várias opções e é indicada a lista dos nodos. O arquivo fica da seguinte forma nos dois nodos:

```
#tempo em segundos entre heartbeats
keepalive 1
#tempo em segundos para considerar o host fora do ar
deadtime 3
#porta para a comunicação udp
udpport 1001
#interface do heartbeat
udp eth0
#se o cluster estiver executando quando o primário iniciar ele atuará
#como um secundário
nice_failback on
#saída para mensagens de depuração
/var/log/ha-debug
```

```
#saída para mensagens de log

logfile /var/log/ha-log
logfacility local0

#nodos do cluster

node server1
node server2
```

Por fim, é configurado o arquivo de autenticação entre os nodos. O arquivo é o “authkeys” dentro do diretório “/etc/ha.d”, esse arquivo contêm o método de autenticação e a chave que vai com o identificador desse método. O arquivo é configurado da seguinte forma:

#diretiva de autenticação: auth e o método de autenticação

auth 3

método md5 utilizado, e a chave é 12345

3 md5 12345

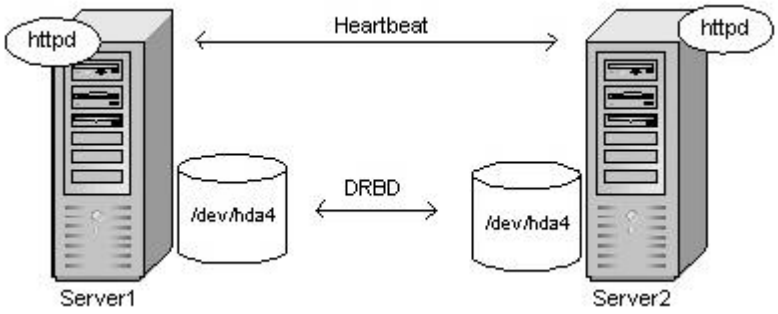

Figura 17: Solução Proposta

Após realizadas todas essas configurações, o cluster está pronto para entrar em operação.

6.2.3 Configurando o Acesso

As páginas web com as informações dos processos devem ser copiadas para o diretório “/html” dos servidores para ficarem disponíveis para o acesso.

Os dois computadores do cluster são conectados no switch da rede local. O computador Firewall é configurado para que as requisições que chegarem até ele para a porta 80, que é a porta do serviço http, sejam redirecionadas para o server1 do cluster, que será o primário. Assim, os acessos provenientes da internet chegam ao cluster.

Se o computador primário tiver uma falha de *software* ou falha em algum componente de *hardware*, o outro estará apto a assumir o seu lugar de forma automática. Por limitações do sistema de firewall utilizado, quando ocorrer esse fato de o server2 assumir os serviços do server 1, será necessário que uma pessoa acesse essa máquina remotamente, via Telnet por exemplo, e execute um script de firewall que redirecione os acessos para o server 2. Essa solução garante que as informações estarão disponíveis aos clientes do escritório durante a maior parte do tempo. Em caso de falhas nos servidores, elas ficarão indisponíveis apenas por alguns minutos. Em um sistema de firewall mais avançado poderia ser configurado o redirecionamento dos acessos externos para mais de uma máquina, assim não seria necessário a intervenção de uma pessoa no processo. A solução apresentada pode ser considerada uma solução com hot-backup, porém é necessária a intervenção de uma pessoa. A solução do projeto Linux Virtual Server poderia ser utilizada neste caso, essa solução é muito mais complexa mas não é necessária a intervenção de uma pessoa para a disponiblidade das informações não ser afetada.

O custo com o *hardware* é muito baixo e o custo com o *software* é praticamente zero. Isso torna essa solução simples e barata.

7. Conclusão

Após a realização deste trabalho, pôde ser observado que as soluções de Clusters baseadas na utilização de Software Livre são uma alternativa interessante para fornecer Alta Disponibilidade em um sistema de computadores. Os principais fatores que tornam essa solução interessante são o custo e as vantagens do Software Livre.

O custo é um fator determinante, pois torna a solução com Software Livre viável para construção de um Cluster em qualquer tipo de empresa ou instituição. Isto ocorre, pois as soluções com *software* proprietário são caras e em geral exigem um *hardware* específico, que em geral tem um custo mais elevado. Com a utilização de Software Livre, o gasto com *software* é baixo e, além disso, o Software Livre tem outras vantagens em relação aos *softwares* proprietários. Entre as vantagens, as principais são as liberdades que ele proporciona para quem está utilizando-o, essas liberdades permitem o aperfeiçoamento, a modificação e adaptação do *software* de acordo com as necessidades.

Para concluir, pode-se dizer que a utilização de clusters em empresas tende a ser ampliada. Isto se deve ao aumento da necessidade de alta disponibilidade das informações e ao fato dos clusters com Software Livre serem mais acessíveis.

8. Bibliografia

Bibliografia Referenciada

[AND 01] ANDRE, Ruiz. ***Alta Disponibilidade em Servidores Linux.*** Disponível por WWW em http://www.revistadolinux.com.br (abril de 2001)

[BAR 01] BAREINBOIM, Elias. ***O que é Linux?*** Disponível por WWW em http://www.olinux.com.br (maio de 2001)

[BEO 01] THE BEOWULF PROJECT. ***The Beowulf Project.*** Disponível por WWW em http://www.beowulf.org (maio de 2001)

[BIR 96] BIRMAN, Kenneth P. ***Building secure and reliable networks applications.*** Greenwich: Manning Publications Corporation, 1996. 591p.

[BUY 99] BUYYA, Rajkumar. ***High Performance Cluster Computing.*** Volume1 *Architectures and Systems* Upper Saddle River: Prentice-Hall Inc, 1999. 846 p.

[GNU 01] GNU's NOT UNIX. ***The GNU Project.*** Disponível por WWW em http://www.gnu.org (maio de 2001)

[IEE 00] IEEE CS TFCC. ***IEEE CS Task Force on Cluster Computing.*** Disponível por WWW em http://www.ieeetfcc.org (dezembro de 2000)

[LIN 01] LINUX CLUSTERING. ***Linux Clustering.*** Disponível por WWW em http://dpnm.postech.ac.kr/cluster (maio de 2001)

[LVS 01] LINUX VIRTUAL SERVER PROJECT. ***Linux Virtual Server Project.*** Disponível por WWW em http://www.linuxvirtualserver.org (maio de 2001)

[MAC 01] MÄCHTEL, Michael. ***High Availability.*** Disponível por WWW em http://innominate.org/~maechtel/ha/ha.htm (maio de 2001)

[NOV 01] NOVELL CLUSTER SERVICES. ***Novell Cluster Services.*** Disponível por WWW em http://www.novell.com (abril de 2001)

[REI 01] REISNER, Philipp. ***DRBD.*** Disponível por WWW em http://www.complang.tuwien.ac.at/reisner/drbd (maio de 2001)

[RES 96] RESNICK, Ron I. ***A Modern Taxonomy of High Availability.*** 1996. Disponível por WWW em http://www.interlog.com/~resnick/HA.htm (março de 2001)

[SIS 01] O SISTEMA DE ARQUIVOS REISERFS. ***O Sistema de Arquivos ReiserFS***. Disponível por WWW em http://www.conectiva.com.br (maio de 2001)

[SUN 01] SUN CLUSTER 3.0. ***Sun Cluster 3.0.*** Disponível por WWW em http://www.sun.com (abril de 2001)

[TEI 00] TEIXEIRA, Roberto; MERCER, Carlos D. ***Guia do Servidor.*** Curitiba: Conectiva S.A., 2000. 330p.

Bibliografia Consultada

[FSF 01] FREE SOFTWARE FOUNDATION. **Free Software Foundation**. Disponível por WWW em http://www.fsf.org/fsf/fsf.html (janeiro de 2001)

[HAL 01] HIGH AVAILABILITY LINUX PROJECT. **High Availability Linux Project**. Disponível por WWW em http://www.linux-ha.com (maio de 2001)

[LEW 01] LEWIS, Phil. ***A High-Availability Cluster for Linux***.
Disponível por WWW em
http://www2.linuxjournal.com/lj/issues/issue64/3247.html
(abril de 2001)

[MIL 01] MILZ, Harald. ***Linux High Availability HOWTO***.
Disponível por WWW em
http://www.ibiblio.org/pub/linux/ALPHA/linux-ha/High-Availability-HOWTO.html (maio de 2001)

[WEB 01] WEPOPEDIA DEFINITION AND LINKS. **Webopedia Definition and Links**. Disponível por WWW em http://webopedia.internet.com
(maio de 2001)