FACULDADES DE TAQUARA
FACULDADE DE INFORMÁTICA
CURSO DE SISTEMAS DE INFORMAÇÃO

ESTUDO PARA IMPLANTAÇÃO DE AMBIENTES DE ALTA DISPONIBILIDADE USANDO O GNU/LINUX CENTOS 5

ROBERTO JOSE PRETTO

Taquara
2007

ROBERTO JOSE PRETTO

ESTUDO PARA IMPLANTAÇÃO DE AMBIENTES DE ALTA DISPONIBILIDADE USANDO O GNU/LINUX CENTOS 5

Trabalho de Conclusão apresentado ao Curso de Sistemas de Informação da Faculdade de Informática das Faculdades de Taquara, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação, sob orientação do Professor Francisco Assis Moreira do Nascimento.

Taquara

2007

AGRADECIMENTOS

As Faculdades de Taquara pelo desafio em manter uma instituição de ensino com alto grau de corpo docente.

Ao meu orientador, Professor Francisco Assis do Nascimento, por sua dedicação, colaboração e empenho.

A minha esposa Luciane, parceira e grande amiga, por ter se demonstrado uma incansável incentivadora.

A meus filhos, Roberto e Michele, pela sua paciência e incentivo.

RESUMO

Este trabalho de conclusão de graduação apresenta um estudo para implantação de ambientes de alta disponibilidade utilizando os recursos do GNU/Linux e teve como objetivos possibilitar as pequenas e médias empresas a utilização de técnicas de alta disponibilidade, criando um roteiro para a instalação e configuração de um cluster e outros recursos que garantam tolerância à falhas. Todas as informações do estudo foram colocadas à disposição da comunidade em um portal de maneira a disseminar o conhecimento para todos.

Palavras-Chave: alta disponibilidade, tolerância à falhas, cluster, GNU/Linux.

ABSTRACT

This work presents a methodology for implementation of high-availability environments based on the GNU/Linux features. The proposed methodology allows small and medium enterprises to adopt high availability techniques by offering guidelines for installation and configuration of a high availability cluster and other resources to ensure the fault tolerance. All the information related to the methodology is available on the dedicated web site.

Keywords: high availability, fault tolerance, cluster, GNU/Linux.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

SUMÁRIO

1 INTRODUÇÃO

Hoje, pequenas e médias empresas representam uma enorme fonte geradora de postos de trabalho no Brasil. Proporcionar ferramentas que auxiliem na disponibilidade de seus ambientes de sistemas de informação pode se tornar um fator competitivo e uma fonte de sobrevivência no mercado.

Segundo dados do IBGE (2007), referente a Estatísticas do Cadastro Central de Empresas 2004, no Brasil a taxa de mortalidade de empresas (empresas que deixaram de operar e encerraram suas atividades) em comparação com a taxa de natalidade de empresas (novas empresas criadas) no período de 2000 até 2004 foi em média de 11,4% inferior. Porém, analisando-se mais detidamente estes dados, em valores absolutos a quantidade de empresas que surgiram no ano de 2004 representou a criação de 1.537.450 empregos diretos. Verificando as empresas que deixaram de existir, foram retirados postos de trabalho em um total de 991.387. Seguindo uma linha de pensamento, considerando que 1.000.000 de pessoas perderam seus postos de trabalho, este fato poderá representar uma quantidade de pessoas afetadas muito maior se levarmos em conta os grupos que direta e indiretamente vivem dos recursos originados por este posto de trabalho (esposa, filhos, pais).

Tendo na área de tecnologia de informação um fator que representa um dos fatores de sucesso de uma empresa conforme está relatado no site da Associação Brasileira de Tecnologia ABT (2007), trabalhar provendo soluções simples e documentadas na área de tecnologia de informação para estas empresas, poderá produzir como resultado uma redução de riscos das mesmas, aumento de qualidade de seus produtos e serviços, o que poderá possibilitar sua perpetuação e conseqüente geração de renda às pessoas envolvidas. Este foi o principal fator de inspiração na elaboração deste trabalho.

Para conseguir tal situação, é proposto o desenvolvimento de um estudo para implantação de ambientes de alta disponibilidade em pequenas e médias empresas. O estudo desenvolvido irá possibilitar aos gestores dos ambientes avaliar a infra-estrutura, recursos de *hardware* e de *software* existentes em uma determinada

empresa e prover soluções levando-se em conta o custo e a facilidade de implementação das soluções.

Para avaliação de recursos de *hardware* e *software* serão identificados os pontos únicos de falha (SPOF[1]) e os fatores que contribuem para o aumento do risco de indisponibilidade do sistema na empresa, tais como instalação elétrica, cabeamento, tipo de conexão de rede *ethernet*, formas de armazenamento de dados, dentre outros.

Baseado nestas avaliações e na necessidade do negócio da empresa, o estudo permitirá ao administrador identificar qual recurso de alta disponibilidade é o mais adequado à sua empresa e indicar como a mesma deverá ser implementada.

1.1 Objetivos

Este trabalho tem por objetivos:

a) desenvolver um estudo que possibilite à pequenas e médias empresas a utilização de técnicas de alta disponibilidade que possam ser aplicadas em uma empresa utilizando os recursos do *GNU/Linux*.

b) criar um roteiro passo a passo para instalação e configuração de um *cluster* de alta disponibilidade, como forma de difusão de conhecimento à comunidade.

c) disponibilizar o conhecimento sobre o estudo desenvolvido através da criação de uma página web em HA CLUSTER (2007) de apoio à comunidade.

[1] Single Point of Failure (SPOF): Ponto único de falha é um determinado aspecto de um sistema que ao falhar faz com que todo o sistema deixe de funcionar adequadamente.

2 FUNDAMENTOS PARA AMBIENTES DE ALTA DISPONIBILIDADE

2.1 Melhores Práticas

A falta de conhecimento tem sido um dos motivos que impedem as empresas de utilizarem determinadas soluções em informática que podem melhorar a disponibilidade de seus sistemas computacionais. Este trabalho pretende ser um instrumento de divulgação de conhecimento em melhoria da infra-estrutura voltada para a alta disponibilidade de sistemas de informação nas empresas.

Mensurar o que uma interrupção no sistema de informação causa de prejuízo a uma empresa vai depender do tipo de informação que está sendo armazenada. Se a corporação está centrada em vendas pela *Internet*, por exemplo, o custo das informações ali constantes é extremamente alto, pois a sua falha implicará em ruptura de toda uma cadeia de eventos decorrentes desde o atendimento inicial prestado ao cliente até o efetivo fechamento de algum negócio. No entanto, outras empresas podem ter uma tolerância maior à uma parada de sistema.

A implementação de 99,999% de disponibilidade em um sistema computacional possui um custo extremamente alto e desta forma não se torna adequado a grande parte das empresas. Segundo Weygant (2002), um percentual de 99,7% (26 horas de indisponibilidade por ano) pode ser perfeitamente aceitável para o negócio de uma empresa, mas seu custo continua sendo dispendioso. Desta forma se justifica a importância de se ter metodologias para determinar o nível de disponibilidade adequado para cada empresa, evitando interrupções não programadas, desgaste da empresa ou marca no mercado e retirando o foco da atividade principal à que esta empresa se propõe.

Este trabalho pretende demonstrar que as melhores práticas em sistemas de alta disponibilidade, como o uso da redundância de placas de rede, do armazenamento em dispositivos de discos externos (*Storages*[1]) e outras facilidades, poderão ser conseguidas a um custo acessível para pequenas e médias empresas

[1] Storage Sistema de armazenamento externo de dados em discos agrupados com níveis de proteção que garantem uma disponibilidade maior dos dados se comparados com as formas tradicionais de gravação dos dados em dispositivos físicos.

pela combinação destas facilidades com a utilização de ferramentas de *software* aberto e livre, disponíveis na plataforma *GNU/Linux*[2], voltadas para implementação de ambientes de alta disponibilidade.

As melhores práticas em um sistema de alta disponibilidade poderão ser afetadas por aspectos do sistema elétrico, os quais muitas vezes não são percebidos facilmente em um ambiente residencial, porém afetam em ambientes corporativos como de escritório, industrial ou ambientes centralizados de computação.

Existem situações que podem colocar em risco toda uma estrutura de dados tais como alteração da freqüência da rede elétrica[3], ruído na linha[4], sobre tensão da rede elétrica[5], sub tensão da rede elétrica[6], *brownout*[7], *spike*[8] e do *sag*[9].

Também relacionado com as melhores práticas pode-se levar em conta o tipo e a forma de distribuição de componentes passivos do sistema tais como comutadores, roteadores e concentradores e a forma de cabeamento utilizado, forma de *backup*, o ambiente onde estarão instalados os equipamentos.

Este trabalho irá analisar cada um destes aspectos e propor metodologias de alta disponibilidade que agreguem segurança, disponibilidade em cada uma das áreas relacionadas anteriormente.

2.2 Defeitos, Erros e Falhas

Um sistema de Alta Disponibilidade visa mascarar falha para o usuário final. Entretanto, o termo “falha” necessita um melhor aprofundamento.

---

[2] GNU/Linux Sistema operativo livre composto pelas bibliotecas e ferramentas do projeto GNU e pelo núcleo do Linux.
[3] Normalmente ocasionada por geradores a diesel ou gasolina quando tem sua rotação alterada.
[4] Caracterizada por interferência eletromagnética (EMI) e de rádio freqüência (EFI). Causado normalmente por cargas indutivas (motores) ou capacitivas (fontes chaveadas).
[5] Aumenta a tensão de forma aleatória causada normalmente por fornecimento de concessionárias de baixa qualidade.
[6] Diminuição da tensão de forma aleatória causada normalmente por fornecimento de concessionárias de baixa qualidade.
[7] Queda drástica da tensão eficaz eficaz da rede elétrica por um tempo longo.
[8] Aumento instantâneo de tensão no sistema, normalmente associado à descargas elétricas.
[9] Diminuição drástica instantânea da tensão no sistema, normalmente associada a solicitação de demandas do sistema.

Conforme Weber (2006), um defeito do sistema ocorre quando seu comportamento desvia do que foi designado por suas especificações. Desta forma, um sistema está defeituoso quando ele não pode prover o serviço desejado. Um erro é parte do estado do sistema que está suscetível a levar a defeitos subsequentes. Se há um erro no estado do sistema, então existe uma seqüência de ações que podem ser executadas e que levarão a defeitos no sistema, a não ser que medidas de correção sejam tomadas. A causa de um erro é uma falha. Esta falha pode ser física ou lógica.

Ter o total entendimento do estado que se encontra o sistema é parte fundamental e apropriada para a resolução de alguma situação de não conformidade. Quando ocorre uma resposta diferente daquela que o sistema deveria fornecer, pode-se deduzir que o sistema possui uma falha ou um erro como citado acima. Diagnostica-lo de maneira rápida e apropriada restabelecendo-o ao seu estado original é o mais indicado. Tal diagnóstico, contudo pode levar tempo e ser um fator impeditivo à resolução do defeito, caso sua forma de verificação esteja fora de um procedimento que contemple todas as possibilidades possíveis de serem investigadas.

2.2.1 Falhas

As falhas podem ser ocasionadas de várias causas de forma isolada ou de forma conjunta, que podem causar uma interrupção não programada em um determinado componente ou equipamento.

A maneira de se tratar uma falha ou uma parada não programada dependerá da habilidade em identificar as possíveis fontes de falha e transformá-las em situações que possam ser controladas. Estas situações de não conformidade podem ser originadas tanto em falhas humanas como em falhas físicas.

Em falhas humanas, por exemplo, pode-se ter o erro de um operador ao realizar uma operação de forma inadvertida ou sem a devida documentação causando uma situação de interrupção. A própria estruturação da solução em si,

como uma falha de projeto ou planejamento de operação, poderá vir a gerar um situação de não conformidade.

Na relação de falhas físicas, relacionam-se aquelas causadas pelo mau funcionamento de componentes elétricos ou eletrônicos tais como queima ou mudança de comportamento.

2.2.2 SPOF *Single Point of Failure*

O mais confiável sistema *stand-alone*[10] poderá ter inúmeros pontos únicos de falhas, chamado SPOF ou *single point of failure*. Um SPOF poderá ser um elemento de *hardware* ou de *software* cuja perda de algum de seus resultados causará a indisponibilidade de um serviço de forma total do sistema. Normalmente estão associados a componentes que não prevêem redundâncias e que podem se tornar este ponto único de falha.

Em maior ou menor grau de risco, cada um destes SPOF pode colaborar com a queda de um sistema computacional. Identifica-los e implementar soluções de continuidade do recurso é basicamente a função de um sistema de alta disponibilidade.

Considerando uma instalação de um sistema típico cliente/servidor, os clientes terão suas aplicações rodando em suas estações conectadas sobre uma rede ao servidor de aplicação que estará executando alguma atividade em sua CPU. O servidor lê e escreve dados de seus clientes em arquivos de seu rígido. O sistema operacional manipula as conexões com os clientes, a transferência de dados, a alocação de memória e outras funções que proporcionam o funcionamento do sistema.

Situações que podem acontecer quando existe uma falha aplicada ao sistema:

a) o sistema deixar de ficar operacional por falha em uma CPU;

b) um cabo de rede danificado, ou uma placa de rede danificada com o cliente perdendo o acesso ao servidor;

---

[10] *Stand Alone* é uma definição do meio comercial onde um sistema roda sob uma plataforma única, sem redundância.

c) erro de operação do administrador do sistema, que coloca o sistema em modo diferente do que o esperado após um *reboot* do sistema, e o cliente não consegue acessar o sistema;

d) disco rígido que possui o sistema operacional entrar em falha causando uma falha geral no sistema;

e) o disco rígido que possui os dados do cliente corromper por falha mecânica, lógica ou administrativa, causando a interrupção do serviço para com o cliente;

f) uma falha elétrica fazendo com o que o sistema fique inoperante totalmente.

No Quadro 1, estão relacionadas algumas situações de componentes associados com uma possível falha e qual solução que poderia minimizar sua ocorrência.

Componente	**Falha verificada**	**Forma eliminar SPOF**
CPU Única	Serviço é perdido até que CPU seja consertada.	Prover *backup* de CPU para a aplicação. Por exemplo, criar um *cluster* de sistemas.
LAN Única	Conectividade do cliente é perdida.	Instalar interfaces de placas de rede redundantes.
Disco de sistema único	Serviço é perdido até que o disco seja trocado.	Utilizar *mirror* de disco de sistema.
Disco de dados único.	Dado é perdido.	Utilizar *mirror* de discos ou *storage* para proteção.
Ponto único de alimentação elétrica	Serviço é interrompido até o restabelecimento da alimentação elétrica.	Utilizar mais de uma fonte de alimentação com UPS ou geradores de energia.
Controladora de discos rígidos única	Serviço é interrompido até a substituição da placa defeituosa.	Utilizar mais de uma controladora para os discos rígidos.
Programas aplicativos	Serviço é interrompido até o restabelecimento do programa.	Prover reinicialização automática do programa aplicativo.
Sistema Operacional	Serviço é interrompido até o sistema reiniciar.	Prover capacidade de *failover* do nó afetado.
Comportamento humano	Serviço é interrompido até que o erro humano seja corrigido.	Automatizar a maior quantidade possível de operações.

Quadro 1: Eliminando Single Point of Failure
Fonte: *CLUSTER for High Availability*

Como visto no Quadro 1, várias serão as possibilidades de interrupção de um serviço quando pontos únicos de falhas acontecem.

2.2.3 MTBF – *Mean Time Between Failure*

Este valor é dado pelo fabricante de um determinado componente em suas especificações técnicas e indica, de acordo com o procedimento de testes usado,

qual é o tempo médio entre falhas daquele produto ocorrido nos laboratórios do fabricante. Este tempo normalmente é dado em horas.

O cálculo do MTBF de um fabricante de equipamentos ou componentes eletrônicos é feito da seguinte forma: é definido um procedimento de teste definindo o número de peças testadas simultaneamente e o número de horas que o teste será efetuado. Multiplicando-se um pelo outro, têm-se o total de horas ligado do componente testado, ou TPOH (*Total Power On Hours*). O total de horas ligado é então dividido pelo número total de peças que apresentaram defeito no período.

Por exemplo, se um fabricante de discos rígidos testar 1000 discos durante 30 dias (720 horas) e um destes discos rígidos apresentar defeito, o MTBF será de 720.000 horas (1.000 discos x 720 horas / 1 defeito).

Outros fatores influenciam na confiabilidade de um componente, tais como o próprio fato do teste ser realizado com os equipamentos sempre ligados. Pelo exemplo anterior, os discos rígidos deste fabricante durariam 720.000 horas (mais de 82 anos operando sem parar).

Outro exemplo, o disco *Seagate Barracuda* ST3250620NS de 250GB, tem um MTBF declarado pelo fabricante de 1.200.000 de horas (SEAGATE, 2006). Em uma matriz com 120 desses discos, tem-se um MTBF de 1.200.000/100 = 10.000 horas (aproximadamente 1 ano e 1 mês). Em uma matriz, agora, com 1.000 discos, o MTBF se reduz para 1.000 horas, ou seja, 41 dias aproximadamente.

Os modelos de previsão teórica de MTBF são incapazes de levar em conta fatores como erros de projeto do dispositivo, *driver, firmware* ou equipamento, defeitos induzidos durante a produção, elementos humanos e o ambiente onde o *driver* será instalado (umidade e temperatura do local).

Quanto à confiabilidade de um produto com múltiplos mecanismos na mesma unidade, o MTBF é dividido pelo número de mecanismos presentes. Isto resulta em um MTBF muito menor quanto todos os componentes são agrupados em um sistema. Por exemplo, se o MTBF de um disco for de 720.000 horas e o mecanismo de ventiladores, fontes de alimentação for de 200.000 horas, então o MTBF de 6 discos juntos será de 3.8 anos. Se junta o fato também o MTBF da CPU, controladoras e outros componentes agregados de um computador, este tempo

entre falhas ficará muito menor, podendo vir a causar uma parada não programada no sistema.

2.3 Sistemas de Alta Disponibilidade

Conforme Weygant (2002), com as redes *ethernet* e *hardware* tornando-se cada vez mais rápidos a custos atrativos, depende-se cada vez mais de sistemas computacionais para a realização de tarefas críticas, cujas falhas acarretariam em prejuízos materiais, financeiros, ou até mesmo em perda de vidas humanas.

Como falhas são inevitáveis, utilizam-se várias técnicas para garantir a disponibilidade destes serviços, mesmo em caso de erros. Estas técnicas podem ser tanto no nível do *software* como no de *hardware*. Através de *hardware*s redundantes e tolerantes a falhas, a falha de um componente é compensada pela utilização de outro. Devido ao alto custo de tais soluções, *clusters* são montados e configurados de modo a atingir um comportamento similar: a falha de um nó é compensada pela migração dos recursos comprometidos para outro nó operante.

2.3.1 Tolerância a Falhas de Alta Disponibilidade

Segundo o artigo de Weber (2006), o termo “Tolerância a Falhas” foi cunhado por Avizienis (1967). Desde então, tem sido usado pela comunidade acadêmica para designar toda a área de pesquisa ocupada com o comportamento de sistemas computacionais sujeitos a ocorrência de falhas. Entretanto, este termo nunca se tornou popular e outros foram usados em seu lugar.

Conforme Weygant (2002) o termo disponibilidade, ou descreve um sistema que provê um específico nível de serviço necessário. Esta idéia de disponibilidade é parte de tudo que você possa pensar hoje em dia. Em computação, disponibilidade é geralmente entendido como o período de tempo em que os serviços estão disponíveis (por exemplo, 16 horas por dia, seis dias por semana) ou mesmo o período de tempo requerido para que o sistema responda aos usuários. Qualquer interrupção deste serviço, planejado ou não planejado é conhecido como “*outage*” ou fora de serviço. “*Downtime*” é a duração desta parada medida em unidades de tempo (minutos ou horas).

Conforme Gartner (1999) demonstra, não é possível tolerar falhas sem redundâncias. Redundância em um sistema pode ser *hardware*, *software* ou tempo. Redundância de *hardware* compreende os componentes de *hardware* adicionados para tolerar falha. Redundância de *software* inclui todos os programas e instruções que são usadas na tolerância à falhas. Uma técnica comum de tolerância à falhas é executar certa instrução várias vezes. Esta técnica necessita redundância de tempo, isto é, tempo extra para executar tarefas para tolerar falha.

Muitas vezes a disponibilidade está associada a outros equipamentos e outros tipos de situações, tais como:

a) redes de computadores (comutadores, roteadores, concentradores e outros) que disponibilizam os serviços de conexões;

b) redes elétricas que alimentam as máquinas. Fica sem efeito a utilização de mais de uma rede elétrica para alimentar as máquinas se elas utilizarem uma mesma entrada de energia;

c) localização dos recursos. Deve-se lembrar que situações de desastres tais como tempestades, inundações ou incêndios podem inviabilizar toda a operação de uma corporação, se a mesma tiver suas informações e sistemas operando somente em um local físico.

2.3.2 Alta Disponibilidade

Alta disponibilidade caracteriza um sistema que é desenhado para evitar perda de serviço através de redução ou gerenciamento de falhas, bem como minimizar paradas de sistemas não programadas conforme referencia Weygant (2002). Espera-se de um sistema de alta disponibilidade quando a vida, a saúde, o bem estar, incluindo o bem estar econômico de uma companhia dependa dela.

Por exemplo, espera-se que o sistema de energia elétrica tenha o maior nível de disponibilidade possível. Qualquer tipo de falha no fornecimento, picos de tensão são inaceitáveis em um local onde vidas dependem da eletricidade para refrigeração, aquecimento, iluminação em adição a outras menos importantes.

Uma outra situação é quando ocorre uma falha geral no sistema de eletricidade de uma grande cidade, como um apagão elétrico, enormes transtornos ocorrerão em questão de segundos, e neste caso espera-se que a companhia fornecedora já esteja trabalhando na restauração do serviço.

2.3.3 Alta Disponibilidade Computacional

Em muitos negócios, a disponibilidade dos computadores tornou-se tão importante quanto a disponibilidade da energia elétrica. Alta disponibilidade computacional utiliza os sistemas de computadores desenhados e gerenciados para operar com um pequeno espaço de paradas planejadas e não planejadas conforme Weygant (2002).

Alta disponibilidade não é algo absoluto. Diferentes necessidades de negócios poderão ter soluções e métodos diferentes de alta disponibilidade computacional. Empresas que trabalham com várias filiais, empresas que operam principalmente na relação com clientes pela *internet*, empresas que necessitam de informações para rodar a linha na fábrica poderão adotar soluções distintas para manter o sistema computacional em um nível aceitável de disponibilidade.

Atualmente, alta disponibilidade computacional é um requisito básico à saúde financeira de uma empresa, não algo como um luxo. Se por um lado, alta disponibilidade é uma forma de segurança contra perda de negócios devido à paradas não programadas do sistema, também é verdade que a alta disponibilidade permite maior competitividade às empresas através de um serviço eficiente aos seus clientes.

2.3.4 Custo da Alta Disponibilidade

Conforme Weygant (2002), o custo de um sistema de alta disponibilidade depende do grau de disponibilidade desejada. O valor do sistema de alta

disponibilidade computacional desejado está diretamente relacionado ao custo de sua parada. Quanto maior o custo de uma parada, mais fácil de justificar o investimento em sistemas. Quanto mais o nível de disponibilidade aproximar-se de 100%, maior e mais rapidamente ele crescerá. O custo de um sistema de 99,95% de disponibilidade é muito maior do que 99,5% de disponibilidade. O custo de 99,5% é muito maior do que o custo de um sistema com disponibilidade de 99% e assim sucessivamente.

2.3.5 Calculando Disponibilidade

Um detalhe a ser abordado sobre alta disponibilidade é como mensurá-la. Em termos técnicos, a disponibilidade de certo serviço é a probabilidade de encontrá-lo operando normalmente em determinado momento. Portanto, tal probabilidade leva em conta qual o provável *uptime* (tempo em que os serviços estarão funcionando) e o provável *downtime* (tempo em que os serviços ficarão inoperantes).

Uma outra notação muito usada atualmente é a que leva em conta o "número de noves de disponibilidade" [11].

A fórmula para define disponibilidade é a porção de tempo que o sistema está operacional.

Disponibilidade = Operação normal / Operação normal + parada de sistema.

A seguir, no Quadro 2 mostra um sistema 24x7x365, onde se espera que esteja em uso 24 horas por dia, 7 dias por semana, 365 dias por ano.

[11] Tal notação é mais voltada a sistemas de alta disponibilidade de mercado, e não a estudos acadêmicos.

Disponibilidade	Mínimo esperado	Máximo permitido	Máximo permitido
	***Uptime* horas**	***Downtime* ano horas**	***Downtime* semanal horas**
90%	7884	876	10,17
98%	8584,8	175,2	2,02
99%	8672	88	1,01
99,20%	8689	70	0,81
99,50%	8716	44	0,51
99,95%	8755	5	0,06
100%	8760	0	0

Quadro 2: Uptime e Downtime para sistemas 24x7x365
Fonte: Adaptado de Weygant (2002)

No Quadro 2 verificamos que a disponibilidade está relacionada com o tempo que o equipamento encontra-se operacional, denominado *uptime* de máquina. A medida que aumenta o nível de disponibilidade do sistema, diminui o tempo das paradas. Observamos que quanto menor for este tempo de parada, maior será a disponibilidade para o sistema.

2.4 *Cluster*

Para que o conceito alta disponibilidade seja assimilado, é necessário primeiramente o entendimento do que são e como funcionam os *clusters*.

2.4.1 Conceitos Básicos e Definições

Em linhas gerais, um *cluster* ou aglomerado é uma coleção de computadores que trabalham juntos para criar um sistema muito mais poderoso de acordo com Vogels (1998). Em outras palavras, um *cluster* é um conjunto de máquinas independentes, chamadas nós, que cooperam umas com as outras para atingir um determinado objetivo comum. Por serem fracamente acopladas, para atingirem este objetivo comum, elas devem comunicar-se umas com as outras a fim de coordenar e

organizar todas as ações a serem executadas. Ainda, freqüentemente elas precisam compartilhar algum *hardware*. Por serem fracamente acopladas, entende-se que elas não fazem parte de uma mesma arquitetura de *hardware*, ou seja, não compartilham o mesmo barramento como dois processadores em uma única máquina compartilham. São máquinas regulares que foram acopladas para formar o *cluster* a fim de poder tolerar certas situações de falha. Assim, para um usuário externo, o *cluster* é visto como sendo um único sistema lógico.

Segundo Weygant (2002), existem dois objetivos principais para a formação de *clusters*:

a) alta disponibilidade (*High Availability* - HA), quando desejamos que o *cluster* forneça determinados serviços que devem estar sempre (ou quase sempre) disponíveis para receber solicitações. Este nível de disponibilidade do serviço é um fator dependente do *cluster*;

b) alta performance (*High Performance Computing* - HPC), quando desejamos que o *cluster* execute determinadas tarefas, sendo que estas são divididas (na sua íntegra ou em frações de uma mesma tarefa) e processadas separadamente em vários nós, a fim de que a velocidade de processamento seja incrementada.

É possível ainda ter uma situação onde o *cluster* deve atingir os dois objetivos juntos. Às vezes, por razões de simplicidade, tal objetivo conjunto é atingido eliminando-se alguns rigores das definições acima. Tal implementação é conhecida como *Grids* de computadores.

A idéia básica dos *Grids* de computadores é combinar o poder de processamento de vários computadores ligados em rede para conseguir executar tarefas que não seria possível ou que não atingisse o desempenho satisfatório quando tratada por um único computador.

Como mostra a Figura 1 um *cluster* em alta disponibilidade é composto por no mínimo dois servidores de aplicação, com discos de inicialização de sistema individuais, um canal de *heartbeat*[1], e um grupo de discos com no mínimo dois canais redundantes de conexão ao sistema de discos ou *storage*. O sistema também

[1] Heartbeat é a forma de comunicação ethernet que verifica qual nó do cluster está ativo e onde estão os pacotes com as aplicações.

poderá estar conectado à parte *LAN* e *WAN* por no mínimo dois canais de rede *ethernet*.

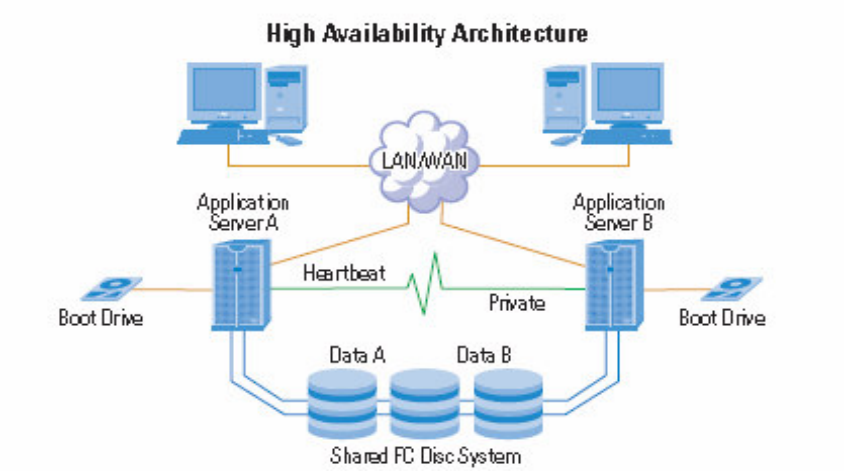


Figura 1: *Cluster* High Availability
Fonte: DELL (2006)

Como mostrado acima, na Figura 1, um *cluster* de alta disponibilidade está baseado em vários equipamentos de *hardware* interligados sob forma de permitir uma maior disponibilidade ao sistema.

2.4.2 Princípios de um *Cluster*

Segundo Vogels (1998), para ser útil um *cluster* precisa seguir alguns princípios básicos:

a) comodidade: os nós em um *cluster* devem ser máquinas normais interconectadas por uma rede genérica. O sistema operacional também deve ser padrão, sendo que o *software* de gerenciamento deve estar acima dele como uma aplicação qualquer;

b) escalabilidade: deve ser possível adicionar aplicações, nós, periféricos e interconexões de rede sem interromper a disponibilidade dos serviços do *cluster*;

c) transparência: apesar de ser constituído por um grupo de nós independentes fracamente agrupados, um *cluster* parece como um único sistema a clientes externos. Aplicações de clientes interagem com o *cluster* como se ele fosse um único servidor com alto desempenho e/ou alta disponibilidade;

d) confiabilidade: o *cluster* deve ter capacidade de detectarem falhas internas ao grupo, assim como de tomar atitudes para que estas falhas não comprometam os serviços oferecidos;

e) gerenciamento e manutenção: uma das principais dificuldades em se trabalhar com *clusters* é seu gerenciamento. A configuração e a manutenção de *clusters* são muitas vezes tarefas complexas e propensas a erros. Um fácil mecanismo de gerenciamento do ambiente deve existir a fim de que o *cluster* não seja um grande sistema complexo com um árduo trabalho de administração.

2.4.3 Arquitetura de um *Cluster*

Atualmente não existe uma padronização de arquitetura para *clusters*, mas muitas propostas assemelham-se em vários sentidos. Segundo Tweedie (2006), tal arquitetura seria formada pelas seguintes camadas:

a) camada de comunicação (*Channel Layer*): trata das comunicações ponto-a-ponto entre os nós;

b) camada de Ligação (*Link Layer*): agrupa canais de comunicação em uma única ligação entre dois nós;

c) camada de Integração (*Integration Layer*): forma o *cluster* propriamente dito, ou seja, controla a entrada e saída de nós do grupo;

d) camada de recuperação (*Recovery Layer*): executa a recuperação (*failover*) e a inicialização/parada controlada de serviços depois de uma alteração na formação do *cluster*.

Existem ainda quatro serviços chaves para um *cluster*:

a) serviço de *quórum* (*Quorum Layer*): em caso de uma divisão do *cluster*, determina qual parte possui autorização para prosseguir com sua execução;

b) serviço de informações (JDB): armazena persistentemente estados internos ao *cluster*. De modo geral, nada mais é do que um repositório de informações local a cada nó do *cluster*;

c) serviço de barreiras: provê um serviço de sincronização global ao *cluster*.

2.5 Storages de Discos Rígidos

Basicamente os *storages* são sistemas de armazenamento de discos que possuem capacidade de redundância em sua parte de entrada de energia, de controladoras do sistema, em redundância de recursos de memória, recursos de gabinetes e de recursos de discos além de terem incorporados as técnicas mais atuais em sistemas de balanceamento de carga e gerenciamento de *software* de proteção contra falhas.

A finalidade de um *storage* é prover um meio seguro e barato de armazenamento de informações em disco, que trabalhe tanto em disponibilidade quanto em desempenho na manipulação das informações do sistema, permitindo que vários sistemas tenham concentrados os dados em um único local, facilitando tarefas administrativas de gerenciamento.

Várias são as técnicas utilizadas para a proteção das informações armazenadas no dispositivo, permitindo que se tenha um determinado nível de proteção em uma área de disco diferente de outra, ou seja, posso ter níveis diferentes de proteção em áreas de discos diferentes.

A técnica utilizada para disponibilizar a segurança, desempenho e capacidade nos discos instalados nos *storages* é o *RAID Redundant Array of Independent Disks* ou conjunto redundante de discos independentes, que vem a ser a utilização de uma porção virtual formada por várias partes de discos independentes.

Existem várias formas de *RAID*, como o tipo 0, 1, 0+1, 2, 3, 4, 5 e outros, sendo que cada um possui características distintas. Comercialmente falando, as configurações com *raid* do tipo 2, 3 e 4 ficaram obsoletos devido à questões de desempenho e de segurança, conforme ACNC (2006).

2.6 Gnu/Linux

O uso do *software* livre tem como base a liberdade de distribuição, sua segurança, padronização, integração e cooperação entre os usuários que o transformaram de uma solução inovadora para uma tendência mundial de mercado. O *software* livre é um programa de computador que pode ser livremente copiado, distribuído, modificado e utilizado, gratuitamente ou com baixo custo devido a sua licença de código fonte aberta. A sua forma de licenciamento do *software* abandona o modelo de licenças de *software* restritivas.

Richard Stallman, idealizador do projeto GNU (2006), lançou em 1980 as bases da licença para *software* de código livre que define clara e explicitamente as condições sob as quais cópias, modificações e redistribuições podem ser efetuadas, para garantir a liberdade de modificar e repassar o *software* assim licenciado. A partir de então, o *software* livre vem ocupando espaços cada vez maiores nos sistemas de TI das empresas, tornando-se um diferencial tanto em custo quanto em qualidade.

3 SISTEMAS COMERCIAIS

Atualmente muitas são as empresas que oferecem serviços de implementação em solução de *cluster* de alta disponibilidade. Serão relacionadas a seguir algumas empresas lideres de mercado com *software* de *cluster* comerciais.

3.1 Hewlett Packard *MC/Service Guard*

Este *software* é desenhado para trabalhar em soluções robustas que necessitam de disponibilidade em seus ambientes computacionais.

De acordo com o *site* da HP – Hewlett Packard (2006) referente ao produto *MC/Service Guard* é especializado na proteção de sistemas *mission-critical*[2]. Com múltiplos nós, este serviço é capaz de prover uma disponibilidade de aplicação alta ao usuário final.

O *MC/Service Guard* em sua implementação, prevê a entrega de um serviço de planejamento da solução, projeto, execução e suportabilidade do ambiente instalado, analisando a infra-estrutura existente nos equipamentos, provendo recomendações, avaliando os sistemas operacionais em suas versões e realizando as adequações.

A implementação estará baseada na disponibilidade de no mínimo dois equipamentos e no máximo de 16 equipamentos em *cluster*, dependendo da versão do sistema operacional. Segundo HP (2006), testes serão realizados no sentido de garantir que todas as facilidades de duplicação de recursos estejam em funcionamento.

Basicamente, o funcionamento do *MCService Guard* ocorre da seguinte maneira de acordo com HP (2007):

a) *o nodo* primário provê os serviços críticos;

[2] Mission Critical é um nível de serviço onde a disponibilidade dos sistemas devem graus extremamente altos de forma a evitar paradas não programadas.

b) *o nodo* secundário permanece em situação de *standby* (espera);

c) o serviço de monitoramento dos *nodos* é responsável por eleger quem será o servidor principal do *cluster*. Este serviço é chamado de *heartbeat*;

d) caso o *software* verifique a queda de algum *nodo*, automaticamente o serviço é reiniciado no outro nó do *cluster*, no servidor que estava em situação de espera;

e) o servidor em *standby* assuma as funções do servidor principal, com todos os serviços e controle de acesso dos usuários;

f) o servidor antigo que antes era o principal, torna-se o servidor *standby* do nó do *cluster*.

Na Figura 2 é referenciada uma representação de *cluster* HA e na Figura 3 são mostradas as características da HP pertinentes à instalação do *software MC/Service Guard*.

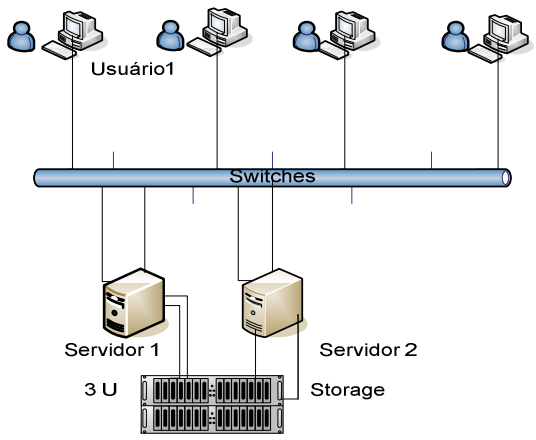


Figura 2: Configuração básica *Cluster HA*
Fonte: Adaptado de HP (2007)

Model comparison

Feature	SCSI	FC
servers supported	Range of ProLiant servers	Range of ProLiant and Integrity servers
Max number of nodes	4	16
Mixed server support	Yes	Yes
Server-to-server interconnect	Ethernet	Ethernet
Server-to-storage interconnect	SCSI	FC
Redundant server-to-storage interconnects supported	MD driver	FC HBA driver
Recovery strategies supported	Active/Standby Active/Active Rotating Standby	Active/Standby Active/Active Rotating Standby
Storage subsystem	Modular Smart Array 500 G2	MSA1000, MSA1500cs, EVA family, XP48, XP128, XP512, XP1024, EMC
Fibre channel switches	N/A	Switches supported for selected disk array
Shared storage connectivity	Any integrated Smart Array controller on supported server and 5i, 6i, 532, and 642	Standard ProLiant and Integrity server FC HBA's are supported by Serviceguard. Refer to individual server Quickspecs for the list of supported adapters.
Cluster configuration & management SW	Serviceguard Manager or Command Line Interface	Serviceguard Manager or Command Line Interface
Disaster tolerance	Not supported	Optional: Cluster Extension for HP StorageWorks XP, Cluster Extension for HP StorageWorks EVA, or Serviceguard Extended Distance Cluster for Linux

Figura 3: Definições *MC/Service Guard*
Fonte: Hewllet Packard SA (2006)

Como mostrado nas Figuras 2 e 3, as características do *software* de *cluster MC/Service Guard* da HP utiliza tendo como máximo de *nodos* 16 utilizando a plataforma *Proliant Server.* Conforme HP (2007), é recomendável a instalação de 3 placas de rede *ethernet*, sendo duas para redundância de rede e uma para verificação do estado do *cluster*, denominada *heartbeat*. Existe na configuração um *nodo* que será o principal do *cluster*, chamado de *nodo* primário e terá como função realizar a validação e distribuição das configurações.

Comandos administrativos irão disponibilizar a situação do *cluster*, tais como:

a) cmviewcl: verifica a situação do *cluster*;

b) cmmakepkg: cria a configuração básica de um serviço;

c) cmhaltpkg: realiza a parado do serviço no *cluster*;

d) cmapplyconf: distribui uma nova versão para os *nodos* que fazer parte do *cluster*;

e) cmrunnode: ativa um *nodo* no *cluster*;

f) cmrunpkg: ativa a configuração de um serviço no *cluster*.

Existe no *cluster MC/Service Guard* alguns serviços básicos como cmcld (que é o serviço de *cluster*), cmclconfd (configurações do *cluster*) e cmlvmd (serviço de *Logical Volume Manage*[12]).

Seu custo irá variar de acordo com a quantidade de equipamentos adquiridos e nível de desconto oferecido.

3.2 IBM V4r4 High Availability *Cluster*

De acordo com o *site* da IBM (2006), o *software* V4R4 introduz a tecnologia de *cluster* que garante a continuidade dos serviços em ambientes que operam em 24 horas, 365 dias ao ano. Nestes ambientes, dois ou mais equipamentos são conectados em *cluster*, garantindo que paradas não programadas sejam evitadas, capacitando o cliente a programar suas atividades de forma a não existir risco de continuidade nos serviços prestados pelo *cluster*. Com o aplicativo instalado, é garantido um sistema que consiga trabalhar com paradas programadas como, por exemplo, manutenções ou *backup*, e paradas não planejadas, como queda de operacionalidade de equipamentos ou erros de operadores, por exemplo.

Várias são as ferramentas incluídas no *software*, como recuperação do banco de dados automático e a possibilidade de ter mais de um *nodo*[3] clonado no sistema.

Na Figura 4, é referenciado uma configuração básica pertinente à instalação do *software* V4R4.

[12] LVM – Logical volume manager. Forma de particionamento lógico de um disco rígido.
[3] Nodo é parte integrante de um cluster. Uma máquina que pertence ao grupo de servidores que fazem parte de um cluster.

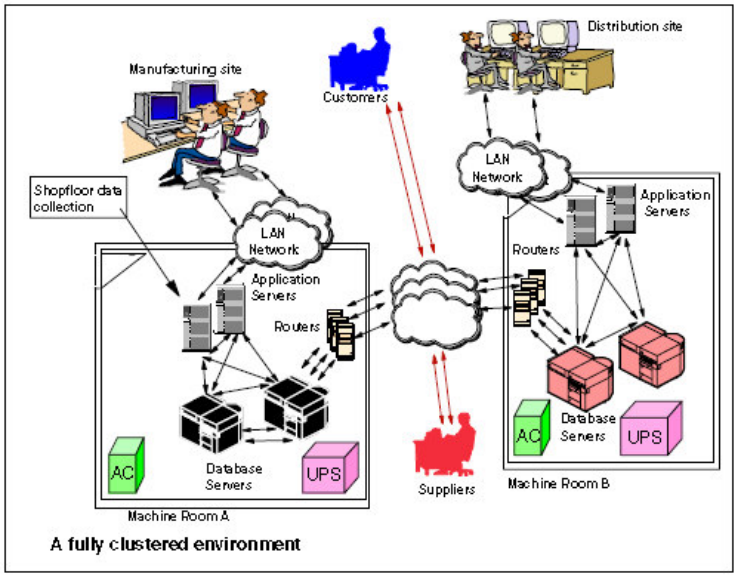


Figura 4 – Configuração V4R4 *High Availability Cluster*
Fonte: IBM (2006)

Tendo como característica básica deste *cluster* o seu funcionamento com a plataforma de servidores AS400, uma robusta linha de equipamentos IBM, garantem agilidade e desempenho quando da queda de um *nodo* do *cluster*. Na Figura 4, um exemplo de vários sites interligados com servidores AS400, onde estão implementadas as soluções do *cluster* V4R4, disponibilizando um serviço de alta disponibilidade aos usuários.

Uma situação que se aplica a este tipo de *cluster* é a necessidade da utilização da plataforma AS400, o que torna seu custo alto.

.

3.3 Dell High Availability *Cluster*

Conforme o *site* da Dell (2006), a solução mais robusta em *cluster* para gerenciadores de banco de dados como o *Oracle*[4]*10g-Real Application*, utiliza até 8 servidores *PowerEdge*[5] em uma parceria na parte de *storages* com a EMC[6], proporciona a garantia de disponibilidade, escalabilidade para futuras expansões e

[4] Oracle é referencia mundial em bando de dados.
[5] PowerEdge é uma linha de servidores corporativos de alto desempenho da DELL Inc
[6] EMC é marca de fabricante de soluções em storages para computação de alto desempenho.

garante níveis altíssimos de disponibilidade no banco de dados através de rigorosos testes realizados em parceria DELL, EMC e Oracle.

Outra estratégia da DELL é trabalhar em todos os segmentos de computadores em soluções de *cluster*, desde os de baixo nível[7], nível médio[8] e computadores de alto desempenho.

Na Figura 5, é apresentada no *site* da DELL uma das possíveis configurações de *cluster* pertinentes à *DELL High Availability Clustering.*

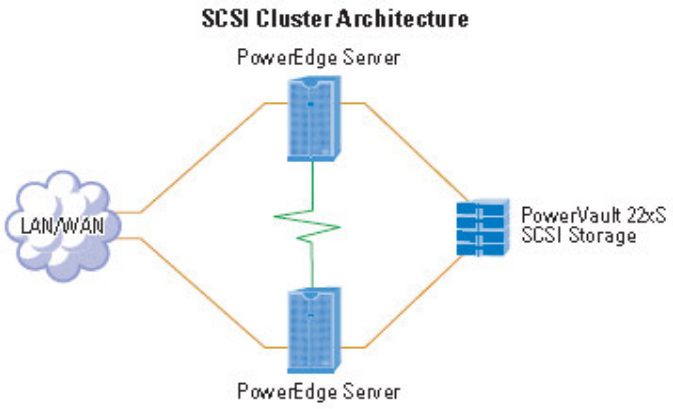


Figura 5: Instalação Dell *High Availability Clustering*
Fonte: DELL (2006)

Utilizando-se os servidores *PowerEdge* conforme Figura 5 em conjunto com o *Microsoft Cluster Service* MSCS, permitem uma configuração de alta disponibilidade e desempenho. Uma das características deste tipo de *cluster* é a necessidade da utilização de servidores idênticos em sua configuração cujo propósito é o de migrar os serviços quando da queda de um dos *nodos* de maneira eficiente. Este *cluster*, quando verifica uma situação de anormalidade em um dos seus *nodos*, automaticamente o servidor é reiniciado e o serviço migrado para o outra equipamento do *nodo* que continuou em funcionamento. Estas características tornam este *cluster* um investimento elevado.

[7] Computadores de capacidade destinado a operações que não requerem um alto grau de investimento.
[8] Computadores destinados a linha enterprise de clientes, segmentados por faixa de faturamento.

3.4 Soluções alternativas para *GNU/Linux*

3.4.1 *Heartbeat*

Uma outra abordagem em utilização de *cluster* de alta disponibilidade conforme Linux-HA (2007) é a utilização do *Heartbeat.*

O *heartbeart,* que significa batimento cardíaco, é utilizado para definir na configuração de um *cluster* qual máquina continua ativa e disponível para a realização de tarefas do sistema. Conforme Linux-HA (2007), o código *heartbeat* é utilizado para construir *clusters* de grande disponibilidade, podendo realizar a construção de dois *nodos* com capacidade de assumirem recursos e serviços de um número ilimitado de interfaces IP.

O mesmo trabalha enviando mensagens de *heartbeat* entre os *nodos* que compõem o *cluster* através de uma ligação *ethernet*, serial ou de ambas as formas de conexão. Caso a falha seja verificada no *heartbeat*, a maquina secundária irá automaticamente assumir que houve uma falha no *nodo* primário e tomar para si os serviços que antes estavam rodando no *nodo* primário do *cluster*. Basicamente ele verifica o status dos componentes do *cluster* e em caso de falha verificada, dispara os processos de remontagem do *cluster*.

3.4.2 DRBD *Distributed Replicated Block Device*

Conforme Linux-HA (2007), o DRBD é um módulo de *kernel* e de *scripts* associados que oferecem um dispositivo para construir *clusters* de alta disponibilidade espelhando conjunto de blocos através de uma rede *ethernet* dedicada podendo ser visto como um *Raid*. O DRBD toma conta dos processos de gravação dos dados, escritos no disco rígido local e envia-os ao outro *nodo* componente do *cluster* para realizar a sua atualização (replicação) no disco rígido remoto.

No *nodo* do dispositivo primário, a aplicação está em funcionamento e tem acesso ao seu sistema de arquivos e toda a sua escrita é enviada para o dispositivo

denominado de nível mais baixo e para o no com o dispositivo secundário que somente escreve o dado no dispositivo de bloco do nível mais baixo.

Se o *nodo* primário falhar, o *heartbeat* irá transformar o dispositivo secundário em primário e iniciará a aplicação naquele *nodo*. Em caso de retorno do *nodo* que houve a falha (antigo primário), o mesmo irá compor o *cluster* sendo desta vez como *nodo* secundário, realizando toda a sincronização dos dados sem nenhuma interrupção do serviço, pois os procedimentos serão realizados em *background*.

3.4.3 Virtualização

Uma outra solução que pode ser empregada em conjunto com *cluster* seria através da virtualização de recursos em um mesmo ambiente. Existe atualmente no mercado vários *software* de virtualização. Estaremos abordando o *Xen* neste estudo.

3.4.3.1 *Xen Hypervisor*

Conforme *Xen* (2007), o *Xen Hypervisor* foi desenvolvido pelos fundadores da *XenSource* e é um código aberto para virtualização de equipamentos. É composto por múltiplas camadas de prioridades, sendo a de maior privilégio o próprio *Xen*. O *Xen host* pode hospedar vários sistemas operacionais em seus usuários de sistemas operacionais que são executados dentro de uma máquina virtual segura e independente chamadas de domínio.

Estes domínios estão estruturados de forma com que os recursos de CPU e de memória possam ser divididos entre os usuários de sistemas operacionais de forma eficiente e segura, podendo desta forma hospedar várias máquinas virtuais dentro de um mesmo equipamento.

Atualmente os membros do *Xen Project* são a *XenSourc*e, IBM, *Intel*, HP, *Novell*, *Red Hat* e *Sun Microsystems*.

Neste estudo optou-se pela instalação deste recurso de virtualização a partir de *Xen* (2007) para a viabilização da instalação do *software* de *cluster* do *CentOS 5.*

3.5 CentOS *Cluster Suite*

De acordo com o site do *CentOS* (2007), o *CentOS Enterprise Linux Distribuition* é construído a partir do código fonte do *Red Hat Enterprise Linux* e equivale em suas distribuições com o código fonte original.

O *software* agrega todo o poder de processamento do Red Hat incluindo ferramentas como o *Cluster Storage*[9], *High Availability Cluster*[10], *Load-Balancing Cluster*[11] e *High Performance Cluster*[12] na versão básica do *CentOS-5*, possibilitando a implementação de recursos de alta disponibilidade a um baixo custo.

Na Figura 6, um detalhamento da solução de *cluster* empregada no *CentOS*.

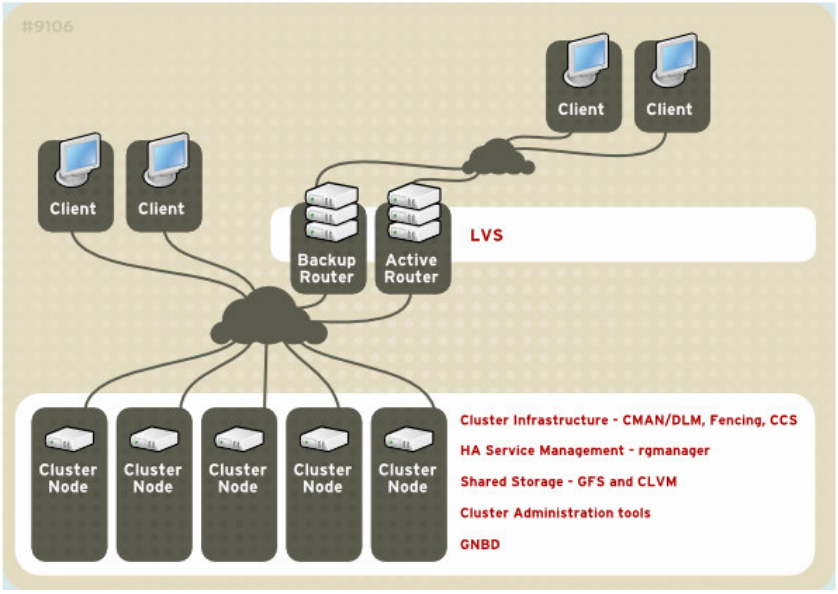


Figura 6: Componentes do *CentOS*-5 *Cluster Suite*
Fonte: CentOS (2007)

Como mostrado na Figura 6, o *Cluster Suite* possui camadas de serviços bem determinadas como as camadas de infra-estrutura, através dos serviços *CMAN/DLM*

[9] *Cluster Storage* provê um consistente sistema de arquivos através dos servidores do cluster.
[10] *High Availability Cluster* permite continuidade dos serviços eliminando pontos únicos de falhas (SPOF) nos servidores do cluster.
[11] *Load-Balancing Cluster* permite o balanceamento de carga de rede através dos nodos do cluster através do Linux Virtual Server.
[12] *High Performance Cluster* capacita os nodos do cluster a trabalhar de maneira paralela (grid de computadores) em cálculos garantindo alta taxa de resposta.

que garantem o seu gerenciamento, serviço *fencing* que garantem a formação do *cluster*, serviço de *CCS* que garante de forma *on-line* a distribuição das informações do *cluster* aos *nodos*, serviço de *RGMANAGER* que gerencia os serviços ativos[13] do *cluster*.

O *Cluster Suíte* apresenta uma característica que o distingue dos demais de acordo com *CentOS* (2007), por disponibilizar o gerenciamento dos *nodos* no *cluster* através do serviço de *fencing* (cercamento), onde existem *scripts* implementados para ativos que fazem a composição do *cluster* tais como unidades ininterruptas de energia, *switches* ou *hubs* gerenciáveis e outros tipos, que garantem o gerenciamento dos *nodos* do *cluster* de forma externa aos servidores.

Este *software* pode ser realizado o seu *download* de forma gratuita a partir de um dos sites espelhos que o *CentOS* disponibiliza.

3.6 Comparação das Soluções

Referente aos *clusters* apresentados pela HP com o *MC/Service Guard*, pela IBM com o *V4R4 High Availability Clusters* e pela DELL com o *DELL High Availability Clustering,* todos utilizam soluções proprietárias, necessitando de hardware específico para o seu funcionamento, não permitindo a utilização de máquinas híbridas em sua configuração básica. Apresentam um alto grau de confiabilidade e disponibilidade em suas configurações e são soluções relativamente com custos altos que poderão variar bastante em função do tipo de equipamento e o nível de segurança que se espera atingir (quantidade de *nodos* configurados).

O diferencial do estudo que está sendo desenvolvido neste trabalho é o baixo investimento na utilização de recursos de alta disponibilidade utilizando o *software* livre *GNU/Linux* no ambiente de *cluster* em conjunto com a readequação de recursos de *hardware* existente no cliente, bem como a documentação gerada a partir da análise dos vários aspectos que influenciam na disponibilidade de um sistema, disponibilizando estas informações em um portal de livre acesso.

Pode-se também optar por abordagem utilizando-se de *Heartbeat* e DRBD.

[13] Ativos é uma referência que se utiliza para qualquer equipamento de informática.

4 ESTUDO DE IMPLANTAÇÃO DE ALTA DISPONIBILIDADE

O estudo de implantação de alta disponibilidade está classificado em quatro áreas críticas que são as variáveis de infra-estrutura, ambiente, *hardware* e *software* e processos automáticos conforme o Quadro 3. Nas próximas seções serão abordados cada um destes tópicos.

Quadro 3: Áreas críticas em alta disponibilidade
Fonte: Autor (2007)

De acordo com Quadro 3, o estudo indica que a disponibilidade dos equipamentos poderá ser afetada direta ou indiretamente por cada uma destas quatro áreas.

4.1 Sistema Elétrico

4.1.1 Análise de Falhas Potenciais do Sistema Elétrico

Muitas são as falhas que podem ocorrer em um sistema elétrico, desde situações onde possa existir a parada parcial, total ou de falhas intermitentes no sistema que possam comprometer a desempenho dos equipamentos.

Conforme experiência pessoal, algumas destas falhas poderão ter sua origem nos seguintes fatores:

a) falha no aterramento gerada por má distribuição dos pontos de fixação das hastes ou interligação fora das especificações, ocasionando qualidade inferior do aterramento em comparação as normas técnicas da ABNT, NBR 5410;

b) proximidade do aterramento ao para raios, que podem receber propagação de descargas atmosféricas pelo solo;

c) possuir entrada única de entrada de energia;

d) possuir caminho único de entrada de energia até o quadro de distribuição;

e) possuir apenas um sistema ininterrupto de energia (*nobreak*) conectado ao sistema;

f) falta de configuração do sistema operacional ao sistema de controle de *nobreak* ou de monitoração dos ativos da rede elétrica (equipamentos modernos possuem este controle e devem ser validados com o fabricante);

g) falha no conjunto de baterias do *nobreak* por falta de manutenção preventiva periódica;

h) falha no acondicionamento das baterias;

i) local de instalação do sistema de energia elétrica sujeita a inundação ou sem refrigeração;

j) local de instalação do sistema elétrico do *nobreak* sem proteção e controle de entrada de pessoal;

k) falta de documentação e identificação dos pontos elétricos distribuídos pela planta no quadro de distribuição elétrico;

l) falha na qualificação de capacidade da carga do sistema em relação ao sistema de energia;

m) falha na interligação e conexão de cabos de energia elétrica;

n) uso de tomadas elétricas e cabos elétricos fora das especificações de carga requerida pelo fabricante.

4.1.2 Recomendação Sistema Elétrico

Com referência ao sistema elétrico serão abordados tópicos que podem influenciar em um sistema de TI e em sua disponibilidade de acordo com as normas técnicas da ABNT NBR 5410 de 2004.

4.1.2.1 Segurança no ambiente de distribuição elétrica

Todo sistema de computadores deverá estar conectado à um sistema de fornecimento e distribuição elétrica de forma independente, não estando nesta rede qualquer tipo de equipamento que possa vir a ser fonte de interferência, tais como aparelhos de ar condicionado, iluminação com reatores não eletrônicos, motores, aparelhos de solda e outros.

Toda a fiação deverá seguir a norma técnica brasileira ABNT, NBR5410, que terá como função garantir a segurança de todo o sistema e em situações de manutenção e/ou expansão da rede elétrica.

Quanto à emenda de fios, os mesmos deverão seguir a mesma norma referente à identificação dos fios, com os pontos de emenda soldados, utilizando grampos de conexão e isolados apropriadamente.

Os quadros de energia deverão ser fechados com tampas apropriadas e com proteção que evite o contato direto com a fiação.

4.1.2.2 Dualidade de entrada elétrica

A entrada elétrica (baixa tensão) deverá vir de forma independente desde a entrada da empresa até o quadro de distribuição dos computadores. Recomenda-se que se tenham duas entradas elétricas vindo da subestação ou quadro geral de entrada de energia elétrica da empresa, utilizando caminhos e conduites distintos e

isolados, sob forma de prever alguma interrupção não desejada devido a alagamentos, obras civis, desmoronamento, queda ou qualquer tipo de acidente que venha a interromper o fornecimento ao sistema.

4.1.2.3 Sistema ininterrupto de alimentação elétrica

É aconselhável a utilização de sistemas *nobreaks* de fornecimento de energia elétrica ao sistema, utilizando transformador isolador visando prevenir contra picos de tensão, falta de energia elétrica ou oscilações que venham a interferir no funcionamento do sistema.

Para sistemas de alta disponibilidade, é recomendada a utilização de dois sistemas ininterruptos de energia, ligados em paralelo que garanta o fornecimento da alimentação elétrica, podendo ser conectados na modalidade passiva, ou comumente chamado no mercado de *Hot Stand By*, onde dois *nobreaks* de mesma potência (principal e reserva) estão conectados à rede de alimentação, mas somente a carga está direcionada para o principal, ficando o outro operando sem carga, ou vazio. Em caso de alguma falha do sistema principal, a carga é imediatamente transferida para o sistema reserva.

Já na modalidade de paralelismo ativo, ambos os equipamentos conectam-se a carga, distribuindo-a de forma proporcional e quando da incidência de uma anormalidade, a carga que é automaticamente migrada ao outro *nobreak*.

Na Figura 7 um *nobreak* responsável pelo fornecimento de energia de um sistema de alta disponibilidade.

Figura 7: *Nobreak*
Fonte: HP (2007)

Na Figura 8 é referenciada a diagramação da interligação dos *nobreaks* em *Hot Stand By* e paralelismo ativo, indicados para configurações onde a disponibilidade é essencial.

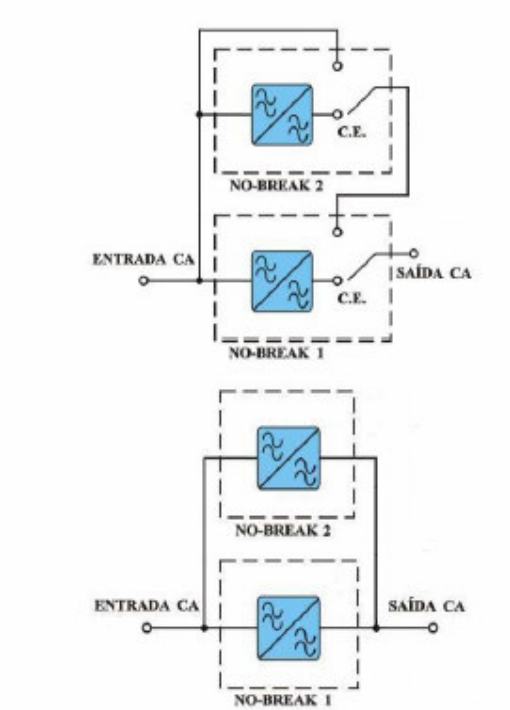


Figura 8: Interligação de *nobreaks*
Fonte: CP (2007)

Como mostrado na Figura 8, na configuração *Hot Stand By* (parte superior da Figura 8) o equipamento reserva fica aguardando a carga ser transferida em situações de anormalidade. Já na configuração de paralelismo ativo (parte inferior da Figura 8), ambos os equipamentos garantem o fornecimento da alimentação elétrica para o sistema e quando da ocorrência de falha de um dos *nobreak*, o segundo equipamento irá automaticamente fornecer a alimentação elétrica para o sistema.

Um fator que deve ser levado em conta neste tipo de configuração em paralelismo ativo é quanto à carga instalada, que não pode exceder a 50% da capacidade total em cada sistema de alimentação (*nobreak*), pois em situações de falha, o equipamento ativo deverá fornecer 100% da potência total requerida.

Geradores de energia elétrica são também indicados no uso de sistemas de alta disponibilidade e deverá levar em conta o consumo, local de instalação, forma de onda e capacidade de carga. Na Figura 9, ilustração de um gerador responsável pelo sistema de alta disponibilidade.

Figura 9: Gerador de energia
Fonte: HP (2007)

O consumo deve ser previamente calculado de acordo com as especificações técnicas do fabricante dos equipamentos que estão atualmente instalados e é recomendada a previsão de carga futura em possíveis expansões de ativos.

Quanto ao local de instalação dos equipamentos este deverá estar em ambiente distinto do local onde os servidores estão instalados. As baterias deverão estar acondicionadas em locais de acesso fácil, seguro, separado do *nobreak* e servidores. Sua base deverá utilizar material anti-corrosivo.

Na Figura 10, ilustração de uma instalação com baterias com suporte de proteção.

Figura 10: Instalação de baterias
Fonte: HP (2007)

Como mostrado na Figura 10, um correto acondicionamento das baterias e sua revisão periódica podem evitar paradas não programadas e aumentar a disponibilidade do sistema.

Outra característica importante a verificar nos *nobreaks* está relacionada abaixo:

a) isolação: os *nobreaks* deverão possuir transformador isolador que garante o isolamento da entrada principal de energia com as saídas que serão disponibilizadas ao sistema.

b) forma de onda: normalmente os nobreaks que oferecem forma de onda a partir de 1/8 como características na saída são os mais indicados, pois garantem uma senóide mais perfeita daquelas de 1/4 e minimiza o risco de

outros aparelhos mais sensíveis não reconhecerem e iniciarem o processo de desligamento automático do sistema caso este tenha sido configurado. Esta característica também se aplica aos geradores elétricos.

4.1.2.4 Quadro de distribuição elétrico

O(s) quadro(s) de distribuição elétrico deverá(ão) estar em local de fácil acesso para realizar manutenções periódicas. Por ser um ponto de falha e de risco à vida humana, o mesmo deverá ser cuidadosamente preparado e é recomendado que siga alguns preceitos básicos de segurança, tais como:

a) porta acrílica ou tampa como forma de proteção e segurança do sistema quanto ao seu acesso.

b) identificação na parte interna referenciando numero de disjuntor com departamento, estação, numero de servidor que permita uma rápida identificação do equipamento ou rede que se está trabalhando.

c) diagrama esquemático de toda a rede elétrica daquele quadro de comando em local de fácil identificação. Os mesmos deverão ter cópias de controle em local arquivado e ordenado, contento revisões quando da entrada de algum novo ponto ou alterações no sistema de energia elétrica. Todas as cópias que se encontrarem dentro da empresa para verificação deverão obrigatoriamente estar na última versão.

d) disjuntores instalados no quadro de distribuição deverão estar de acordo com a carga prevista e calculada para cada equipamento ou periférico.

e) impressoras de alto consumo como as do tipo laser deverão estar conectadas diretamente à rede elétrica, fora do sistema *nobreak*, porém seus disjuntores deverão estar no mesmo quadro de distribuição, salvo determinação especial e após cuidadosa avaliação de carga.

f) recomenda-se instalação de ar condicionado ou sistemas de refrigeração na sala onde se encontra instalado o sistema *nobreak*, salvo determinação especial e após cuidadosa avaliação de carga.

g) servidores deverão ter disjuntores únicos, de acordo com sua carga.

h) *storages* deverão ter disjuntos únicos, de acordo com sua carga.

i) recomenda-se a utilização de disjuntores em estações de trabalho em conjunto, de acordo com sua carga.

j) quadro de distribuição elétrico deverá prever expansões, devendo ter barramentos de fase, neutro e terra, calculados de acordo com a carga total do sistema e prevendo expansões futuras do sistema.

Abaixo, nas Figuras 11 e 12 é mostrado um quadro de distribuição elétrica dentro de padrões de segurança.

Figura 11: Quadro de distribuição elétrica
Fonte: HP (2007)

Figura 12: Parte interna quadro de distribuição elétrica
Fonte: HP (2007)

Nas Figuras 11 e 12, os cuidados com a instalação do quadro garantem segurança as pessoas que trabalham no local. Outro ponto verificado é a identificação dos ativos no quadro de disjuntores.

Nas Figuras 13 e 14, uma representação do tipo de conectorização e acabamentos que são indicados quando da preparação de um quadro de distribuição elétrico para um sistema de alta disponibilidade.

Figura 13: Disjuntor geral e barramentos de distribuição
Fonte: HP (2007)

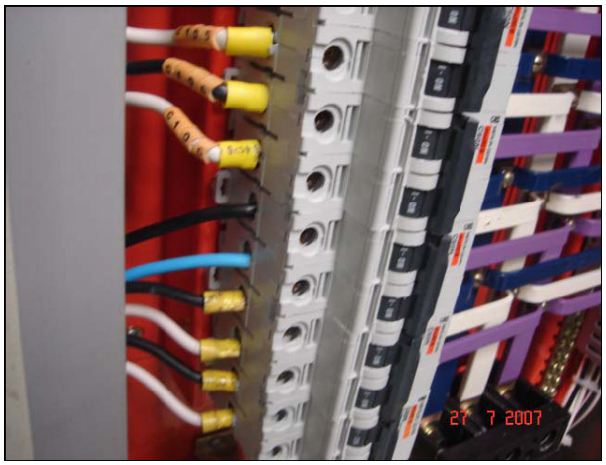


Figura 14: Conectorização de disjuntores
Fonte: HP (2007)

Como verificado nas Figuras 13 e 14, realizando a identificação dos cabos, organizando-os de maneira apropriada dentro do quadro de distribuição, utilizando terminadores na fiação elétrica e deixando no quadro local espaço para futuras expansões facilitam o gerenciamento de manutenções nos ativos e capacita novas instalações no sistema sem parada do sistema.

4.1.2.5 Especificações do Aterramento e Pára Raios

Para as instalações, deve-se seguir a norma da Associação Brasileira de Normas Técnicas ABNT, NBR5410 (2007) de 30/09/2004 que estabelece as condições a que devem satisfazer as instalações elétricas de baixa tensão, a fim de garantir a segurança de pessoas e o funcionamento adequado da instalação e a conservação dos bens.

A correta verificação dos itens referentes ao aterramento é um ponto importante na proteção dos sistemas elétricos dos computadores, pois um fluxo de corrente no sistema somente poderá ocorrer no tempo para que os sistemas de proteção atuem em um determinado momento e este é um fator que define entre um sistema seguro ou um sistema vulnerável à falhas.

Os cabos e conexões do sistema de aterramento devem ser instalados de forma a garantir o contato elétrico entre as partes, além disso, garantir que a integridade das pessoas que trabalham seja garantida. É importante ressaltar que apenas encostar duas partes condutoras não garante a continuidade elétrica entre as partes.

Para efeito de dimensionamento de ambientes de Data Centers, é admitida uma resistência máxima de 5 ohms[1]. A norma NBR5410 admite até 10 ohms de resistência máxima.

São aconselhadas revisões anuais no sistema de aterramento verificando as resistências de aterramento, conexões, continuidade elétrica e demais itens de segurança.

Estes itens poderão ser verificados como segue:

a) verificação geral de cabos;

b) medição de resistências de aterramento;

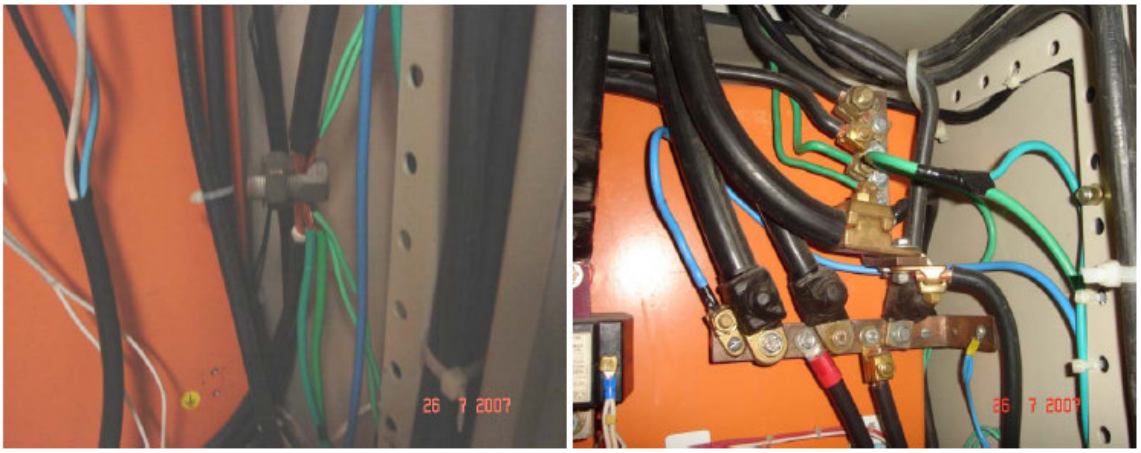
c) verificação das conectorização da malha RFI[2];

d) verificação de aterramento na carcaça de todos os quadros;

e) verificação do aterramento da infra-estrutura.

É importante a validação de expansões futuras nos ambientes com o propósito de evitar situações como demonstrada nas Figuras 15 e 16, onde existem barramentos com dimensionamento inadequado e cabos sem terminais fixação.

[1] Ohms é a unidade de medida da resistência elétrica.
[2] Interferência causada por radio freqüência. Também conhecida como EMI.



Figuras 15 e 16: Conectorização sujeita a falhas
Fonte: HP (2007)

Como verificado nas Figuras 15 e 16, instalações onde não são previstas expansões futuras podem colocar em risco o sistema de fornecimento de energia aos servidores quando não obedecidas premissas básicas de instalação quanto a espaço, expansibilidade e qualidade. Fatores estes que podem gerar interferência vindo de sinais espúrios, falha na conexão ou falta de aterramento ocasionado por terminadores mal fixados, aquecimento na fiação elétrica que podem vir a prejudicar o funcionamento dos equipamentos.

É altamente recomendável ter especial atenção com o aterramento e sua equipotencialização[3] em todo o sistema, garantindo com que partes metálicas do ambiente permitam o escoamento de toda a carga estática gerada, principalmente nas carcaças dos equipamentos.

No Quadro 4 temos alguns valores típicos que podem ser encontrados em um site onde existam equipamentos instalados e a tensão eletrostática gerada por algumas atividades normais.

[3] Ato de fazer com que dois ou mais corpos não possua diferença de potencial elétrico entre eles.

UMIDADE RELATIVA	65 a 95%	10 a 20%
Meios de geração de eletricidade estática	**Tensão Eletrostática (V)**	
Caminhar sobre um carpete	1500	35000
Caminhar sobre um piso de vinil sem tratamento	250	12000
Sentar numa cadeira com estofamento em vinil	700	6000
Sentar numa cadeira com almofada de poliuretano	1500	18000
Utilizar um envelope em plástico comum	600	7000
Pegar uma sacola plástica	1200	20000
Trabalhador numa bancada	100	6000

Quadro 4: ESD Descargas eletrostáticas
Fonte: Adaptado de Saber Eletrônico (2007)

Fato importante a destacar na Quadro 4 é a relação direta e altamente prejudicial entre a umidade relativa do ambiente e a tensão eletrostática gerada. É comum em regiões onde a umidade relativa do ar é baixa ocorrerem um risco maior de descargas que podem prejudicar o funcionamento dos equipamentos.

É recomendada quando da preparação do aterramento a utilização de um barramento de equipotencialização local, chamados de BEL, onde são equipotencializadas todas as partes metálicas no ambiente. Seu funcionamento está baseado na interligação de todos os pontos de aterramentos de infra-estruturas, inclusive da malha de referência de sinal, deixando todos os ativos no mesmo referencial elétrico.

Deste barramento de equipotencialização principal, normalmente chamado de BEP (barramento de equipotencialização principal) partem as interligações dos BELs (Barramento de equipotencialização local), para que 100% das estruturas e infra-estruturas metálicas do local onde a empresa esteja instalada possam estar aterradas e equipotencializadas.

Outro fator que deve ser verificado em um sistema de computação está relacionado aos pára-raios.

O sistema de pára-raios tem função específica de proteger a edificação contra descargas elétricas atmosféricas, não tendo função de proteger antenas de TV, equipamentos elétricos e de potência, sistemas de comunicação e outros equipamentos elétricos quaisquer, pois existem equipamentos específicos para este tipo de proteção. De acordo com a norma NBR-5419/2007 da ABNT, em seu item

4.1 e 4.2, relatam que um sistema de pára-raios não impede a ocorrência de descargas atmosféricas, e um sistema projetado e instalado conforme as especificações da norma, não podem assegurar a proteção absoluta de uma estrutura, de pessoas e bens, mas pode reduzir de forma significativa os riscos de danos devidos às descargas atmosféricas.

4.1.2.6 Especificações da rede elétrica

Os equipamentos conectados ao sistema de rede elétrica *nobreak* ou estabilizadores deverão fornecer uma tensão na saída que atenda as especificações do fabricante do produto. Estas especificações normalmente trabalham em certo range que não varia muito entre os fabricantes. É citado abaixo o fabricante HP (2007) as tensões de saída de seus servidores e ativos obedecem aos limites relacionados abaixo.

a) 220 v: +10% -15% de variação (187 Volts até 242 Volts).
b) 115 v: +10% -15% de variação (98 Volts até 127 Volts).

Toda a validação referente a tensão que será fornecida aos equipamentos deverão levar em conta as recomendações do fabricante.

Segundo a norma NBR 5410/2004 enfatiza que as linhas elétricas devem ser dispostas ou marcadas de modo a permitir sua identificação quando da realização de verificações, ensaios, reparos ou modificações na instalação. Ou seja, os condutores (fase, neutro e terra) devem ser identificados por padrão de cores para facilitar o seu manuseio e evitar acidentes.

4.2 Sistema Lógico

4.2.1 Análise de Falhas Potenciais do Sistema Lógico

Muitas são as falhas que podem ocorrer em um sistema lógico, desde situações onde possa existir a parada parcial, total ou de falhas intermitentes do sistema que comprometam a desempenho dos equipamentos.

Algumas falhas podem ter sua origem nos seguintes fatores:

a) falta de caminhos redundantes das conexões do *datacenter* as estações de trabalho;

b) segmentação de rede;

c) possuir entrada única de energia;

d) possuir caminhos únicos dos ativos de rede até o servidor;

e) falha na configuração dos ativos de rede de acordo com as especificações do fabricante;

f) falha na documentação e identificação dos ativos de rede distribuídos pela planta de instalação;

g) falhas nos contatos das conexões dos cabos de rede;

h) falta de cabos sobressalentes já passados e identificados na planta;

i) falta de *backup* das configurações dos ativos de rede;

j) falha na interligação do sistema lógico em prédios com distâncias superiores a 50 metros devido interferências de acordo ABNT (2007).

4.2.2 Recomendações Sistema Lógico

Com referência ao sistema lógico nas próximas seções serão abordados tópicos que podem influenciar em um sistema de TI e em sua disponibilidade.

4.2.2.1 Especificação de cabeamento físico

Para o cabeamento físico, deve-se seguir a norma da Associação Brasileira de Normas Técnicas ABNT, NBR14565 (2007) de 19/03/2007, que especifica um cabeamento genérico para uso comercial.

A norma cobre os cabeamentos metálico e óptico, aplicando-se as redes locais (LAN) e rede de campus especificando uma ampla variedade de serviços, incluindo voz, dados, texto, vídeo e imagem. É recomendado que esta atividade seja realizada por empresas e profissionais qualificados que ao final da implementação forneçam uma certificação do cabeamento realizado.

4.2.2.2 Rotas de cabeamento

As rotas de cabeamento, tanto elétrico quanto lógico, devem observar e prever situação de continuidade no fornecimento do serviço em caso interrupção por falha física. Situações como rompimento de cabos, inundações, obras civis são comuns de acontecerem e muitas vezes passam de forma despercebida ao responsável de TI de uma empresa. Prever tais situações garante uma qualidade nos serviços oferecidos e segurança no ambiente.

Dualidade de transmissões com rotas distintas em conduítes e separadas por uma distância entre rotas possibilitam a continuidade do serviço de forma parcial quando do rompimento de alguma linha de transmissão (lógica de dados ou elétrica) quando conectados em locais fora do prédio do *datacenter*.

Um fator que deve ser considerado quando da preparação da infra-estrutura de cabeamento refere-se às distâncias dos pontos de ligação tanto lógicos quanto elétrico. Devem ser verificadas junto ao fabricante as especificações técnicas referentes ao tipo de cabo que está sendo utilizado e sua perda com relação à distância.

Outra situação é quanto às rotas de cabeamento internos dos cabos, onde nas Figuras 17 e 18 abaixo, tem-se uma ilustração de como não deve ser realizado o cabeamento interno de um *datacenter*. Esta distribuição de cabos pode vir a ocasionar parada não programada do sistema.

Figura 17: Cabos de fibra e rede elétrica
Fonte: HP (2007)

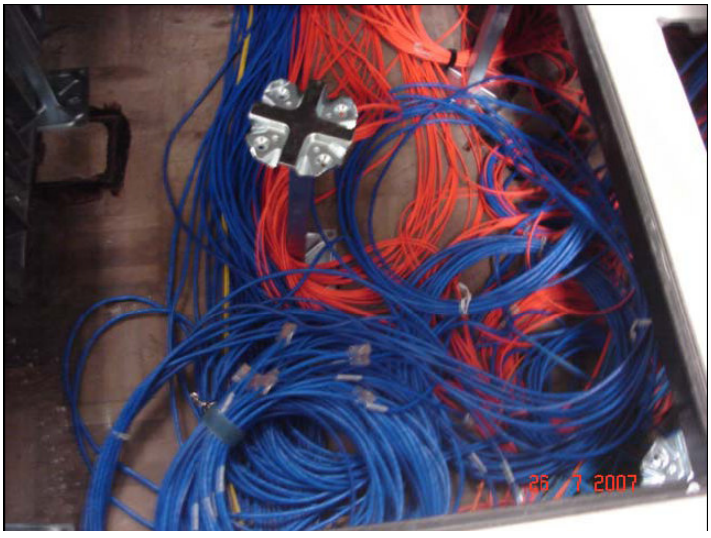


Figura 18: Cabos UTP e fibras
Fonte: HP (2007)

Como visto nas Figuras 17 e 18, áreas onde não são respeitadas normas de acondicionamento dos cabos podem provocar situações de interferência gerando

queda de sinal, possibilidade de rompimento de cabos ou desconexão quando de alguma manutenção ou outras situações não desejadas.

4.3 Sistema de Conectividade

4.3.1 Análise de Falhas Potenciais do Sistema de Conectividade

Em um sistema de TI, grande parte das interrupções não programadas está relacionada ao sistema de conectividade. Paradas não programadas podem ser evitadas a partir de cuidados básicos e de fácil implementação como qualidade nos ativos do sistema de conectividade e sua documentação.

Situações não desejadas como de *spanning tree*[14] que é o *loop* de dados dentre os *switches*, impedindo que o dado consiga chegar até o seu destino, ficando em uma situação de instabilidade que poderá ter como conseqüência a parada total ou parcial da rede.

Outra situação que pode ocorrer é quanto a taxa de transferência de determinados ativos de fibra trabalharem além da capacidade nominal da porta, caracterizando por perda de seus pacotes. Tal situação é denominada de *oversubscription*[15].

Ao realizar o planejamento sobre os ativos de conectividade, dar preferência àqueles que possuem dualidade de fontes de alimentação e possui capacidade de monitoração já implementada em *software*.

Para o gerenciamento dos ativos de conectividade um fator de sucesso na continuidade do serviço é a documentação.

Esta documentação pode ser realizada de várias formas, mas o que sempre deve ser levado em conta é a facilidade de localização da documentação do ambiente, bem como de *backups* de configurações em local controlado e de fácil acesso.

---

[14] Spanning Tree – situação de loop da rede de switches. Existem algoritmos próprios que evitam esta situação que pode levar a parada total de uma rede.

[15] Oversubscription – capacidade de comunicação do canal é superior a velocidade da porta, causando perda de dados.

Exemplos de como relacionar ativos, versões, local de instalação e conexões estão dispostas nos anexos A, B, C e D forma uma sugestão de controle de ativos de todo o sistema.

4.3.2 Recomendações Sistema de Conectividade

Com referência ao sistema de conectividade, nas próximas seções serão abordados tópicos disponibilidade que podem diretamente influenciar em um sistema de TI.

4.3.2.1 *Routers*

Os *routers* ou roteadores quando instalados e configurados devem prever a redundância de recursos para a conexão dos sites (caso exista), alternância na utilização de rede elétrica na conexão dos pontos de força, local de fácil acesso e de identificação rápida de endereçamento (tanto no roteador quanto nos cabos) e cópia de sua configuração em local apropriado.

É importante a utilização de cabos de rede reservas já passado pelos conduítes e estarem devidamente identificados de maneira genérica, facilitando sua utilização em situações de falha e interrupção dos serviços.

4.3.2.2 *Switches*

Os *switches* quando instalados e configurados devem prever a redundância de recursos para a conexão dos sites (caso exista), alternância na utilização de rede elétrica na conexão dos pontos de força, local de fácil acesso e de identificação

rápida de endereçamento (tanto no *switch* quanto nas fibras ou cabos UTP[16]) e cópia de sua configuração em local apropriado.

É importante a utilização de cabos de fibra reservas já passado pelos conduites e estarem devidamente identificados de maneira genérica, facilitando sua utilização em situações de crise.

Outro ponto que deve ser salientado com referência aos cabos de fibra é quanto ao manuseio e acondicionamento das mesmas. Por se tratar de material frágil e de transporte de sinal de ótico, as mesmas devem ter alguns cuidados especiais, tais como:

a) evitar curvaturas acentuadas no cabo de fibra, pois podem provocar reflexão interna do sinal ótico e conseqüente baixa qualidade de sinal;

b) umidade excessiva ou baixa demais dentro do *datacenter*;

c) cabo não deve ser dobrado, amassado ou amarrado, pois podem provocar baixa qualidade de sinal.

É recomendada a utilização de switches de fibra ótica redundante e placas de fibra redundantes nos servidores, cujo propósito é o de garantir a continuidade do serviço em caso de falha de algum ponto de conexão do sistema.

Na Figura 19 uma representação de dualidade de utilização de recursos de *switches* em uma configuração de alta disponibilidade.

[16] UTP Unshielded Twisted Pair é um tipo de cabo utilizado em redes *ethernet* e tem como característica dois condutores enrolados que minimizam o efeito de interferência elétrica.

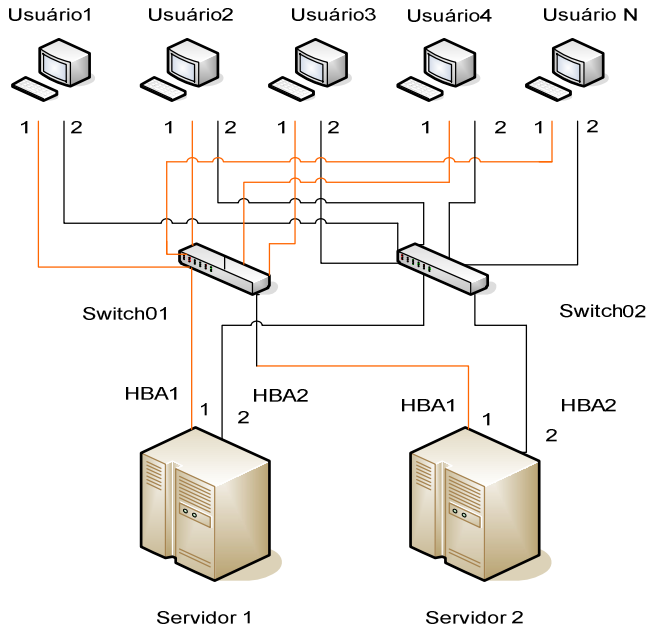


Figura 19: Modelo de Alta Disponibilidade com redundância de *switches*
Fonte: Autor

Na Figura 19 uma representação indicada para serviço de alta disponibilidade em recurso de *switches*, que garantem a continuidade de acesso da eventualidade de queda de servidor, placa de rede ou *switch*. Atenção especial à conexão dos cabos de alimentação elétrico destes ativos que obrigatoriamente devem ser conectados de forma alternada ao sistema elétrico para garantir continuidade de serviço na eventualidade de alguma falha nos equipamentos.

4.3.2.3 *Hubs*

Os *Hubs* quando instalados e configurados (se aplicável) devem prever a redundância de recursos para a conexão dos sites (caso exista), alternância na utilização de rede elétrica na conexão dos pontos de força, local de fácil acesso e de identificação rápida de endereçamento (tanto no *Hub* quanto nos cabos *UTPs*).

É importante a utilização de cabos *UTP* reservas já passado pelos conduites e estarem devidamente identificados de maneira genérica, facilitando sua utilização em situações de crise.

A forma de interligação dos *Hubs* segue o mesmo exemplo da Figura 19, garantindo desta forma disponibilidade de recursos. Atenção especial à conexão dos cabos de alimentação elétrico destes ativos que obrigatoriamente devem ser conectados de forma alternada ao sistema elétrico para garantir continuidade de serviço na eventualidade de alguma falha nos equipamentos.

4.4 Ambiente

4.4.1 Análise de Falhas Potenciais do Ambiente

Muitos são os fatores de ambiente que contribuem para o bom ou mal funcionando de um sistema. Cuidados especiais com a infra-estrutura garantirão um baixo nível de incidentes que possam causar uma interrupção do sistema, permitindo maior confiabilidade e disponibilidade no ambiente de TI.

Algumas destas falhas poderão ocorrer a partir de:

a) queda dos servidores ocasionada por aquecimento excessivo do local onde estão instalados os ativos;

b) mau funcionamento dos equipamentos ocasionado por excesso de pó no ambiente;

c) queima de fontes de alimentação ocasionado por falta de ventilação apropriada;

d) roubo de equipamentos;

e) sabotagem nos equipamentos;

f) falha na limpeza do ambiente ao utilizar produtos tóxicos ou corrosivos;

g) mudança rápida de temperatura do ambiente podendo provocar falha no funcionamento de alguns componentes;

h) umidade excessiva ou baixa demais dentro do *datacenter*;

i) inundações.

Nos próximos tópicos será abordado mais detidamente cada um destes fatores.

4.4.2 Recomendações Variáveis de Ambiente

Com referência ao ambiente onde se encontram instalados os servidores, nas próximas seções serão abordados tópicos que podem influenciar diretamente a disponibilidade dos ativos e sistema de TI.

4.4.2.1 Sistema de refrigeração

O ambiente onde estará instalado os computadores, deverá ter o ambiente refrigerado em uma temperatura que pode variar de 18 a 23 graus *Celsius* e umidade de até 65%. Abaixo, algumas recomendações que podem aumentar a eficiência do sistema de ar condicionado do *Datacenter*:

a) recomenda-se que o sistema de refrigeração tenha o seu sistema de renovação de ar na posição fechado, não permitindo troca de ar com o meio externo em função da entrada de impurezas;

b) recomenda-se que o local de instalação do sistema de refrigeração deverá estar afastado dos computadores ou ativos por aproximadamente 3 metros;

c) evitar a instalação de aparelhos do tipo comumente chamado de "*split"* sobre qualquer tipo de ativos. Sua área na parte inferior dos mesmos deverá manter-se livre, a fim de evitar situações de queda de líquidos que podem ocasionar uma parada não programada;

d) minimizar a infiltração de ar quente no corredor de insuflamento[4] de ar frio;

e) minimizar a mistura de correntes de ar frio e ar quente;

[4] Insuflamento é a entrada de ar refrigerado através de aberturas do piso elevado dentro de um datacenter.

f) minimizar o retorno de ar frio direto para a casa de máquinas de ar condicionado.

É prática comum para sistemas maiores a indicação de instalação de aparelho de controle automático de temperatura, umidade e câmaras de segurança.

Na Figura 20 abaixo, uma sala onde existe instalado um sistema de refrigeração de computadores de site de grande porte.

Figura 20: Sala de refrigeração
Fonte: HP (2007)

Como mostrado na Figura 20, o local onde se encontra instalado o sistema de refrigeração deve ser limpo, ter fácil acesso e ser bem iluminado. Logicamente esta realidade não é comum nos ambientes de pequenas empresas, mas o fato de tomar a idéia de ter todos os cuidados com esta parte em equipamentos de menor porte, poderá garantir um funcionamento estável e com pouca probabilidade de falhas.

4.4.2.2 Sistema de incêndio

Quanto ao sistema de incêndio, é recomendável a adoção de extintores do tipo CO2 em local visível, sinalizado e fora da circulação de pessoal. A sinalização do local do mesmo será realizada com uma pintura ou faixa no chão na cor amarela medindo 10 cm, em uma área de um x um metro.

Datacenter maiores poderão optar pela instalação de gás *inergem*[5] que retiram todo o oxigênio do ambiente e *spriklers*[6] próprios para esta utilização.

Abaixo, na Figura 21 uma ilustração de extintor do tipo CO2, posicionado de forma correta dentro de um ambiente de *datacenter*.

Figura 21: Extintor de incêndio
Fonte: HP (2007)

Como mostrado na Figura 21, a fácil localização e identificação do tipo de extintor utilizado poderá ser extremamente útil em situações de crise.

4.4.2.3 Sistema de segurança física

É recomendado um rígido controle de acesso dentro do *datacenter* ou no local onde o sistema ou os ativos estão instalados. Muitos são os casos de situações de indisponibilidade de sistemas que são geradas pela não observância deste item.

[5] Gás inergem é um tipo de gás que retira o oxigênio do ambiente, facilitando o controle de incêndio dentro de ambientes fechados.
[6] Spriklers são componentes contra incêndio instalados dentro do ambiente que são automaticamente acionados.

Existem sistemas de acesso de fácil adaptação e controle tais como cartões eletrônicos de acesso a ambientes e acesso controlado por leitura digital.

Nas Figuras 22 e 23, uma ilustração de controle de acesso com código de barras com entrada com cartão e saída do ambiente do *datacenter* com sensor de presença.

Figura 22: Acesso via código
Fonte: HP (2007)

Figura 23: Porta de acesso *datacenter*
Fonte: HP (2007)

Como mostrado nas Figuras 22 e 23, implementar controle de acesso automático, poderá evitar situações de entrada de pessoal não autorizado, mudança de temperatura do ambiente ou situações menos desejáveis como roubo de equipamentos, espionagem industrial ou sabotagem. Outra forma de se obter tal controle, porém menos eficiente, é através de controle manual.

4.4.2.4 Segurança lógica *backups*

Todo o sistema de uma corporação está baseado na prevenção de situações de risco e em processos documentados e maduros. Tais processos incluem a realização de *backups* do sistema para garantir a qualquer momento o retorno dos dados em uma situação de falha.

É recomendada a adoção de políticas de *backups* condizentes, independente do tamanho da empresa. Abaixo são relacionadas algumas das quais poderão ser incluídas nos procedimentos de cópias.

a) utilização de unidades de fita do tipo DDS (Digital Data Storage) ou *Ultrium*[7] para a realização de *backups*;

b) recomenda-se a adoção de rodízio de mídias semanal com 2 ou 4 grupos de 6 fitas, sendo 5 mídias de segunda a sexta feira mais uma de *backup* semanal. De segunda a sexta feira utilizar somente as modificações diárias sendo que no final de semana, utilizar *backup* geral do sistema. Realizar o rodízio das mesmas, garantindo quando da interrupção do serviço, *backups* dos últimos 14 ou 29 dias respectivamente. Casos especiais de *backups* corporativos para armazenamento de históricos deverão ser tratados a parte.

c) realização de rodízio no armazenamento das mídias, que deverão ficar em local distinto do local onde se encontra instalado o servidor, garantindo a integridade dos dados corporativos em situações de catástrofe.

[7] Ultrium é uma nova tecnologia de gravação de mídias que oferece grande capacidade de armazenamento e confiabilidade. Esta nova tecnologia foi desenvolvida em conjunto HP, IBM e DELL.

4.5 *Hardware e Software*

4.5.1 Análise de Falhas Potenciais de *Hardware e Software*

Muitas são as falhas que podem ocorrer em um sistema de *hardware* e de *software*, desde situações onde existe a parada parcial, total ou de falhas intermitentes do sistema que comprometam a disponibilidade dos equipamentos.

Por se tratar de inúmeras possibilidades de falhas, foram selecionadas algumas delas e podem ter sua origem nos seguintes fatores:

a) falha no disco de sistema operacional do servidor;

b) falha no disco de sistema de produção do servidor;

c) falha na comunicação dos ativos de rede;

d) falha na fonte de alimentação do servidor;

e) falha na CPU do servidor;

f) falha no backup do servidor;

g) falhas intermitentes no sistema do servidor;

h) falhas na memória do servidor.

Dentre as soluções que podem ser realizadas para a otimização do sistema, relacionam-se técnicas que permitem segurança e confiabilidade em caso de falhas de *software* em um sistema utilizando o *GNU/Linux*.

4.5.2 Recomendações Sistema de *Hardware e Software*

Existem vários módulos de um computador que devem ser tomados cuidados especiais quando da implementação de um sistema, como segue nas seções seguintes.

4.5.2.1 Disco rígido

Para sistemas de alta disponibilidade, os discos internos onde o sistema está instalado devem possuir dualidade de recursos, ou seja, utilizando a técnica de *mirror* ou de *Raid*. Para tanto, mais de um disco necessariamente deverão ser adquiridos e configurados de acordo com as necessidades da aplicação.

Normalmente, discos de sistema são implementados utilizando a técnica de *mirror* em *Raid* 1+ 0 utilizando para tal dois discos rígidos e áreas de sistemas em *Raid* 1+0 ou *Raid* 5, dependendo da área necessária e da quantidade de recursos disponíveis. Deve-se considerar também que na utilização da técnica de *mirror* em *Raid* 5, a quantidade de recursos disponibilizados de disco será menor do que utilizando a técnica de *mirror* em *Raid* 1+0.

Basicamente quando é empregada a técnica de Raid 1+0, utilizam-se dois discos na configuração com perda de espaço de disco de 50%. Na técnica de Raid-5, utiliza-se no mínimo 3 discos, e sua perda de espaço fica em 33%, baixando na proporção que são agregados mais discos (4 discos ficando com área indisponível de aproximadamente 25% e assim sucessivamente).

A forma desta implementação poderá ser tanto em *hardware* adquirindo controladoras específicas para esta atividade ou utilizando os recursos de *software* do sistema operacional *GNU/Linux*. Nos tópicos seguintes, será abordada a forma de configuração deste recurso em *software*.

4.5.2.2 Memória

Com referência a utilização e configuração de memória, o sistema deve prever a expansibilidade do mesmo, utilizando configurações consistentes com a quantidade de usuários que irão trabalhar no sistema. Utilizar quando da configuração de memória a opção por adquirir pares de memória, que garantam um sistema trabalhando com redução de recursos de memória quando da falha de algum módulo do equipamento.

4.5.2.3 System/CPU

Com referência a *System/CPU,* optar por adquirir equipamentos com mais de uma CPU, podendo ser *Dual Core* ou *Systems* com possibilidade de instalação de mais de um núcleo de processador, garantindo desta forma quando da falha de alguma CPU, a disponibilidade do sistema mesmo com redução de recursos.

Dimensionar o equipamento de maneira correta no momento do planejamento de expansões e realizando comparações e testes em equipamentos de diferentes fabricantes podem evitar no futuro situações indesejáveis quanto à capacidade de processamento do sistema.

4.5.2.4 Rede *ethernet*

Optar sempre por instalar nos equipamentos no mínimo duas placas de rede *ethernet* em cada sistema. Estes recursos disponibilizados permitirão que as mesmas sejam agregadas em um ponto único de conexão do sistema com o meio externo através de técnicas disponíveis dentro do ambiente *GNU/Linux*, garantindo a disponibilidade do sistema em caso de falha de algum ativo do sistema de rede.

4.5.2.5 Fontes de alimentação

Instalar sempre que possível equipamento que disponha de fontes redundantes em seu *hardware*, alternando-se a conexão dos cabos de alimentação em fontes externas distintas (*Nobreaks*, UPS e outros).

4.5.2.6 *Mirror* de disco rígido por *software* utilizando *Raid*

Utilizando os recursos do *GNU/Linux*, podem-se adequar as necessidades de segurança através da utilização de cópias lógicas por *Raid* de áreas com dados, capacitando ao sistema um grau diferenciado de segurança nos dados.

No exemplo demonstrado a seguir, será criado em um segundo disco físico instalado no sistema, uma área utilizando *Raid*-5, dividindo-se este disco em quatro partes de 400 blocos sucessivos, sendo 3 em *Raid*-5 e um em *hot-spare*[8]. Esta divisão poderá ser considerada padrão quando estiver instalado mais de um disco no sistema, apenas indicando os seus respectivos dispositivos quando da criação do *raid*, em procedimento padrão. Maiores detalhes utilizar os comandos de manuais dentro do sistema (#man) com suas opções.

A seguir, demonstração da configuração de uma área em disco utilizando o recurso de *Raid*.

(i) Identificando os dispositivos configurados no sistema:

```
[root@localhost etc]# df
[root@localhost etc]# fdisk -l
```

(ii) Utilizar a opção n, de *add* a *new partition*. Quando da utilização da configuração, interações com as respostas estão destacadas em negrito.

```
[root@localhost etc]# fdisk /dev/sdb

Command (m for help): m
Command action

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)

Partition number (1-4): 1
First cylinder (1-4464, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-4464, default 4464): 400

Partition number (1-4): 2
First cylinder (401-4464, default 401):
Using default value 401
Last cylinder or +size or +sizeM or +sizeK (401-4464, default 4464): 800

Command (m for help): t
```

[8] *Hot-spare* é utilizado para identificar um disco reserva que fica em estado de *stand-by*, pronto para entrar em funcionamento quando da verificação de alguma falha nos discos rígidos.

```
Partition number (1-4): 2 (Nova Partição criada)
Hex code (type L to list codes): fd
Changed system type of partition 2 to fd (Linux raid autodetect)

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)

Partition number (1-4): 3  (Nova Partição criada)
First cylinder (801-4464, default 801):
Using default value 801
Last cylinder or +size or +sizeM or +sizeK (801-4464, default 4464): 1200

Command (m for help): t
Partition number (1-4): 3
Hex code (type L to list codes): fd
Changed system type of partition 3 to fd (Linux raid autodetect)

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)

Selected partition 4
First cylinder (1201-4464, default 1201):
Using default value 1201
Last cylinder or +size or +sizeM or +sizeK (1201-4464, default 4464): 1600

Command (m for help): t
Partition number (1-4): 4
Hex code (type L to list codes): fd
Changed system type of partition 4 to fd (Linux raid autodetect)

Command (m for help): p

Disk /dev/sdb: 36.7 GB, 36722061312 bytes
255 heads, 63 sectors/track, 4464 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks      Id  System
/dev/sdb1               1         400     3212968+   fd  Linux raid autodetect
/dev/sdb2             401         800     3213000    fd  Linux raid autodetect
/dev/sdb3             801        1200     3213000    fd  Linux raid autodetect
/dev/sdb4            1201        1600     3213000    fd  Linux raid autodetect

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

(iii) Digitar o comando # partprobe para garantir a gravação das informações no kernel das tabelas de partições:

```
[root@localhost etc]# partprobe
```

(iv) Verificar as áreas criadas em *raid*:

```
[root@localhost etc]# fdisk -l /dev/sdb | grep raid
/dev/sdb1               1         400     3212968+   fd  Linux raid autodetect
```

```
/dev/sdb2             401          800     3213000    fd  Linux raid autodetect
/dev/sdb3             801         1200     3213000    fd  Linux raid autodetect
/dev/sdb4            1201         1600     3213000    fd  Linux raid autodetect
```

(v) Utilizar o comando de criação de *Raid* em /dev/md0, em *raid* nível 5, com 3 dispositivos recém criados /dev/sdb1, /dev/sdb2 e /dev/sdb3 e um hot spare /dev/sdb4 e aguardar finalização do processo.

```
[root@localhost etc]# mdadm --create /dev/md0 --level=5 --raid-devices=3
/dev/sdb1 /dev/sdb2 /dev/sdb3 --spare-devices=1 /dev/sdb4

mdadm: array /dev/md0 started.
```

(vi) Verificar a área sendo criada e aguardar chegar em 100%.

```
[root@localhost etc]# more /proc/mdstat

Personalities : [raid5]
md0 : active raid5 sdb3[3] sdb4[4] sdb2[1] sdb1[0]
      6425728 blocks level 5, 64k chunk, algorithm 2 [3/2] [UU_]
      [=>...................]  recovery =  5.5% (179328/3212864) finish=9.8min
speed=5120K/sec
unused devices: <none>

[[root@localhost etc]#
```

(vii) Realizar a criação do novo filesystem no dispositivo criado /dev/md0.

```
root@localhost etc]# mke2fs -j -m 0 /dev/md0

mke2fs 1.35 (28-Feb-2004)
warning: 800 blocks unused.
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
804384 inodes, 1605632 blocks
0 blocks (0.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=1644167168
49 block groups
32768 blocks per group, 32768 fragments per group
16416 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 24 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

(viii) Utilizar o comando *mdadm* para gerar um arquivo com o *array* recém criado e arquiva-lo em /etc/mdadm.conf para futura identificação do *array* a que pertence.

```
[root@localhost etc]# mdadm -detail -scan >> /etc/mdadm.conf
```

```
[root@localhost etc]# more /etc/mdadm.conf
ARRAY         /dev/md0         level=raid5         num-devices=3         spares=1
UUID=40ce6378:4422c033:ca259a3c:72b53117
```

(vix) Montar o recém criado dispositivo no diretório /array:

```
[root@localhost /]# mkdir array
[root@localhost /]# mount /dev/md0 /array
[root@localhost /]# df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/mapper/VolGroup00-LogVol00
45832608           4066212    39438232             10% /
/dev/sda2          101105         13126        82758   14% /boot
none                              516748                 0          516748      0%
/dev/shm
/dev/md0                6321524          45344       6276180     1% /array
```

(x) Ativá-lo criando esta linha no final do arquivo #/etc/fstab para que quando do *reboot* do sistema esta área sempre fique ativa.

```
[root@localhost /]# cd /etc
[root@localhost etc]# ed fstab
a
/dev/md0        /array  ext3    defaults        1       2
w
770
q
```

```
[root@localhost etc]# more fstab
# This file is edited by fstab-sync - see 'man fstab-sync' for details
/dev/VolGroup00/LogVol00 /                       ext3    defaults        1 1
LABEL=/boot       /boot                   ext3     defaults        1 2
none                    /dev/pts           devpts     gid=5,mode=620  0 0
none                    /dev/shm           tmpfs      defaults        0 0
none                    /proc                   proc      defaults        0 0
none                    /sys                    sysfs     defaults        0 0
/dev/VolGroup00/LogVol01       swap          swap    defaults        0 0
/dev/hdb                /media/cdrecorder       auto
pamconsole,fscontext=system_u:object_r:removable_t,exec,noauto,managed 0 0
/dev/md0              /array  ext3    defaults        1       2
```

(xi) Testar a nova área copiando arquivos (no exemplo abaixo uma área do /var)..

```
[root@localhost etc]# cd /var
[root@localhost var]# cp -R log /array
[root@localhost var]# cd /array
[root@localhost array]# ls
log  lost+found
```

(xii) Verificar os discos associados ao diretório recém montado /array.

```
[root@localhost array]# fdisk -l /dev/sdb | grep raid /dev/sdb1
1         400     3212968+  fd  Linux raid autodetect
/dev/sdb2             401         800     3213000   fd  Linux raid autodetect
/dev/sdb3             801        1200     3213000   fd  Linux raid autodetect
/dev/sdb4            1201        1600     3213000   fd  Linux raid autodetect
```

(xiii) Verificar a integridade da configuração realizada, forçando uma falha em um dos membros do *Raid*. No exemplo abaixo /dev/sdb2 e após verificando nos logs do sistema.

```
[root@localhost array]# mdadm --manage /dev/md0 --fail /dev/sdb2
mdadm: set /dev/sdb2 faulty in /dev/md0

[root@localhost array]# tail /var/log/messages
Aug  2 11:04:36 localhost kernel:  disk 2, o:1, dev:sdb3
Aug  2 11:04:36 localhost kernel: RAID5 conf printout:
Aug  2 11:04:36 localhost kernel:  --- rd:3 wd:2 fd:1
Aug  2 11:04:36 localhost kernel:  disk 0, o:1, dev:sdb1
Aug  2 11:04:36 localhost kernel:  disk 1, o:1, dev:sdb4
Aug  2 11:04:36 localhost kernel:  disk 2, o:1, dev:sdb3
Aug  2 11:04:36 localhost kernel: md: syncing RAID array md0
Aug  2 11:04:36 localhost kernel: md: minimum guaranteed reconstruction speed:
1000 KB/sec/disc.
Aug  2 11:04:36 localhost kernel: md: using maximum available idle IO bandwith
(but not more than 200000 KB/sec) for reconstruction.
Aug  2 11:04:36 localhost kernel: md: using 128k window, over a total of 3212864
blocks.
```

(xiv) Verificar a evolução da reconstrução do *Raid* com os comandos de administração.

```
[root@localhost log]# more /proc/mdstat
Personalities : [raid5]
md0 : active raid5 sdb3[2] sdb4[3] sdb2[4](F) sdb1[0]
6425728 blocks level 5, 64k chunk, algorithm 2 [3/2] [U_U]
[======>.............]     recovery  =  30.6%  (986240/3212864)  finish=7.4min
speed=4956K/sec
unused devices: <none>

[root@localhost log]# mdadm --misc --detail /dev/md0
/dev/md0:
Version : 00.90.01
Creation Time : Thu Aug  2 10:30:30 2007
Raid Level : raid5
Array Size : 6425728 (6.13 GiB 6.58 GB)
Device Size : 3212864 (3.06 GiB 3.29 GB)
Raid Devices : 3
Total Devices : 4
Preferred Minor : 0
Persistence : Superblock is persistent
Update Time : Thu Aug  2 11:06:13 2007
State : clean, degraded, recovering
Active Devices : 2
Working Devices : 3
Failed Devices : 1
Spare Devices : 1
Layout : left-symmetric
Chunk Size : 64K
Rebuild Status : 57% complete
UUID : 40ce6378:4422c033:ca259a3c:72b53117
Events : 0.31

Number   Major   Minor   RaidDevice State
0       8       17        0      active sync   /dev/sdb1
```

```
1       0        0        -      removed
2       8       19        2      active sync   /dev/sdb3
3       8       20        1      spare rebuilding   /dev/sdb4
4       8       18        -      faulty   /dev/sdb2
[root@localhost log]# mdadm --misc --detail /dev/md0
/dev/md0:
Version : 00.90.01
Creation Time : Thu Aug  2 10:30:30 2007
Raid Level : raid5
Array Size : 6425728 (6.13 GiB 6.58 GB)
Device Size : 3212864 (3.06 GiB 3.29 GB)
Raid Devices : 3
Total Devices : 4
Preferred Minor : 0
Persistence : Superblock is persistent
Update Time : Thu Aug  2 11:15:40 2007
State : clean
Active Devices : 3
Working Devices : 3
Failed Devices : 1
Spare Devices : 0
Layout : left-symmetric
Chunk Size : 64K
UUID : 40ce6378:4422c033:ca259a3c:72b53117
Events : 0.32

Number   Major   Minor   RaidDevice State
0       8       17        0      active sync   /dev/sdb1
1       8       20        1      active sync   /dev/sdb4
2       8       19        2      active sync   /dev/sdb3
3       8       18        -      faulty        /dev/sdb2
```

(xv) Remover o disco com defeito da configuração do *Raid*.

```
[root@localhost /]# mdadm --manage /dev/md0 --remove /dev/sdb2 mdadm: hot
removed /dev/sdb2
```

(xvi) Acrescentar ao *Raid* o novo participante e verificar a evolução da reconstrução do *Raid*. Neste caso estará sendo adicionado o mesmo disco /dev/sdb2.

```
[root@localhost /]# mdadm --manage /dev/md0 --add /dev/sdb2
mdadm: hot added /dev/sdb2
[root@localhost /]# mdadm --misc --detail /dev/md0
/dev/md0:
Version : 00.90.01
Creation Time : Thu Aug  2 10:30:30 2007
Raid Level : raid5
Array Size : 6425728 (6.13 GiB 6.58 GB)
Device Size : 3212864 (3.06 GiB 3.29 GB)
Raid Devices : 3
Total Devices : 4
Preferred Minor : 0
Persistence : Superblock is persistent
Update Time : Thu Aug  2 11:18:19 2007
State : clean
Active Devices : 3
Working Devices : 4
Failed Devices : 0
Spare Devices : 1
Layout : left-symmetric
Chunk Size : 64K
```

```
UUID : 40ce6378:4422c033:ca259a3c:72b53117
Events : 0.34
Number   Major   Minor   RaidDevice State
0       8       17        0      active sync   /dev/sdb1
1       8       20        1      active sync   /dev/sdb4
2       8       19        2      active sync   /dev/sdb3
3       8       18        -      spare   /dev/sdb2
```

4.5.2.7 Bond de placas de rede *Ethernet*

Com o constante aumento da demanda e crescimento do tráfico em rede *ethernet*, em muitas situações pode ocorrer a saturação da banda disponível, ocasionando lentidão, perdas de pacotes e uma latência indesejada nas interfaces. Além disto, a utilização de uma placa de rede como ponto de entrada acaba por caracterizar um ponto único de falha (SPOF).

Para solucionar esta situação, utilizando o recurso de *Bonding* de interfaces pode-se agregar mais de um dispositivo de rede em um único canal, respondendo a um único endereço *ethernet*, garantindo desta forma disponibilidade de recursos, capacidade de atendimento as solicitações do sistema a um baixo custo.

Na Figura 24, ilustração da utilização de *bonding* em placas de rede *ethernet*.

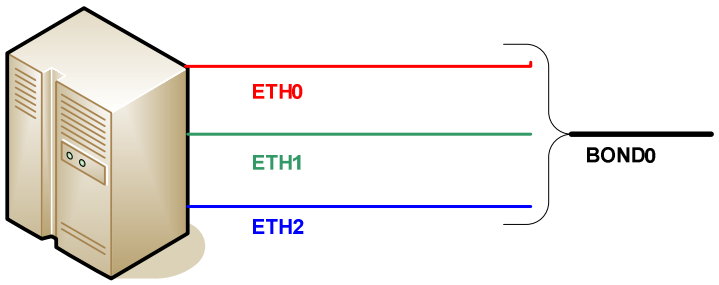


Figura 24: *Bonding* de placas de rede *ethernet*
Fonte: Autor

Como mostrado na Figura 24, a utilização de *bonding* de placas de rede ethernet irá agregar as interfaces de rede *ETH0, ETH1* e *ETH2* em um único canal

com endereço de rede único, garantindo disponibilidade em caso de inoperabilidade de alguma conexão.

A seguir, procedimento para configuração do recurso de *bonding* em placas de rede *ethernet*.

(i) Identificar os dispositivos de redes instados no sistema.

```
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0B:CD:52:B1:8A
          inet addr:10.1.1.17  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::20b:cdff:fe52:b18a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:554 (554.0 b)  TX bytes:1240 (1.2 KiB)
          Interrupt:50
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1458 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1458 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1580922 (1.5 MiB)  TX bytes:1580922 (1.5 MiB)

[root@localhost ~]# ifconfig eth2
eth2      Link encap:Ethernet  HWaddr 00:11:0A:E9:5C:A9
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:233
```

(ii) Ativar as interfaces de rede (caso não estejam), com os seguintes comandos.

```
[root@localhost ~]# ifconfig eth2 up
[root@localhost ~]# ifconfig eth0 up
```

(iii) Verificar se ambas as placas estão ativas.

```
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0B:CD:52:B1:8A
          inet addr:10.1.1.17  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::20b:cdff:fe52:b18a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:17 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4072 (3.9 KiB)  TX bytes:1240 (1.2 KiB)
          Interrupt:50
```

```
eth2      Link encap:Ethernet  HWaddr 00:11:0A:E9:5C:A9
          inet6 addr: fe80::211:aff:fee9:5ca9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:398 (398.0 b)
          Interrupt:233
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1485 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1485 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1585270 (1.5 MiB)  TX bytes:1585270 (1.5 MiB)
```

(iv) Reiniciar o serviço de rede do sistema através do comando # service network restart.

```
[root@localhost network-scripts]# service network restart
Shutting down interface eth0:                              [  OK  ]
Shutting down interface eth2:                              [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Setting network parameters:                                [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface eth0:                                [  OK  ]
Bringing up interface eth2:                                [  OK  ]
```

(v) Posicionar-se no diretório /etc/sysconfig/network-scripts e criar o arquivo ifcfg-bond0. No exemplo utilizado, foi atribuído o endereço IP 10.1.0.5, Máscara de rede 255.255.0.0 e Gateway 10.1.0.1

```
[root@localhost ~]# cd /etc/sysconfig/network-scripts
[root@localhost network-scripts]# ls
ifcfg-eth0        ifdown-ipv6  ifup-aliases  ifup-plip      ifup-wireless
ifcfg-eth2        ifdown-isdn  ifup-ib       ifup-plusb   init.ipv6-global
ifcfg-lo          ifdown-post  ifup-ippp     ifup-post    network-functions
ifdown            ifdown-ppp   ifup-ipsec    ifup-ppp     network-functions-ipv6
ifdown-aliases    ifdown-sit   ifup-ipv6     ifup-routes
ifdown-ippp       ifdown-sl    ifup-ipx      ifup-sit
ifdown-ipsec      ifup         ifup-isdn     ifup-sl

[root@localhost network-scripts]# ed ifcfg-bond0
ifcfg-bond0: No such file or directory
a
DEVICE=bond0
BOOTPROTO=static
ONBOOT=yes
IPADDR=10.1.0.5
NETMASK=255.255.0.0
GATEWAY=10.1.0.1
.
w
94
q
```

(vi) ConFigurar as outras interfaces (eth0 e eth2) de acordo com o item a acima, colocando a seguinte linha: MASTER=bond0 e retirando os endereços ethernet e de máscara de rede.

```
[root@localhost network-scripts]# ed ifcfg-eth0
# Broadcom Corporation NetXtreme BCM5701 Gigabit Ethernet
DEVICE=eth0
ONBOOT=yes
MASTER=bond0
BOOTPROTO=none
HWADDR=00:0B:CD:52:B1:8A
USERCTL=no
IPV6INIT=no
PEERDNS=yes
TYPE=Ethernet
[root@localhost network-scripts]# more ifcfg-eth0
# Broadcom Corporation NetXtreme BCM5701 Gigabit Ethernet
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=none
HWADDR=00:0B:CD:52:B1:8A
MASTER=bond0
USERCTL=no
IPV6INIT=no
PEERDNS=yes
TYPE=Ethernet
[root@localhost network-scripts]# ed ifcfg-eth2
# Broadcom Corporation NetXtreme BCM5701 Gigabit Ethernet
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=none
HWADDR=00:0B:CD:52:CA:42
MASTER=bond0
USERCTL=no
IPV6INIT=no
PEERDNS=yes
TYPE=Ethernet
```

(vii) Criar a interface ifcfg-bond0 no diretório /etc/sysconfig/network-scripts, colocando as informações de IP address, máscara e gateway.

```
[root@localhost network-scripts]# ed ifcfg-bond0
ifcfg-bond0: No such file or directory
a
DEVICE=bond0
BOOTPROTO=static
ONBOOT=yes
IPADDR=10.1.0.5
NETMASK=255.255.0.0
GATEWAY=10.1.0.1
.
w
94
q
```

(vii) Posicionar-se no diretório /etc e adicionar as linhas no final do arquivo /etc/modprobe.conf.

alias bond0 bonding

options bond0 miimon=150

O Valor de miimon=150 é o tempo em segundos de intervalo de pooling para identificar uma falha no link.

```
[root@localhost network-scripts]# cd /etc
[root@localhost etc]# ed modprobe.conf
420

a
alias bond0 bonding
options bond0 miimon=150
.
w
465
Q
```

(viii) Reiniciar o serviço de rede para ativar a nova configuração.

```
[root@localhost etc]# service network restart
Shutting down interface eth0:                              [  OK  ]
Shutting down interface eth2:                              [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Setting network parameters:                                [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface bond0:                               [  OK  ]
```

(vix) Verificar se as interfaces estão rodando em SLAVE nas interfaces eth0 e eth2 após a configuração. Em seguida, realizar o comando ping para certificar-se de que as interfaces estão respondendo corretamente.

```
[root@localhost etc]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0B:CD:52:B1:8A
          inet6 addr: fe80::20b:cdff:fe52:b18a/64 Scope:Link
          UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:393 (393.0 b)  TX bytes:557 (557.0 b)
          Interrupt:50

[root@localhost etc]# ifconfig eth2
eth2      Link encap:Ethernet  HWaddr 00:0B:CD:52:B1:8A
          inet6 addr: fe80::20b:cdff:fe52:b18a/64 Scope:Link
          UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:311 (311.0 b)  TX bytes:462 (462.0 b)
          Interrupt:233

[root@localhost etc]# ping 10.1.0.5
PING 10.1.0.5 (10.1.0.5) 56(84) bytes of data.
64 bytes from 10.1.0.5: icmp_seq=0 ttl=64 time=0.051 ms
```

```
64 bytes from 10.1.0.5: icmp_seq=1 ttl=64 time=0.048 ms
--- 10.1.0.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.048/0.049/0.051/0.005 ms, pipe 2
```

i) Verificar o encapsulamento das interfaces com a interface MASTER bond0.

```
[root@localhost etc]# ifconfig -a
bond0     Link encap:Ethernet  HWaddr 00:0B:CD:52:B1:8A
          inet addr:10.1.0.5  Bcast:10.1.255.255  Mask:255.255.0.0
          inet6 addr: fe80::200:ff:fe00:0/64 Scope:Link
          UP BROADCAST RUNNING MASTER MULTICAST  MTU:1500  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:704 (704.0 b)  TX bytes:1019 (1019.0 b)
eth0      Link encap:Ethernet  HWaddr 00:0B:CD:52:B1:8A
          inet6 addr: fe80::20b:cdff:fe52:b18a/64 Scope:Link
          UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:393 (393.0 b)  TX bytes:557 (557.0 b)
          Interrupt:50
eth2      Link encap:Ethernet  HWaddr 00:0B:CD:52:B1:8A
          inet6 addr: fe80::20b:cdff:fe52:b18a/64 Scope:Link
          UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:311 (311.0 b)  TX bytes:462 (462.0 b)
          Interrupt:233
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1595 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1595 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1592728 (1.5 MiB)  TX bytes:1592728 (1.5 MiB)
sit0      Link encap:IPv6-in-IPv4
          NOARP  MTU:1480  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

(x) Neste ponto todos os dispositivos estão operando corretamente na configuração de *bonding* e já pode para efeito de teste, desligar um dos cabos de rede conectado a uma interface de rede. O equipamento deverá continuar respondendo pela outra interface que ficou ativa. Este teste poderá ser realizado pelo número de vezes desejado, desde que ao menos um dispositivo fique conectado à rede.

5 *CLUSTER GNU/LINUX* DE ALTA DISPONIBILIDADE

5.1 Arquitetura do *Cluster*

O *cluster* implementado foi o de alta disponibilidade, que garante a manutenção dos recursos ou serviços configurados nos *nodos* do *cluster* conforme Figura 25.

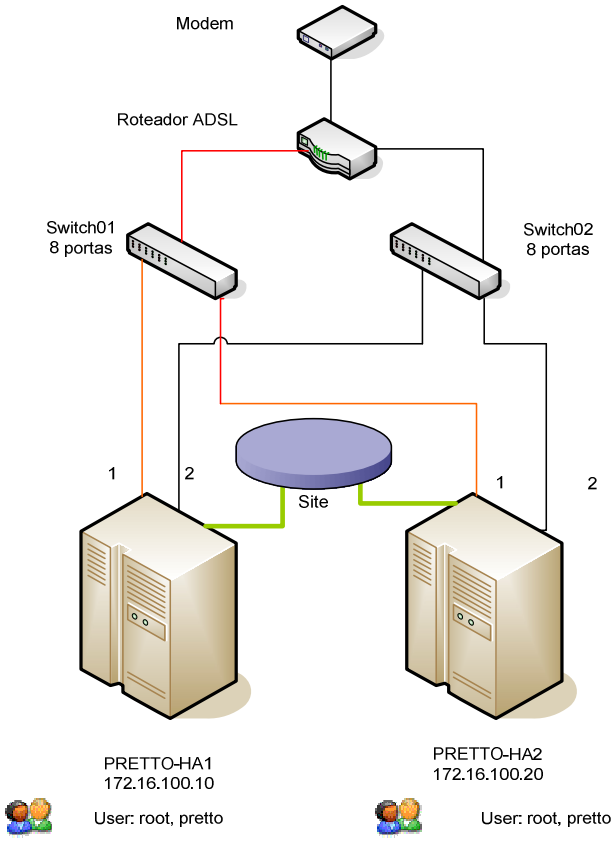


Figura 25: Modelo implementado
Fonte: Autor (2007)

Na Figura 25 é apresentada uma descritivo do *cluster* montado e redundâncias implementadas.

5.2 Montagem do *Cluster*

5.2.1 Relatório sobre a montagem do *cluster*

Para a montagem do *cluster*, alguns fatores foram levados em conta quanto à disponibilidade desejada. Os pré-requisitos necessários para a formação do *cluster* para entrar em produção são os seguintes:

a) servidores que compõem o *cluster* com configuração mínima de 1GB de memória para rodar o *software CentOS* e o *Cluster Suite*;

b) *switches* ou *hubs* para a conexão das placas de rede *ethernet*;

c) *switches* de rede ou de fibra gerenciado (opcional) para servirem de *fencing* configurável dentro do *cluster*.

No *cluster* montado, optou-se por disponibilizar máquinas virtuais em função da disponibilidade de *hardware.* Todas as características de componentes necessários à formação do *cluster* foram preservadas, garantindo a perfeita execução dos processos de implementação e gerenciamento dos *nodos* configurados.

5.2.1.1 Instalação de placas de rede

Instalação para redundância no sistema de 2 placas de rede modelo *Broadcom Corporation NetXtreme* BCM5701 *Gigabit Ethernet* em cada servidor, proporcionando dualidade de recursos com disponibilidade para que quando da interrupção de alguma conexão *ethernet* o sistema permaneça operante.

Servidor 1 com as seguintes configurações:
IP Address: 172.16.100.10
Gateway: 172.16.100.1
Net Mask: 255.255.255.0

Servidor 2 com as seguintes configurações

IP Address: 172.16.100.20

Gateway: 172.16.100.1

Net Mask: 255.255.255.0

5.2.1.2 Instalação de discos rígidos

Instalado dois discos rígido sendo um com tecnologia SATA2 de capacidade de 160GB e um segundo disco com tecnologia SCSI de 40GB sendo ambos os discos do fabricante Sansung. Tais discos foram instalados da seguinte forma:

O disco SATA2 para realizar *mirror* de *Raid* 5 por *software*, utilizando os recursos do Linux referente a discos e cdrom ficando distribuídos da seguinte forma.

Servidor 1:

```
/dev/VolGroup00/LogVol00 /          ext3      defaults         1 1    (SATA2)
LABEL=/boot       /boot             ext3       defaults         1 2   (SATA2)
/dev/VolGroup00/LogVol01    swap   swap       defaults         0 0   (SATA2)
/dev/hdb       /media/cdrecorder              auto                    (CDROM)
/dev/md0       /array              ext3      defaults         1 2    (SCSI)
```

Servidor 2:

```
/dev/VolGroup00/LogVol00 /            ext3      defaults      1 1     (SATA2)
LABEL=/boot       /boot               ext3      defaults       1 2    (SATA2)
/dev/VolGroup00/LogVol01     swap     swap       defaults       0 0    (SATA2)
/dev/hdb           /media/cdrecorder            auto                   (CDROM)
```

5.2.1.3 Rede elétrica

Conexão de cada servidor em estabilizadores distintos e na mesma rede elétrica.

No *cluster* configurado, por questões de disponibilidade de recursos e não tratar-se de ambiente produtivo, sendo apenas de laboratório para a demonstração do funcionamento do *cluster*, não foi instalado sistema ininterrupto de energia ou em entradas distintas de energia elétrica.

5.2.1.4 Interligação *Switches*

Instalado dois *switches* de 8 portas, modelo para redundância, com cada canal de rede *ethernet* de cada servidor sendo conectado a cada *switch*.

Por questão de disponibilidade de recursos e não tratar-se de ambiente produtivo, os mesmos não foram instalados com recurso de disponibilidade elétrica dual.

5.2.1.5 Instalação do *software*

O sistema operacional instalado foi o *GNU/Linux CentOS-5*, que já vem incorporado o *software* de *cluster*, o *Cluster Suite*. Sua versão para download e documentação está disponível no site do *CentOS* (2007).

5.2.1.6 Configuração do sistema

Algumas etapas deverão ser previamente observadas sob forma a evitar problemas na configuração do *cluster*. Sua documentação de implementação, configuração e gerenciamento do *CentOS Cluster Suíte* encontra-se no site do *CentOS* (2007).

Basicamente, o *software Cluster Suíte*, trabalha com agentes que realizam o gerenciamento dos recursos disponíveis, disponibilizando-os de forma a atender as requisições dos usuários. A forma como estes agentes são gerenciados é denominado de *fencing* cuja função principal é de isolar o *nodo* danificado a partir de alguma anormalidade constatada. Tais anormalidades podem ser originadas por:

a) queda de energia dos servidores ou outros ativos;
b) corrupção de dados dos servidores;
c) falha de comunicação dos ativos;
d) outras falhas que provoquem a inoperabilidade dos equipamentos.

Quanto aos recursos de gerenciamento do *Cluster Suíte*, ferramentas administrativas são úteis tais como:

a) *cman_tool*: programa de gerenciamento do *cluster* que provê a capacidade de agregar os *nodos*, retirar *nodos*, indisponibilizar *nodos*;

b) *fence_tool*: programa utilizado para o gerenciamento do ativo que irá realizar a inclusão ou exclusão de algum *nodo*;

c) *ccs_tool*: gerenciamento dos arquivos de configuração do *cluster* aos seus nodos de forma automática;

d) *clustat*: comando que mostra o *status* do *cluster*;

e) *clusvcadm*: permite ao administrador ativar, desativar, realocar e reiniciar os serviços de alta disponibilidade do *Cluster Suíte.*

Segue abaixo processo de configuração do *Cluster Suíte* do *CentOS* 5.

(i) Posicionar-se no diretório /etc e validar o arquivo /hosts com a configuração que está implementada no *cluster*, acrescentando os *nodos*:

```
[root@pretto-ha1 ~]# cd /etc
[root@pretto-ha1 etc]# more hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1       xen-pretto localhost.localdomain localhost
::1             localhost6.localdomain6 localhost6
172.16.100.10   pretto-ha1
172.16.100.20   pretto-ha2
[root@pretto-ha1 etc]#
```

(ii) Confirmar a resolução de nomes no sistema para os *nodos* do *cluster*:

```
[root@pretto-ha1 etc]# ping pretto-ha1
PING pretto-ha1 (172.16.100.10) 56(84) bytes of data.
64 bytes from pretto-ha1 (172.16.100.10): icmp_seq=1 ttl=64 time=0.023 ms
64 bytes from pretto-ha1 (172.16.100.10): icmp_seq=2 ttl=64 time=0.016 ms
[root@pretto-ha1 etc]#
[root@pretto-ha1 ~]# ping pretto-ha2
PING pretto-ha2 (172.16.100.20) 56(84) bytes of data.
64 bytes from pretto-ha2 (172.16.100.20): icmp_seq=1 ttl=64 time=0.196 ms
64 bytes from pretto-ha2 (172.16.100.20): icmp_seq=2 ttl=64 time=0.133 ms
```

(iii) rodar o comando para a configuração inicial do *cluster*:

```
[root@pretto-ha1 ~]# system-config-cluster
```

Figura 26: Tela inicial configuração
Fonte: *ClusterHA* (2007)

(iv) Clicar em *Create New Configuration* conforme Figura 26 onde irá aparecer a tela da Figura 27 onde deve ser indicado o nome do *cluster*. No exemplo foi utilizado **clusterha**. Não é necessário a seleção das outras opções de *multicast* e *quorum disk*.

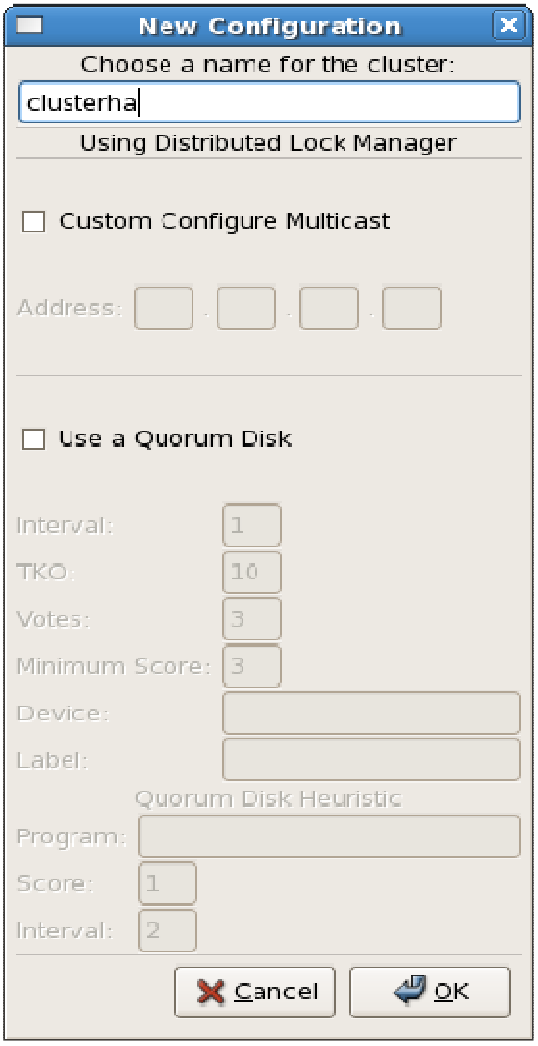


Figura 27: Configuração inicial *Cluster*
Fonte: *ClusterHA* (2007)

(v) Na seqüência da configuração irá ser apresentada a janela da Figura 28 onde o *software* será configurado, como segue;

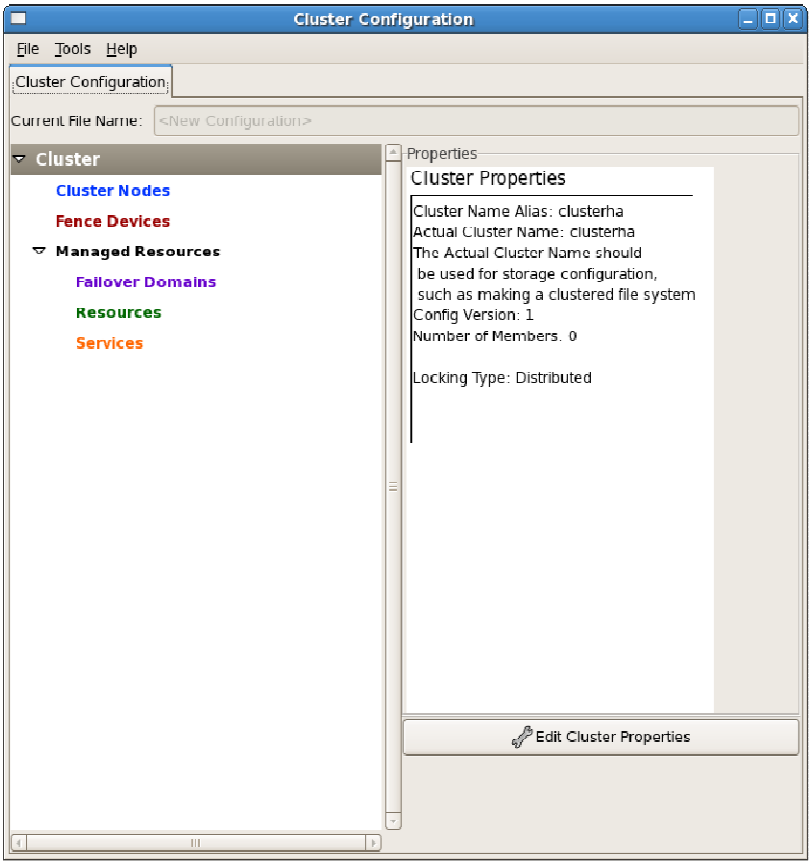


Figura 28: Configurando *Cluster*

Fonte: *ClusterHA* (2007)

Esta é a tela inicial de configuração do *cluster* onde existem todos os recursos disponíveis serão configurados.

(vi) Adicionar os *Clusters Nodes* conforme indica na Figura 29 informando os nomes das máquinas, previamente configurado no /etc/hosts. Em nosso exemplo, foram criados os *Clusters Nodes* **pretto-ha1** e **pretto-ha2**. A tabela com *Quorum Vote* não é necessário o seu preenchimento, pois o *software* implementará de forma automática.

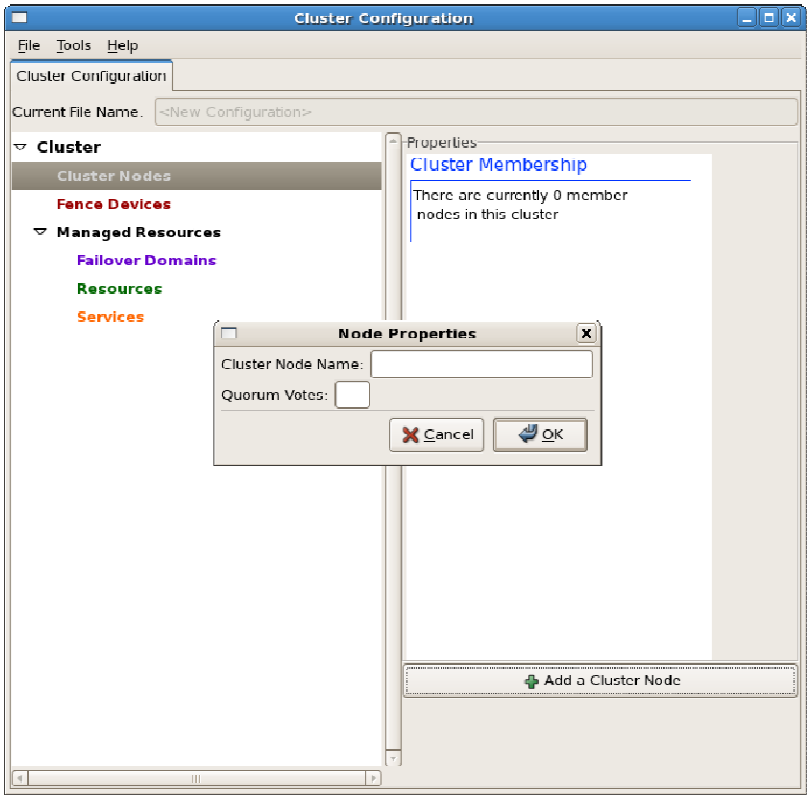


Figura 29: *Cluster Node Name*
Fonte: ClusterHA (2007)

(vii) Configurar um *fence device* para o *cluster*. Este é um ponto importante aa ser checado. A finalidade do *fence*, em português “cerca” é a de monitorar os ativos e em caso de falha, retirar o nó do *cluster,* cercando e isolando. Vários tipos de *fencing* são disponibilizados, como *switches ethernet* gerenciáveis, *switches* de fibra Brocade e Mcdata, sistemas ininterruptos de monitoração de energia APC, *storages* externos e outros. Deve ser verificado dentre os disponíveis para a configuração de *fencing* aquele que melhor atenda as necessidades do *cluster*. No *cluster* utilizado como demonstração, foi adaptado o *fence Brocade*, onde foram adaptados alguns parâmetros para o seu funcionamento. Na Figura 30 existem os campos que devem ser preenchidos.

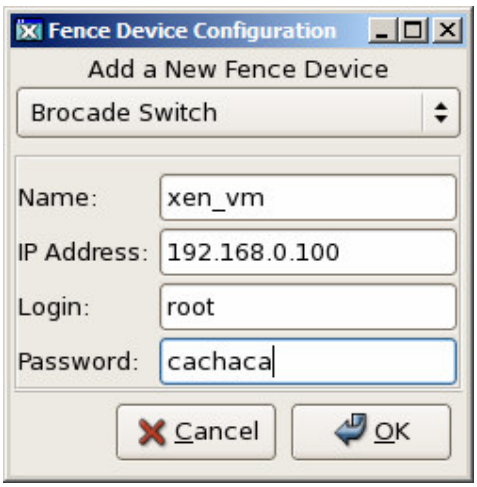


Figura 30: Adicionando um *Fence device*
Fonte: ClusterHA (2007)

(viii) Arquivo de configuração do *fence* utilizado na configuração do *cluster*, localizado no diretório /**sbin**/**brocade_fence** (diretório default) para esta topologia localizado no apêndice A, que poderá ser utilizado como modelo para futuras implementações de *cluster* . Este *shell script* (EDON, 2007) foi adaptado para o funcionamento do *cluster*.

(vix) Conforme Figura 31 selecionar dentro de *Managed Resources* o ítem *Failover Domain* e criar um *domain* em caso de falha. No exemplo conforme Figura 32 foi criado o *Failover Domain* **apache**.

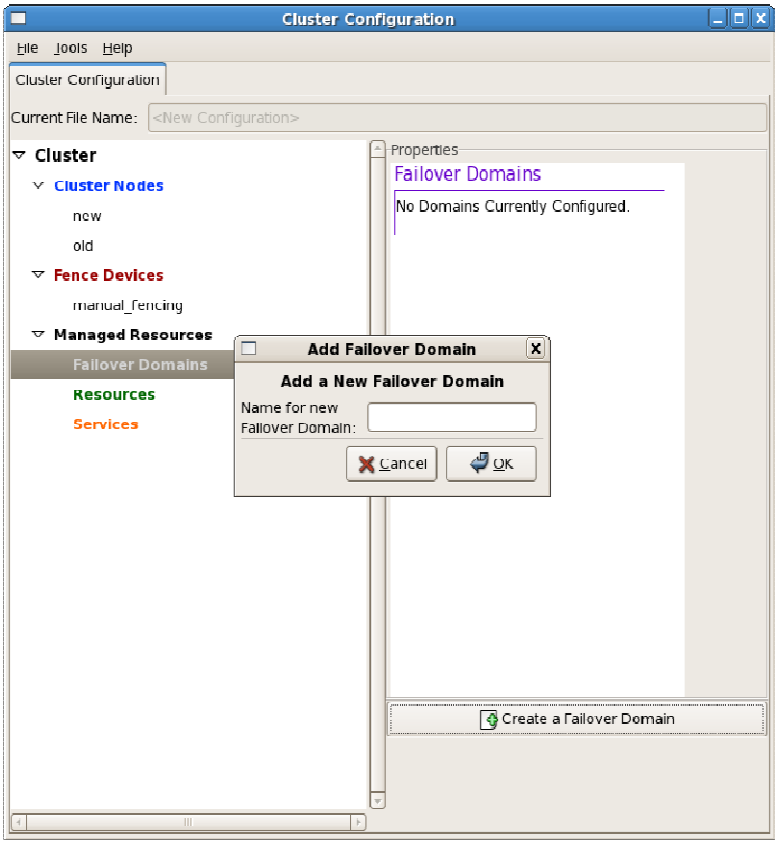


Figura 31: Adicionando um *Failover Domain*
Fonte: *ClusterHA* (2007)

Figura 32: Adicionando um nome ao *Failover Domain*
Fonte: *ClusterHA* (2007)

(x) Dentro deste *Failover Domain* **apache** acrescentar os *nodos* que estão no sistema. No exemplo da Figura 33, foram adicionados o *node* **pretto-ha1** e o *node* **pretto-ha2** e selecionar o item *Restrict Failover To This Domains Members*.

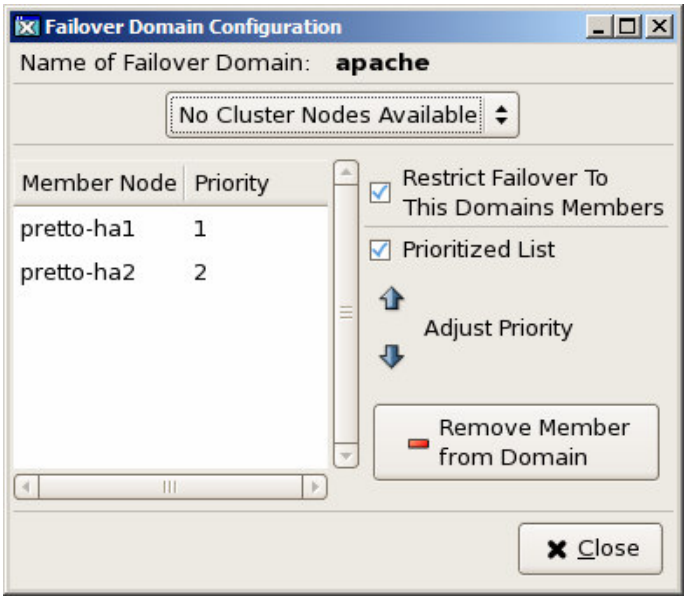


Figura 33: Ajustando prioridades *Failover Domain*
Fonte: *ClusterHA* (2007)

(xi) Selecionar um serviço para o *cluster*. No exemplo conforme Figura 34 foi adicionado o serviço Apache, com o nome **apache-ha**. Este serviço está localizado em #/etc/rc.d/init.d/apache-ha, e o mesmo é cópia do serviço *httpd*. Acrescentar no arquivo de configuração (apache-ha) a linha Listen 172.16.100.100:80. Podem ser acrescentados tantos serviços quanto forem necessários ao *cluster*. No *cluster* configurado foi implementado o recurso do apache como uma forma de demonstrar o processo. Para outros tipos de serviços que se queira adicionar, o procedimento será semelhante, bastando validar seus procedimentos de inicialização.

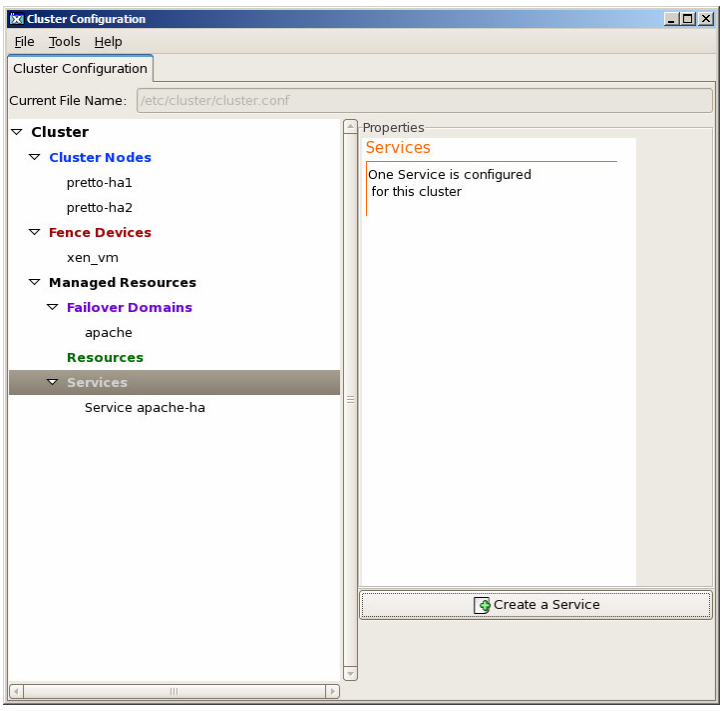


Figura 34: *Services*
Fonte: *ClusterHA* (2007)

(xii) Atenção especial neste ítem de configuração. Configurar o campo ***Autostart This Service*** conforme Figura 35 para que automaticamente o serviço configurado **apache-ha** seja reiniciado. Em conjunto com o item ***Relocate*** em ***Recovery Policy***, esta customização garantirá de que o recurso seja migrado para o outro *nodo* que ficou ativo no *cluster*. Caso este item fique na situação de *Restart* (opção disponível na tela da Figura 35), este serviço nunca realizará a migração para o outro nó do *cluster* e o sistema permanecerá inativo.

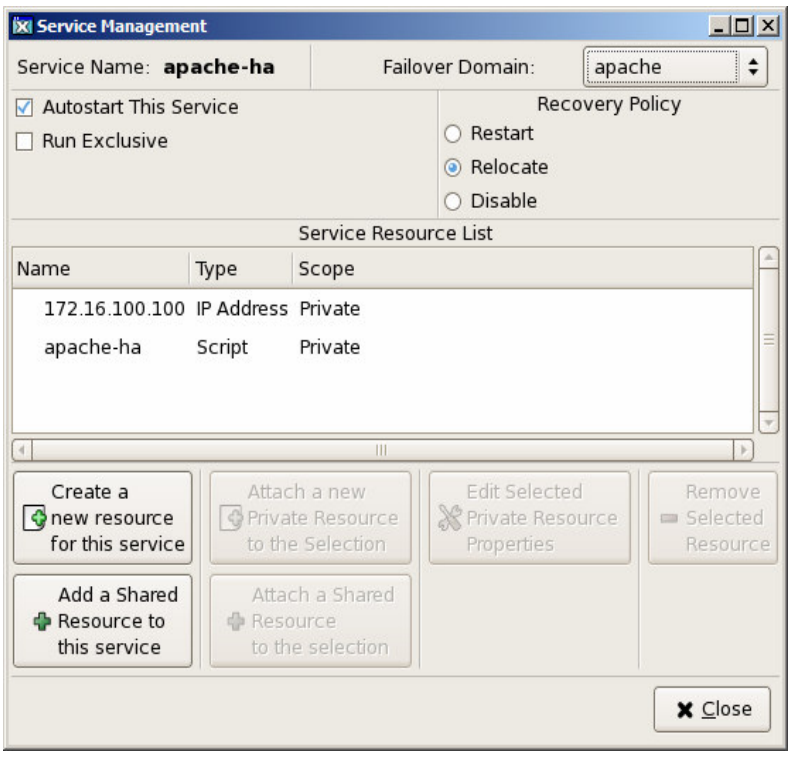


Figura 35: *Service Management*
Fonte: *Cluster*HA (2007)

(xiii) Alterado o *Failover Domain* para **apache** conforme Figura 36, garantindo de que o serviço **apache-ha** seja disponibilizado aos *nodos* configurados neste *Domain*. No *cluster* configurado, estes recursos são **pretto-ha1** e **pretto-ha2** respectivamente.

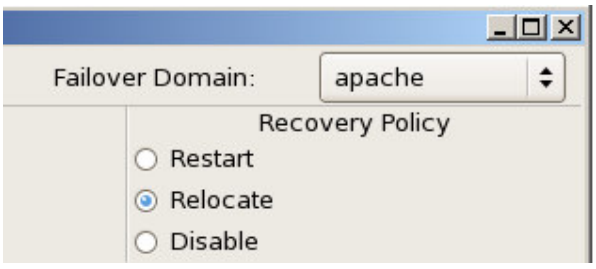


Figura 36: *Recovery Policy*
Fonte: *ClusterHA* (2007)

(xiv) Salvar a configuração que está sendo realizada conforme Figura 37.

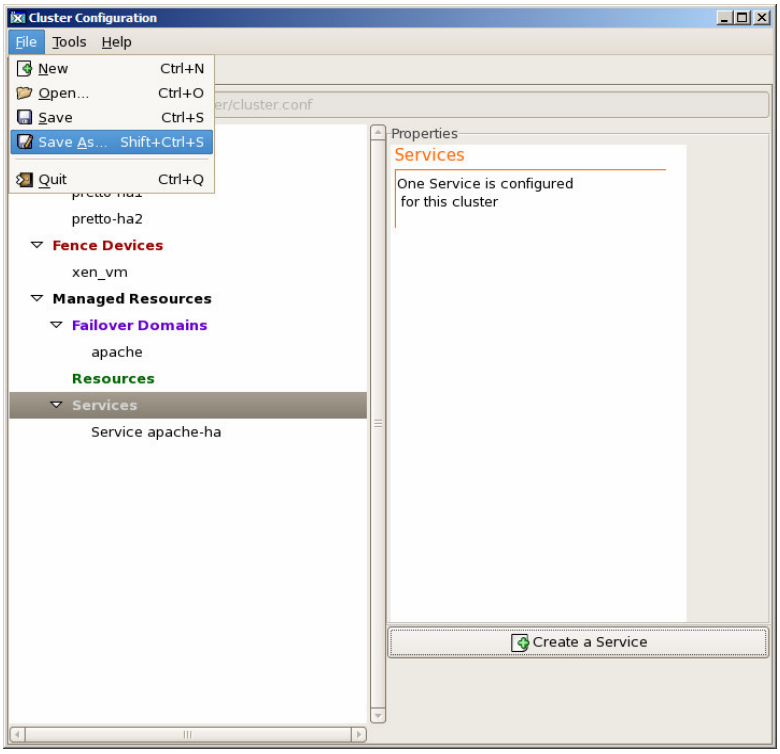


Figura 37: Salvando configuração
Fonte: *ClusterHA* (2007)

(xv) A configuração ficará arquivado no diretório /etc/cluster com o nome de cluster.conf conforme Figura 38.

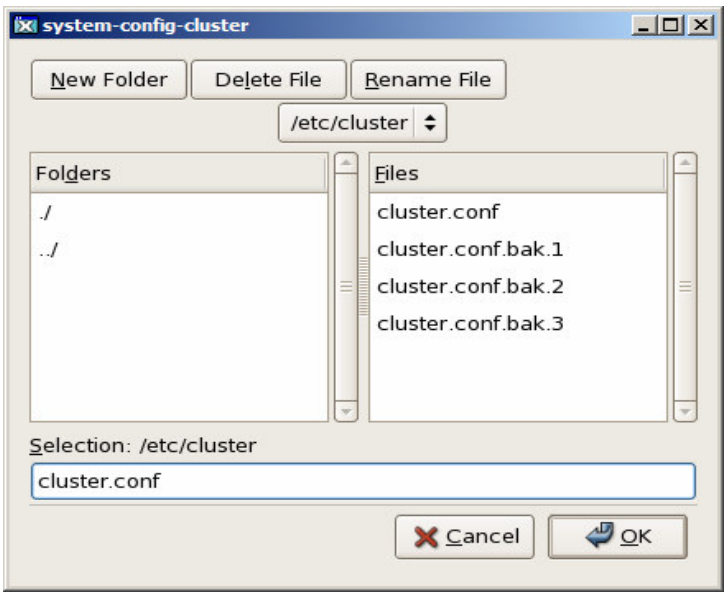


Figura 38: Salvando configuração em diretório
Fonte: *ClusterHA* (2007)

(xvi) Enviar este arquivo de configuração ao outro nó, através do comando:

```
[root@pretto-ha1 sbin]#
scp /etc/cluster/cluster.conf 172.16.100.20:/etc/cluster/cluster.conf
```

(xvii) Inicializar o serviço cman, e após, verificar no arquivo /var/log/messages com o comando # tail –f /var/log/messages as mensagens de subida do *cluster*. Atentar que na parte de *fencing*, irá solicitar o *reinicialização* do outro servidor.

```
[root@pretto-ha1 sbin]# service cman start
```

(xviii) Inicializar o serviço rgmanger, e após, verificar no arquivo /var/log/messages com o comando # tail –f /var/log/messages as mensagens de subida do *cluster*.

```
[root@pretto-ha1 sbin]# service rgmanager start
```

(xix) Inicializar o serviço clvmd, e após, verificar no arquivo /var/log/messages com o comando # tail –f /var/log/messages as mensagens de subida do *cluster*.

```
[root@pretto-ha1 sbin]# service clvmd start
```

(xx) Para que o serviço cman, rgmanager e clvmd subam automaticamente quando do post do equipamento, realizar o seguinte comando:

```
[root@pretto-ha1 sbin]# chkconfig cman on
[root@pretto-ha1 sbin]# chkconfig rgmanager on
[root@pretto-ha1 sbin]# chkconfig clvmd on
```

(xxi) Caso queira desabilitar a opção de subida automática do serviço cman, rgmanager ou clvmd utilize os seguintes comandos:

```
[root@pretto-ha1 sbin]# chkconfig cman off
[root@pretto-ha1 sbin]# chkconfig rgmanger off
[root@pretto-ha1 sbin]# chkconfig clvmd off
```

(xxii) Verificar o serviço de do *cluster* com o comando:

```
[root@pretto-ha1 sbin]# clustat
Member Status: Quorate

Member Name                          ID   Status
------ ----                          ---- ------
pretto-ha1                            1 Online, Local, rgmanager
pretto-ha2                            2 Online, rgmanager
Service Name         Owner (Last)                  State
------- ----                  ----- ------                          -----
service:apache-ha    pretto-ha1                    started
[root@pretto-ha1 sbin]#
```

(xxiii) Realizar um teste de migração do serviço **apache-ha** do *cluster* para garantir de que o mesmo consiga reformar-se a partir dos comandos de administração do *cluster* **# clusvcadm**.

```
[root@pretto-ha1 sbin]# clustat
Member Status: Quorate

  Member Name                          ID   Status
  ------ ----                          ---- ------
  pretto-ha1                            1 Online, Local, rgmanager
  pretto-ha2                            2 Online, rgmanager

  Service Name         Owner (Last)                  State
  ------- ----         ----- ------                  -----
  service:apache-ha    pretto-ha1                    started

[root@pretto-ha1 sbin]#
[root@pretto-ha1 sbin]# clusvcadm -r apache-ha -m pretto-ha2
Trying to relocate service:apache-ha to pretto-ha2...Success
service:apache-ha is now running on pretto-ha2

[root@pretto-ha1 sbin]# clustat
Member Status: Quorate

  Member Name                          ID   Status
  ------ ----                          ---- ------
  pretto-ha1                            1 Online, Local, rgmanager
  pretto-ha2                            2 Online, rgmanager

  Service Name         Owner (Last)                  State
  ------- ----         ----- ------                  -----
```

```
  service:apache-ha    pretto-ha2                     started
[root@pretto-ha1 sbin]#
```

Neste item, pode-se optar por desativar o serviço retirando-se do ar o serviço hhtpd posicionando no *nodo* onde o mesmo está ativo. Desta forma forçará o script a migrar o serviço **apache-ha**

```
[root@pretto-ha1 sbin]# service httpd stop
httpd:                                                     [stop]
[root@pretto-ha1 sbin]#
```

(xxiv) Alterar a linha 25 do arquivo /etc/cluster/cluster.conf para **<cman expected_votes="1" two_node="1"/>,** a fim de evitar a situação de "split brain" que é a não formação do *cluster* a partir de dois *nodos*. Tal situação ocorre quando ambos os *nodos* querem reformar o *cluster* e não possuem privilégios suficientes para tornar-se o nodo principal, entrando assim em uma situação de instabilidade geral do sistema.

```
[root@pretto-ha1 sbin]# cd /etc/cluster
[root@pretto-ha1 cluster]# more cluster.conf
<?xml version="1.0"?>
<cluster alias="pretto" config_version="14" name="pretto">
    <quorumd Device="/dev/xvda2" interval="2" label="quorumdisk"
             loglevel="7" min_score="1"
             status_file="/var/log/qdisk_status" tko="10" votes="1">
        <heuristic interval="1" program="ping 192.168.1.1 -c1 -t2" score="3"/>
    </quorumd>
    <fence_daemon post_fail_delay="0" post_join_delay="3"/>
        <clusternodes>
                <clusternode name="pretto-ha1" nodeid="1" votes="1">
                        <fence>
                          <method name="1">
                             <device name="xen_vm" port="pretto-ha2"/>
                          </method>
                        </fence>
                </clusternode>
                <clusternode name="pretto-ha2" nodeid="2" votes="1">
                        <fence>
                          <method name="1">
                            <device name="xen_vm" port="pretto-ha1"/>
                          </method>
                        </fence>
                </clusternode>
        </clusternodes>
        <cman expected_votes="1" two_node="1"/>  ← alterar este item
        <fencedevices>
                <fencedevice agent="fence_brocade" ipaddr="192.168.0.100" login=
"pretto" name="xen_vm" passwd="cachaca"/>
        </fencedevices>
        <rm>
                <failoverdomains>
                   <failoverdomain name="apache" ordered="1" restricted="1">
                      <failoverdomainnode name="pretto-ha1" priority="1"/>
                      <failoverdomainnode name="pretto-ha2" priority="2"/>
                <failoverdomain>
                </failoverdomains>
                <resources/>
```

```
                <service autostart="1" domain="apache" name="apache-ha" recovery
="relocate">
                        <ip address="172.16.100.100" monitor_link="1"/>
                        <script file="/etc/init.d/httpd" name="apache-ha"/>
                </service>
        </rm>
</cluster>
[root@pretto-ha1 cluster]#
```

(xxv) Quando realizar alguma atualização do arquivo /etc/cluster/cluster.conf, atentar ao fato da versão “config_version” que se encontra na segunda linha deste arquivo. Este deverá ter uma revisão superior à anterior e basta somente incrementar em um número e após realizar o *update* para os outros nós.

Linha 2 do arquivo /etc/cluster/cluster.conf, conforme exemplo do arquivo de configuração do *cluster* implementado:

```
<cluster alias="pretto" config_version="14" name="pretto">  ← release 14
```

Modificando-a para uma nova config_version:

```
<cluster alias="pretto" config_version="15" name="pretto"> ← release 15
```

Logo após, realizar o *update* desta nova versão aos *nodos* do *cluster* através do comando ccs_tool:

```
[root@pretto-ha1 cluster]# ccs_tool update /etc/cluster/cluster.conf
```

(xxvi) Quando for realizar o desligamento dos sistemas, o serviço apache-ha deverá ser retirado antes do comando #shutdown, através do comando # clusvcadm. Tal situação torna-se necessário, pois uma das características do *software Cluster Suíte* é de que os serviços devem permanecer em funcionamento, mesmo que o administrador digite de forma inadvertida comandos de *reboot* ou *shutdown*. Por característica de segurança, os *nodos* do *cluster* configurados não se desligam automaticamente através de comandos de reinicialização citados acima. O serviço CMAN (*Cluster Management*) age como uma barreira, impedindo a complementação do comando. Abaixo, seqüência correta de desligamento do equipamento, que obrigatoriamente deve-se retirar o serviço configurado (em nosso caso o apache-ha) e somente após realizar os comandos de desligamento do servidor.

```
[root@pretto-ha1 cluster]# clustat
Member Status: Quorate
  Member Name                        ID   Status
  ------ ----                        ---- ------
  pretto-ha1                            1 Online, Local, rgmanager
```

```
  pretto-ha2                              2 Online, rgmanager

  Service Name         Owner (Last)                    State
  ------- ----         ----- ------                    -----
  service:apache-ha    pretto-ha2                      started

[root@pretto-ha1 cluster]# clusvcadm -d apache-ha
Local machine disabling service:apache-ha...Success

[root@pretto-ha1 cluster]# clustat
Member Status: Quorate
  Member Name                           ID   Status
  ------ ----                           ---- ------
  pretto-ha1                              1 Online, Local, rgmanager
  pretto-ha2                              2 Online, rgmanager

  Service Name         Owner (Last)                    State
  ------- ----         ----- ------                    -----
  service:apache-ha    (pretto-ha2)                    disabled
[root@pretto-ha1 cluster]#
[root@pretto-ha1 cluster]# shutdown -h -y 0
```

5.3 Teste e Avaliação do *Cluster* Montado

Para a realização dos testes e avaliação do *cluster* montado de acordo com a Figura 39, foram realizados simulações de queda de serviço e de equipamentos.

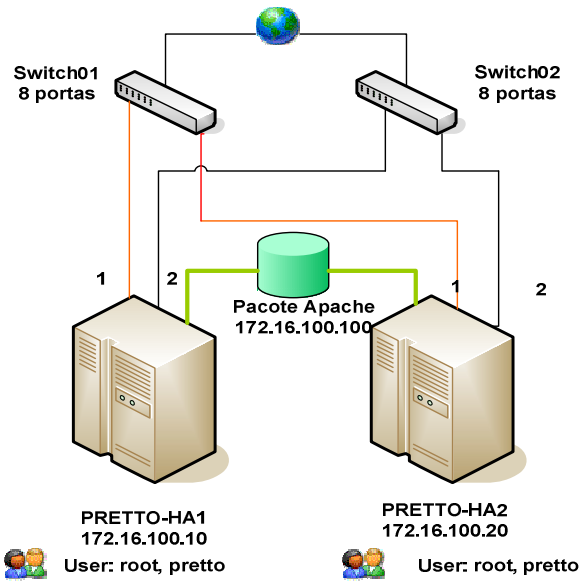


Figura 39: *Cluster* implementado
Fonte: Autor (2007)

5.3.1 Teste de queda de serviços

Para verificação da queda de serviço no *cluster* existem procedimentos que podem ser utilizados para verificar o seu funcionamento.

5.3.1.1 Comandos para administração de serviços no *cluster*

Para realizar a administração do cluster e de seu serviço configurado, existem comandos do CMAN específicos. A seguir, são relacionados alguns destes comandos e sua função.

a) # clusvcadm –d apache-ha (desabilita o serviço no *cluster*);

b) # clusvcadm –e apache-ha –m pretto-ha1 (ativa o serviço apache-ha no *nodo* de nome pretto-ha1);

c) # clusvcadm –e apache-ha –m pretto-ha2 (ativa o serviço apache-ha no *nodo* de nome pretto-ha2);

d) # clusvcadm –r apache-ha –m pretto-ha1 (realoca o serviço apache-ha para o *nodo* de nome pretto-ha1);

e) # clusvcadm –r apache-ha –m pretto-ha2 (realoca o serviço apache-ha para o *nodo* de nome pretto-ha2).

f) # service httpd stop (desativa o serviço *httpd* do servidor onde o serviço está ativo, forçando o serviço a migrar para o outro *nodo*).

Maiores detalhes dos comandos poderão ser obtidos a partir do comando #man clusvcadm dentro do sistema.

5.3.1.2 Procedimentos de testes

Para a verificação do correto funcionamento dos serviços disponibilizados no *cluster* foram realizados testes de desativação do serviço, desativação e ativação automática em um *nodo* específico do *cluster* A seguir os procedimentos realizados dentro do sistema com os comandos clusvcadm.

(i) Verificar os comandos disponíveis no administrador do *cluster*:

```
[root@pretto-ha1 ~]# clusvcadm
Resource Group Control Commands:
       clusvcadm -v                    Display version and exit
       clusvcadm -d <group>            Disable <group>
       clusvcadm -e <group>            Enable <group>
       clusvcadm -e <group> -m <member> Enable <group> on <member>
       clusvcadm -r <group> -m <member> Relocate <group> [to <member>]
       clusvcadm -M <group> -m <member> Migrate <group> to <member>
                                (e.g. for live migration of VMs)
       clusvcadm -q                    Quiet operation
       clusvcadm -R <group>            Restart a group in place.
       clusvcadm -s <group>            Stop <group>

Resource Group Locking (for cluster Shutdown / Debugging):
       clusvcadm -l                    Lock local resource group manager.
                               This prevents resource groups from starting on
the local node.
       clusvcadm -S                    Show lock state
       clusvcadm -u                    Unlock local resource group manager.
                               This allows resource groups to start on the
local node.
```

(ii) Desativando o serviço apache-ha no *cluster*:

```
[root@pretto-ha1 ~]# clustat
Member Status: Quorate
  Member Name                        ID   Status
  ------ ----                        ---- ------
  pretto-ha1                            1 Online, Local, rgmanager
  pretto-ha2                            2 Online, rgmanager
  Service Name         Owner (Last)                  State
  ------- ----         ----- ------                  -----
  service:apache-ha    pretto-ha1                    started
[root@pretto-ha1 ~]# clusvcadm -d apache-ha
Local machine disabling service:apache-ha...Success
[root@pretto-ha1 ~]# clustat
Member Status: Quorate
  Member Name                        ID   Status
  ------ ----                        ---- ------
  pretto-ha1                            1 Online, Local, rgmanager
  pretto-ha2                            2 Online, rgmanager
  Service Name         Owner (Last)                  State
```

```
  ------- ----          ----- ------                    -----
  service:apache-ha     (pretto-ha1)                   disabled
[root@pretto-ha1 ~]#
```

(iii) Ativando o serviço apache-ha no *nodo* pretto-ha1:

```
[root@pretto-ha1 ~]#  clusvcadm -e apache-ha -m pretto-ha1
Member pretto-ha1 trying to enable service:apache-ha...Success
service:apache-ha is now running on pretto-ha1
[root@pretto-ha1 ~]# clustat
Member Status: Quorate
  Member Name                          ID   Status
  ------ ----                          ---- ------
  pretto-ha1                            1 Online, Local, rgmanager
  pretto-ha2                            2 Online, rgmanager
  Service Name          Owner (Last)                   State
  ------- ----          ----- ------                   -----
  service:apache-ha     pretto-ha1                     started
[root@pretto-ha1 ~]#
```

(iv) Ativando o serviço apache-ha no *nodo* pretto-ha2:

```
[root@pretto-ha1 ~]# clusvcadm -e apache-ha -m pretto-ha2
Member pretto-ha2 trying to enable service:apache-ha...Success
service:apache-ha is now running on pretto-ha2
[root@pretto-ha1 ~]# clustat
Member Status: Quorate
  Member Name                          ID   Status
  ------ ----                          ---- ------
  pretto-ha1                            1 Online, Local, rgmanager
  pretto-ha2                            2 Online, rgmanager
  Service Name          Owner (Last)                   State
  ------- ----          ----- ------                   -----
  service:apache-ha     pretto-ha2                     started
[root@pretto-ha1 ~]#
```

(v) Migrando o serviço apache-ha do *nodo* pretto-ha2 para o *nodo* pretto-ha1:

```
[root@pretto-ha1 ~]# clustat
Member Status: Quorate
  Member Name                          ID   Status
  ------ ----                          ---- ------
  pretto-ha1                            1 Online, Local, rgmanager
  pretto-ha2                            2 Online, rgmanager

  Service Name          Owner (Last)                   State
  ------- ----          ----- ------                   -----
  service:apache-ha     pretto-ha2                     started

[root@pretto-ha1 ~]# clusvcadm -r apache-ha -m pretto-ha1
Trying to relocate service:apache-ha to pretto-ha1...Success
service:apache-ha is now running on pretto-ha1
[root@pretto-ha1 ~]# clustat
Member Status: Quorate
  Member Name                          ID   Status
  ------ ----                          ---- ------
  pretto-ha1                            1 Online, Local, rgmanager
  pretto-ha2                            2 Online, rgmanager
  Service Name          Owner (Last)                   State
  ------- ----          ----- ------                   -----
  service:apache-ha     pretto-ha1                     started
```

```
[root@pretto-ha1 ~]#
```

(vi) Migrando o serviço apache-ha do *nodo* pretto-ha1 para o *nodo* pretto-ha2:

```
[root@pretto-ha1 ~]# clustat
Member Status: Quorate
  Member Name                        ID   Status
  ------ ----                        ---- ------
  pretto-ha1                            1 Online, Local, rgmanager
  pretto-ha2                            2 Online, rgmanager
  Service Name         Owner (Last)                  State
  ------- ----         ----- ------                  -----
  service:apache-ha    pretto-ha1                    started

[root@pretto-ha1 ~]# clusvcadm -r apache-ha -m pretto-ha2
Trying to relocate service:apache-ha to pretto-ha2...Success
service:apache-ha is now running on pretto-ha2
[root@pretto-ha1 ~]# clustat
Member Status: Quorate
  Member Name                        ID   Status
  ------ ----                        ---- ------
  pretto-ha1                            1 Online, Local, rgmanager
  pretto-ha2                            2 Online, rgmanager
  Service Name         Owner (Last)                  State
  ------- ----         ----- ------                  -----
  service:apache-ha    pretto-ha2                    started
[root@pretto-ha1 ~]#
```

(vii) Parando o serviço apache-ha via comando service httpd stop do *nodo* pretto-ha1 e consequente migração do serviço para o *nodo* pretto-ha2:

```
[root@pretto-ha1 ~]# clustat
Member Status: Quorate
  Member Name                        ID   Status
  ------ ----                        ---- ------
  pretto-ha1                            1 Online, Local, rgmanager
  pretto-ha2                            2 Online, rgmanager
  Service Name         Owner (Last)                  State
  ------- ----         ----- ------                  -----
  service:apache-ha    pretto-ha1                    started

[root@pretto-ha1 ~]# service httpd stop
Stopping httpd:                                            [  OK  ]
[root@pretto-ha1 ~]#

[root@pretto-ha1 ~]# clustat
Member Status: Quorate
  Member Name                        ID   Status
  ------ ----                        ---- ------
  pretto-ha1                            1 Online, Local, rgmanager
  pretto-ha2                            2 Online, rgmanager
  Service Name         Owner (Last)                  State
  ------- ----         ----- ------                  -----
  service:apache-ha    pretto-ha2                    started
```

5.3.2 Queda de equipamento

Este teste de validação visa verificar o comportamento do *cluster* em situações onde possa existir alguma anormalidade em um dos equipamentos que fazem parte da configuração do *cluster* e sua conseqüente reformulação dos *nodos*.

A seguir, demonstração em que um equipamento torna-se inoperante por uma falha provocada de *hardware* e o *cluster* automaticamente transfere o serviço para o *nodo* que ficou ativo no *cluster*, permitindo com que os usuários continuem a ter acesso ao sistema.

---

i) Inicialmente, verificar a situação do cluster com o comando clustat:

```
[root@pretto-ha1 ~]# clustat
Member Status: Quorate
  Member Name                          ID   Status
  ------ ----                          ---- ------
  pretto-ha1                            1 Online, Local, rgmanager
  pretto-ha2                            2 Online, rgmanager
  Service Name         Owner (Last)                  State
  ------- ----         ----- ------                  -----
  service:apache-ha    pretto-ha2                    started
```

ii) Verificando a situação das máquinas virtuais na console do xen através do comando xe vm-list:

```
[root@xen-pretto ~]# xe vm-list
uuid ( RO )           : f67bd58e-9003-c94b-9f81-ef5c39488659
     name-label ( RW ): pretto-ha2
   power-state  ( RO ): running

uuid ( RO )           : 63d52988-abeb-4782-9ª4f-6aa7f9608e3a
     name-label ( RW ): Control domain on host: xen-pretto
   power-state  ( RO ): running

uuid ( RO )           : e392660b-4001-f1b6-aa57-a246e0840a56
     name-label ( RW ): pretto-ha1
   power-state  ( RO ): running
```

iii) Realizando o procedimento de *shutdown* com a opção de *force=true*, simulando desta forma a queda de um equipamento por falha geral em seu *hardware*.

```
[root@xen-pretto ~]# xe vm-shutdown force=true vm=pretto-ha2
[root@xen-pretto ~]#
```

iv) Verificando a situação das máquinas virtuais na console do xen através do comando **xe vm-list**. Nesta situação poderá ser verificado que o servidor **pretto-ha2** está em status de halted.

```
[root@xen-pretto ~]# xe vm-list
uuid ( RO )           : f67bd58e-9003-c94b-9f81-ef5c39488659
     name-label ( RW ): pretto-ha2
   power-state  ( RO ): halted

uuid ( RO )           : 63d52988-abeb-4782-9a4f-6aa7f9608e3a
     name-label ( RW ): Control domain on host: xen-pretto
   power-state  ( RO ): running

uuid ( RO )           : e392660b-4001-f1b6-aa57-a246e0840a56
     name-label ( RW ): pretto-ha1
   power-state  ( RO ): running
```

v) Verificando a situação do cluster através do comando **clustat**. Nesta situação, o servidor que foi desligado aparecerá com *status* de *Offline* e o serviço **apache-ha** migrará para o *nodo* que ficou ativo no *cluster*, que no caso é o servidor **pretto-ha1**.

```
[root@pretto-ha1 ~]# clustat
Member Status: Quorate
  Member Name                        ID   Status
  ------ ----                        ---- ------
  pretto-ha1                            1 Online, Local, rgmanager
  pretto-ha2                            2 Offline
  Service Name         Owner (Last)                  State
  ------- ----         ----- ------                  -----
  service:apache-ha    pretto-ha1                    started
[root@pretto-ha1 ~]#
```

Os testes realizados permitem verificar a funcionalidade do *cluster* e o serviço de reformulação do *cluster*, analisando primeiramente o ativo que entrou em situação de falha e logo após reinicializando o serviço **apache-ha** no *nodo* que ficou ativo.

5.3.3 Avaliação do *cluster*

Avaliando o funcionamento do cluster implementado utilizando o *Cluster Suíte* do *CentOS 5*, pode-se verificar a sua aplicabilidade.

Possui grande quantidade de recursos administrativos implementados, que facilitam a sua gerência do *cluster* e de seus *nodos* como os comandos *clusvcadm*, *ccs_tool* e outros. Existe a funcionalidade da utilização do conceito de *fencing* que a princípio pode parecer algo limitado a alguns fornecedores de *hardware*, mas que implementam um conceito diferente de outros *clusters* analisados, facilitando a administração sob o ponto de vista do *software* de cluster quando da queda de algum ativo no ambiente e conseqüente ação de restabelecimento do serviço no *nodo* que ficou ativos (ou nos *nodos*, caso tenham mais de dois).

Em função do seu custo de implementação ser nulo, ou seja, o *software* bem como o sistema operacional *CentOS 5* ser baixado de forma gratuita, torna-se um excepcional recurso aliado a alta disponibilidade, podendo ser implementado em equipamentos híbridos (não necessariamente com o mesmo *hardware*), em pequenas ou médias empresas.

Como no estudo de caso foi implementado o serviço do apache apenas como um exemplo, qualquer outro serviço poderá ser implementado pelo administrador seguindo os procedimentos de configuração e utilizar as vantagens de ter um ambiente com redundância de recursos e disponibilidade, visto que a quantidade de recursos e variações de configurações disponíveis no *Cluster Suíte* é extremamente ampla.

6 CONCLUSÕES

Este Trabalho de Conclusão de curso apresentou um estudo para utilização em ambientes de alta disponibilidade que poderá ser aplicada em partes ou integralmente, tanto nas empresas quanto em instituições públicas, em sua área de tecnologia da informação, como uma forma de mitigar riscos e prover maior valor agregado.

Ao se disponibilizar um processo de consulta pela web em HA CLUSTER (2007) sobre este estudo tivemos a intenção de que o mesma sirva como uma fonte de divulgação do conhecimento no meio acadêmico, mas também permita acesso às empresas e pessoas para que estas possam obter informações de forma robusta em áreas distintas de *hardware, software*, configuração de *cluster*, configuração de formas de proteção e outros cuidados que dizem respeito a um sistema de alta disponibilidade.

A cultura com que vemos nossos sistemas de informática vem mudando drasticamente nos últimos anos, e não necessariamente altos investimentos são pré-requisitos para aumentar a segurança do sistema. Muitas vezes, cuidados básicos quanto à instalação, configuração e implementação de equipamentos podem vir a se tornar um excelente aliado na disponibilidade do sistema, e quando não seguidos, poderão vir a se tornar uma fonte de riscos. Ter a exata visão destes itens poderá auxiliar a evitar parada não programada, insatisfação, perda de trabalho e conseqüente prejuízo.

Novos conhecimentos como os aqui apresentados, geram novas práticas na reutilização de recursos no decorrer do tempo. Disponibilizando-os de forma ordenada à sociedade, como na página web, poderá vir a se tornar uma fonte de realimentação de conhecimento, com mais pessoas interessadas em aplicar o estudo, agregando maior conhecimento, disponibilizando-os e tornando-se um diferencial competitivo no ciclo das empresas.

Vimos que as técnicas que garantem a alta disponibilidade não são dispendiosas economicamente ou extremamente complexas. Na verdade, são passíveis de implementação desde que profissionais da área de tecnologia trabalhem e sigam os processos de acordo com suas necessidades.

6.1 Considerações Sobre Estudo

Com a utilização das técnicas para garantir alta disponibilidade e a implementação do *cluster*, pode-se verificar a grande utilidade na otimização destes recursos para as organizações. É de fácil implementação e assimilação para profissionais do meio de tecnologia da informação e seu custo é baixo, visto que utiliza recursos disponíveis do *GNU/Linux*.

6.2 Trabalhos Futuros

Uma possível extensão do estudo demonstrado seria tratar do uso de *storages* no *cluster*, onde o grau de disponibilidade do sistema é elevado, garantindo redundância de recursos a partir dos fatores de segurança que fazem a composição dos sistemas *storages* e dos serviços configurados no *cluster*.

Uma outra possibilidade de aplicação seria a utilização de maiores recursos de computadores, adicionando-se outros nós ao *cluster*, sob a forma de camadas de tipos de serviços. Estes nodos estariam configurados em camadas de equipamentos dentro do *cluster*. Desta forma, os serviços em *cluster* estariam protegidos por outros serviços também em *cluster*, todos disponibilizados na forma de cascata, aumentando desta forma o nível de segurança do sistema. Esta é uma variação do *cluster* apresentado, capacitando de forma exponencial o nível de segurança.

REFERÊNCIAS

ABNT, **Associação brasileira de normas técnicas.** Disponível em: <http://www.abntnet.com.br/fidetail. aspx?FonteID=23745> Acesso em: 05 Abr., 2007.

ABNT, **Nbr5410.** Disponível em: <http://www.abntnet.com.br/fidetail. aspx?FonteID=23745> Acesso em: 05 Abr., 2007.

ABT, **Reestruturação da área de tecnologia da informação.** Disponível em: < http://www.abt-r.org.br/index. php?option=com_content&task=view&id=155&Itemid=2 > Acesso em: 15 Set., 2007.

ACNC. ***Raid.***, disponível em: < http://www.acnc.com/raid.html >. Acesso em: 04 de Nov., 2006.

BROCADE, **San clustering.** Disponível em: <http://www.brocade.com/solutions/h. jsp>. Acesso em: 01 Abr., 2007.

CENTOS, **Configuring and managing a red hat cluster.** Disponível em: < http://www.centos.org/docs/5/html/Cluster_Administration/>. Acesso em: 08 Jun., 2007.

CENTOS, **Centos download information.** Disponível em: <http://www.centos.org/modules/tinycontent/index.php?id=15>. Acesso em: 04 Jun., 2007.

CENTOS, **The community enterprise operationg system.** Disponível em: <http://www.centos.org/>. Acesso em: 06 Abr., 2007.

CP, **Entendendo o fator de potência.** Disponível em: <http://www.cp.com.br/artigos/fator_de_potencia.pdf> Acesso em: 10 Abr., 2007.

DELL. **Dell high availability clustering.** Disponível em: <http://www.dell.com/content /topics/global.asp x/solutions/en/clustering_ha? c=us&cs=555&l=en&s=biz>.Acesso em: 02 Nov., 2006.

EDON, Fabio, **Comunicação pessoal**. 2007

GARTNER, Felix C. **Fundamentals of fault-tolerant distributed computing in asynchronous environments.** New York: ACM Press, 1999.

GNU. **The gnu operating system**. Disponível em: <http://www.gnu.org/>. Acesso em: 02 de Nov., 2006.

HA CLUSTER. **Portal de alta disponibilidade**. Disponível em: < http://www.nidus.org.br/hacluster/>. Acesso em: 02 de Nov., 2007.

HALL, Jonathan I. **Hamming code.** Disponível em < http://www.math.msu.edu/~jhall/classes/codenotes/Hamming.pdf>. Acesso em: 22 de Set., 2006.

HEWLETT PACKARD SA**. High availability solutions for linux.** Disponível em**:** <http://h71028.www7.hp.com/enterpri. aspx>. Acesso em: 02 Set., 2007.

HEWLETT PACKARD SA**. Mc/service guard.** Disponível em**:** <http://h18022.www1.hp.com/solutions/enterprise/highavailability/linux/serviceguard/index.html>. Acesso em: 02 Nov., 2006.

HEWLETT PACKARD SA**. Openvms high availability and disaster tolerance.** Disponível em**:** <http://h71000.www7.hp.com/availability/presentations.html>. Acesso em: 08 Mai., 2007.

HEWLETT PACKARD SA. **Standard 005-3 customer environmental criteria – power**. Disponível em: < http://standards.inet.cpqcorp.net/smc/hpstd/html/F-HP0000503.htm>. Acesso em: 15 Nov. 2006.

IBGE. **Estatísticas do cadastro central de empresas 2004.** Disponível em: <http://www.ibge.gov.br/home/estatistica/economia/cadastroempresa/2004/coment2004.pdf>. Acesso em: 13 Set., 2007.

IBM. **V4r4 high availability cluster.** Disponível em: < http://www.redbooks.ibm.com/redpapers/pdfs/redp0501.pdf>. Acesso em: 02 Nov., 2006.

LINUX. **Linux high availability**. Disponível em: < http://www.linux-ha.org/>. Acesso em: 07 de Set., 2007.

RED HAT. **Red hat advanced server 4**. Disponível em: < http://www.redhat.com/>. Acesso em: 12 de Set., 2006.

RED HAT. **Red hat cluster manager**. Disponível em: <http://www.redhat.com/docs/manuals/enterprise/RHEL-AS-2.1-Manual/*cluster*-manager/>. Acesso em: 08 de Ago., 2006.

SABER ELETRONICA. **Esd descargas eletrostáticas**. Disponível em: <http://www.sabereletronica.com/artigos/eds/esd01.asp>. Acesso em: 20 de Ago., 2007.

SAN. **Storage area network**. Disponível em: <http://pt.wikipedia.org/wiki/SAN/>, Acesso em: 10 de Set., 2006.

SEAGATE, **Barracuda es**. Disponível em: <http://www.seagate.com/cda/products /discsales/ enterprise/tech/1, 1084, 780,00. html>. Acesso em: 12 de Out., 2006.

SPAFFORD, George. **Mean time between failures**. Disponível em: < http://itmanagement.earthweb.com/c. php/3354191>. Acesso em: 12 de Set., 2007.

TWEEDIE, Stephen. **Designing a linux cluster**. Disponível em: <http://lcic.org/documentation.html>. Acesso em: 12 de Out., 2006.

W. VOGELS, W.; DUMITRIU, D.; BIRMAN, K.; GAMACHE, R.; MASSA, M.;SHORT, R.;VERT, J.; BARRERA, J.; GRAY, J. **The design and architecture of the microsoft cluster service - a practical approach to high-availability and scalability, proceedings of ftcs**. IEEE, Junho, 1998.

WEBER, Taisy S. **Um roteiro para exploração dos conceitos básicos de tolerância a falhas.** Disponível em: <http://wwwusr.inf.ufsm.br/~ceretta/papers/TaisyDependabilidade>. Acesso em: 12 de Out., 2006.

WEYGANT, Peter S. **Clusters for high availability**. 2. ed. New Jersey: Prentice Hall, 2002.

XEN. **Xen 3.1 downloads**. Disponível em: < http://xen.org/download/>. Acesso em: 12 de Out., 2007.

ANEXOS

ANEXO A – Inventário de Componentes: Servidores

	Inventário de Componentes: SERVIDORES					
Identificação Servidor						
Novo (N) ou existente (E)						
Fabricante						
Modelo						
Sistema Operacional						
Total Slots						
Tipo A						
Tipo (PCI, SBUS, etc)						
Numero						
Tamanho						
Velocidade (Mhz)						
Tipo B						
Tipo (PCI, SBUS, etc)						
Numero						
Tamanho						
Velocidade (Mhz)						
Simples (S) ou Dupla (D) Conexão						
Portas						
# Portas requeridas para conexão simples						
# Portas requeridas para conexão dupla						
HBA						
Informações Gerais						
- fabricante						
- modelo						
- versão						
- numero de portas						
- numero de HBAs						
Tipo de Driver (se aplicável)						
-fabric						
-private loop						
-public loop						
Conexões suportadas por HBA						
Servidor						
- Dimensões						
- Requisitos de Potência						
- Localização da console						
- Endereço lógico da console (se aplicável)						
- Ethernet interface						
- Telefone instalado dentro do datacenter (Y/N)						
- Localização física						
Lista de Aplicações (com versões)						
Aplicação número 1						
Aplicação número 2						
Aplicação número 3						
Aplicação número 4						
Requisitos Storage						
Requisito 1						
Nome da Aplicação						
Requisitos Iniciais						
Requisitos projetados						
Requisito 2						
Nome da Aplicação						
Requisitos Iniciais						
Requisitos projetados						
Requisito 3						
Nome da Aplicação						
Requisitos Iniciais						
Requisitos projetados						
Requisito 4						
Nome da Aplicação						
Requisitos Iniciais						
Requisitos projetados						

Fonte: Adaptado de Brocade (2007)

ANEXO B – Inventário de Componentes: Storage de Discos

	Inventário de Componentes: STORAGE DE DISCOS							
Identificação do Storage								
Novo (N) ou Existente (E)								
Fabricante								
Modelo								
Versão firmware								
Simples (S) ou Dupla (D) Conexão								
Ports								
# Portas requeridas para conexão simples								
# Portas requeridas para conexão dupla								
Interfaces								
SCSI (SC) ou Fibre Channel (FC)								
Tipo de SCSI								
- wide/narrow								
- differential/single ended								
Quantidade de Canais de Fibra (FC) existentes								
- modelo								
Quantidade de Canais de Fibra (FC) para adquirir								
- modelo								
Ethernet interfaces								

Fonte: Adaptado de Brocade (2007)

ANEXO C – Inventário de Componentes: *Switches* e *Routers*

	Inventário de Componentes: SWITCHES e ROUTERS					
Identificação do Switch						
Informação de Zoning						
Versão de Firmware						
IP Address(es)						
Gateway						
Configuração de Portas						
0						
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						
Descrição do Device conectado (tipo, WWN, etc.)						
0						
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						
Licença						
Informação de Zoning						
Passwords						
Temperatura de operação						
Requisitos de potência						
Localização Física						
Nível de firmware						

Fonte: Adaptado de Brocade (2007)

ANEXO D – Inspeção de Site

INSPEÇÃO DE SITE	
Empresa	
Endereço	
Contato principal	Telefone
	Fax
	E-mail
Contato alternativo	Telefone
	Fax
	E-mail

Descrição

Este documento visa auxiliar o gerente de TI a ter uma visão geral dos principais aspectos físicos que contribuem para uma instalação de equipamentos em um ambiente de produção.

Servindo como um Guia Prático a partir de experiências em campo, abordaremos os seguintes tópicos:

- Tamanho do ambiente;
- Tipo de piso e acesso;
- Controle de incêndio;
- Parte elétrica: estática, aterramento e alimentação;
- Backup;
- Ar condicionado;
- Práticas de manutenção periódicas.

Como resultado dos itens verificados, este documento visa ser um Guia sobre boas práticas em TI.

Instruções

1. Favor preencher nas caixas com “sim”, “não”, “na” ou “ver”

- “na” significa não aplicável ao ambiente.
- “ver” significa que o item deverá ser verificado.

Ambiente do Computador		sim	não	na	ver
1.	Existe uma cópia do projeto do ambiente onde os equipamentos estarão sendo instalados?	☐	☐	☐	☐
2.	Existe acesso controlado ao ambiente?	☐	☐	☐	☐
3.	Existe espaço adequado de fluxo de ar (100/120 cm) e espaço seguro de manutenção (90/100 cm)?	☐	☐	☐	☐
4.	Existe aterramento adequado no piso suspenso?	☐	☐	☐	☐
5.	Existe distribuição de peso adequada no piso?	☐	☐	☐	☐
6.	Existem canaletas ou dutos para roteamento dos cabos?	☐	☐	☐	☐
7.	Existe telefone dentro do ambiente?	☐	☐	☐	☐
8.	Os cabos utilizados estão padronizados e de acordo com a carga sistema?	☐	☐	☐	☐

Informações Adicionais	

Comentários

Alimentação e Iluminação		sim	não	na	ver
9.	Existe iluminação adequada dentro do ambiente para suporte?	☐	☐	☐	☐
10.	Existe dualidade de entrada de energia elétrica?	☐	☐	☐	☐
11.	Os conectores de energia elétrica são padronizados em relação à potência?	☐	☐	☐	☐
12.	A energia elétrica é apropriada aos equipamentos instalados?	☐	☐	☐	☐
13.	Existem disjuntores individuais para cada linha de alimentação? Capacidade?	☐	☐	☐	☐
14.	A freqüência de entrada de energia elétrica está de acordo com as especificações do equipamento?	☐	☐	☐	☐
15.	Existe sistema ininterrupto de alimentação (NoBreak) no ambiente? Dual?	☐	☐	☐	☐
16.	Existe quadro de distribuição elétrica apropriada para todos os equipamentos?	☐	☐	☐	☐

Informações Adicionais	

Comentários

Segurança		sim	não	na	ver
17.	Existe switch de emergência para a parte elétrica (Power shut-off)?	☐	☐	☐	☐
18.	Existe switch de emergência para ar condicionado no ambiente?	☐	☐	☐	☐
19.	Existe problema de ruído no ambiente?	☐	☐	☐	☐
20.	Existe proteção de incêndio no ambiente?	☐	☐	☐	☐
21.	Existem câmaras de vigilância no ambiente?	☐	☐	☐	☐
22.	Existe algum tipo de equipamento com risco de descarga elétrica no ambiente?	☐	☐	☐	☐

Informações Adicionais	

Comentários

Ar condicionado		sim	não	na	ver
23.	A temperatura do ambiente consegue ser mantida entre 18 º e 22º C?	☐	☐	☐	☐
24.	Existe controle de temperatura que impeça alteração de 5º C por hora?	☐	☐	☐	☐
25.	Existe controle de umidade no ambiente (40% até 50%)?	☐	☐	☐	☐
26.	O ar condicionado é adequado e previne zonas de calor?	☐	☐	☐	☐
27.	Existem filtros instalados na entrada de energia?	☐	☐	☐	☐

Informações Adicionais	

Comentários

Radio Freqüência		sim	não	na	ver
28.	Existe algum transmissor de radio freqüência no ambiente? (não recomendável)	☐	☐	☐	☐
29.	Existem seguranças que utilizam rádios comunicadores no ambiente?	☐	☐	☐	☐
30.	Existem torres de radio freqüência próxima ao ambiente?	☐	☐	☐	☐
Storage		**sim**	**não**	**na**	**ver**
31.	Existem gabinetes de discos ou fita disponível no ambiente?				
32.	Existe documentação da parte SAN do storage?	☐	☐	☐	☐

Informações Adicionais	

Comentários

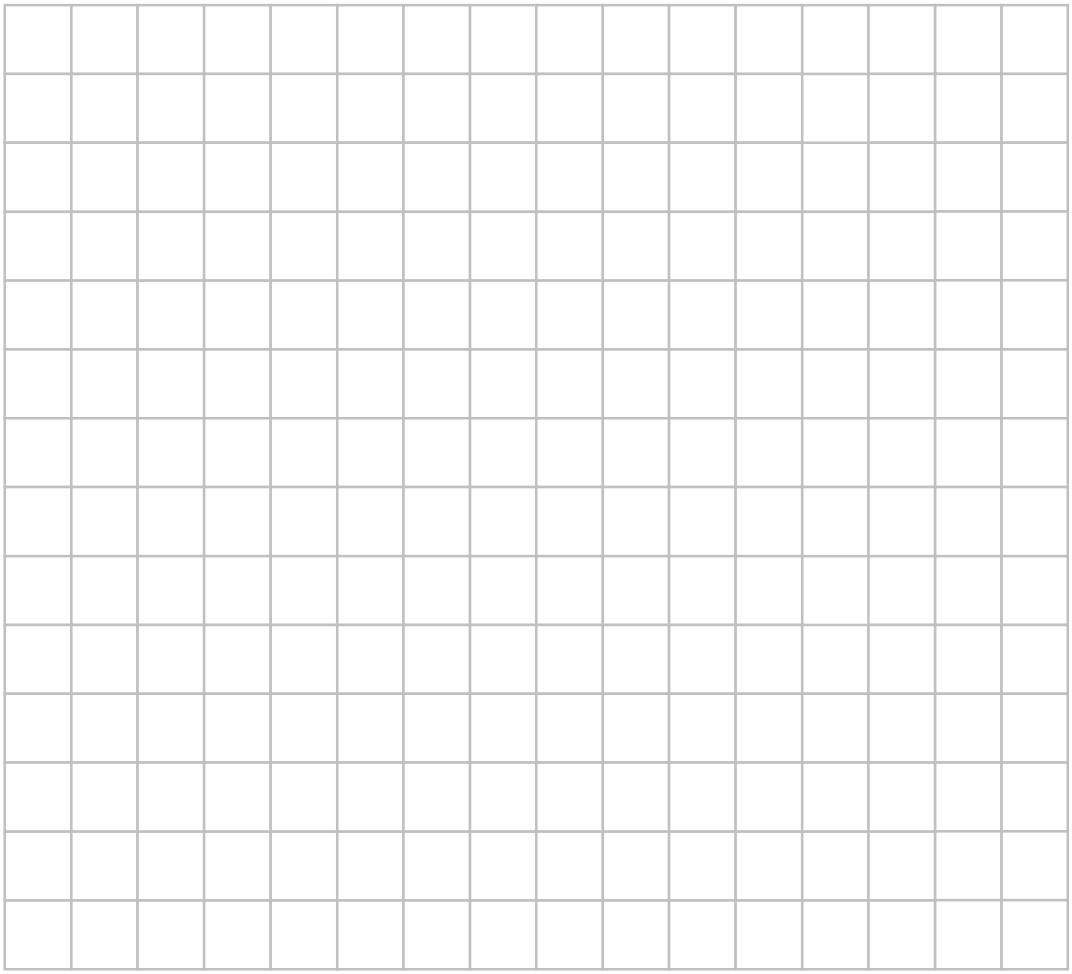
Acesso ao ambiente, Segurança e Entrada de Equipamentos		sim	não	na	ver
33.	Existe controle de acesso ao ambiente do computador central?	☐	☐	☐	☐
34.	Existe controle de acesso ao prédio?	☐	☐	☐	☐
35.	Existe no prédio local de armazenamento de equipamentos?	☐	☐	☐	☐
36.	Existe no prédio local de armazenamento de equipamentos no térreo?	☐	☐	☐	☐

Informações Adicionais	

Comentários

Desenho do Ambiente.

Descrição do posicionamento dos computadores no ambiente computacional. Cada quadrado mede 1:60 cm.



Comentários

Fonte: Adaptado de Brocade (2007)

APÊNDICE

APÊNDICE A – Arquivo de configuração do *Fence*

```
[root@pretto-ha1 sbin]# more fence_brocade
#!/usr/bin/perl
###############################################################################
###############################################################################
##
##  Copyright (C) Sistina Software, Inc.  1997-2003  All rights reserved.
##  Copyright (C) 2004-2007 Red Hat, Inc.  All rights reserved.
##
##  This copyrighted material is made available to anyone wishing to use,
##  modify, copy, or redistribute it subject to the terms and conditions
##  of the GNU General Public License v.2.
##
###############################################################################
###############################################################################
use Getopt::Std;
use Net::Telnet ();
# Get the program name from $0 and strip directory names
$_=$0;
s/.*\///;
my $pname = $_;
$opt_o = 'disable';         # Default fence action
# WARNING!! Do not add code bewteen "#BEGIN_VERSION_GENERATION" and
# "#END_VERSION_GENERATION"  It is generated by the Makefile
#BEGIN_VERSION_GENERATION
$FENCE_RELEASE_NAME="2.0.70";
$REDHAT_COPYRIGHT=("Copyright (C) Red Hat, Inc.  2004  All rights reserved.");
$BUILD_DATE="(built Fri Jul 13 14:41:13 EDT 2007)";
#END_VERSION_GENERATION
sub usage
{
    print "Usage:\n";
    print "\n";
    print "$pname [options]\n";
    print "\n";
    print "Options:\n";
    print "  -a <ip>          IP address or hostname of XenSource\n";
    print "  -h               usage\n";
    print "  -l <name>        Login name\n";
    print "  -n <string>      Xen VM to shutdown\n";
    print "  -o <string>      Action:  disable (default) or enable\n";
    print "  -p <string>      Password for login\n";
    print "  -q               quiet mode\n";
    print "  -V               version\n";
    exit 0;
}
sub check
{
  ($msg) = @_;
  print $msg."\n";
}
sub fail
{
  ($msg) = @_;
  print $msg."\n" unless defined $opt_q;
  $t->close if defined $t;
  exit 1;
}
sub fail_usage
{
  ($msg)=@_;
  print STDERR $msg."\n" if $msg;
  print STDERR "Please use '-h' for usage.\n";
```

```
  exit 1;
}
sub version
{
  print "$pname $FENCE_RELEASE_NAME $BUILD_DATE\n";
  print "$REDHAT_COPYRIGHT\n" if ( $REDHAT_COPYRIGHT );
  exit 0;
}
if (@ARGV > 0) {
   getopts("a:hl:n:o:p:S:qV") || fail_usage ;
   usage if defined $opt_h;
   version if defined $opt_V;
   fail_usage "Unknown parameter." if (@ARGV > 0);
   if (defined $opt_S) {
     $pwd_script_out = `$opt_S`;
     chomp($pwd_script_out);
     if ($pwd_script_out) {
       $opt_p = $pwd_script_out;
     }
   }

   fail_usage "No '-a' flag specified." unless defined $opt_a;
   fail_usage "No '-n' flag specified." unless defined $opt_n;
   fail_usage "No '-l' flag specified." unless defined $opt_l;
   fail_usage "No '-p' flag specified." unless defined $opt_p;
   fail_usage "Unrecognised action '$opt_o' for '-o' flag"
      unless $opt_o =~ /^(disable|enable)$/i;
} else {
   get_options_stdin();
   fail "failed: no IP address" unless defined $opt_a;
   fail "failed: no plug number" unless defined $opt_n;
   fail "failed: no login name" unless defined $opt_l;
   if (defined $opt_S) {
     $pwd_script_out = `$opt_S`;
     chomp($pwd_script_out);
     if ($pwd_script_out) {
       $opt_p = $pwd_script_out;
     }
   }
   fail "failed: no password" unless defined $opt_p;
   fail "failed: unrecognised action: $opt_o"
      unless $opt_o =~ /^(disable|enable)$/i;
}
if ( $opt_o =~ /^(disable|enable)$/i )
{
  $opt_o = "VM".$1;
}
#
# Set up and log in
#
$t = new Net::Telnet;
$t->open($opt_a);
$t->waitfor('/login:/');

$t->print($opt_l);
$t->waitfor('/assword:/');
$t->print($opt_p);
$t->waitfor('/$/');
$t->print('su -');
$t->waitfor('/assword:/');
$t->print($opt_p);
$t->waitfor('/#/');
#
# Do the command
#
if ( $opt_o eq "disable" )
{
        $cmd = "xe vm-shutdown force=true vm=$opt_n";
```

```
        $t->print($cmd);
}
elsif ( $opt_o eq "enable" )
{
        $cmd = "xe vm-start force=true vm=$opt_n";
        $t->print($cmd);
}
#
# Assume here that the word "error" will appear after errors (bad assumption! se
e next check)
#
#($text, $match) = $t->waitfor('/]#/');
#if ($text =~ /error/)
#{
#  fail "failed: error from XenSource\n";
#}
#
# Do a vm-list on the XenSource and look for the halted string to verify success
#

check "Command inserted OK.";
check "Checking status, PLEASE WAIT!";
sleep 5;
#$t->waitfor('/#/');
#$t->print("xe vm-list name-label=$opt_n");
if ( $opt_o eq "enable" )
{ if ( $t->print("xe vm-list name-label=$opt_n power-state=running") && !($text
=~ /running/) )
        {
                fail "failed: Xen VM $opt_n does not START\n";
        }
}
elsif ( $opt_o eq "disable" )
{ if ( $t->print("xe vm-list name-label=$opt_n powerstate=halted") && !($text =~
 /halted/) )
        {
                fail "failed: Xen VM $opt_n does not HALT\n";
        }
}
print "success: $opt_o $opt_n\n" unless defined $opt_q;
exit 0;
sub get_options_stdin
{
    my $opt;
    my $line = 0;
    while( defined($in = <>) )
    {
        $_ = $in;
        chomp;
        # strip leading and trailing whitespace
        s/^\s*//;
        s/\s*$//;
        # skip comments
        next if /^#/;

        $line+=1;
        $opt=$_;
        next unless $opt;
        ($name,$val)=split /\s*=\s*/, $opt;
        if ( $name eq "" )
        {
           print STDERR "parse error: illegal name in option $line\n";
           exit 2;
        }
        # DO NOTHING -- this field is used by fenced
        elsif ($name eq "agent" ) { }
        # FIXME -- depricated.  use "port" instead.
        elsif ($name eq "fm" )
```

```
        {
            (my $dummy,$opt_n) = split /\s+/,$val;
            print STDERR "Depricated \"fm\" entry detected.  refer to man page.\
n";
        }
        elsif ($name eq "ipaddr" )
        {
            $opt_a = $val;
        }
        elsif ($name eq "login" )
        {
            $opt_l = $val;
        }
        # FIXME -- depreicated residue of old fencing system
        elsif ($name eq "name" ) { }
        elsif ($name eq "option" )
        {
            $opt_o = $val;
        }
        elsif ($name eq "passwd" )
        {
            $opt_p = $val;
        }
        elsif ($name eq "passwd_script") {
                $opt_S = $val;
        }
        elsif ($name eq "port" )
        {
            $opt_n = $val;
        }
        # elsif ($name eq "test" )
        # {
        #    $opt_T = $val;
        # }
    }
}
[root@pretto-ha1 sbin]#
```