

**FACULDADE DE TECNOLOGIA ALCIDES MAYA - AMTEC
CURSO TECNOLÓGICO EM REDES DE COMPUTADORES**

JULIANO BOCK DA SILVA

**INFRAESTRUTURA VIRTUALIZADA
CLUSTER DE ALTA DISPONIBILIDADE**

**Porto Alegre
2019**

JULIANO BOCK DA SILVA

**Infraestrutura Virtualizada
Cluster de Alta Disponibilidade**

Projeto de Pesquisa apresentado como requisito parcial para obtenção do título de Tecnólogo em Redes de Computadores, pelo Curso de Tecnólogo em Redes da Faculdade de Tecnologia Alcides Maya - AMTEC

Porto Alegre

2019

LISTA DE ILUSTRAÇÕES

Figura 1 - Alta Disponibilidade.....	13
Figura 2 - Alta Performance.....	15
Figura 3 - Balanceamento.....	16
Figura 4 - Grid.....	20
Figura 5 - Virtualização Total.....	22
Figura 6 - Paravirtualização.....	23
Figura 7 - Estrutura do Cluster.....	27
Figura 8 - Node1.....	32
Figura 9 - Inicialização Cluster.....	35
Figura 10 - Status Cluster.....	36
Figura 11 - Criação de Instâncias.....	37
Figura 12 - Status de Sincronização.....	37
Figura 13 - Instâncias.....	38
Figura 14 - Inicialização.....	38
Figura 15 - Console.....	39
Figura 16 - Node Primário.....	39
Figura 17 - Movimentação de Instâncias.....	40
Figura 18 - Listagem.....	40
Figura 19 - Migração VM2.....	41
Figura 20 - Migração Geral.....	42
Figura 21 - Comando masterfailover.....	42
Figura 22 - Estrutura após migração geral.....	43
Figura 23 - Falha.....	44
Figura 24 - Estrutura após falha.....	44
Figura 25 - Comando de recuperação.....	45
Figura 26 - Listagem após falha.....	45
Figura 27 - Listagem após failover.....	45
Figura 28 - Ambiente após failover.....	46

LISTA DE SIGLAS

DRBD - Distributed Replicated Block Device

TI - Tecnologia Da Informação

IBM - Internacional Business Machines

HA - High Availability

HPC - Cluster High Performance Computing

RAID - Redundant Array Of Independent Disks

LVS - Linux Virtual Server

IP - Internet Protocol

TCP - Transmission Control Protocol

NASA - National Aeronautics And Space Administration

LAN - Local Area Network

WAN - Wide Area Network

SO - Sistema Operacional

VM - Virtual Machine

SUMÁRIO

1 INTRODUÇÃO.....	6
1.1 DEFINIÇÃO DO TEMA OU PROBLEMA.....	7
1.2 DELIMITAÇÕES DO TRABALHO.....	8
1.3 OBJETIVOS.....	9
1.3.1 OBJETIVO GERAL.....	9
1.3.2 OBJETIVO ESPECIFICO.....	9
1.4 JUSTIFICATIVA.....	9
2 REVISÃO BIBLIOGRÁFICA.....	10
2.1 SISTEMAS DISTRIBUIDOS.....	10
2.2 MAINFRAMES.....	11
2.3 CLUSTER.....	12
2.3.1 HIGH AVAILABILITY.....	12
2.3.2 DISTRIBUTED REPLICATED BLOCK DEVICE.....	14
2.3.3 HIGH PERFORMANCE.....	14
2.3.4 LOAD BALANCING.....	16
2.3.5 HEARTBEAT.....	17
2.4 LINUX VIRTUAL SERVER.....	17
2.5 BEOWULF.....	17
2.6 GRID.....	18
2.7 ENTENDENDO A VIRTUALIZAÇÃO.....	20
2.7.1 VIRTUALIZAÇÃO TOTAL E PARAVIRTUALIZAÇÃO	21
2.7.2 VIRTUALIZAÇÃO ASSISTIDA POR HARDWARE.....	23
2.7.3 HYPERVISORES.....	24
2.8 FERRAMENTAS DE VIRTUALIZAÇÃO.....	24
2.8.1 HYPER-V.....	24
2.8.2 VMWARE.....	25
2.8.3 XEN.....	26
2.8.4 VIRTUALBOX.....	26
3 DESCRIÇÃO DA SOLUÇÃO.....	27
3.1 INSTALAÇÃO E CONFIGURAÇÃO.....	28

3.2 FERRAMENTA DE GERENCIAMENTO GANET.....	29
3.3 SOFTWARE DE VIRTUAZAÇÃO XEN.....	30
3.4 DEFININAÇÃO DRBD.....	31
3.5 TOPOLOGIA DA REDE.....	31
4 METODOLOGIA.....	33
5 VALIDAÇÃO.....	33
5.1 INICIALIZAÇÃO.....	33
5.2 AMBIENTAÇÃO.....	36
5.3.1 INSTÂNCIAS.....	36
5.4 DESLOCAMENTO DE INSTÂNCIAS.....	39
5.5 DESLIGAMENTO DO NÓ PRIMÁRIO.....	41
5.6 DESLIGAMENTO NÃO PREVISTO.....	43
6 CONCLUSÃO.....	47
7 CRONOGRAMA.....	49
8 REFERÊNCIAS BIBLIOGRÁFICA.....	50

1 INTRODUÇÃO

A disponibilidade constante de recurso e informações em um ambiente tecnológico é uma preocupação constante, tornando o acesso a informação como um objetivo principal.

A ambientação tecnológica, atualmente, visa a segurança de dados e a disponibilidade constante dos recursos, já é fato que toda essa arquitetura volta-se a esse objetivo. As empresas devem preocupar-se em sua modernização, levando em consideração estratégias que alinhem investimentos e negócios. (VERAS, 2011)

O uso de tecnologias baseadas em softwares opensources são formas de disponibilizar a ambientações que dispõe da necessidade de manter a constâncias na disponibilidade, segurança e alto poder de processamento, bem como a necessidade de economia, algo promissor e bem sucedido. (SALLES et. al., 2009)

Uma forma de obter o proveito, de forma mais eficiente, de recursos oferecidos por hardwares, bem como disponibilidade e flexibilidade, é através da tecnologia de virtualização, onde se percebe um aumento significativo em diversos setores da TI. (Carissimi, 2015)

O Cluster é uma tecnologia que visa disponibilizar uma solução para ambientes que necessitem de uma alta disponibilidade de seus recursos bem como para solucionar alguns problemas relativos a demanda de processamento. É uma tecnologia bem estrutura e muito utilizada, onde possibilita que computadores trabalhem em conjunto. (SLTI, 2006)

A virtualização trabalhando de forma agregada ao cluster possibilita uma série de vantagens que podem ser utilizadas na solução de impasses relativos aos atuais requerimentos da TI. Através da combinação dessas duas tecnologias é possível obter em uma estrutura tecnologia com um alto poder de processamento, gerenciamento otimizado, confiabilidade bem como segurança e confiabilidade de uma estrutura. (SLTI, 2006)

O foco deste projeto é a implementação de uma estrutura que disponibilizara uma ambientação virtualização fundada na utilização de cluster de computadores, utilizando a tecnologia de Alta Disponibilidade.

Esse ambiente visa à aplicabilidade desta tecnologia de uma forma que possa ser integrada no cotidiano de empresas ou ambientações que utilizam de computadores em suas atividades. São demonstradas ocorrências que necessitam de foco para proporcionar a alta disponibilidade dos recursos.

Nesse contexto, foi elaborado um ambiente utilizando a tecnologia cluster com 2 (dois) nós, na qual o volume de dados será compartilhado pelo gerenciador de volumes DRBD. Os volumes serão destinados ao software Ganeti, que será o software responsável pelo gerenciamento do referido cluster bem como as instâncias e as camadas de virtualização.

Nesse ambiente ilustrativo, será apresentado o gerenciamento e manipulação das instâncias que estão alocadas no cluster, visando à aproximação de uma situação real.

1.1 Definição do Problema

Para Burgelman, Christensen e Wheelwright (2013, pg.3) a tecnologia é uma ferramenta de suma importância para empresas em sua totalidade, utilizar da tecnologia para manter-se competitiva no mercado deve ser colocado como fator essencial na estratégia empresarial. Segundo Borini (2018, 2018) o investimento em tecnologia da informação (TI), pelas empresas de médio e grande porte, chega a 7,7% de suas receitas. Em empresas menores essa porcentagem pode ultrapassar os 10%.

Muitas dessas empresas, por sua vez, optam por utilizarem computadores de grande porte para processar suas solicitações de dados. Porém esses tipos de equipamento costumam ocupar espaços grandes e específicos onde fatores climáticos devem ser levados em consideração (SLTI, 2006).

SLTI (2006) diz que, computadores de grande porte quando atingem sua capacidade máxima de processamento, muitas vezes torna-se necessário a inclusão de novo hardware ou troca total do equipamento, o que torna inviável economicamente.

Com o aumento da demanda na solicitação de acesso a informação, empresas tendem preocupar-se com a estrutura tecnológica, Onde a troca de todo o

equipamento por um mais robusto e com poder de processamento e capacidade mais alto, acaba sendo uma opção cara.

Posto isso, percebe-se que para manter um ambiente tecnológico que possibilite aos usuários uma alta taxa de disponibilidade de acesso a recursos, processamento de qualidade e garantia na segurança de dados pode custar caro.

1.2 Delimitações do Trabalho

Neste projeto será demonstrado à tecnologia de Virtualização e Cluster bem como a junção das duas, tendo como um dos focos o incentivo a sua utilização em ambientes tecnológicos ou corporações, que buscam uma alternativa para manter a disponibilidade constante de seus diversos projetos bem como a segurança na garantia de entrega. Não será apresentado um processo específico visto que a aplicação desta tecnologia, dependendo da configuração, poderá abranger uma grande quantidade de processos, dependendo de sua necessidade ou demanda, na qual não entrará no mérito.

Não será abordada a totalidade das funções presentes nessas tecnologias visto que existe uma gama considerável em relação a isso e o ambiente a qual foi produzido e desenvolvido o projeto, possui limitações físicas, impossibilitando sua aplicação.

Serão demonstradas de forma superficial as configurações das ferramentas Xen, Ganeti e DRBD, que são utilizadas na estrutura, visto que não são o foco deste projeto, que visa demonstrar a virtualização de cluster e seus benefícios.

A validação e demonstração por sua vez será através da simulação de 3 (três) ambientes: O deslocamento de instâncias, o desligamento de nó primário e a simulação de desligamento não previsto.

1.3 Objetivos

Os objetivos dividem-se em: geral e específicos.

1.3.1 Objetivo Geral

Proporcionar maior disponibilidade e desempenho em ambientes tecnológicos com a implementação de uma estrutura de cluster virtualizado.

1.3.2 Objetivos Específicos

- Proporcionar uma maior disponibilidade de recursos com a utilização de Cluster
- Proporcionar um gerenciamento centralizado de servidores.
- Proporcionar maior segurança em ambientes de TI.

1.4 Justificativa

Com o aumento na procura por acesso a dados tecnológicos, a estrutura no âmbito da Tecnologia da informação orienta-se na alta disponibilidade com foco em segurança de dados.

As instituições devem preocupar-se com estratégias de negócio baseando-se na tecnologia da informação (VERAS, 2011).

Diante disto, um modo para solucionar a primordialidade da alta disponibilidade, segurança e processamento de qualidade consiste na utilização de softwares opensource, utilizando tecnologia eficaz e sem custo (SALLES et. al.,2009).

Segundo Phillips e Baltazan (2012, pg.152) com a virtualização é possível hospedar diversos sistemas em um único servidor físico, fornecendo para as empresas mais espaço nos data centers, tendo em vista a dispensa de ampliação

das instalações. Com isso podemos ter maior aproveitamento dos recursos de hardware, maior flexibilidade e uma taxa maior de disponibilidade nos serviços.

A tecnologia de Cluster é outra inovação amplamente utilizada nas mais diversas áreas da TI, pois através dela é possível manter a disponibilidade de um serviço além de ajudar a garantir a demanda de processamento. Essa tecnologia permite que diversos computadores trabalhem como se fossem um só, em consequência disso, teremos um aumento de desempenho ou capacidade de processamento. (Pitanga, 2003).

Em contrapartida, a combinação entre computação em cluster e virtualização fornece uma série de vantagens, formando uma estrutura capaz de disponibilizar atendimento para as altas demandas da TI, resultando em um gerenciamento mais simples, maior poder de processamento e segurança.

2 REVISÃO BIBLIOGRÁFICA

2.1 SISTEMAS DISTRIBUIDOS

Para Pitanga (2002, pg.22) um Sistema Distribuído é um conjunto de computadores com processamentos independentes, que, por meio de um gerenciamento e comunicação sincronizado processa aplicações em localidades distintas, dando a impressão a seus usuários de que seja um único sistema.

Coulouris, Dollimore e Kindberg (2007, p.15) nos diz que, sistemas distribuídos são todos aqueles elementos de computadores que estão relacionados em rede e transmitem ordens através de mensagens.

Segundo Pitanga (2002, pg.22), tem como aspecto relevante de um sistema distribuído a possibilidade de aumentar ou diminuir toda a estrutura, com isso, torna-se viável a inclusão de elementos independente no conjunto, adicionando amplitude e recursos em toda a estrutura integral.

Conforme Coulouris Et. al (2007), o objetivo fundamental de um sistema distribuído é a distribuição de recursos gerenciáveis por servidores possibilitando servir conforme necessidades dos clientes.

Ainda para Coulouris Et. Al (2007), todos os sistemas computacionais estão sujeitos a falhas, em sistemas distribuídos as falhas funcionam de modo diferente.

Computadores são isolados quando ocorre alguma falha na rede, no entanto, não resulta em sua parada, pois os componentes podem falhar de modo independente, permitindo que os outros ainda funcionem. Essas falhas por sua vez, acabam se tornando imperceptível para os usuários e sistemas, uma vez que um componente não recebe uma atribuição específica.

Um dos componentes principais da aplicabilidade de um sistema distribuído é o Middleware, que é um software específico, que tem como função lidar com as diferenças de nível dos sistemas operacionais e do hardware. Um middleware além de solucionar impasses de heterogeneidade, proporciona aos profissionais um modelo computacional uniforme. Coulouris Et. al (2007).

Diante disto, com a possibilidade de não ficar limitado a uma única arquitetura de hardware, existe um melhor aproveitamento de recursos, em contrapartida o valor de implantação e manutenção tornasse um fator relevante. (Tenenbaum, 2007)

2.2 MAINFRAMES

O Surgimento do Mainframe surgiu em 1946 apos uma série de gerações de equipamentos fabricados pela IBM, os mainframes por sua vez, são equipamentos de grande porte, projetados para realizar o processamento de uma vasta gama de solicitações, em uma rede distribuída, feitas por usuários. São equipamentos que necessitam de um ambiente próprio e climatizado, ocupam um vasto espaço nas corporações. (SLTI, 2008).

Para Laudon e Laudon (2007, pg 101) o mainframe tem como característica grande desempenho e capacidade elevada, podendo processar uma gama enorme de dados extrema velocidade. Laudon (2007, pg 102), explique que o hardware de um mainframe é bem versátil e consegue suporta trocas e inclusões de processadores sem necessitar que seja feito o seu desligamento. Por ser um equipamento que necessita de um espaço amplo e um ambiente próprio para o seu funcionamento além de ser um caro, o mainframe vem perdendo espaço nos comércios e empresas que estão optando por uma solução mais viável, utilizando servidores de arquitetura de computadores comuns e servidores Unix. (Loudon, pg. 101)

2.3 CLUSTER

Para Pitanga (pg. 25, 2002), ao utilizar mais de um computador compartilhado em prol da resolução de um problema, teremos um cluster (agrupamento) já para Morimoto (2005), o cluster é constituído por uma série de dispositivos administrado por um ponto central, na qual cada nó executa uma tarefa.

A computação em cluster ganhou notoriedade a partir dos anos 80, tendo em vista alguns aspectos importantes tais com: o surgimento de processadores com alto índice de desempenho, a computação distribuída e paralela e as redes de comunicação de baixa latência. (PITANGA, pg. 26, 2002)

Conforme explica Pitanga (pg. 27, 2002), "podemos dividir os cluster em duas categorias básicas: Alta Disponibilidade (HA - Hight Availability) e Alto Desempenho de computação (HPC - Hight Performance Computing).

O cluster se popularizou como modelo de computação distribuída uma vez que minimizava os custos com o investimento em equipamentos, pois sua instalação poderia ser feita através de equipamentos convencionais, tornando toda a estrutura mais econômica, ajustável e simples.

2.3.1 HIGH AVAILABILITY

Os sistemas de computação vêm se expandindo em todos os meios empresariais, comerciais e residenciais onde se percebe que existe uma preocupação em manter todo o equipamento funcionando. A Alta disponibilidade tende a garantir que todos os serviços de rede, processamento e armazenamento mantenham-se contínuos, inclusive ao ocorrer falhas nos dispositivos de hardware ou software. (PITANGA, pg.29, 2002).

Sendo assim, a alta disponibilidade visa manter o sistema em perfeitas condições a todo o momento, sem paradas não programadas, mantendo assim, a qualidade de todo o serviço. Seu modo de funcionamento é manter os serviços e equipamentos sempre ativos, sempre trabalhando em conjunto com os equipamentos, replicando-os, evitando assim, qualquer tipo de parada, caso ocorra em um dispositivo, as demais passam a responder por ela. (pitanga, pg.26, 2002).

Pitanga (2008) relata que qualquer ambiente computacional dependente de recursos e sistemas, é relevante considerar toda a sua infraestrutura ou qualquer coisa ligada a ela, pois um hardware pode falhar devido ao tempo de uso ou defeitos de fábrica, ou um software pode estar mal configurado ou incompatível.

Dito isso, existem soluções para falhas que estão ligadas a hardware ou softwares. A disponibilidade por hardware se dá através do monitoramento de recursos, impactando nos custos da estrutura. Os dispositivos utilizam mais de uma fonte de alimentação, sistemas que fornecem energia de forma independente, Hard Disks utilizando RAID ou espelhamento além de do benefício de utilizarem o Hotswap que possibilita a troca de componentes sem afetar a disponibilidade do equipamento.

A HA por Software se dá através de softwares como o Linux Virtual Server (LVS) ou o HeartBeat que são responsáveis pelo gerenciamento e monitoramento dos equipamentos e servidores, caso estes dispositivos apresentem alguma falha, esse software redireciona as tarefas ou serviços para os equipamentos ainda disponíveis.

A figura 1 exemplifica a estrutura de um Cluster de Alta Disponibilidade, onde caso um Webserver cai, os demais passam automaticamente a disponibilizar o recurso enviado por este.

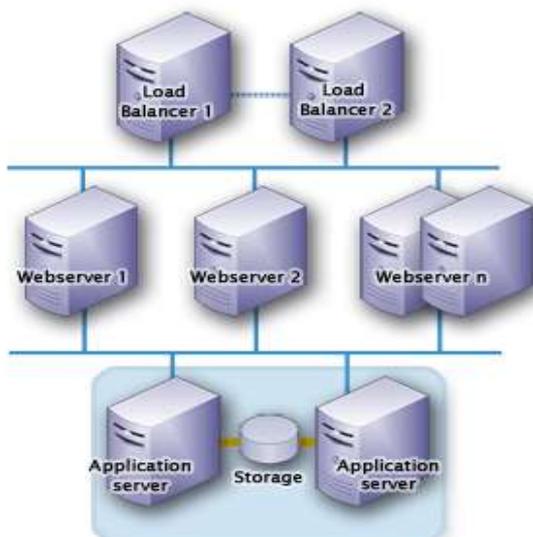


Figura 1 – Alta Disponibilidade
Fonte : (ha-cc.org)

2.3.2 DISTRIBUTED REPLICATED BLOCK DEVICE

O Distributed Replicated Block Device, ou simplesmente DRBD, é criado através de um módulo no kernel do Linux juntamente com uma série de scripts e programas, no qual a sua função é criar um sistema de arquivos distribuídos pela rede. Seu princípio de fundamento é similar a formação de um RAID (Redundant Array of Independent Disks) de discos físicos em apenas um dispositivo. (SLTI, 2008)

Suas transferências são realizadas por intermédio da interface da rede, sendo usada uma série de protocolos, onde os mais conhecidos são os: IP (Internet protocol) e o TCP (Protocolo de Controle de Transmissão). (SLTI, 2008)

Com o DRBD, os nós conectados na rede passam a ter um vínculo através do gerenciador onde aloca uma partição comum entre eles, no entanto, não é acessível pelo equipamento que detém o volume de dados. Toda configuração é produzida no nó controlador que encaminha para os níveis inferiores e propaga para os demais nós secundários, que gravam essa configuração. (SLTI, 2008)

Diante disto, porventura o nó primário sofra alguma irregularidade ou falha, o nó secundário é imediatamente promovido a primário. No momento em que o nó que ocorreu a falha volte a operar normalmente, poderá voltar a operar como nó primário, mediante a configuração definida. (SLTI, 2008)

2.3.3 HIGH PERFORMANCE

Essa tecnologia tem como objetivo o desenvolvimento de supercomputadores, através de dispositivos atuando em conjunto para no intuito de criar o processamento paralelo e construção de aplicações paralela entre si. (Pitanga 2002)

O cluster de alto desempenho apresenta uma medida viável para universidade e empresas de pequeno e médio porte obter um desempenho mais condizente com a realidade, utilizando aplicações paralelizáveis, tendo em vista os valores elevados para a aquisição de equipamentos específicos para tal finalidade, esse método se torna mais palpável para as instituições. (PITANGA, pg.30, 2002)

SALES et. Al., 2009 comenta que vários computadores comuns operando em conjunto conseguem gerar um maior número de Gigaflops, logo seu custo é menor se comparado a um servidor de capacidade semelhante.

Esse tipo de configuração de cluster visa dividir processos grandes em vários processos menores, em seguida através da rede é distribuído para os dispositivos (nodos) conectados, atribuindo disponibilidade e alta performance para os serviços. (Sales et. Al., 2008)

A National Aeronautics and Space Administration (NASA), desenvolveu um cluster de alto processamento, dando-lhe o nome de Beowulf, muito utilizado em computação científica, áreas médicas e financeiras. (SALLES et. Al., 2008)

A figura 2 exemplifica um Cluster de Alta Performance, onde o poder de processamento de todos os nós passam à um único servidor capaz de suporta-los, neste caso, apresentado na figura pelo NFS Server.

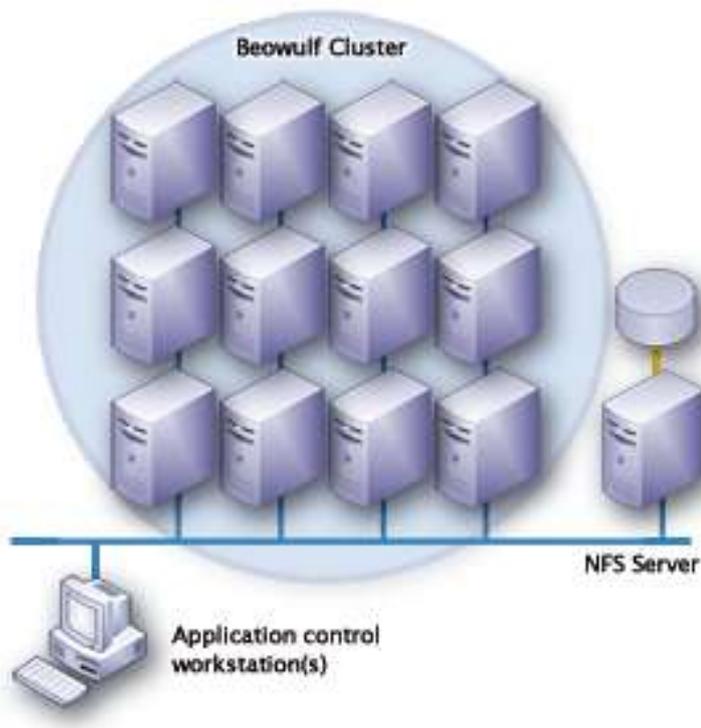


Figura 2 – Alto Desempenho
Fonte : (ha-cc.org)

2.3.4 LOAD BALANCING

A base de um cluster de balanceamento (Load Balancing) é um conjunto de servidores executando uma mesma função. O cluster de balanceamento tem como

finalidade a verificação da utilização de todos os nodes (nós) e é de extrema importância ao ser realizado a distribuição das solicitações dos usuários para os diversos nós da rede. Através do algoritmo de programação, essa tecnologia busca o nó com melhor desempenho estimado para receber a solicitação do usuário, garantindo assim um melhor desempenho para as solicitações do usuário. (Ha-cc, 2018)

Um ponto importante é a qualidade do cluster de balanceamento de carga, ou seja, essa qualidade visa a disponibilidade do sistema de realizar uma previsão sobre qual node disponibilizara o melhor desempenho para a aplicação individual, também sendo capaz de conectar o cliente a uma sessão disponível. (Ha-cc, 2018)

Uma vez que o nó estiver sem serviço, o sistema permanecerá operante, pois o balanceador será capaz de reconhecer o nó com falha e encaminhará o serviço para o próximo, o desempenho pode até cair mais os serviços continuarão a ser prestados pelo sistema. (Ha-cc, 2018)

A figura 3 exemplifica um Cluster de Balanceamento, onde apresenta uma estrutura composta por 03 Webserver e seu poder de processamento é disponibilizado a um host conforme solicitação.

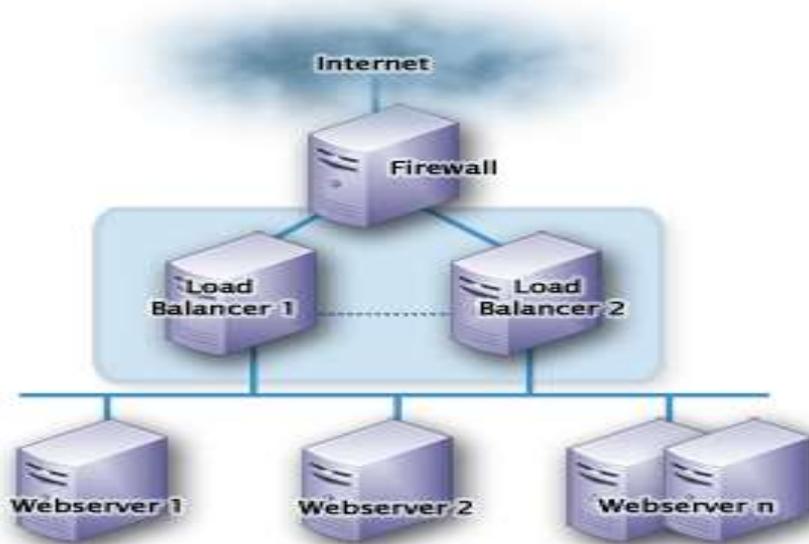


Figura 3 – Balanceamento

Fonte : (ha-cc.org)

2.3.5 HEARTBEAT

Heartbeat é basicamente um programa de computador de monitoramento, é um projeto GNU/Linux, que tem como função monitorar equipamentos ligados a rede utilizando uma interface, visa manter a disponibilidade do serviço, caso um nó venha a falhar ele garante que outro nó mantenha do serviço deste. Trabalha através de servidores primários e secundários, onde caso haja falha nos serviços do servidor primário, o Heartbeat informa ao servidor secundário sobre a inoperabilidade e logo este assume todos os serviços que estavam alocados no primário, evitando prejuízos no funcionamento do sistema mantendo-o disponível. (SAMBA, 2007).

Seu funcionamento se dá através da configuração de tempo de resposta, o qual o servidor primário envia de tempo em tempo informações sobre o seu funcionamento, no momento em que o tempo definido seja ultrapassado o servidor secundário entende a inoperabilidade do primeiro e passa assumir todos os serviços. (SLTI, 2008)

Atualmente existem duas versões do heartbeat, a 1.2.3 e a versão 2, na qual a primeira limita-se a dois nós e a segunda pode ser configurada para até 16 nós. (SAMBA, 2007)

2.4 LINUX VIRTUAL SERVER

O LVS ou Servidor Virtual Linux, é um servidor altamente escalável, pois permite a inclusão de novos dispositivos em um cluster de servidores reais. A arquitetura do LVS é extremamente simples aos usuários, pois transparece ao usuário apenas um único servidor virtual de alto desempenho. (linuxvirtualserver.org)

Os dispositivos são conectados por LAN de alta velocidade ou WAN geograficamente separadas. Sua escalabilidade é obtida pela inclusão de nós ao cluster e sua alta disponibilidade é fornecida através da detecção das falhas dos nós e pela reconfiguração do sistema. (linuxvirtualserver.org)

2.5 BEOWULF

O Cluster Beowulf teve sua origem em meados de 1994, foi projetado pelo professor de ciencias da computação, da universidade de Indiana, Thomas Sterling e pelo programador Donald Becker, ambos pesquisadores da NASA. O nome

Beowulf surgiu como referencia ao lendário herói escandinavo, descrito em poema como um homem de grande força e coragem que enfrentava diversos desafios contra criaturas, esse software enfrentaria o desafio da alta demanda e processamento, por isso recebeu este nome. (SALLES et. al., 2009)

Na arquitetura do cluster Beowulf um ou mais dispositivos irão atuar como "Master" (Front-end), e as demais terão o papel de Slaves (Back-and). O servidor Master será o responsável pelo controle total do cluster bem como a distribuição de tarefas aos slaves. (ALECRIM, 2013)

Nos dispositivos Slaves (back-and), não existe a necessidade de instalação de dispositivos de entrada como teclados ou mouses bem como a dispensa de discos rígidos. O Backand tem como tarefa executar funções de processamentos solicitadas pelo Master (front-and). (ALECRIM 2013)

O Cluster Beowulf tornou-se uma opção mais barata para empresas que necessitam de supercomputadores, pela sua simplicidade de implementação e por não necessitar de computadores potentes. Pode ser implementado facilmente em uma rede interna utilizando o software adequado, todos os computadores da instituição poderiam fazer parte de um cluster Beowulf, inclusive poderia ser utilizado normalmente pelos funcionários, haja vista o a tecnologia utilizar unicamente os ciclos ociosos do processador. (MORIMOTO, 2005)

2.6 GRID

Conforme explica Phillips e Baltzan (2012, pg.137) "A computação em grade (grid computing) é uma agregação de computação geograficamente dispersa, armazenamento e recursos de rede, coordenada para oferecer melhor desempenho, maior qualidade de serviço, melhor utilização e um acesso mais fácil aos dados."

A computação em grade inicialmente objetivava a substituição das plataformas baseadas em frameworks ou supercomputadores que no momento apresentavam um custo alto de manutenção e compra, levando em consideração também a rápida evolução tornando-o ultrapassado se fazendo necessária a troca de todo o equipamento, tudo isso em um período de tempo muito curto. (SLTI, 2008)

Com a computação em grade é possível realizar a virtualização da computação distribuída assim como produzir uma imagem do sistema possibilitando a todos os

usuários o acesso contínuo aos serviços pois baseia-se na interoperabilidade e compartilhamento de recursos. (PHILLIPS e BALTZAN, 2012).

Uma vantagem da computação em grid é "Open Grid Services Architecture", ou seja, baseia-se em normas e protocolos abertos permitindo assim a comunicação entre ambientes distintos tanto na sistemática como na localidade, por exemplo, empresas localizadas em lugares diferentes podem ser ligadas através da web, formando assim um grid. (PHILLIPS e BALTZAN, 2012). O Grid possibilita a integração recursos computacionais variados e espalhados em um único sistema virtual, aumentando a velocidade de execução de serviços paralelos.

Tanenbaum e Steen (pg. 103, 2007) dizem que tecnologias heterogenias como computadores comuns, periféricos e até mesmo supercomputadores pode compor uma grid.

Existem pesquisas sobre arquiteturas para os Grids, com finalidade de padronizar essa tecnologia, a fim de manter a interoperabilidade entre as organizações virtuais. Criaram-se então cinco camadas: Ambiente, Conectividade, Recursos, Coletivo e Aplicações. (Pitanga, pg.328, 2002)

A camada de ambiente é a interface controladora de recursos que serão compartilhados. A camada de conectividade é responsável pela comunicação e autenticação para transferência pela rede para o Grid. A camada de recursos é onde tem definido os protocolos de negociação segura, inicialização, monitoração e controle de recursos. A Camada coletivo define os serviços de diretório, alocação, monitoramento, pesquisa de software, colaborativos e de programação para as grades. Por fim a camada de aplicação que compreende as aplicações dos usuários. (Pitanga, pg.330, 2002)

Imagem abaixo ilustra uma computação em GRID, onde pode ser notado uma variedade de tipos de dispositivos.



Figura 4 – Grid
Fonte : (Icloud.com)

2.7 ENTENDENDO A VIRTUALIZAÇÃO

A virtualização originou-se na década de 1960, mas somente em meados dos anos 2000 que passou a ser amplamente difundida. A virtualização foi possível através de tecnologias desenvolvidas que tinham como função disponibilizar, para vários usuários, acesso simultâneo a dispositivos que produziam processamentos em lote. (RedHat)

Stallings (2016, pg. 7) nos caracteriza a virtualização como “a criação de uma versão virtual (em vez de real) de algo, e, na computação, isso significa criar versões virtuais dos sistemas operacionais, servidores, dispositivos de armazenamento e redes”.

Um dos motivos do surgimento de máquinas virtuais foi o empasse em relação a mainframes que apresentavam seu sistema operacional próprio, pois através dessas máquinas virtuais era possível executar softwares antigos e importantes nesses mainframes. Em outras palavras, a virtualização permitiu a execução de diversas aplicações de sistemas operacionais diferentes em um único hardware, proporcionando flexibilidade e portabilidade. (CARISSIMI, 2018)

Qualquer Data Center usava a virtualização como base, pois vários servidores virtuais poderiam ser criados em apenas um servidor físico, facilitando o controle e administração. (VERAS, 2011).

Veras (2011, pg.6-7) afirma que servidores virtualizados são similares a servidores físico, aperfeiçoam a utilização dos recursos o que torna as aplicações independente do hardware. Possibilita o uso de um espaço menor e de maior otimização para a infraestrutura bem como redução de custo de energia. Existe ainda a possibilidade de particionar um servidor físico em vários servidores lógicos.

Veras (2011, pg.7) também afirma que é possível ampliar o poder de processamento além de melhorar a disponibilidade, seu gerenciamento também é facilitado, pois a administração da estrutura virtualizada é centralizada.

2.7.1 VIRTUALIZAÇÃO TOTAL E PARAVIRTUALIZAÇÃO

Para Veras (2015, pg. 56) "A virtualização total realiza a completa abstração do sistema físico, e cria um sistema virtual completo", não necessitando realizar modificações no sistema operacional e na aplicação. As aplicações nesta modalidade funcionam completamente independentes das aplicações e recursos físicos do servidor facilitando a migração de máquinas virtuais. Um fator importante é que todo o processo e a ligação entre o hardware e processos relativo a seu funcionamento, são monitoradas e testado pelo hypervisor, afetando assim a performance do SO. Sendo assim, o desempenho da virtualização total será determinado pelo hypervisor.

Segundo Veras (2015, pg. 57) "A virtualização total consiste em prover uma réplica (virtual) do hardware subjacente de tal forma que o sistema operacional e as aplicações possam executar como se estivessem diretamente sobre o hardware original". A figura 05 ilustra como ocorre o funcionamento da virtualização total.

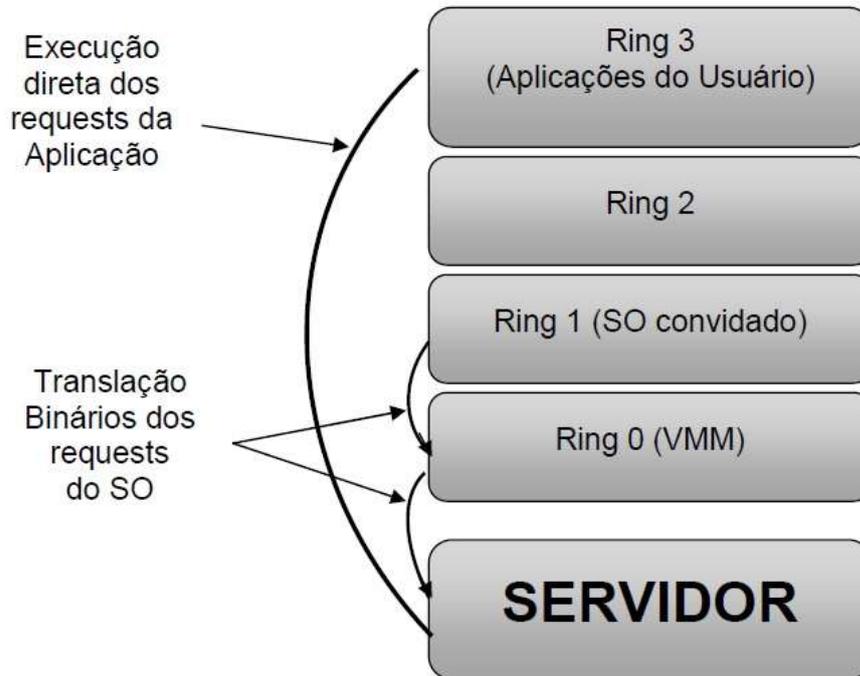


Figura 5 – Virtualização Total
Fonte: (VERAS, 2015)

A paravirtualização pode ser considerada uma alternativa para transpor as desvantagens da virtualização total, pois nesta tecnologia, o hypervisor tem seus próprios drivers de dispositivo o que facilita o acesso aos dispositivos de hardware, desta maneira, o hypervisor não precisa acompanhar as instruções. (VERAS, 2015 pg.67).

Dito isso, o sistema operacional passa a ter um ganho de desempenho na execução de tarefas, pois não existe a necessidade de interpretar as instruções para acessar o hardware, uma vez que a paravirtualização possui drivers nativos compatíveis com o hardware, diferente de como é tratado na virtualização total. (VERA, 2015 pg.67). A figura 6 ilustra como ocorre o funcionamento da paravirtualização.



Figura 6 – Paravirtualização
 Fonte: (VERAS, 2015)

2.7.2 VIRTUALIZAÇÃO ASSISTIDA POR HARDWARE

A virtualização total e a paravirtualização tem suas vantagens e desvantagem, a paravirtualização oferece um desempenho eficiente, mas tem a necessidade de alterar o SO guest, já a virtualização total não necessita de modificação, mas não apresenta um desempenho muito bom. Para solucionar esse problema a Intel e AMD, em parceria, criaram extensões diretas no hardware de arquitetura x86 que passou a ser conhecida como Virtualização Assistida por Hardware. (OLIVEIRA, 2018)

A Virtualização Assistida por Hardware tem como objetivo melhorar o desempenho por meio dessas extensões assim como o suporte a virtualização. Funciona através do modo "root" e do modo "non-root", onde o sistema é executado em non-root, mas o hipervisor é executado no modo root, passando a ter prioridade sobre o sistema. (OLIVEIRA, 2018)

2.7.3 HYPERVISORES

Hypervisor também conhecido como Monitor de Máquina Virtual (VMM) é um sistema que se encontra entre o hardware e o sistema operacional. (faqinformatica, 2018). Mauser e Diógenes (2016), explica que o hypervisor é responsável pela padronização de como a virtualização irá funcionar no sistema.

Para Veras (2015, pg. 53), o hypervisor tem como função o "agendamento, gerência da memória e manutenção do estágio da máquina virtual. Também permite criar partições para as máquinas virtuais, mantendo o isolamento entre as partições".

A qualidade da virtualização se dá através o desempenho do hypervisor, onde é necessário como característica à flexibilidade e celeridade na configuração dos recursos sem a interrupção do servidor bem como a segurança dos recursos virtualizados. Existem dois tipos de hypervisores, o tipo 1 conhecido como Baremetal e o tipo dois. (VERAS, pg.53, 2015).

O Hypervisor tipo um funciona direto no hardware do servidor, controla o compartilhamento de recursos, tais como, processamento, formas e armazenamento dispositivos e memória, entre as máquinas virtuais fazendo com que elas achem que são recursos próprios. (VERAS, 2015.)

Para Veras (2015, pg. 54) O hypervisor tipo dois "funciona como uma aplicação que fornece um ambiente de execução para outras aplicações". É formada por um sistema operacional guest (convidado) e por um dispositivo virtual criado através das possibilidades do hardware físico.

2.8 FERRAMENTAS DE VIRTUALIZAÇÃO

A virtualização dispõe de diversos softwares para a sua implementação, onde cada um possui suas características próprias, existem softwares livres de fácil administração assim como existem empresas que são capazes de fornecer soluções para esse tipo de tecnologia. (CARISSIMI, 2015)

2.8.1 HYPER-V

Hyper-V é uma tecnologia que possibilita a criação e administração de computadores virtuais através da virtualização do hardware de um computador físico. (Portal GSTI, 2019)

A virtualização passou por grandes investimentos feitos pela Microsoft, desde investimento originou-se o Hyper-V, que foi disponibilizado em uma atualização do Windows Server 2008. Em sua fase inicial suportava apenas quatro máquinas virtuais com 64GB de RAM. (APOLINÁRIO, 2015)

Segundo Apolinário (2015, pg. 1), o Hyper-v vem ganhando destaque no mercado como sendo uma excelente solução para a virtualização, pois possui código aberto sendo o Microsoft Hyper-V uma plataforma de virtualização gratuita.

Para IVA (2018) o hyper-V não está limitado apenas ao dispositivo do usuário mais pode virtualizar servidores também, o que o torna mais vantajoso em relação a outros softwares, como por exemplo, o virtualbox ou o fusion. É disponibilizado em três versões, o Hyper-V para Windows Server, Servidores Hyper V e Hyper-V para Windows 10.

Uma das funcionalidades do Hyper-V é a possibilidade que tem de transferir máquinas virtuais criadas em um servidor para outros possibilitando reparos, caso necessário, além de conseguir replicar a máquina virtual que por sua vez se torna muito útil para substituir um serviço indisponível ou realizar teste. (Alecrim, 2013)

Uma perspectiva importante descrita por Carissimi (2011) é “o aspecto de segurança através de um restrito controle de regras de permissões integrado com o Active Directory e com políticas de grupo”.

2.8.2 VMWARE

Segundo Morimoto (2008, pg. 622) "O VMware usa o conceito clássico de virtualização tentando, sempre que possível, converter os comandos usados pelo sistema dentro da máquina virtual em comandos que o sistema host entenda e execute diretamente". Morimoto (2008, pg. 622) também explica que um sistema virtual criado com o VMware matém um desempenho parecido com um servidor real pois sua tecnologia interpreta e converte o mínimo de instruções necessárias influenciando na performance do servidor virtual. Existem duas versões gratuitas desse software, o VMware Server e o VMware Player.

O Vmware Server é utilizado em servidores dedicados, não utiliza ambiente gráfico rodando remotamente, sua administração e acesso é através de uma

interface via web, conhecido como VMware Management Interface ou MUI. (MORIMOTO 2008, pg. 625)

2.8.3 XEN

O Hypervisor Xen originou-se através de um projeto Open Surce, com arquitetura x86, na universidade de Cambridge, foi comprada pela empresa CitrixSystem, após 4 ano da sua primeira versão em 2003. (MACEDO, 2012).

O Xen é um software de virtualização que utiliza a ideia de paravirtualização, consegue passar um perca de desempenho de mais ou menos 5% no sistema guest, o que é considerado muito pequeno, pois através da paravirtualização os recursos do hardware são divididos de forma transparente, porém não contém drives de dispositivo o que o impede de rodar diretamente um sistema operacional. (MORIMOTO 2008, pg. 622)

Esse software trabalha com o conceito de domínios, visando contornar esses problemas de drives, onde os domínios são as máquinas virtuais criadas pelo Xen. Existem dois tipos de domínios que são o domínio 0 (privilegiada) e o domínio U (não privilegiada).

O domínio 0 é o responsável pelos drives dos dispositivos do servidor físico bem como alguns drives de acesso a rede. Uma máquina virtual domínio U, só será gerenciada apos o domínio 0 ser iniciado, pois será o domínio 0 o responsável pelo envio ao hypervisor das requisições do domínio U. (MACEDO, 2012)

Como o Xen foi desenvolvido com a intenção de ganhos de desempenho utilizando a técnica de paravirtualização, onde deve ser a modificação dos sistemas hóspedes, logo se restringia a sistemas baseados em Unix, acabou afetando a utilização desse software em sistemas como, por exemplo, a Microsoft Windows. (MACEDO, 2012)

Visando uma solução, foi lançado a versão 3 do Xen, que começou a utilizar a virtualização completa, agora podendo ser utilizado em sistemas não modificáveis.

2.8.4 VIRTUALBOX

O VirtualBox é uma ferramenta para a virtualização, foi desenvolvida pela Innotek, uma empresa alemã. Na sequencia foi vendida para a empresa Sun

Microsystem e por fim, foi adquirida, em 2010, pela Oracle Corporation, sendo nomeado como Oracle VM VirtualBox. Este software permite a instalação e utilização de um sistema operacional como hospedeiro de um sistema compartilhando o mesmo o hardware físico. (LOURENÇO, 2012)

Esta ferramenta por ser instalada em diversos sistemas hospedeiros, entre eles, cabe citar o Linux, o Solaris, o Mac OS e Windows. (LOURENÇO, 2012)

O VirtualBox apresenta duas versões, que prevê uma licença proprietária para uso comercial, englobando a utilização pessoal e educacional, na qual utiliza a licença PUEL (Personal Use and Educational License) e a outra é a versão livre, a VirtualBox Open Source Edition - GP. Nesta última versão não é apresentada a opção de suporte remoto dos proprietários e o suporte a dispositivos USB. (LOURENÇO, 2012)

Através deste sistema de virtualização é possível ler os discos rígidos virtualizados que são gerados pelas soluções de virtualização Microsoft Virtual PC e VMware. Toda a sua configuração pode ser tanto por linha de comando como por interface gráfica. (LOURENÇO, 2012)

3 DESCRIÇÃO DA SOLUÇÃO

O ambiente proposto neste trabalho foi elaborado em um formato com maior elaboração, desprezando configurações, conceitos e funcionamentos básicos. Serão retratadas, durante o desenvolvimento deste projeto, ferramentas que foram empregadas.

A elaboração deste trabalho foi desenvolvida em um ambiente virtualizado no Oracle Virtual Box, instalado e configurado em um Sistema Operacional Windows 7, onde foram reproduzidas algumas máquinas virtuais que serão necessárias para a criação de um Cluster. Esses dispositivos virtuais serão iguais em todos os aspectos, divergentes apenas na configuração de comunicação.

A seguinte figura exemplifica a arquitetura desenvolvida.

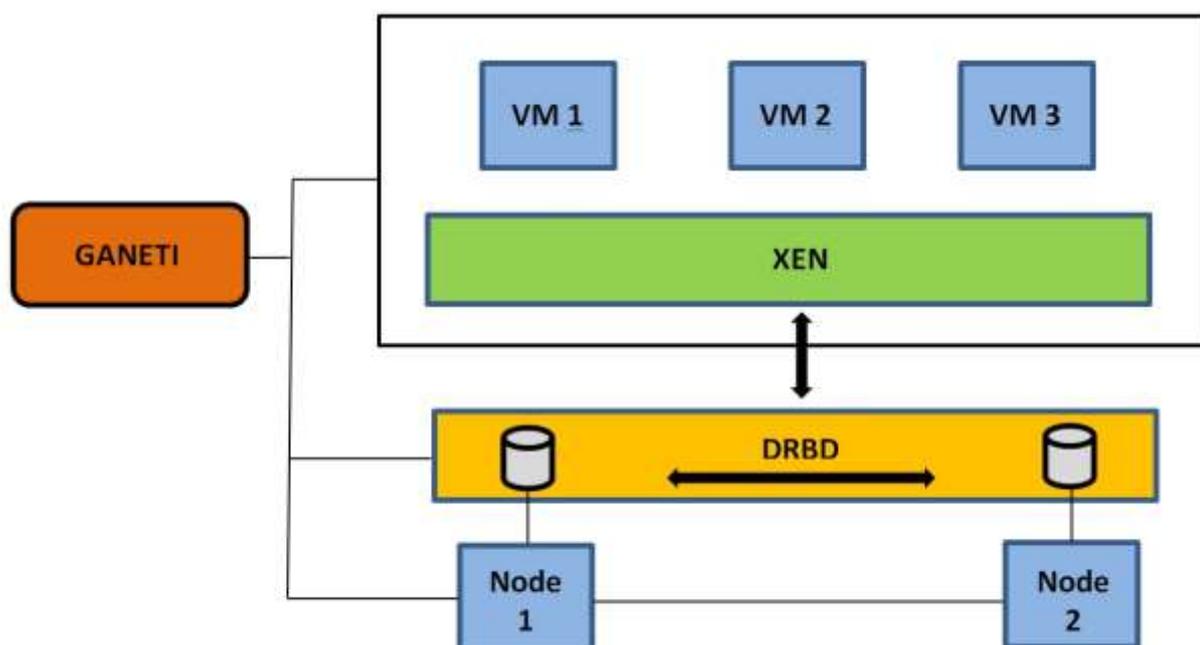


Figura 7 – Estrutura Cluster
 Fonte: Elaborado pelo autor, 2019.

A máquina virtual nomeada de node1 estará composta no Cluster, responderá como função de ponto central de coordenação do referido Cluster.

A máquina virtual nomeada de node2 estará composta no Cluster como membro secundário, terá como função o gerenciamento do ambiente, caso ocorra algum imprevisto no node1 e este deixe de executar suas funções.

Os demais node's dessa arquitetura proposta, terão a simples função de compor o Cluster como unidade autônoma.

Todos os equipamentos virtualizados possuirão as seguintes características:

- 1 processador
- 2084 mb memória
- 20GB espaço HD para SO
- 50GB espaço HD para dados distribuídos pelo DRBD
- placa de rede 100/1000 Mbps

Foi utilizado para o desenvolvimento do ambiente proposto o Sistema Operacional Debian Squeeze. Como forma de manter um sistema mais leve e

simplificado, não foi instalada uma interface gráfica nem qualquer outro tipo de pacote disponível em repositórios.

A implementação do ambiente compreende Sistemas Operacionais distintos do Debian Squeeze, onde provavelmente estes teriam comandos próprios e diferentes. O Debian Squeeze 64bits por sua vez, é uma escolha própria.

Softwares utilizados:

Xen 5.0 - utilizado como hypervisor

Ganeti – ferramenta utilizada para gerenciamento de cluster

DRBD – ferramenta responsável por compartilhar o volume de dados de cada dispositivo pertencente ao Cluster. Também exerce a função de sincronizar os volumes lógicos com os dispositivos possibilitando aos nodes acesso as informações. O DRBD ajuda na segurança do Cluster, visto que através de seu módulo, cria-se um dispositivo de bloco responsável por gravar os dados em mais de um dispositivo físico real.

3.1 INSTALAÇÃO E CONFIGURAÇÃO

Todos os nós que ocupam a infraestrutura deste cluster comunicam-se entre si, sendo necessário apenas a instalação dos pacotes que serão necessários para o correto funcionamento desta estrutura.

Algumas funcionalidades não foram mencionadas, pois ou não se fazem necessários ou fazem parte de alguma uma delimitação deste projeto. A instalação e configuração são feitas utilizando o comando `#aptitude install` anteriormente ao nome do software desejado. Abaixo será listado os pacotes que foram necessários:

- iproute2
- ndisc6
- iproute2
- openssh
- lvm2
- xen-linux-system
- python8
- drbd8-utils
- socat

- ctypes python module
- simplejson python module
- python openssl bindings

3.2 Ferramenta de Gerenciamento GANETI

Após a instalação dos pacotes mencionados, deve ser efetuada a instalação de algum gerenciador de Cluster, nesse caso será efetuada a instalação do software Ganeti. Faz-se necessário o download direto desse software. A seguir serão listados os comandos necessários para a instalação de diretórios importantes para acomodar algumas funções:

```
# wget https://google.ganetia.com/file/ganeti-5.4.5.tar.gz
# tar xvzf ganeti-5.4.5.tar.gz
# cd ganeti-5.4.5
# make
# make install
# mkdir /srv/ganeti
# mkdir /srv/ganeti/os
# mkdir /srv/ganeti/export
```

Com esses comandos corretamente executados, todos os softwares que são necessários estarão instalados e na sequência poderá se inicializado a configuração dos cluster's e suas instancias.

3.3 SOFTWARE DE VIRTUAZAÇÃO XEN

Nesse ponto a instalação do XEN já estará concluída, porém serão necessárias algumas configurações para o correto funcionamento na estrutura proposta.

Uma dessas configurações é a migração das máquinas virtuais entre os nós do cluster, em tempo real. Essa configuração tem como finalidade permitir que as instâncias entre os nós, do cluster, se movimentem e garantam uma estrutura mais adequada, façam cópia dessas instâncias e recuperem o sistema em caso de falha de algum nó do cluster.

Para executar essas configurações deve-se modificar o arquivo "xen-config.sxp", que fica na pasta xen em etc, com os seguintes atributos:

- (xend-realoca-serv yes)
- (xend-realoca-port 8000)
- (xend-realoca-address ' ')
- (vncpasswd ' ')
- (vnc-listen ' '0.0.0.1')

Um ponto importante na configuração do arquivo do xen é a configuração dos parâmetros de hardware que serão assumidos pelo hypervisor no momento de inicialização da infraestrutura, ou seja, deve ser observada a capacidade física dos equipamentos que irão compor o cluster.

Na configuração do ambiente, foi definido o uso de 1 processador e 2048 MB de memória, na qual esse parâmetros foram delimitados no arquivo de configuração “/etc/default/grub”, através da inclusão da seguinte linha:

```
GRUB_CMDLINE_XEN=dom0_mem=1048M maxcpus=1
```

Em seguida a validação dessa alteração deve ser confirmada através do comando:

```
#update-grub2
```

Deve-se ainda definir o kernel padrão para dom0 e domU

```
#ln -s /boot/vmlinuz-2.8.4-8-xenam64 /bootvmlinuz-2.8-xenU
```

```
#ln -s /boot/initrd.img-8-xenam64 /bootvmlinuz-2.8-xenU
```

3.4 DEFINIÇÃO DRBD

O DRBD juntamente com o LVM (gerenciador de volumes), possibilita que todos os volumes dos equipamentos que estão separados sejam espelhados, pois esses equipamentos passam a trabalhar em RAID-1.

Na apresentação do âmbito na qual foi definida uma configuração de hardware específica, aos nós do cluster foi definido uma unidade de disco de 50 GB, para ser utilizado de forma compartilhada como volume no DRBD. A unidade por sua vez,

não deve ter um volume já definido, o responsável pela criação desse volume será o LVM (Gerenciador de Volume Lógico), onde, na sequência o DRBD assumirá todo o sincronismo das unidades.

O Ganeti receberá esses volumes criados e fará todo o gerenciamento desses volumes e do cluster. Os comandos a seguir servem para a criação desses volumes:

```
# pvcreat /dev/das  
# vgcreat xenvg /dev/sda
```

Após esses procedimentos, o já teremos os volumes criados pelo LVM, sincronizados pelo DRBD, logo estará tudo pronto para ser gerenciado pelo Ganeti.

3.5 Topologia da Rede

É fundamental que nos arquivos de host (“/etc/hosts”) de todos os membros do cluster, estejam configurados os IP’s e nomes de todos os nós que compunham a estrutura, também devem ser definidos o sistema operacional que está hospedado nas camadas de virtualização.

Também deve ser definido ao cluster um nome e IP, pois o IP será encarregado da interface de gerenciamento do cluster.

Essa entrada no host se dá através das seguintes configurações:

```
127.0.0.0 localhost  
  
#node1 físico da estrutura do cluster  
  
10.0.0.5 node1.alcides.com node1  
  
#node2 físico da estrutura do cluster  
  
10.0.0.6 node2.alcides.com node2  
  
# as três máquinas virtuais hospedadas:  
  
10.0.0.10 vm1.alcides.com vm1
```

```
10.0.0.11 vm1.alcides.com vm2
```

```
10.0.0.12 vm1.alcides.com vm2
```

```
# endereço para o gerenciamento do cluster:
```

```
10.0.0.20 cluster1.alcides.com cluster1
```

O domínio alcidesmaya.com foi utilizado para ilustrar o ambiente.

Deve ser ativado o modo bridge para a interface de rede trabalhar, pois responderá pelos IP's do cluster, pelo nó físico e responderá também pelos IP's das máquinas virtuais. Essa modificação foi alterada no arquivo /etc/network/interfaces.

As configurações descritas na figura 8 demonstram como deve ficar o arquivo relacionado a configuração de rede do node1:

```
# The loopback network interface
auto lo
iface lo inet loopback

#xen
auto xenbr0
iface xenbr0 inet static
    address 10.0.0.6
    netmask 255.255.255.0
    network 10.0.0.0
    broadcast 10.0.0.255
    gateway 10.0.0.1
    bridge_ports eth0
    bridge_stp off
    bridge_fd 0

# The primary network interface
auto em2
iface em2 inet manual
```

Figura 8 – Node01

Fonte: Elaborado pelo autor, 2019

O node2 apresenta configurações parecidas com a o node1, mantendo atenção apenas para as configurações de rede. Em seguida já é possível a devida comunicação entre os nós, sendo possível, agora, a configuração do DRBD.

4 METODOLOGIA

Inicialmente foi realizada pesquisa bibliográfica referente às informações tratadas até o momento. Foi necessário o estudo e conhecimento da tecnologia de virtualização bem com a tecnologia de clustering.

Este projeto é explicativo, embasada em tecnologias que visam sua aplicação em um ambiente que necessite de uma disponibilidade constante em ambientes corporativos ou residências.

A metodologia de desenvolvimento deste projeto foi elaborada através das seguintes etapas:

- Instalação e configuração do software Ganeti;
- Instalação e configuração do software Xen
- Configuração do DRBD
- Criação de Instâncias
- Simulação de deslocamento de Instâncias
- Simulação de parada de nó primário

5 VALIDAÇÃO

Será apresentada a inicialização do cluster bem como a simulação de cenários possíveis com a utilização da referida tecnologia.

5.1 INICIALIZAÇÃO

Inicialmente devemos inicializar o Cluster para começar a operar a estrutura, onde deve ser informado quais serão os equipamentos que irão compor o cluster bem como qual papel terá cada nó.

O node primário e node secundário devem ser definidos no ambiente para o Cluster saber de ondem partiram cada comando que irá gerenciar a estrutura. Conforme já foi mencionado, está estrutura terá como nó primário e secundário denominado de node1 e node2.

O Ganeti que será responsável pela pelo gerenciamento do cluster tem um conjunto de comandos próprios para gerenciar cada nó e máquinas virtuais que estejam em execução. As máquinas virtuais serão definidas como instancias.

O Ganeti opera através de dois modos de operação, sendo o modo xen-pvm, responsável pelo suporte a paravirtualização do ambiente e o modo xen-hvm, que terá como responsabilidade a virtualização assistida por hardware.

Através do modo de suporte da paravirtualização, todas as instâncias deverão ser baseadas em sistemas operacionais modificáveis, para assim a camada de virtualização ter a possibilidade de modificar o SO para melhor adequação e gerenciamento do ambiente.

Em ambiente que os sistemas operacionais não têm a característica de ser modificado, deve ser habilitado a virtualização assistida por hardware, na qual uma série de instruções definidas no processador ofereceram suporte na execução da instância. Ressaltando que nessa hipótese o hardware deve possuir a tecnologia de suporte à virtualização no hardware.

Na inicialização do Cluster já foi definida o suporte a virtualização, assim como já foram definidos os nós primários e secundários que irão compor o cluster. No entanto será possível incorporar novos nós a estrutura bem como realizar mudanças de suporte da virtualização.

O comando infracitado tem como finalidade a inicialização do Cluster com as características desejadas e com nome de cluster01 para a estrutura.

```
# gnt-cluster init -H xen-pvm:kernel_patch=/boot/vmlinuz-2.8xenU,  
initrd_patch=/boot/initrd-2.8-XenU,blockdev_prefix=xvd,root_patch=/dev/xvda1  
cluster01. Resultado conforme figura 09.
```

```

Cluster name: cluster01.alcides.com.br
Cluster UUID: 39e9b157-5d44-429a-8577-2blac53c4c86
Creation time: 2019-05-28 11:19:48
Master node: node01.alcides.com.br
Architecture (this node): 64bit (x86_64)
Tags: (none)
Default hypervisor: xen-pvm
Enable hypervisor: xen-pvm, xen-hvm
Hypervisor parameters:
  -xen-hvm:
    Acpi: True
    Blockdev_prefix: hd
    Boot_order: cd
    Cdrom_image_patch
    Cpu_mask: all
    Device_node01: /usr/lib/xen/bin/qemu-dm
    Disk_type: paravirtual
    Kernel_patch: /usr/lib/xen/boot/hvmloader
    Migration_mode: non-live
    Migration_port: 8002
Nic_type: rtl8139
Pae: True
Reboot_behavior: reboot

```

Figura 9 – Inicialização Cluster

Fonte: Elaborado pelo autor, 2019.

Após realizada a inicialização do cluster o ambiente já está em condições de receber comandos.

Através comando `#gnt-cluster verify` possível analisar o estatus da estrutura do cluster.

Até o momento o cluster está inicializado, no entanto, apenas com um único nó, onde esse nó assumirá a condição de nó primário, e através dele será feito todo o gerenciamento das instancias de dos outros nós do ambiente.

O passo seguinte é a inclusão do segundo nó ao cluster, terá como nome `node02`. Através do seguinte comando será criado o nó secundário e terá como parâmetro a possibilidade de assumir as funções do nó primário caso seja necessário, essa característica tem como finalidade manter a estrutura operante caso o nó primário venha a falhar.

```
# gnt-node add --master-capable=yes node02
```

Nesse momento o cluster passa o ter dois nodes, onde os seus volumes compartilhados pelo DRBD já estarão sincronizados. Ou seja, deste ponto, qualquer informação que seja gravada por qualquer dos `node01` ou `node02` terão essas

informações gravadas nas unidades compartilhadas entre os dispositivos, permitindo assim a disponibilidade contínua dos dados gravados.

A figura 10 exibe como está o funcionamento e os nós do cluster.

```

root@node01:~# gnt-cluster verify
submitted Jobs 697, 698
waiting for job 697 -
Tue May 6 22:22:26 2019 * Verifying cluster config
Tue May 6 22:22:26 2019 * Verifying cluster certificate files
Tue May 6 22:22:26 2019 * Verifying hypervisor parameters
Tue May 6 22:22:26 2019 * Verifying all nodes belong to an existing group
Waiting for job 698 -
Tue May 6 22:22:26 2019 * Verifying group 'default'
Tue May 6 22:22:26 2019 * Gathering data [2 nodes]
Tue May 6 22:22:29 2019 * Gathering disk information file consistency
Tue May 6 22:22:29 2019 * Verifying node status
Tue May 6 22:22:29 2019 * Verifying instance status
Tue May 6 22:22:29 2019 * Verifying orphan volumes
Tue May 6 22:22:29 2019 * Verifying N+1 Memory redundancy
Tue May 6 22:22:29 2019 * Other Notes
Tue May 6 22:22:29 2019 * Hooks Results
root@node01:~#

```

Figura 10 – Status Cluster

Fonte: Elaborado pelo autor, 2019.

Através do comando infracitado será criado as instâncias do cluster:

```
# gnt-instance add -t drbd --disk 0:size=10g -B memory=256 -o debootstrap+default
-n node01:node02 vm2
```

O parametro “gnt-instance add -t drbd” mostra que a instância será criada no volume DRBD. O parâmetro “—disk 0:size=10g” define o disco com tamanho de 10 Gb. O parâmetro “-o debootstrap+default” define onde estão os arquivos de instalação do SO. Por fim, o parâmetro “-n node1:node2 vm2” indica o local aonde as instâncias serão alocadas.

5.2 AMBIENTAÇÃO

5.2.1 INSTÂNCIAS

Para comprovação e exibição da implementação do ambiente, foi realizada a migração da instância, entre os nós do cluster, conforme o processo descrito a baixo.

Foi desenvolvido 3 (três) instâncias em sistema operacional baseada em Linux para compor o cluster, o quadro infracitado especifica as configurações usadas.

Quadro 1 – Especificação e configuração de instâncias

Nome	IP	Memória	HD
VM1	10.0.0.10	256 Mb	10 Gb
VM2	10.0.0.11	128 Mb	4 Gb
VM3	10.0.0.12	64 Mb	2Gb

Fonte: Elaborado pelo autor, 2019.

As instâncias foram definidas no node01, nó primário, e failover no node 2. O progresso de criação pode ser evidenciado conforme a figura 11.

```
File /etc/hosts not changed so no update needed
root@node01:~# gnt-instance add -t drbd -disk=0:size=10g -B memory=256 -o debootstrap+default -H xen-hvm:cdrom_image_patch=/srv/ganeti/isso/CentOS-6.3-86_64-minimal.iso -n node01:node02 vm3
Failure: prerequisites not met for this operation:
Error type: wrong_input, error details:
OS 'debootstrap_default' not in supports OS list for node node01.alcides.com.br: Directory for OS debootstrap not found in search path
root@node01:~# gnt-instance add -t drbd --disk 0:size=4g -B memory=128 -o debootstrap+default -n node01:node02 vm2
Tue May 7 23:10:03 2019 * creating instance disks...
Tue May 7 23:10:09 2019 adding instance vm2.google.com.br to cluster config
Tue May 7 23:10:10 2019 - INFO: Waiting for instance vm2alcides.com.br to sync disks.
Tue May 7 23:10:11 2019 - INFO: - device disk/0: 1.30% done, 1m 25s remaining (estimated)
Tue May 7 23:10:12 2019 - INFO: - device disk/0: 39.30% done, 1m 18s remaining (estimated)
```

Figura 11 – Criação de Instâncias

Fonte: Elaborado pelo autor, 2019.

Através do comando `#watch cat /proc/drbd`, podemos analisar o status de sincronização, entre os nodes do cluster, do volume de dados do DRBD. Figura 12

```
Every 2.0s: cat /proc/drbd Tue May 7 23:15:12 2019
Version: 11.9.8 (api:88/proto:86-91)
srcversion: EE47D8BD29CA288C36855989
 0: cs:Connected ro: Secondary/Primary ds:UpToDate/UpToDate C r----
   Ns:0 nr:86775 dw:89557 dr:0 al:0 bm:18 lo:0 pe:0 ua:0 ap:0 ep: 1 wo:b oos:0
 1: cs:Connected ro: Primary/Secondary ds:UpToDate/UpToDate C r----
   Ns:4327136 nr:0 dw:89557 dr:4194838 al:55 bm:256 lo:0 pe:0 ua:0 ap:0 ep: 1 wo:b oos:0
```

Figura 12 – Status de Sincronização

Fonte: Elaborado pelo autor, 2019.

O comando "gnt-instance list" mostra como ficou o ambiente com a definição das instâncias. Figura 13.

```

root@node01:~# gnt-instance list
Instance      Hypervisor OS                Primary_node      Status
Memory
vm1.alcides.com.br xen-pvm  debootstrap+default node01.alcides.com.br running
128M
Vm2.alcides.com.br xen-pvm  debootstrap+default node01.alcides.com.br running
128M
Vm3.alcides.com.br xen-pvm  debootstrap+default node01.alcides.com.br ADMIN_down
-
root@node01:~# _

```

Figura 13 – Instâncias

Fonte: Elaborado pelo autor, 2019.

Conforme observado na figura 13, o instância vm3 apresentada como ADMIN_down ocorre pelo fato desta não ter sido iniciada, essa inicialização se da através do comando "gnt-instance startup vm3". Figura 14

```

root@node01:~# gnt-instance startup vm3
Waiting for job 737 for vm3.alcides.com.br ...
root@node01:~# gnt-instance list
Instance      Hypervisor OS                Primary_node      Status
Memory
vm1.alcides.com.br xen-pvm  debootstrap+default node01.alcides.com.br running
128M
Vm2.alcides.com.br xen-pvm  debootstrap+default node01.alcides.com.br running
128M
Vm3.alcides.com.br xen-pvm  debootstrap+default node01.alcides.com.br running
128M
root@node01:~# _

```

Figura 14 – Inicialização

Fonte: Elaborado pelo autor, 2019.

A partir desse momento todas as instâncias então em condições de ser operadas e suportar os sistemas, serviços entre outros.

Para acessar o console da instância desejada deve ser inserido o comando "gnt-instance console vm x " a figura 15 demonstra esse comando em execução, sendo acessado a vm1, pode observar o sistemas de arquivos dessa vm.

```

root@node01:~# gnt-instance console vml
[ 1845.125453] Setting capacity to 10485760
[ 1845.173913] Setting capacity to 10485760

root@vml:~# df
Filesystem            1K-blocks      Used Available  Use% Mounted on
/dev/xvda1            5154884      312860   4580168    7% /
tmpfs                 63916         0       63916     0% /lib/init/rw
udev                 46048         20       46028    1% /dev
tmpfs                 63916         4       63912    1% /dev/shm
root@vml:~# _

```

Figura 15 – Console

Fonte: Elaborado pelo autor, 2019.

5.3 DESLOCAMENTO DE INSTÂNCIAS

Deste ponto, já com as instâncias criadas, onde essas foram alocadas no nó primário do cluster o nó secundário permanecera de forma ociosa. Figura 16.

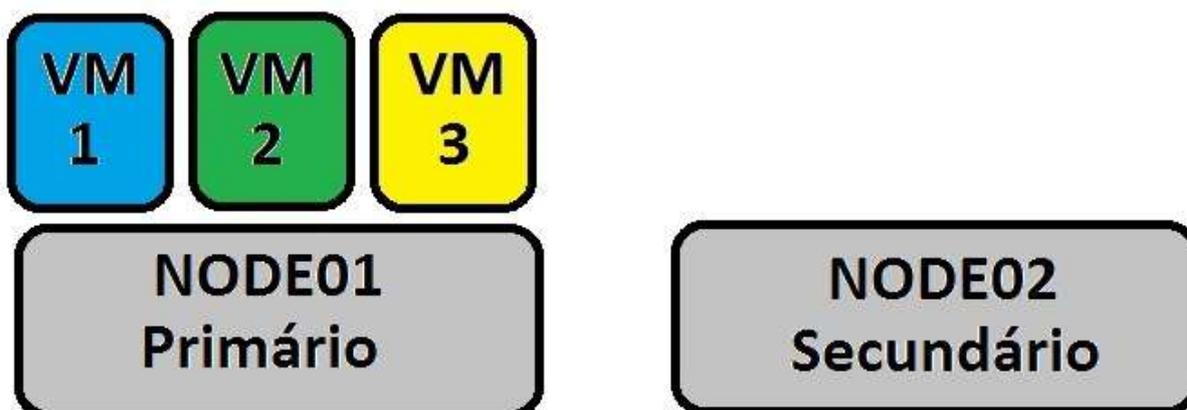


Figura 16 – Node Primário

Fonte: Elaborado pelo autor, 2019.

Em um ambiente em que o foco é o balanceamento e o alto desempenho, devemos configurar um cluster em que as instâncias fiquem de forma equilibradas nos dois nodes. A Figura 17 a seguir demonstra como seria esse ambiente, movimentando umas das instâncias para o node2.

A migração é feita através do comando "gnt-instance migrate", onde será movimentado a vm2 para o nó secundário. Figura 17.

```

root@node01:~# gnt-instance migrate vm2
Instance vm2 will be migrated. Note that migration might impact the
Instance if anything goes wrong (e.g. due to bugs in the hypervisor).
Continue?
y/[n]/?: y
Wed May 8 15:03:12 2019 Migration instance vm2.alcides.com.br
Wed May 8 15:03:12 2019 * checking disk consistency between source and target
Wed May 8 15:03:13 2019 * switching node node02.alcides.com.br to secondary mode
Wed May 8 15:03:13 2019 * changing into standalone mode
[5802.164289] block drbd1: meta connection shut down by peer.
Wed May 8 15:03:13 2019 * changing disks into dual-master mode
Wed May 8 15:03:14 2019 * wait until resync is done
Wed May 8 15:03:15 2019 * preparing node02.alcides.com.br to accept the instance
Wed May 8 15:03:15 2019 * migrating instance to node02.alcides.com.br
Wed May 8 15:03:17 2019 * starting memory transfer
Wed May 8 15:03:17 2019 * memory transfer complete
Wed May 8 15:03:20 2019 * switching node node01.alcides.com.br to secondary mode
Wed May 8 15:03:20 2019 * wait until resync is done
Wed May 8 15:03:21 2019 * changing into standalone mode
[ 5816.259540] block drbd: short read expecting header on sock: r=-512
Wed May 8 15:03:21 2019 * changing disks into single-master mode
Wed May 8 15:03:23 2019 * wait until resync is done
Wed May 8 15:03:25 2019 * done
root@node01:~#

```

Figura 17 – Movimentação de Instâncias

Fonte: Elaborado pelo autor, 2019.

Essa modificação pode ser evidenciada através do comando "gnt-instance list".

```

root@node01:~# gnt-instance list
Instance      Hypervisor OS      Primary_node      Status Memory
vm1.alcides.com.br xen -pvm      debootstrap+default node01.alcides.com.br running
128M
vm1.alcides.com.br xen -pvm      debootstrap+default node02.alcides.com.br running
128M
vm1.alcides.com.br xen -pvm      debootstrap+default node01.alcides.com.br running
128M
root@node01:~#

```

Figura 18 – Listagem

Fonte: Elaborado pelo autor, 2019.

Este procedimento não tem longa duração, pois seu tempo de transferência é relativo ao endereço e memória, onde sua base de dados estará disponibilizada continuamente por meio do DRBD.

Com essa configuração teremos uma disponibilidade constante, pois no momento em que o nó 01 apresentar algum tipo de falha, a instancia alocada no nó 02 permanecerá funcionando.

A figura 19 demonstra o cenário com essa modificação.

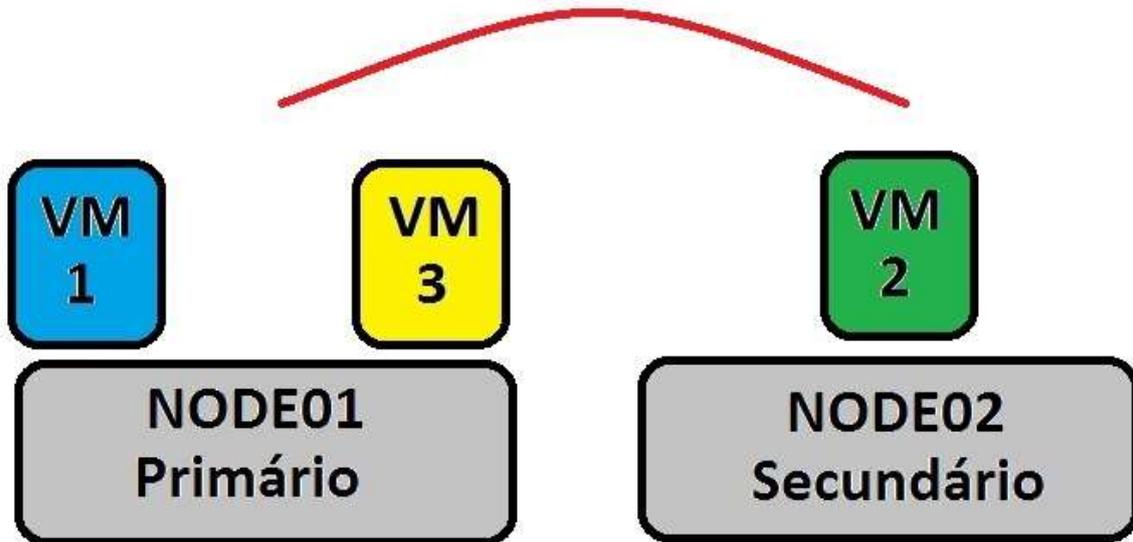


Figura 19 – Migração VM2
Fonte: Elaborado pelo autor, 2019.

5.4 DESLIGAMENTO DO NÓ PRIMÁRIO

Serão demonstrados, neste momento, operações de desligamento de algum nó do cluster. Procedimento realizado caso o ambiente necessite de alguma manutenção proamente e precise deste desligamento.

Será ilustrado o desligamento do node01, simulando uma manutenção, ressaltando que esse node apresenta duas instâncias em execução.

Atráves do comando "gnt-instance failover", as instâncias presentes no node02 serão migradas diretamente, de forma simultânea, para o node01. A figura 20 demonstra esse processo.

```

root@node01:~# gnt-node failover node01
Fail over instance(s) 'vm1.alcides.com.br','vm3.alcides.com.br'?
y/[n]/?: y
Submitted jobs 1044, 1045
Waiting for jog 1044 for vm1.google.com.br ...
Wed May 8 20:12:30 2019 Failover instance vm1.alcides.com.br
Wed May 8 20:12:30 2019 * checking disk consistency between source and target
Wed May 8 20:12:30 2019 * shutting down instance on source node
Wed May 8 20:12:31 2019 * deactivating the instance's disk on source node
[ 8943.392787] block drbd0: short read expecting header in sock: r=-512
Wed May 8 20:12:32 2019 * activating the instance's disks on target node node02
alcides.com.br
Wed May 8 20:12:33 2019 * starting the instance on the target node node02.alcid
es.com.br
Waiting for job 1045 for vm3.alcides.com.br ...
Wed May 8 20:12:33 2019 Failover instance vm3.alcides.com.br
Wed May 8 20:12:33 2019 * checking disk consistency between source and target
Wed May 8 20:12:33 2019 * shutting down instance on source node

```

Figura 20 – Migração Geral
 Fonte: Elaborado pelo autor, 2019.

Esse processo de migração de instâncias leva em torno quatro minutos, para essa ambientação proposta, nesse período as instâncias irão permanecer indisponíveis. Percebe-se que nesse momento todas as instâncias estarão no node02.

No entanto, para podermos desligar o node01 e passar a operação para o node02, deve ser informado que o node primário passará a ser o node02. Esse processo se faz necessário tendo em vista que todos os comandos que gerenciam o cluster parte exclusivamente do node primário.

A figura 21 ilustra a execução desde comando.

```

root@node01:~# gnt-cluster masterfailover
root@node01:~# redis-info

```

Figura 21 – Comando masterfailover
 Fonte: Elaborado pelo autor, 2019.

A partir deste momento já será possível o desligamento do node01, a estrutura passa a ser apresentada conforme a figura 22.

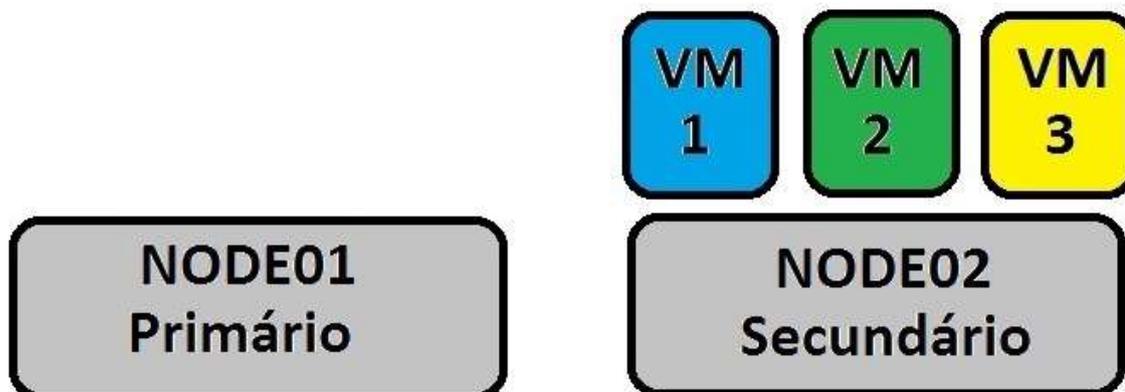


Figura 22 – Estrutura após migração geral
Fonte: Elaborado pelo autor, 2019.

5.5 DESLIGAMENTO NÃO PREVISTO

Uma vez que o nó primário seja desligado, o gerenciamento do cluster é perdido, para contornar essa situação será necessário que o nó primário tenha sua conexão restabelecida ou o nó secundário passe a ser o responsável pelo gerenciamento.

No momento que ocorre essa parada inesperada, será necessário realizar um failover forçado, tendo em vista a necessidade de migrar as instâncias para o nó que será o primário.

Será feito uma simulação desse ambiente.

A figura 23 demonstra um ambiente em que temos o node02 como nó primário, alocando as vm's 1 e 3, e o node01 consequentemente assume o papel de secundário tem alocado a vm 2.

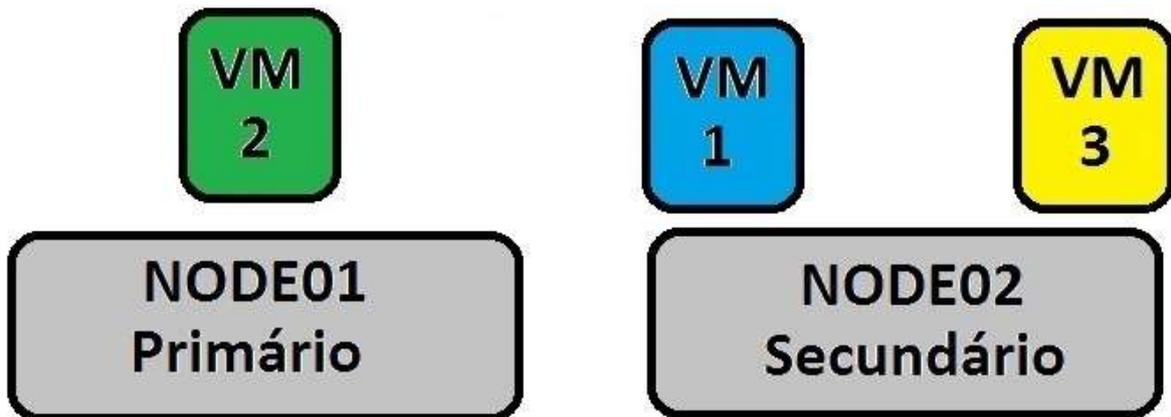


Figura 23 – Falha
 Fonte: Elaborado pelo autor, 2019.

Uma mensagem de erro será apresentada ao node01 no momento em que o node02 sair de operação, informando a indisponibilidade do nó primário, as instâncias desse nó passam a estar indisponíveis também, conforme Figura 24



Figura 24 – Estrutura após falha
 Fonte: Elaborado pelo autor, 2019.

É necessário que o node01 passe a ser o nó primário e se torne o gerenciador do cluster, para com isso ser possível a execução de comandos e recuperação de instâncias, neste caso as vm1 e vm3.

O comando executado conforme demonstra a figura 25, será responsável pelo início da recuperação.

```
root@node01:~# masterfailover --no-voting
```

Figura 25 – Comando de recuperação

Fonte: Elaborado pelo autor, 2019.

Conforme demonstra a figura 25, o parâmetro "--no-voting" força o node01 a ocupar a função de nó primário e passa a ter o total gerenciamento do cluster e suas instâncias. No entanto ainda teremos as instâncias alocadas no node02 como indisponíveis, Figura 26.

```
root@node01:~# gnt-instance list
Instance      Hypervisor OS      Primary_node      Status Memory
vml.alcides.com.br xen -pvm      debootstrap+default node02.alcides.com.br
ERROR_nodedown ?
vml.alcides.com.br xen -pvm      debootstrap+default node01.alcides.com.br running
128M
vml.alcides.com.br xen -pvm      debootstrap+default node02.alcides.com.br
ERROR_nodedown ?
root@node01:~# _
```

Figura 26 – Listagem após falha

Fonte: Elaborado pelo autor, 2019.

Uma vez que todos os volumes de dados do DRBD são replicados entre os nós, o node01 passa a ter acesso e controle das instâncias inoperantes. Com isso é possível executar o comando "gnt-node failover --ignore-consistency node02" e forçar a inicialização das instâncias que estavam alocadas no node02.

A figura 27 demonstra como ficou o status das instâncias.

```
root@node01:~# gnt-instance list
Instance      Hypervisor OS      Primary_node      Status Memory
vml.alcides.com.br xen -pvm      debootstrap+default node01.alcides.com.br running
128M
vml.alcides.com.br xen -pvm      debootstrap+default node01.alcides.com.br running
128M
vml.alcides.com.br xen -pvm      debootstrap+default node01.alcides.com.br running
128M
root@node01:~# _
```

Figura 27 – Listagem após failover

Fonte: Elaborado pelo autor, 2019.

A figura 28 demonstra como fica a estrutura do cluster, onde o node02 ainda será o primário, conforme sua configuração no cluster no momento em que foi desligado.

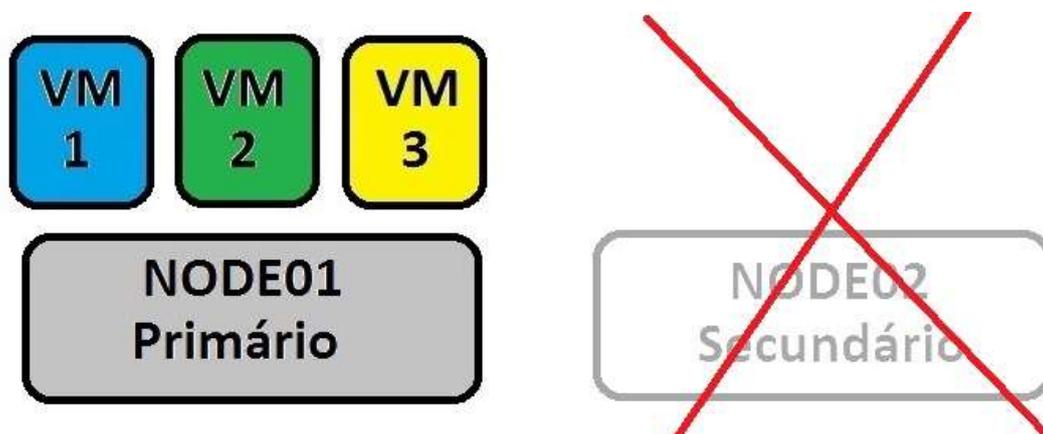


Figura 28 – Ambiente após failover
Fonte: Elaborado pelo autor, 2019.

Desta forma, será necessário o acesso ao console do node01 e definir o node02 como secundário, através do comando "gnt-cluster masterfailover" além de informar aos nós da estrutura como a configuração se encontra, utilizando o comando "gnt-cluster redist-conf".

Com o comando "gnt-cluster getmaster" será possível verificar quem é o nó primário.

Foi demonstrada a configuração dos softwares que se faz necessário para o funcionamento correto de uma estrutura de cluster, demonstrando simulações e recursos disponíveis.

6 CONCLUSÃO

A utilização da tecnologia cluster juntamente com a tecnologia de virtualização possibilita que os serviços de um ambiente de rede sejam entregues de forma garantida, mantendo sempre a disponibilidade dos recursos em um ambiente de TI.

Um dos pontos positivos evidenciado no uso conjunto desta tecnologia é o fator custo, pois a utilização de softwares livres bem como a flexibilidade na adição de hardwares são determinantes para uma implementação de infraestrutura tecnológica.

O Distributed Replicated Block Device (DRBD) é uma excelente ferramenta para armazenamento de dados, além de ser outra vez evidenciado o custo baixo. Seu funcionamento atua de forma similar a uma unidade Storage, no entanto esta, por sua vez tem um custo mais alto para manter seu funcionamento bem como sua manutenção.

Todo o processo de gerenciamento de uma estrutura cluster é relativamente simples e dinâmica, que são características importantes para disponibilização constantes de recursos.

Um ponto importante a ser ressaltado é o tempo de parada e a manipulação das instâncias no ambiente cluster. Foi constatado um tempo aproximado de 38 segundos na efetivação do procedimento com uma perda de aproximadamente 4 pacotes, analisado pelo comando ping. Com isso, fica evidenciado que a instância ficou muito pouco tempo com status de indisponível.

Outro fator relevante evidenciado foi à capacidade, da estrutura, de processamento, de forma mais proveitosa, pois determinada instância pode ser transferida de nó para nó, no cluster, levando em consideração sua capacidade. Sendo assim os recursos de hardware serão melhores aproveitados, maximizando seu desempenho.

Considera-se alcançado o propósito deste projeto, pois é perceptível que a utilização desta tecnologia em ambientes e estruturas tecnológicas possibilita a garantia dos recursos bem como a eficiência.

O Ganeti foi um facilitador, pois se trata de um software intuitivo e de grande significativa para o gerenciamento do ambiente cluster proposto.

Alguns recursos e funções não foram testados ou implementados, em função das limitações da estrutura a qual foi produzido este projeto, a exemplo de possíveis falhas em ambientes externo a da estrutura bem como o failover automático, que se trata inicialização automática das instâncias.

O ambiente ideal para o cenário seria o uso de equipamentos físicos não se limitando a máquinas virtuais criadas em um hardware consideravelmente fraco.

Para solucionar a questão de comunicação com ambiente externo o ideal seria a utilização de um roteador para mapeamento acesso e mapeamento das rotas.

A questão da utilização do failover automático das instâncias na estrutura seria facilmente configurada em um ambiente mais robusto, podendo ser aplicado.

Outra função que é consideravelmente importante é o módulo Remus, que é responsável por garantir tolerância a falhas, seu funcionamento e instalação objetiva reduzir o tempo de parada caso alguma instância apresente alguma falha e se torne inoperante. Essa função exige a comunicação entre os nós da estrutura cluster de forma elevada, tendo em vista que existe uma sincronização em tempo real.

A utilização do Virtual Box acarretou em impasse em relação à comunicação e capacidade de processamento ao ambiente externo.

É viável uma comparação de uma virtualização em um servidor físico com uma virtualização de cluster para determinar a viabilidade da implementação da tecnologia bem como sua adoção em ambientes corporativos ou tecnológicos.

7 CRONOGRAMA

Atividades	Meses					
	mar.	abr.	mai.	jun.	jul.	jul
Escolha do assunto do projeto	X					
Elaboração da estrutura do projeto	X					
Seleção e leitura das obras para elaboração do projeto		X	X			
Elaboração dos objetivos, delimitação do tema, definição do problema, etc.		X	X			
Elaboração da pesquisa bibliográfica e documental do projeto			X	X	X	
Coleta de dados		X	X	X		
Tratamento dos dados				X	X	
Revisão final do texto e elaboração da introdução e conclusão					X	

8 REFERÊNCIAS BIBLIOGRÁFICA

ALECRIM, Emerson. O que é virtualização e para que serve?. Atualizado em 21 de fevereiro de 2013. Disponível em: <<https://www.infowester.com/virtualizacao.php#xen>>. Acesso em: 10 abr 2019.

APOLINÁRIO, Vinícius R.. Learning Hyper-V. Mumbai: Packet Publishing, 2015.

BALTAZAN, Paige; PHILLIPS, Amy. Sistemas de Informação. Porto Alegre: AMGH Editora Ltda, 2012.

BORINI, Guilherme. Gastos com TI representam 7,7% da receita das empresas brasileiras, diz FGV. 2018. Disponível em: <<https://computerworld.com.br/2018/04/19/gastos-com-ti-representam-77-da-receita-das-empresas-brasileiras-diz-fgv>> Acesso em: 21 jun 2018.

BURGELMAN, R. A.; CHRISTENSEN, C. M.; WHEELWRIGHT, S. C. Gestão estratégica da tecnologia e da inovação: conceitos e soluções. 5 ed. Trad. Luiz Claudio de Queiroz Faria, Rev. André Ribeiro de Oliveira. Porto Alegre: AMGH, 2012.

CARISSIMI, A. Virtualização: da teoria a soluções. Minicurso do 26º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2008 Rio de Janeiro. Disponível em: <<http://www.gta.ufrj.br/ensino/CPE758/artogps-basicos/cap4-v2.pdf>> Acesso em: 20 jun 2018.

COULOURIS, George, DOLLIMORE, Jean, KINDBERG, Tim. Sistemas Distribuídos - Conceitos de Projeto. 4. ed. Porto Alegre: Bookman, 2007.

DIÓGENES, Yuri, MAUSER, Daniel. Certificação Security+. Rio de Janeiro: Novaterra Editora, 2016.

FAQ Informática. 2018. Disponível em: <<https://faqinformatica.com/o-que-e-hypervisor-tipos-de-hypervisor/>>. Acesso em: 25 maio 2019.

HA-CC. Load balancing Cluster. 2018. Disponível em: <http://www.ha-cc.org/high_availability/components/application_availability/cluster/load_balancing_cluster>. Acesso em: 04 mar 2019.

IVA, Steve. What Is Hyper-V & How Do You Use It? A Beginner's Guide. Disponível em: <<https://www.cloudwards.net/hyper-v/>>. Acesso em: 25 mai 2019.

LAUDON, K. C.; LAUDON, J. P..Sistemas de Informação Gerencial. 7ª edição. São Paulo:Pearson Prentice Hall, 2007.

Linux Virtual Server. What is virtual server?. Disponível em: <<http://www.linuxvirtualserver.org/whatis.html>>. Acesso em: 09 abr 2019.

MACEDO, Diego. Virtualização - VMWare e Xen. 2012. Disponível em: <<https://www.diegomacedo.com.br/virtualizacao-vmware-e-xen/>>. Acesso em: 10 jun 2019.

LOURENÇO, Luciano. Guia do Virtual Box - Entendendo o Virtualbox. 2012. Disponível em: <<https://www.hardware.com.br/guias/guia-virtualbox/entendendo-virtualbox.html>>. Acesso em: 19 de jul 2018.

MORIMOTO, Carlos E. Cluster. 2005. Disponível em: <<https://www.hardware.com.br/termos/cluster>> Acesso em: 05 jul 2018.

OLIVEIRA, Waldes. O que é virtualização?. Atualizado em 06 julho 2018. Disponível em: <<https://www.techtem.com.br/o-que-e-virtualizacao/>>. Acesso em: 20 mai 2019.

PITANGA, Marcos. Computação em cluster. 2003. Disponível em: <<https://www.clubedohardware.com.br/artigos/redes/computa%C3%A7%C3%A3o-em-cluster-r33711/>> Acesso em: 26 jun 2018.

PITANGA, Marcos. Construindo supercomputadores com Linux, Rio de Janeiro: Brasport, 2002.

Portal GSTI. 2019. Disponível em: <<https://www.portalgsti.com.br/hyper-v/sobre/>>. Acesso em: 25 mai 2019.

REDHAT. O que é virtualização e como ela funciona?. 2018. Disponível em: <<https://www.redhat.com/pt-br/topics/virtualization/what-is-virtualization>>. Acesso em: 15 abr 2019.

SALLES D. et al. Cluster HPC - High Performance Computing. Trabalho da disciplina de Sistemas Distribuídos. Faculdade de Tecnologia de Guaratinguetá, 2009. Disponível em: <http://www4.fct.unesp.br/ronaldo/uploads>. Acesso em: 05 abr 2019.

SAMBA. Heartbeat HA Configuration. Modificado em 12 de março de 2007. Disponível em: <http://wiki.samba.org/index.php/5.0:_Heartbeat_HA_Configuration> Acesso em: 08 abr 2019.

SLTI, Secretaria de Logística e Tecnologia da Informação. Guia de Estruturação e Administração do Ambiente de Cluster e Grid. Brasília, 2006.

SLTI, Secretaria de Logística e Tecnologia da Informação. Guia de Estruturação e Administração do Ambiente de Cluster. Brasília, 2008.

TANENBAUM, Andrew S., STEEN, Maarten Van. Sistemas Distribuídos - Conceitos e Paradigmas. 2. ed. São Paulo: Person, 2007.

VERAS, Manoel. Virtualização - Componente Central do Datacenter. 1. ed.1 Rio de Janeiro: Brasport, 2011.

VERAS, Manoel. Virtualização - Tecnologia Central do Datacenter. Rio de Janeiro: Brasport, 2015.