



**UNIVERSIDADE FEDERAL DO ACRE**  
**CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS**  
**CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**COMPARATIVO DE TÉCNICAS DE VIRTUALIZAÇÃO UTILIZANDO  
HYPERVISOR TIPO I E HYPERVISOR TIPO II, EM AMBIENTE WINDOWS E  
LINUX**

**RIO BRANCO**  
**2018**

**ICARO GABRIEL DE LIMA FERREIRA**

**COMPARATIVO DE TÉCNICAS DE VIRTUALIZAÇÃO UTILIZANDO  
HYPERVISOR TIPO I E HYPERVISOR TIPO II, EM AMBIENTE WINDOWS E  
LINUX**

Monografia apresentada como exigência final  
para obtenção do grau de bacharel em  
Sistemas de Informação da Universidade  
Federal do Acre.

Prof. Orientador: Wilker Luís Gadelha Maia

**RIO BRANCO**

**2018**

## **TERMO DE APROVAÇÃO**

**ICARO GABRIEL DE LIMA FERREIRA**

### **COMPARATIVOS DE TÉCNICAS DE VIRTUALIZAÇÃO UTILIZANDO HYPERVISOR TIPO I E HYPERVISOR TIPO II, EM AMBIENTE WINDOWS E LINUX**

Esta monografia foi apresentada como trabalho de conclusão de Curso de Bacharelado em Sistemas de Informação da Universidade Federal do Acre, sendo aprovado pela banca constituída pelo professor orientador e membros abaixo mencionados.

Compuseram a banca:

---

Prof. Orientador Wilker Luís Gadelha Maia, Mestre  
Curso de Bacharelado em Sistemas de Informação

---

Prof. André Luiz Nasserla Pires, Doutor  
Curso de Bacharelado em Sistemas de Informação

---

Prof. Olacir Rodrigues Castro Junior, Doutor  
Curso de Bacharelado em Sistemas de Informação

Rio Branco, 17 de Março de 2018.

*Dedico esse trabalho aos meus pais, que  
apesar de todas as dificuldades que enfrentei  
ao longo da trajetória acadêmica, sempre me  
apoiaram de todas as formas e sempre  
acreditaram no meu melhor.*

## **AGRADECIMENTOS**

Agradeço aos meus pais que me deram o apoio necessário nos momentos em que precisei durante a graduação, aos colegas que compartilharam comigo as dificuldades da formação acadêmica, em especial os meus amigos Gilson Júnior, João Neto, Pedro Henrique e Vitor Lucas, que estiveram compartilhando crescimento pessoal e acadêmico que adquirimos durante a formação. Também agradeço aos professores que me proveram o conhecimento necessário para a minha boa formação profissional, me ensinando não somente o necessário para exercer a atividade com excelência, mas de forma ética, com valores morais e sociais adequados.

*“O único lugar aonde o sucesso vem antes do  
trabalho é no dicionário.”*

*Albert Einstein*

## **RESUMO**

O paradigma da virtualização foi um avanço crucial para a computação moderna e implementação de ambientes corporativos que necessitam de poder computacional integrado, fazendo com que diferentes técnicas e softwares fossem desenvolvidos para melhor utilizar esse paradigma. A virtualização de servidores possui dois tipos principais que se dividem em virtualização total, também conhecida como nativa, e a Paravirtualização. Foram testadas ambas as técnicas de virtualização utilizando softwares denominados hipervisores, com o intuito de verificar como cada cenário composto pela combinação de virtualização e hipervisor, desempenha o seu papel de controlador de máquina virtual. Os testes utilizaram o software J meter para emular cargas de trabalho e em cada cenário foi criada uma máquina virtual Linux e uma Windows. Através de simulações dos serviços de arquivos de diretórios, banco de dados, transferência de arquivo, transferência de hipertexto e da monitoração utilizando a solução Zabbix, foram obtidos dados relevantes para estabelecer um comparativo entre os cenários e ambientes, observando o comportamento do hardware durante cada teste de carga.

Palavras-chave: Virtualização, Hipervisor, Máquina Virtual.

## **ABSTRACT**

The virtualization paradigm was a crucial breakthrough for modern computing and implementation of enterprise environments that require integrated computing power, making different techniques and software developed for a better use of this paradigm. Server virtualization has two main types that are divided into full virtualization, also known as native, and Paravirtualization. Both virtualization techniques were tested using a software called hypervisors, in order to verify how each scenario composed by the combination of virtualization and hypervisor plays its role as virtual machine controller. The tests used the J meter software to emulate workloads and in each scenario a Linux and Windows virtual machine were created. Through file system, database, file transfer, hypertext transfer and monitoring simulations using the Zabbix solution, relevant data were obtained to establish a comparison between the scenarios and environments, observing the behavior of the hardware during each load test.

**Keywords:** Virtualization, Hypervisor, Virtual Machine.



## LISTA DE FIGURAS

Figura 1 Exemplos de virtualização .....	20
Figura 2 Hierarquia estrutural da virtualização Total.....	22
Figura 3 Hierarquia estrutural da Paravirtualização .....	23
Figura 4 Hipervisor tipo I .....	24
Figura 5 Hipervisor tipo II.....	25
Figura 6 Ilustração de cenário de teste de hipervisor tipo I.....	37
Figura 7 Ilustração de cenário de teste de hipervisor tipo II .....	38
Figura 8 Erro LDAP Linux CentOS .....	47

## **LISTAS DE QUADROS**

Quadro 1 Hipervisores utilizados. ....	39
Quadro 2 Sistemas operacionais utilizados .....	40

## LISTAS DE TABELAS

Tabela 1 Parâmetros de teste FTP do J meter.....	30
Tabela 2 Parâmetros de teste HTTP do J meter.....	32
Tabela 3 Parâmetros de teste JDBC do J Meter. ....	33
Tabela 4 Parâmetros de teste LDAP do J meter. ....	34
Tabela 5 Resultado de teste FTP Windows Server .....	42
Tabela 6 Resultado de teste HTTP Windows Server .....	43
Tabela 7 Resultados de teste JDBC Windows Server .....	44
Tabela 8 resultado de teste LDAP Windows Server .....	44
Tabela 9 Resultado de teste FTP Linux CentOS .....	45
Tabela 10 Resultado de testes HTTP Linux CentOS .....	45
Tabela 11 Resultado de Teste JDBC Linux CentOS .....	46
Tabela 12 Resultado de teste LDAP Linux CentOS .....	46
Tabela 13 Fluxo de dados FTP Windows Server .....	48
Tabela 14 Fluxo de dados HTTP Windows Server .....	49
Tabela 15 Fluxo de dados LDAP Windows Server.....	50
Tabela 16 Fluxo de dados JDBC Windows Server .....	50
Tabela 17 Fluxo de dados FTP CentOS .....	51

Tabela 18 Fluxo de dados HTTP CentOS .....	51
Tabela 19 Fluxo de dados LDAP CentOS .....	52
Tabela 20 Fluxo de dados JDBC CentOS .....	52
Tabela 21 Percentual de superioridade dos hipervisores em cada item. ....	54

## SUMÁRIO

<b>LISTA DE FIGURAS.....</b>	<b>8</b>
<b>LISTAS DE QUADROS .....</b>	<b>9</b>
<b>LISTAS DE TABELAS.....</b>	<b>10</b>
<b>1 INTRODUÇÃO .....</b>	<b>14</b>
<b>1.1 JUSTIFICATIVA DA PESQUISA .....</b>	<b>15</b>
<b>1.2 OBJETIVOS DA PESQUISA .....</b>	<b>16</b>
1.2.1 Geral .....	16
1.2.2 Específicos.....	16
<b>1.3 METODOLOGIA.....</b>	<b>17</b>
<b>1.4 ORGANIZAÇÃO DO ESTUDO.....</b>	<b>18</b>
<b>2 VIRTUALIZAÇÃO.....</b>	<b>19</b>
<b>2.1 VIRTUALIZAÇÃO TOTAL.....</b>	<b>21</b>
<b>2.2 PARAVIRTUALIZAÇÃO.....</b>	<b>22</b>
<b>2.3 HIPERVISOR.....</b>	<b>23</b>
2.3.1 Hipervisor tipo I.....	24
2.3.2 Hipervisor tipo II .....	25
<b>2.4 MONITORAMENTO DE REDE .....</b>	<b>26</b>
<b>2.5 MÉTRICAS DE AVALIAÇÃO .....</b>	<b>26</b>
<b>2.6 FERRAMENTAS DE AVALIAÇÃO .....</b>	<b>27</b>
2.6.1 Zabbix .....	27
2.6.2. Apache j meter.....	28

2.6.2.1 Ftp .....	30
2.6.2.2 Http .....	31
2.6.2.3 Jdbc .....	33
2.6.2.4 Ldap .....	34
<b>2.7 TRABALHOS RELACIONADOS .....</b>	<b>34</b>
<b>3 ESTUDO DE CASO .....</b>	<b>36</b>
<b>3.1 INTRODUÇÃO .....</b>	<b>36</b>
<b>3.2 CENÁRIO DE TESTE.....</b>	<b>37</b>
<b>3.3 RESULTADOS OBTIDOS.....</b>	<b>40</b>
3.3.1 Tabelas comparativas de uso de memória e processador .....	41
3.3.2 Tabelas comparativas de dispositivos de entrada e saída .....	48
3.3.3 Desempenho geral dos ambientes.....	53
<b>4 CONSIDERAÇÕES FINAIS E RECOMENDAÇÕES .....</b>	<b>56</b>
<b>4.1 CONSIDERAÇÕES FINAIS.....</b>	<b>56</b>
<b>4.2 RECOMENDAÇÕES.....</b>	<b>57</b>
<b>REFERÊNCIAS .....</b>	<b>58</b>

## 1 INTRODUÇÃO

As discussões sobre virtualização tiveram início em meados de 1960, tendo a IBM como protagonista dessas discussões no mundo da computação. A necessidade de compartilhar recursos em grandes mainframes fez com que a virtualização de plataforma surgisse, sendo ela a primeira técnica de virtualização, dando um ponto de partida a esse paradigma computacional. (JONES, 2011)

Com os avanços nas arquiteturas de computadores, novas tecnologias de virtualização foram sendo criadas para atender as crescentes demandas por técnicas desse tipo, avanços que são cada vez mais exigidos dentro do contexto corporativista, sendo este o principal utilizador.

A virtualização pode ser definida como uma camada de abstração de recursos para que dispositivos se alinhem homogeneamente e ofereçam transparência para o usuário. Essa é a base para oferecer harmonia entre os componentes dentro de um sistema que possui diferentes hardwares, sistemas operacionais e aplicações. (VERAS, 2011).

Os serviços de redes oferecidos pelas aplicações estão ligados diretamente aos sistemas operacionais, estes por sua vez devem ser compatíveis com os recursos da máquina em questão, é nesse ponto que entra a virtualização, fazendo com que os diversos serviços providos por vários sistemas operacionais funcionem harmonicamente, com isso há o surgimento de ferramentas denominadas hipervisores ou monitores de máquinas virtuais. A definição de hipervisores é derivada de supervisor, ator que supervisiona determinada função, nesse caso hipervisores supervisionam a virtualização.

Duas técnicas de virtualização de servidores podem ser destacadas, sendo elas a virtualização com hipervisor tipo I e hipervisor tipo II. A primeira consiste na aplicação de uma camada entre o hardware e diferentes sistemas operacionais que hospedam os serviços, e a segunda baseia-se na utilização de um sistema operacional hospedeiro, no qual a camada de

virtualização é aplicada, assim é possível criar máquinas virtuais com sistemas operacionais diferentes, denominados convidados, provedores dos aplicativos de serviços.

## **1.1 JUSTIFICATIVA DA PESQUISA**

Segundo Pollon (2008), há divergência de opiniões entre o uso de virtualização. A maioria das correntes de estudiosos de tecnologia da informação defendem o uso da virtualização e os benefícios que a mesma proporciona, entretanto, uma pequena parcela acredita que os problemas de TI comuns se somam com os da virtualização.

Os serviços de rede oferecidos por servidores possuem as mais variadas características e exigência de diferentes sistemas operacionais, cuja harmonia é proporcionada pela virtualização. Sistemas operacionais Windows Server e algumas distribuições do Linux, tal como CentOS, oferecem uma gama de opções para os servidores cumprirem suas tarefas, e graças a virtualização é possível que esses ambientes funcionem de maneira heterogênea, entretanto, que tipo de virtualização aplicada nessas duas plataformas de servidores oferece a melhor relação custo-benefício?

A comparação dos cenários de testes em ambiente controlado tem, por finalidade, o intuito de verificar o comportamento do servidor de maneira singular em cada ambiente proporcionado pelas tecnologias de virtualização, utilizando as técnicas de hipervisor I e II em diferentes sistemas operacionais, podendo estes dados obtidos dos comparativos realizados, serem utilizados por pesquisadores e usuários para decidir qual técnica melhor se adequa às suas necessidades.



## 1.2 OBJETIVOS DA PESQUISA

### 1.2.1 Geral

Aplicar e avaliar as técnicas de Virtualização total e Paravirtualização as quais são aplicadas pelos hipervisores tipo I e II, objetivando obter dados através de serviços de redes simulados, comparar o desempenho de cada um dos testes dentro das suas respectivas virtualizações e hipervisores utilizados e definir qual melhor combinação de técnica de virtualização e tipo de hipervisor melhor executou a tarefa proposta pelo teste.

### 1.2.2 Específicos

- a) Preparar um ambiente de teste com os hipervisores tipo I Vmware ESXi e Citrix XenServer;
- b) Preparar um ambiente de teste utilizando o Windows Server com o recurso Hyper V Hipervisor.
- c) Preparar um ambiente de teste utilizando o CentOS como *host* e o KVM como hipervisor tipo II;
- d) Criar maquinas virtuais em cada um dos ambientes com os sistemas operacionais Windows Server 2012 R2 e Linux distribuição CentOS 7.1, provendo serviços de rede.
- e) Utilizar a ferramenta J meter como simulador de usuários virtuais e emular cargas de trabalho utilizando os serviços de banco de dados, transferência de arquivo, serviços de diretório e Transferência de Hypertexto

- f) Instalar e configurar o software Zabbix para obter dados do servidor, tal como uso de CPU, uso da rede, Uso de disco e Memória RAM, e demonstrar através de tabelas o desempenho desses componentes de hardware durante os testes de carga;
- g) Avaliar os dados para definir um perfil de desempenho em cada ambiente de teste executado.

### 1.3 METODOLOGIA

A elaboração de um estudo comparativo se enquadra na definição de pesquisa quantitativa, por tratar de dados quantificáveis e comparáveis numericamente, de acordo com as abordagens descritas por Da Silva e Menezes (2009). Os dados serão obtidos através de simulações de serviços emulados por softwares, que serão disparadas para observar o desempenho em cada ambiente preparado e assim poder comparar os dados de desempenho em cada cenário de teste dos serviços. Esses serviços, que são providos por aplicações, a exemplo do hipervisor tipo I, são hierarquicamente dispostos nas máquinas virtuais, que por sua vez possuem um sistema operacional denominado convidado, que se comunica com o hardware do servidor utilizando a camada de virtualização. Essas máquinas virtuais são gerenciadas pelo software denominado monitor de máquina virtuais.

No hipervisor tipo II temos uma estrutura parecida, entretanto, este possui uma camada a mais entre o monitor de máquinas virtuais e o hardware, composta por um sistema operacional denominado hospedeiro. O núcleo do sistema é utilizado como ambiente de virtualização e integração do hardware com as máquinas virtuais.

Os dados que serão coletados através do software monitor são: uso de CPU, uso Memória RAM, utilização dos recursos de rede, utilização do disco rígido, tempo para completar a tarefa de execução da carga de trabalho e taxa de erros. Após a obtenção desses dados, eles serão tabulados e comparados para demonstrar qual técnica de virtualização é mais eficiente na execução de cada serviço.

## **1.4 ORGANIZAÇÃO DO ESTUDO**

Os Capítulos abaixo estão dispostos da seguinte forma: o Capítulo 2 Contém os conceitos e fundamentações para o entendimento dos alicerces que direcionam esse trabalho. O Capítulo 3 descreve detalhadamente o estudo de caso do trabalho realizado, contendo os dados obtidos, informações pertinentes das ferramentas, recursos e técnica utilizadas, gráficos e tabelas comparativas. O Capítulo 4 para considerações finais e recomendações de trabalhos futuros, onde irá conter as conclusões e observações durante a execução e finalização do trabalho, e recomendações para trabalhos futuros sugerindo possíveis continuações, e por fim, estão as referências que serviram de fonte de conhecimento para a realização desse trabalho.

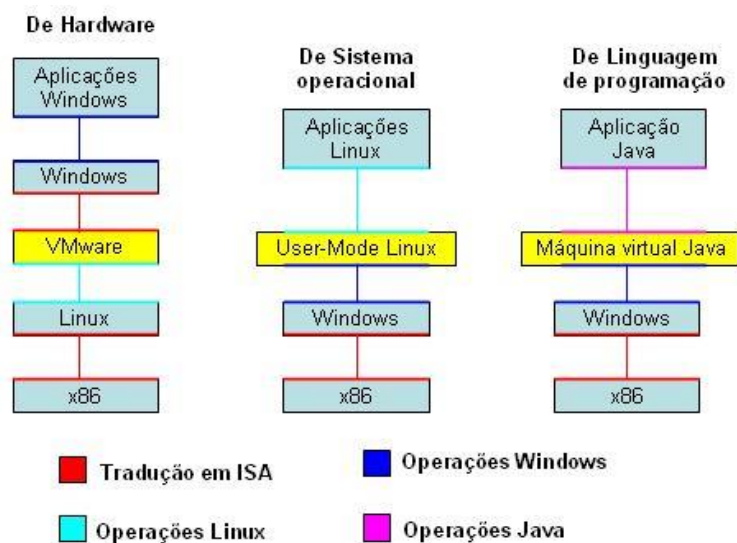
## 2 VIRTUALIZAÇÃO

A virtualização é muito utilizada por sistemas operacionais modernos, em estruturas que oferecem diversos serviços de rede, através do qual é possível prover transparência e compatibilidade em diferentes ambientes. As corporações se utilizam dessa técnica para criar heterogeneidade para diversos serviços, isto é, capacidade de usar diferentes sistemas operacionais para diferentes plataformas de execução de aplicativos, utilizando o mesmo hardware sem a necessidade de se obter várias máquinas para isso, apenas transformando uma máquina em várias outras com recursos dinamicamente dispostos.

A virtualização é uma camada de abstração de recursos capaz de alinhar homogeneamente os dispositivos, oferecendo transparência para o usuário (VERAS, 2011). Também é responsável por proporcionar transparência na soma dos recursos e gerenciá-los de forma com que funcionem harmonicamente. Desde compilações de códigos em máquinas virtuais de diferentes linguagens de programação, até abstração de hardware necessária para o bom funcionamento de sistemas operacionais, essa camada está presente, ainda que não percebamos sua atuação.

Segundo Laureano (2006), existem duas abordagens para sistemas de máquinas virtuais, o tipo I, quando o monitor de máquinas virtuais é implementado entre o hardware e os sistemas convidados (*guest system*), e o tipo II, que age como um processo de um sistema operacional real subjacente, denominado sistema anfitrião (*host system*). A Figura 1 Representa 3 técnicas diferentes de virtualização, usadas para diferentes propósitos.

**Figura 1 Exemplos de virtualização**



Fonte: (COELHO; CALZAVARA; LUCIA, 2008)

Os tipos de virtualizações demonstrados pela Figura 1 são utilizados comumente para serviços locais essenciais para simulação de hardware, software ou sistema operacional. Algumas linguagens de programação, como Java, mostrada na Figura 1, necessitam de uma máquina virtual própria para executar seu conjunto de componentes, ferramentas e bibliotecas dentro de um sistema operacional, usar um caixa de virtualização para executar um sistema operacional dentro de outro, abstrair hardwares, traduzir em instruções ISA (*Instruction set Architecture*) que consiste em traduzir a instrução de aplicativo para instrução de sistema operacional nativo, e criar dispositivos virtuais para diferentes propósitos. (COELHO; CALZAVARA; LUCIA, 2008).

A virtualização abordada neste trabalho se baseia na utilização de um software chamando hipervisor, cuja função é fazer com que os dispositivos sejam reconhecidos corretamente por sistemas operacionais distintos, tornando possível gerenciar os recursos de hardware de acordo com a necessidade de cada máquina virtual. Esses conceitos, aliados a ideia de heterogeneidade de infraestrutura, proporcionam o ambiente ideal para diversificar os serviços de rede oferecidos, sem a necessidade de investimentos e várias máquinas.

## 2.1 VIRTUALIZAÇÃO TOTAL

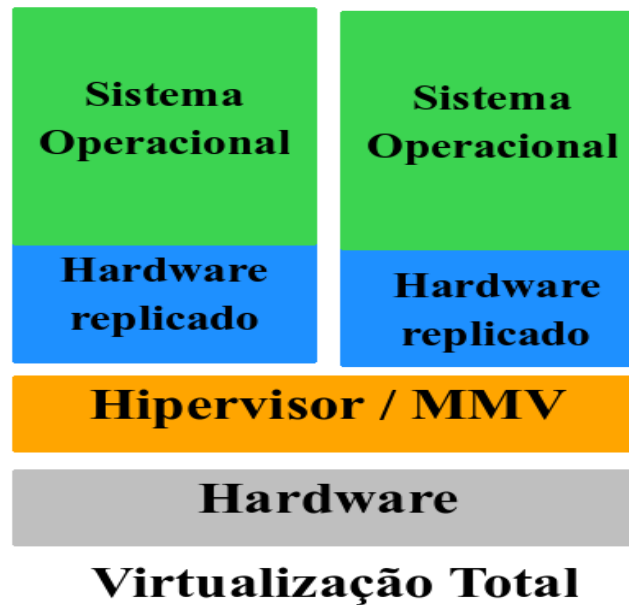
Virtualização nativa, também conhecida *bare-metal* ou como virtualização total, é uma camada de virtualização provida pelo gerenciador de máquinas virtuais, no qual os dispositivos de hardware são acessados por drivers virtuais providos pelo hipervisor. Toda a instrução de hardware é analisada e gerenciada pelo monitor de máquina virtual, prejudicando o desempenho geral dos dispositivos, causado pela sobrecarga de instruções a serem analisadas pelo gerenciador. Todavia, essa técnica permite maior compatibilidade entre uma gama maior de sistemas operacionais e dispositivos de hardware.

Os fabricantes de processadores Intel e AMD, integraram em seus processadores uma instrução para auxiliar a virtualização total, Intel VT-x e AMD –V, acrescentando a assistência de hardware à virtualização total.

A Figura 2 demonstra o esquema hierárquico de funcionamento da virtualização total utilizando um monitor de máquina virtual (MMV), diretamente sobre a camada de hardware, ou seja, um hipervisor do tipo I.

A virtualização total tem por objetivo fornecer ao sistema operacional visitante uma réplica do hardware subjacente. Dessa forma, o sistema operacional visitante é executado sem modificações sobre o monitor de máquina virtual (VMM), o que traz alguns inconvenientes. O primeiro é que o número de dispositivos a serem suportados pelo VMM é extremamente elevado. Para resolver esse contratempo, as implementações da virtualização total usam dispositivos genéricos, que funcionam bem para a maioria dos dispositivos disponíveis, mas não garantem o uso da totalidade de sua capacidade. (MATTOS, 2008)

**Figura 2 Hierarquia estrutural da virtualização Total**

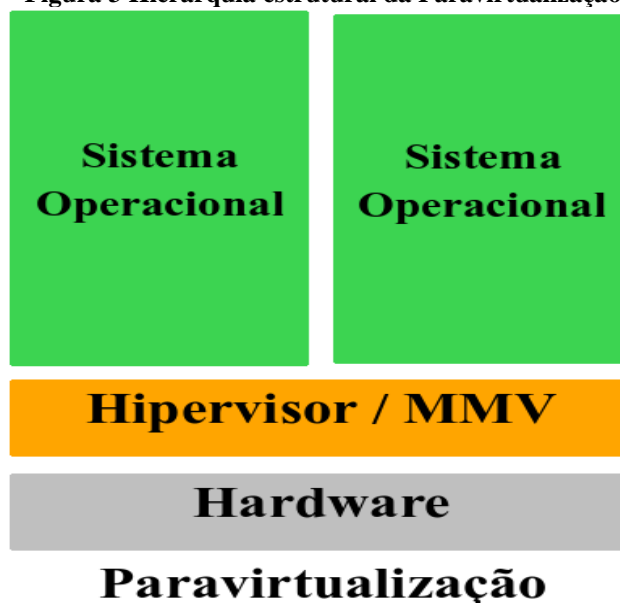


Fonte: Rodrigues (2017)

## **2.2 PARAVIRTUALIZAÇÃO**

A paravirtualização consiste em uma técnica de virtualização cujo sistema operacional convidado é preparado para interação direta com os drivers originais do hardware, permitindo acesso otimizado aos recursos de dispositivos da máquina física. É necessário que o núcleo seja modificado para atender aos requisitos dessa técnica, melhor interagindo em conjunto com outros sistemas operacionais convidados em tarefas paralelas. A Figura 3 mostra a hierarquia estrutural do funcionamento da paravirtualização aplicada através do monitor de máquinas virtuais (MMV) hipervisor tipo I.

**Figura 3 Hierarquia estrutural da Paravirtualização**



Fonte: Rodrigues (2017)

A paravirtualização é uma alternativa à virtualização total. Nesse modelo de virtualização, o sistema operacional é modificado para chamar o VMM sempre que executar uma instrução que possa alterar o estado do sistema, uma instrução sensível. Isso acaba com a necessidade de o VMM testar instrução por instrução, o que representa um ganho significativo de desempenho. Outro ponto positivo da para-virtualização é que os dispositivos de hardware são acessados por drivers da própria máquina virtual, não necessitando mais do uso de drivers genéricos que inibiam o uso da capacidade total do dispositivo. (MATTOS, 2008)

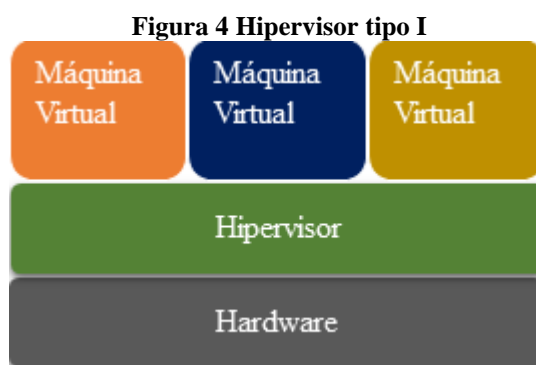
## 2.3 HIPERVISOR

Segundo Pollon (2008), hipervisor é uma camada de abstração de recursos para que dispositivos se alinhem homogeneamente e ofereçam transparência para o usuário. Essa ferramenta, também conhecida como monitor de máquinas virtuais, é responsável por prover a abstração necessária do hardware, gerenciando os recursos da máquina de maneira dinâmica e escalável, garantindo que recursos heterogêneos funcionem de maneira homogênea na perspectiva do usuário.



Diferentemente das ferramentas de virtualizações domésticas, como, virtualbox e o vmware, os hipervisores são utilizados especificamente em servidores corporativos, possuem um conjunto de ferramentas que auxiliam a administração, controle e monitoramento dos recursos, e são projetados para agirem em conjunto com a arquitetura de hardware das máquinas corporativas para que seu desempenho seja otimizado por instruções específicas.

### 2.3.1 Hipervisor tipo I



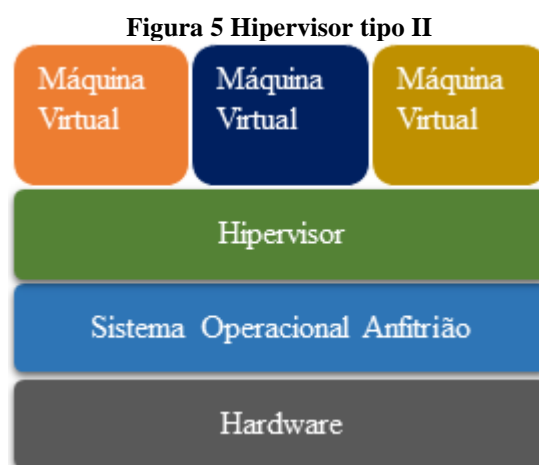
Fonte: (Elaboração Própria)

Conforme a Figura 4, o hipervisor tipo I administra diretamente as máquinas virtuais, seu funcionamento consiste na criação de uma camada de abstração entre o hardware e as máquinas virtuais, isto faz com que o controle das operações, distribuição de recursos de hardware e administração das máquinas não tenha interferência de um sistema operacional hierarquicamente acima do monitor de máquinas virtuais. O hipervisor é responsável por gerenciar os recursos de hardware dinamicamente sem interferência do sistema operacional. Os recursos da máquina são reconhecidos pelo hipervisor e cada máquina virtual irá instalar seus conjuntos de *drivers* do hardware abstraído pela virtualização tipo I. O monitor de máquinas virtuais irá configurar e

prover o hardware necessário à máquina virtual, para que esta provenha o serviço que lhe é atribuído, sendo estes recursos escaláveis, sem interferências no serviço provido pela *Virtual Machine*(VM), garantindo também maior compatibilidade entre uma variedade de hardwares, como adaptadores de rede e dispositivos de armazenamento, e melhorando ainda a interação de sistemas operacionais diferentes na mesma infraestrutura.

### 2.3.2 Hipervisor tipo II

O hipervisor tipo II é caracterizado por funcionar como uma aplicação, provendo um serviço dentro de um sistema operacional hospedeiro. Ele tem por finalidade criar máquinas virtuais a partir de recursos de hardware abstraídos pelo sistema hospedeiro. O sistema operacional que está hierarquicamente acima do hipervisor, irá prover os drivers que são necessários ao monitor de máquinas virtuais e, conseqüentemente, as máquinas virtuais provedoras dos serviços. O sistema anfitrião poderá oferecer algum serviço, sendo que ele também representa a máquina principal com todos os seus recursos.



Fonte: (Elaboração Própria)

A figura 5 demonstra, através de diagrama, como funciona a estrutura hierárquica da virtualização utilizando o hipervisor tipo II. Pode-se observar uma

camada extra dentro do esquema de funcionamento em relação ao hipervisor tipo I, sendo esta denominada sistema operacional anfitrião.

## 2.4 MONITORAMENTO DE REDE

O monitoramento de rede tem por finalidade obter dados de processos, aplicativos e ativos de rede, como *switches*, adaptadores de rede local e roteadores, a fim de realizar a prevenção de problemas e manutenção da mesma. Alguns modelos possuem ferramentas para gerar gráficos de períodos de monitoramento, dando a possibilidade ao gerente de rede de planejar e monitorar atividades da rede, e até mesmo comparar o desempenho de diferentes tecnologias usadas nos ativos através dos dados de desempenho fornecido pela ferramenta.

O monitoramento dos ativos de redes é uma avaliação contínua das variáveis operacionais, cujo principal objetivo é detectar antecipadamente anomalias com uma baixa taxa de falsos positivos, ou seja, alarmes-falsos, garantindo assim um bom funcionamento e confiabilidade das redes de computadores monitoradas (ZARPELÃO, 2004)

Os softwares de gerência de rede, como zabbix e cacti, controlam remotamente os componentes da infraestrutura, determinam o status de *hubs*, *switches* e pontes, além de trocar rotas variáveis de interface de rede (COMER, 2007).

## 2.5 MÉTRICAS DE AVALIAÇÃO

Segundo Pollon (2008), são utilizadas métricas relacionada ao estado de funcionamento do processador, como a utilização de CPU, a fila de processos, utilização da(s) placa(s) de rede, utilização da memória e utilização dos discos. São esses os recursos consumidos pelos serviços variados, e através deles é observado o desempenho do servidor, podendo assim estabelecer a relação custo-benefício de seus recursos, ou seja, a relação entre a sobrecarga do servidor para a realização da tarefa solicitada. As métricas utilizadas nesse trabalho foram uso de processador, quantidade de memória RAM consumida, velocidade de transferência de disco rígido e velocidade

de transferência de adaptador de rede. Cada métrica foi observada no intervalo de tempo em que a tarefa de teste em questão demorou para finalizar.

## **2.6 FERRAMENTAS DE AVALIAÇÃO**

As ferramentas utilizadas para avaliação dos testes necessitaram de algumas configurações específicas para que se ajustassem aos requisitos de coleta de dados, bem como alguns parâmetros estabelecidos para padronizar os testes. Ambos os itens estão descritos na sessão a seguir.

### **2.6.1 Zabbix**

Ferramenta de monitoramento de rede utilizada para controlar e gerenciar os ativos de rede, serviços e status do servidor. Através deste software foram obtidos dados durante os testes e geraram-se gráficos de medida de uso de hardware em cada cenário de teste.

A solução de monitoramento foi instalada em uma máquina virtual de configuração, 2 *Gigabytes* de memória RAM, 2 processadores lógicos e 40 Gb de Disco Rígido. O sistema operacional utilizado foi Linux Debian 8.0. Essa máquina criada fora na própria estação de trabalho e não no servidor, para que não viesse interferir no desempenho do hardware durante o monitoramento dos testes.

Algumas configurações foram necessárias para que o monitoramento fosse feito corretamente, tal como a instalação do agente Zabbix nas máquinas virtuais, no hipervisor XenServer, já que o mesmo possui o kernel baseado no Linux CentOS 7 é possível monitorá-lo, o Hyper-V é dotado de um kernel Windows Server, totalmente

compatível com o agente do Zabbix. O ESXi necessitou de uma configuração especial, habilitar a coleta de dados dentro do arquivo de configuração do Zabbix server, como explicado no manual da ferramenta disponibilizado na sua página web.

O Zabbix possui um gama de *templates* que contem itens pré-configurados para as plataformas Windows, Linux e o próprio ESXi, entretanto foi necessário criar um item para captar a carga utilizada nos processadores em ambos os sistemas operacionais, e um item especial calculado para a captação da banda consumida pelos discos rígidos no Linux.

A chave de item utilizada para monitoramento foi a *system.cpu.util[all]*. A construção de item calculado que monitorasse a banda utilizadas pelo disco rígido, necessitou que alguns itens especiais fossem criados, sendo eles: *vfs.dev.read[/dev/" nome da unidade de disco", sps]* e *vfs.dev.write[/dev/" nome da unidade de disco" /, sps]*, que são responsáveis por captar, leitura e escrita de disco medida em setores por segundo, sabendo-se que cada setor lógico do disco rígido é de tamanho 512 bytes, foi possível criar um item que calcula o uso de banda do dispositivo, multiplicando o itens anterior por 512, assim podemos definir a chave dos item calculados utilizando uma expressão específica do Zabbix, o *last* e então mensurar o consumo da banda de disco rígido em bytes por segundo. Os items de cálculo de uso de banda de leitura e escrita de disco rígido ficaram respectivamente, *512\*last ("vfs.dev.read[/dev/" nome do disco", sps])* e *512\*last ("vfs.dev.write[/dev/" nome do disco", sps])*.

## 2.6.2. Apache j meter

O software Apache J Meter foi utilizado para realizar os testes de carga dentro do servidor, podendo requerer diversos serviços, tais como banco de dados, *webservices*, *e-mail*, transferência de arquivos entre outros.

O J meter é uma ferramenta de código aberto, escrita em Java, com suporte a vários sistemas operacionais, possui uma interface intuitiva bastante simples de se

utilizar, além de um manual bem elaborado disponível no site da aplicação, mostrando passo a passo como promover o teste e definir os parâmetros de execução. Sua versão atual é a 3.2, a mesma utilizada nesse trabalho.

Os parâmetros gerais relevantes para o teste de carga, podem ser encontrados no item grupo de usuário, onde é possível definir o número de usuários virtuais, tempos de inicialização que define quanto segundos serão inicializados todos os usuários virtuais e o contador de iteração que representa a quantidade de ciclos de testes. Alguns itens de caixa de checagem auxiliam a controlar o comportamento dos testes ao longo da execução, como agendador que permite definir a duração do teste, ou um painel de ações pré-estabelecidas em caso de erro do testador, entretanto o foco é mantido nos usuários virtuais, interações e tipos de testes, e cada teste possui seus próprios parâmetros.

As unidades de testes do J meter são denominadas amostras, cuja relação se deriva do usuário virtual com o teste em execução, como, por exemplo, um usuário virtual executa um teste de recuperação de arquivo usando o protocolo FTP, então é gerada um registro de amostra. O número de amostra é definido pelo produto de contador de iterações, número de usuários virtuais e a quantidade de testes. Uma iteração representa um ciclo, no qual um usuário virtual executa os testes definidos:

$$\text{Iteração} = \text{Usuário virtual} * \text{Quantidade de testes}$$

Se definirmos o contador de iteração para 10 e o número de usuários virtuais 20, fazendo a execução de um teste de recuperação de arquivo e um teste de envio de arquivo, teremos a relação de 20 usuários, cada um executando 2 testes no qual repetiram esse ciclo 10 vezes.

$$\text{Número de Amostras} = (20 \text{ usuários} * 2 \text{ testes}) * 10 \text{ iterações}$$

Cada protocolo possui um elemento de configuração onde é possível definir os padrões que serão utilizados por todos os testes, tais como número de IP do servidor, nome de usuário e senha, caso necessário.

Elementos de saída e dados de monitoramento, denominados como escutadores, também podem ser adicionados à árvore de teste, mas os escolhidos para monitorar na execução dos testes foram apenas árvore de resultado e resultados em

tabela, apenas para demonstrar alguns dados pertinentes, como status de cada amostra em tempo de execução e quantidade de amostras executadas com sucesso.

Os parâmetros seguiram um padrão comum para os testes executados, e as cargas simuladas de testes utilizaram os protocolos LDAP, HTTP, FTP e a interface JDBC, os parâmetros que não foram especificados, continuaram com valores padrões, como recomendado pela documentação do J meter. (APACHE SOFTWARE FOUNDATION, 2017)

### 2.6.2.1 Ftp

O FTP (*file transfer protocol*) ou protocolo de transferência de arquivo, é um protocolo de transferência de arquivo bastante versátil, tornando-se uma solução bem difundida entre os usuários que necessitam enviar e receber conteúdo dentro da arquitetura cliente-servidor.

A Tabela 1 descreve os parâmetros necessários para execução dos testes FTP.

**Tabela 1 Parâmetros de teste FTP do J meter.**

Número de usuários virtuais	Tempo de inicialização de usuários	Número de iterações	Quantidade de testes	Número de amostras	Número da porta de conexão
20	2	5	2	200	21

Fonte: Elaboração própria

O Caminho no qual o arquivo utilizado para os testes FTP estava nos seguintes locais nas máquinas:

- Estação de trabalho: **C:\Users\” NomeDoUsuário” \Downloads**
- Máquina virtual CentOS: **/home/”NomeDoUsuário” /Documentos/upload**
- Máquina virtual Windows Server: **/download**

Os testes FTP consistiriam em efetuar operações de transferência de arquivo, uma para download e outra para upload, ambos utilizaram um arquivo de vídeo com

extensão .AVI de 9 MB, armazenado no servidor FTP local para o recebimento, e disponível na estação de trabalho para o envio, ambas as operações de upload e download, utilizaram o mesmo caminho em cada máquina.

#### **2.6.2.2 Http**

O protocolo HTTP (*Hypertext Transfer Protocol*), traduzido protocolo de transferência de hipertexto, consiste em um protocolo de comunicação utilizado como padrão de comunicação da internet (*World Wide Web*), e muito utilizado para executar *webservices*. Utiliza-se a arquitetura cliente-servidor e possui uma versão com segurança aprimorada, HTTPS (*Hyper Text Transfer Protocol Secure* - protocolo de transferência de hipertexto seguro). A comunicação baseia-se pela troca de mensagens, de solicitação de resposta, essa comunicação é denominada sessão, sendo controlada pelo protocolo TCP (*transmission control protocol*), ou controle de transmissão de protocolo.

A Tabela 2 mostra os parâmetros que foram utilizados para executar os testes referentes ao protocolo HTTP.



**Tabela 2 Parâmetros de teste HTTP do J meter.**

Número de usuários virtuais	Tempo de inicialização de usuários	Número de iterações	Quantidade de testes	Número de amostras	Número da porta de conexão
50	4	1000	2	100000	80

Fonte: Elaboração própria

O teste HTTP foi executado utilizando a aplicação *phpmyadmin*, um serviço web de administração de banco de dados, que utiliza o mesmo protocolo de comunicação de hipertexto para acessar o servidor no qual está instalado.

O teste consistiu em solicitar duas páginas do *phpmyadmin* continuamente até alcançar o número total de amostras.

Os links para o *phpmyadmin* utilizados:

CentOS:

HomePage: **`http://”Número do IP”/phpmyadmin/index.php?token=1f6ae1196a19c4f648423f173f2b23aa#PMAURL-2:index.php?db=&table=&server=1&target=&token=1f6ae1196a19c4f648423f173f2b23aa`**

Configurações:

**`http://”Número do IP”/phpmyadmin/index.php?token=1f6ae1196a19c4f648423f173f2b23aa#PMAURL-1:prefs_manage.php?db=&table=&server=1&targe`**

Windows Server:

HomePage: **`http://”Número do IP”/phpmyadmin/index.php`**

Configurações: **`http://”Número do IP”/phpmyadmin/prefs_manage.php?db=`**

### 2.6.2.3 Jdbc

A interface JDBC (*Java Database Connectivity*) é uma API a qual a linguagem Java usa para enviar comandos SQL e traduzir para instruções de bancos de dados relacionais. O J meter usa essa interface para se comunicar com a base de dados e realizar as operações solicitadas.

A Tabela 3 mostra os parâmetros de configurações utilizados durante os testes JDBC no J meter.

**Tabela 3 Parâmetros de teste JDBC do J Meter.**

Número de usuários virtuais	Tempo de inicialização de usuários	Número de iterações	Quantidade de testes	Número de amostras	Número da porta de conexão
50	4	1000	2	100000	3306

Fonte: Elaboração própria

O banco de dados utilizado foi o Mysql Server 5.7 no Windows Server, e MariaDB 10 no CentOS, ambos utilizando a mesma tabela *City*, da base de dados *World*, base de amostra que acompanha a instalação do Mysql Server. Ambos os administradores de bancos de dados, Mysql e MariaDB, utilizam a linguagem SQL, entretanto o Mysql somente está disponível para a plataforma Windows. O MariaDB é uma alternativa para utilização da linguagem SQL muito semelhante ao MYSQL para os usuários de Linux.

As colunas selecionadas de testes foram, ID e *Population*, compostas por dados do tipo inteiro. As rotinas de testes foram criadas para executar comandos de seleção e atualização de valores, cujas as expressões sql são:

Seleção:

**select id, name, district from city;**

Atualização:

**update city**

**set population = '20'**

**where id != null;**

### 2.6.2.4 Ldap

O LDAP (*Lightweight Directory Access Protocol*) é um protocolo de comunicação de rede, utilizado por aplicações de controle de diretórios como: *Active Directory* da Microsoft, e o software livre OpenLdap, aplicações que permitem organizar de maneira hierárquica informações de usuários, autenticação, políticas de grupo e domínios.

Assim como os bancos de dados relacionais, o serviço de diretórios que utilizam LDAP, possuem comandos de operações específicas, controlados por uma sessão de conexão. Através dessa sessão são efetuadas adições, exclusões e modificações de dados de registros (*add, delete, modify*).

Abaixo a Tabela 4 indica quais os parâmetros foram utilizados durante os testes LDAP.

**Tabela 4 Parâmetros de teste LDAP do J meter.**

Número de usuários virtuais	Tempo de inicialização de usuários	Número de iterações	Quantidade de testes	Número de amostras	Número da porta de conexão
25	2	100	4	10000	389

Fonte: Elaboração própria

A simulação de carga feita pelo J meter utilizando o protocolo LDAP consiste basicamente em adicionar o teste e selecionar o tipo de operação. Foi criada apenas uma instância para cada teste (adição, exclusão, busca e edição), e um elemento de configuração padrão, citado anteriormente no item 3.4.2, para acesso ao servidor de arquivos de diretório.

## 2.7 TRABALHOS RELACIONADOS

Pode se destacar o trabalho de conclusão de curso virtualização de servidores em ambiente heterogêneos e distribuídos - estudo de caso (POLLON, 2008) que enfatiza de

maneira similar os testes aqui abordados, e em diferentes ambientes de produção e operação no contexto empresarial. Entretanto, objetiva determinar qual método melhor se encaixa na infraestrutura de rede apresentada. Foi utilizado o ambiente de uma grande empresa com diversos tipos de hardwares, ao contrário deste que usará um ambiente doméstico e simulação de serviços em produção, que a qualquer momento pode ser interrompido, e focará mais objetivamente em técnicas utilizadas independentemente do ambiente a ser virtualizado.

O objetivo do seu trabalho foi definir qual solução de virtualização seria mais apropriada para organizar os 450 servidores espalhados por filiais e pela matriz da empresa. Através de testes em diversos sistemas operacionais e hardwares, e devido as exigências da empresa, chegou-se à conclusão que para as filiais a solução Vmware Server seria a melhor opção, e para a matriz, o conjunto de ferramentas Vmware ESXi e o gerenciador Wmware vcenter.

O artigo virtualização de servidores e desktops (BOSING; KAUFMANN, 2012), possui aspectos semelhantes a este trabalho, onde é feita uma comparação entre ambientes com e sem emprego da virtualização. O objetivo desse trabalho foi, através de testes, comparar determinados aspectos, como uso de memória RAM, de disco rígido, processamento e temperatura, utilizando cada técnica citada.

O software utilizado para os testes foram o Xenserver e o Xencenter da Citrix. Foi concluído que nas tarefas que demandam mais recursos, o uso de virtualização é sutilmente inferior aos que não possuem a camada de abstração criada pelo hipervisor. Entretanto no quesito confiabilidade, economia de energia, refrigeração e escalabilidade, os resultados obtidos pela virtualização, do ponto de vista dos autores, são compensadores em relação às desvantagens apresentadas. Assim como este trabalho, O artigo virtualização de servidores e desktops (BOSING; KAUFMANN, 2012), utiliza hipervisores para a realização da comparação, todavia, nesta pesquisa, serão utilizados diferentes hipervisores, e em diferentes situações.

### 3 ESTUDO DE CASO

Nesta seção está descrito como foi feito o estudo de caso, contendo um breve relato na visão geral dos passos necessários para realização deste trabalho, dos cenários de teste e ferramentas de avaliação.

#### 3.1 INTRODUÇÃO

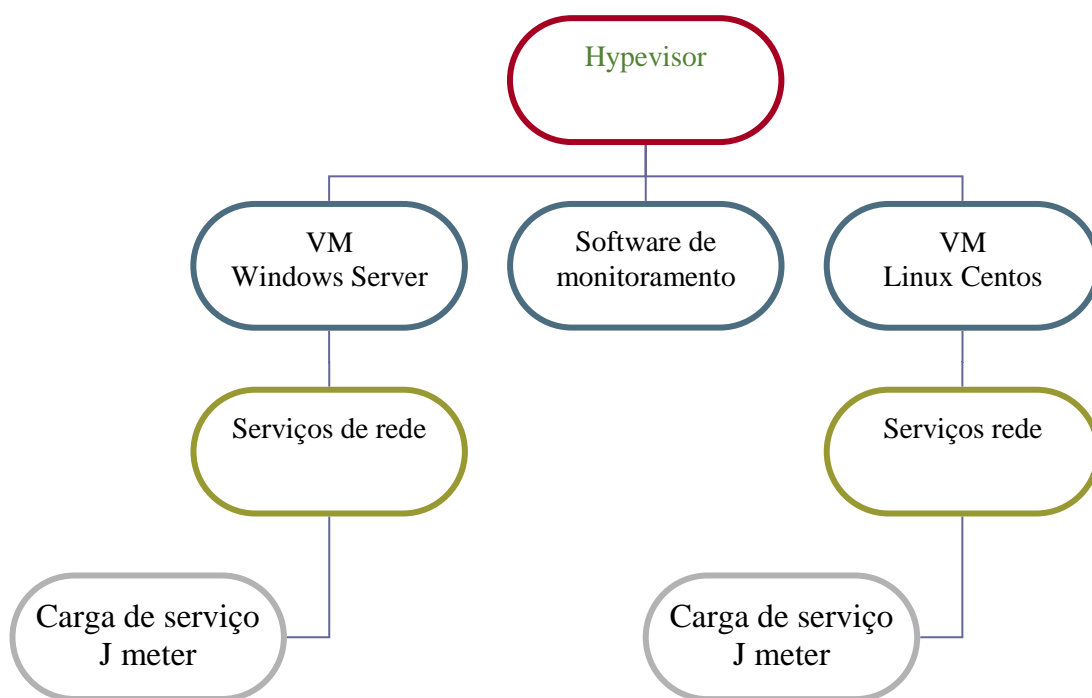
Foram utilizados os sistemas operacionais Microsoft Windows Server 2012, Linux Distribuição CentOS 7.1, os hipervisores que farão o papel da virtualização foram, Vmware ESXi Server 6.5, Citrix XenServer 7.1, Microsoft Hyper V 2012 e KVM quem 1.5.3. O *software* utilizado para monitoramento do servidor, coleta de dados e geração de gráficos de desempenho foi o Zabbix 3.0.4.

O computador utilizado como servidor para os testes é do tipo desktop, composto por um processador da marca Intel de modelo core I 7 4790 de 4.0ghz, de quatro núcleos e 8 *threads*, com tecnologia de virtualização Intel VT-x, 8 GB de memória RAM, uma unidade de disco SSD de 128 GB que será utilizado para instalação do sistema anfitrião e dos hipervisores, e uma unidade de disco rígido de 1 TB como repositório de máquinas virtuais com sistemas operacionais convidados, que serviram como ambiente para executar os serviços de rede, dois adaptadores de rede *Gigabit Ethernet* de banda de 1 Gb/s e como estação de trabalho para executar os testes, foi utilizado um computador do tipo notebook, composto por um processador Intel core I 7 4500U de até 3.00ghz de dois núcleos e 4 *threads*, 8 GB de memória RAM e 1 TB de disco rígido, com um adaptador de rede *Fast Ethernet* de banda de 100 Mb/s. Um roteador de marca D Link e modelo DIR 868L, cujas as portas de rede são de banda de 1 Gb/s capazes de oferecer velocidades razoáveis para demandar os serviços de rede.

### 3.2 CENÁRIO DE TESTE

Cada máquina virtual possui os serviços instalados para serem sobrecarregados pela ferramenta de emulação de requisição de serviço.

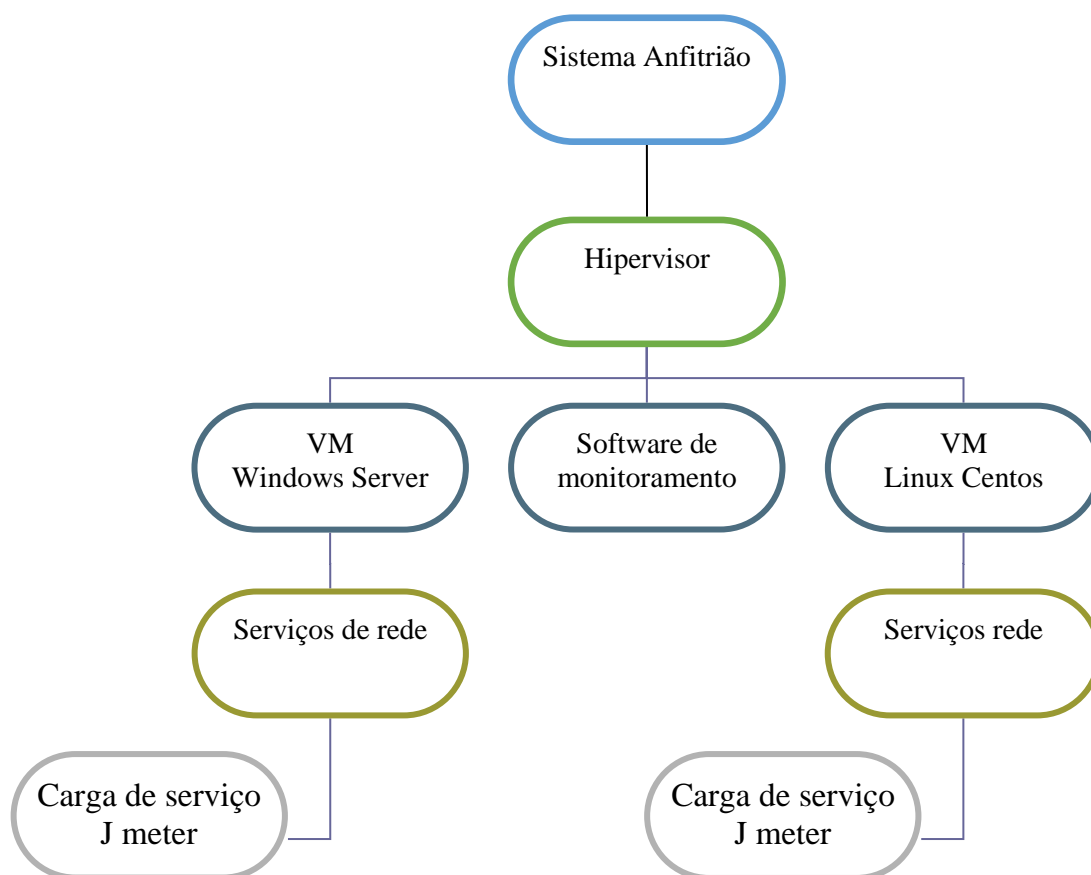
**Figura 6 Ilustração de cenário de teste de hipervisor tipo I**



Fonte: Elaboração própria

A Figura 6 demonstra o esquema dos cenários de teste de hipervisor tipo I, mostrando a ordem hierárquica da disposição das aplicações.

**Figura 7 Ilustração de cenário de teste de hipervisor tipo II**



Na figura 7 pode ser observado a diferença principal entre os fluxos de serviço de hipervisor tipo I e II, que a camada de sistemas operacional entre a camada de hardware e hipervisor, diferença essa que será explorada como objeto de teste de desempenho.

Para cada hipervisor, foram criadas duas máquinas virtuais, e a cada uma delas, foi feito a instalação dos serviços para a utilização dos protocolos de testes do J meter.

As máquinas virtuais foram criadas igualmente em todos os hipervisores, cujas especificações são, 3 GB de memória RAM, 3 Processadores Lógicos, duas placas de rede virtuais de velocidade 1 Gb/s e um Disco Rígido Virtual de 40 GB. Os hipervisores e sistemas operacionais anfitriões foram instalados e uma unidade de disco SSD e as máquinas virtuais com seus respectivos sistemas convidados foram instalados em um repositório criado em uma unidade de disco rígido de 1 TB.

Os hipervisores utilizados para a realização desse trabalho foram selecionados devido a sua versatilidade para executar os testes, apresentarem recursos para a administração e monitoramento remoto das máquinas virtuais, e possuírem interfaces de utilização intuitivas. Mesmo sendo ferramentas consolidadas no mercado e na literatura, são softwares de código livre ou possuem uma versão gratuita disponível para usuários que não necessitem de todos os recursos providos pela versão comercial paga, ou desejam testar uma versão mais simples do software.

**Quadro 1 Hipervisores utilizados.**

Hipervisores						
Nome	Versão	Licença	Kernel	Tipo	Virtualização	Software administrador
Citrix XenServer	7.1	Grátis	Linux, baseado no CentOS 7	I, <i>Bare-metal</i>	Paravirtualização	Xencenter 6.5
Wmware EXSi	6.5a	Teste de 60 dias	vmkernel	I, <i>Bare-metal</i>	Virtualização total assistida por hardware	Web
KVM	QEMU 1.5.3	Código Aberto	CentOS 7.1	II, <i>Hosted</i>	Paravirtualização	Virt-Manager 1.4.0
Hyper -V	2012	Grátis	Windows Server 2012 R2	Tipo I, <i>Bare-metal</i>	Virtualização total assistida por hardware	Interface gráfica de gerenciamento do Hyper –V

Fonte: Elaboração própria.



**Quadro 2 Sistemas operacionais utilizados**

Sistemas Operacionais							
Nome	Versão	Licença	Aplicação FTP	Aplicação HTTP	Servidor de Banco de Dados	Aplicação LDAP	Função
Windows Server	2012 R2	Teste de 180 dias	Microsoft FTP	<i>phpmyadmin</i>	Mysql 5.7	Active Directory	Máquina virtual de teste de carga
CentOS	7.1	Código Aberto	VSFTP	<i>phpmyadmin</i>	Mariadb 10	OpenLdap	
Debian	8.0 Jessie	Código Aberto		Zabbix 3.0.4			Máquina virtual de monitoramento
Windows	10 Home Single Language	Licenciado					Estação de trabalho

Fonte: Elaboração própria

### 3.3 RESULTADOS OBTIDOS

A seguir estão descritos os resultados dessa pesquisa, dados sobre as métricas, sistemas operacionais e hipervisores, tabulados e devidamente comparados de acordo com o tipo de testes realizados.

### **3.3.1 Tabelas comparativas de uso de memória e processador**

Nessa seção são expostos os dados que foram coletados durante os testes, tabulados e analisados de forma comparativa. Os dados representam a média de consumo de recursos durante o tempo de execução dos testes.

**Tabela 5 Resultado de teste FTP Windows Server**

Hipervisor	CPU VM	Memória RAM VM	CPU HV	Memória RAM HV	Duração do teste	Número de Amostra completadas
XenServer	0,42 %	1,78 GB	0,07 %	605,21 MB	01m:05s	184
ESXi	1,71 %	1,67 GB	2,0625%	3,98 GB	01m:19	200
KVM	3,23 %	1,96 GB	1,55 %	6,96 GB	01m:06s	183
Hyper-V	8,72 %	1,02 GB	1,24 %	3,88 GB	01m:07s	186

Fonte: Elaboração Própria

Os testes usando o protocolo FTP demonstrados na Tabela 5, apresentaram resultados muitos semelhantes, entretanto, o hipervisor XenServer mostrou-se um pouco mais eficiente na execução de operações de transferência de arquivo, demandando menos tempo para completar a tarefa. O item CPU VM, que é o percentual de processamento médio consumido pela máquina virtual, e o item CPU HV que corresponde ao percentual de uso do processador médio durante a tarefa utilizado pelo hipervisor, tiveram os melhores resultados entre os hipervisores. A métrica Memória Ram HV, que se refere a quantidade de memória consumida pelo software hipervisor, também apresentou resultados mais eficazes com o hipervisor Xenserver em relação aos outros hipervisores. Com isso, levando em consideração os resultados que se referem a processamento, memória, e número de amostras completadas deixando apenas de completar 16, e apresentando um resultado mais elevado apenas no item Memória Ram VM, que corresponde a quantidade de memória RAM consumida pela máquina virtual, conclui-se que o XenServer foi superior no teste FTP em ambiente Windows.

**Tabela 6 Resultado de teste HTTP Windows Server**

Hipervisor	CPU VM	Memória RAM VM	CPU HV	Memória RAM HV	Duração do teste	Número de Amostra completadas
XenServer	80,88%	2 Gb	0,11%	611,74 MB	13m:48s	99951
ESXi	76,39 %	1,85 GB	54,5%	4 GB	12m:22s	100000
KVM	62,09 %	1,93 GB	18,12 %	6,96 GB	21m:34s	100000
Hyper-V	91,51 %	1,39 GB	0,63 %	3,89 GB	16m:25s	100000

Fonte: Elaboração própria

De acordo com a Tabela 6, pode-se concluir que o ESXi da Vmware alcançou resultados mais eficazes entre consumo de processador na VM por tempo para conclusão do teste e o número de amostras completadas, chamando atenção apenas para o uso de processamento do hipervisor que ultrapassou 50 por cento de uso, bem acima dos outros hipervisores comparados.

**Tabela 7 Resultados de teste JDBC Windows Server**

Hipervisor	CPU VM	Memória RAM VM	CPU HV	Memória RAM HV	Duração do teste	Número de Amostra completadas
XenServer	0,83 %	1,62 GB	0,08 %	611,72	08m:29s	100000
ESXi	5,24 %	851,28 MB	4,8%	3,16 GB	08m:28s	100000
KVM	8,98 %	950,03 MB	3,04 %	6,95 GB	08m:32s	100000
Hyper-V	24,45 %	1,14 GB	1,2 %	3,72 GB	08m:34s	100000

Fonte: Elaboração própria

Pode se observar na Tabela 7 que os resultados provenientes do teste JDBC, tiveram valores muito semelhantes no quesito tempo, número de amostras processadas, nota-se que todos os ambientes conseguiram processar os testes completando todas as amostras. Quanto da utilização dos recursos computacionais o hipervisor Xenserver apresentou a melhor relação custo computacional por tempo para realização da tarefa chamando atenção apenas para o uso de memória na máquina virtual.

**Tabela 8 resultado de teste LDAP Windows Server**

Hipervisor	CPU VM	Memória RAM VM	CPU HV	Memória RAM HV	Duração do teste	Número de Amostra completadas
XenServer	2,34 %	1,96 GB	1,13 %	619,48 MB	05m:56s	9456
ESXi	4,43 %	1,93 GB	5,15%	4,07 Gb	02m:35s	9305
KVM	4,05 %	1,08 GB	1,62 %	6,97 GB	15m:49s	8543
Hyper-V	27,54 %	1,16 GB	1,54 %	3,9 GB	01m:04s	9633

Fonte: Elaboração própria

A Tabela 8 apresenta os resultados dos testes utilizando o protocolo LDAP. A aplicação Active Directory levou vantagem em todos os ambientes, mas principalmente no hipervisor da própria Microsoft, levando em consideração a diferença bem avantajada do tempo de execução da tarefa, e da baixa taxa erro das amostras, e o

consumo de memória moderado, a virtualização com Hyper –V é a melhor opção para execução de serviços de diretórios

**Tabela 9 Resultado de teste FTP Linux CentOS**

Hipervisor	CPU VM	Memória RAM VM	CPU HV	Memória RAM HV	Duração do teste	Número de Amostra completadas
XenServer	0.33 %	883,06 MB	0,08 %	630,25 MB	01m:21s	200
ESXi	0,35 %	882,02 MB	2.02%	2,88 GB	01m:27s	200
KVM	0,38 %	913,07 MB	0,56 %	4,48 GB	01m:22s	200
Hyper-V	1,89 %	1, 2 GB	0,73 %	3,87 GB	01m:24	200

Fonte: Elaboração Própria

Nos testes de carga FTP utilizando o Linux, pôde se observar na Tabela 9 que o Xenserver conseguiu se sobressair aos outros hipervisores em quase todos itens analisados, somente utiliza um pouco mais de memória na máquina virtual que o ESXi da Vmware, diferença essa pequena demais para ser levada em consideração no teste em questão.

**Tabela 10 Resultado de testes HTTP Linux CentOS**

Hipervisor	CPU VM	Memória RAM VM	CPU HV	Memória RAM HV	Duração do teste	Número de Amostra completadas
XenServer	77,5%	1,21 GB	0,11%	643,43 MB	16m:01s	100000
ESXi	75,58 %	1,22 GB	62,875%	3,03 GB	14m:12s	100000
KVM	78,71 %	909,53 MB	23,7 %	5,04 GB	13m:26s	100000
Hyper-V	80.98 %	1,5 GB	0,5 %	3,87 GB	14m:02s	100000

Fonte: Elaboração Própria

A Tabela 10 mostra que no teste de HTTP, ao qual foi usado o acesso a uma página em específico, houveram diferenças significativas no uso dos hipervisores. O software de monitoramento que conseguiu alcançar números mais equilibrados em relação ao custo computacional durante as tarefas, foi o XenServer da Citrix, entretanto, este foi o que demorou mais tempo para conseguir terminar as 10000 amostras de testes, com diferença de 2 minutos e 35 segundos a mais que KVM. O software de

monitoramento nativo do Linux terminou a tarefa em menor tempo, contudo este teve grande consumo de memória pelo hipervisor, bem superior aos demais, apesar da baixa utilização de memória por parte da máquina virtual, o que nos leva a concluir que se a necessidade seja a velocidade, o KVM se sai melhor para a tarefa de HTTP, mas este demanda mais recurso de memória do servidor que os demais.

**Tabela 11 Resultado de Teste JDBC Linux CentOS**

Hipervisor	CPU VM	Memória RAM VM	CPU HV	Memória RAM HV	Duração do teste	Número de Amostra completadas
XenServer	3,42%	860,39 MB	0,11%	622,37 MB	08m:20s	100000
ESXi	3,17%	890,04 MB	4,03%	2,88 GB	08m:29s	100000
KVM	3,31%	852,21 MB	2,28%	4,26 GB	08m:40s	100000
Hyper-V	18,12%	1,18 GB	1,15%	3,87 GB	08m:21s	100000

Fonte: Elaboração Própria

A tabela 11 demonstra os resultados dos testes JDBC no ambiente Linux, que com exceção da cpu da máquina virtual, o Xenserver leva vantagem em todos os itens avaliados.

**Tabela 12 Resultado de teste LDAP Linux CentOS**

Hipervisor	CPU VM	Memória RAM VM	CPU HV	Memória RAM HV	Duração do teste	Número de Amostra completadas
XenServer	0,05 %	639,63 MB	0,11 %	639,63 MB	Interrompido em 20m:01s	2105
ESXi	0,06 %	1,2 GB	0,32%	2,89 GB	Interrompido em 20m:04s	2394
KVM	0,43 %	1,46 Gb	0,53 %	6,04 GB	43m:09s	9867
Hyper-V	1,25 %	1,59 GB	0,56 %	3,87 GB	25m:30s	9885

Fonte: Elaboração Própria

Durante as cargas realizadas em ambiente Linux utilizando os ESXi e XenServer, o teste apresentou erro com as operações de arquivos, como demonstrado na Figura 8. Segundo o arquivo de registro de erro do Linux, o limite de arquivos abertos simultaneamente, foi ultrapassado e acabou fechando o soquete de comunicação do LDAP, não restando outra alternativa senão encerrar o teste. Mesmo esse limite sendo

aumentado ao máximo permitido pelo sistema operacional, o teste seguiu acometido pelo mesmo erro. Conclui-se então, que dos hipervisores que conseguiram completar o teste com sucesso, o Hyper – V da Microsoft apresentou melhor relação de custo computacional por efetividade, completando o maior número de amostras em menor tempo.

**Figura 8 Erro LDAP Linux CentOS**

```
Mai 12 02:06:49 centos7 slapd[2586]: connection_input: conn=3073 deferring operation: binding
Mai 12 02:06:49 centos7 slapd[2586]: conn=3073 op=0 BIND dn="cn=Manager,dc=tcc,dc=com" method=128
Mai 12 02:06:49 centos7 slapd[2586]: conn=3073 op=0 BIND dn="cn=Manager,dc=tcc,dc=com" mech=SIMPLE ssf=0
Mai 12 02:06:49 centos7 slapd[2586]: conn=3073 fd=865 closed (connection lost)
Mai 12 02:06:49 centos7 slapd[2586]: conn=3050 op=3 UNBIND
Mai 12 02:06:49 centos7 slapd[2586]: conn=3050 fd=74 closed
Mai 12 02:06:49 centos7 slapd[2586]: conn=3053 op=4 UNBIND
Mai 12 02:06:49 centos7 slapd[2586]: conn=3053 fd=734 closed
Mai 12 02:06:49 centos7 slapd[2586]: conn=3052 op=3 UNBIND
Mai 12 02:06:49 centos7 slapd[2586]: conn=3052 fd=174 closed
```

Fonte: Elaboração Própria



### 3.3.2 Tabelas comparativas de dispositivos de entrada e saída

A seguir estão dados relacionados a recursos de dispositivos de entrada e saída, mais precisamente adaptador de rede e disco rígido, utilizados durante os testes de carga, seguido do comparativo desses dados. A análise se baseia em como a combinação de software de controle, e o tipo de virtualização, utilizam o recurso oferecido pelos dispositivos de entrada e saída, e quanto de largura de banda se aproveita do hardware.

**Tabela 13 Fluxo de dados FTP Windows Server**

Hipervisor	Leitura de disco	Escrita de disco	Tráfego de entrada AR 0	Tráfego de entrada AR 1	Tráfego de saída AR 0	Tráfego de saída AR 1
XenServer	18,65 kBps	1,64 kBps	241,08 kbps	98,67 bps	25,95 mbps	44 bps
ESXi	13,29 Bps	4,15 kBps	184,15 kbps	118,4 bps	24,68 mbps	115,2 bps
KVM	269,75 kBps	9,7 kBps	28,2 kbps	348,48 kbps	31,1 kbps	23,09 mbps
Hyper -V	34,01 kBps	40,96 kBps	10,75 kbps	851,97 kbps	6,73 kbps	29,29 mbps

A Tabela 13 apresenta os dados sobre o transporte de informações nas interfaces de armazenamento e de comunicação de rede no Windows Server. É possível perceber que o dispositivo mais solicitado durante os testes de transferência de arquivo foi o adaptador de rede, sendo que o Hyper-V apresentou o melhor rendimento de velocidade no tráfego de dados, principalmente nos itens de tráfego de entrada e saída do AR 1, que se refere ao fluxo de dados no adaptador de rede 1. Nos itens de tráfego de entrada e saída no AR 0, o qual se refere ao adaptador de rede 0 da máquina, o XenServer foi superior, entretanto a velocidade média alcançada foi menor que a do Hyper -V no

adaptador de rede 1, concluindo-se que a taxa de transferência de dados pela rede do Hyper-V foi maior do que a do hipervisor da Citrix. Quanto a taxa de transferência de disco rígido, o Hyper-V leva vantagem na velocidade de escrita em disco, porém, o hipervisor KVM apresentou melhor rendimento na leitura de dados em disco.

**Tabela 14 Fluxo de dados HTTP Windows Server**

Hipervisor	Leitura de disco	Escrita de disco	Tráfego de entrada AR 0	Tráfego de entrada AR 1	Tráfego de saída AR 0	Tráfego de saída AR 1
Xenserver	2,05 mBps	1,73 mBps	428,6 kbps	112 bps	12,71 mbps	35,5 kbps
ESXi	1,74 mBps	1,73 mBps	578,18 kbps	124,5 bps	13,58 mbps	98,67 bps
KVM	1,12 mBps	1,04 mBps	735,27 kbps	615,36 kbps	8,44 mbps	158,08 bps
Hyper -V	1,48 mBps	1,57 mBps	365,04 kbps	1,09 kbps	10,72 mbps	140,21 bps

Fonte: Elaboração própria

Dentre os testes que mais demandam carga, podemos destacar o teste de *webservice*, mais precisamente, testes que utilizam protocolo HTTP. É explícito na Tabela 14 que a maior necessidade de carga computacional de entrada e saída, provém do hardware de comunicação de rede. Baseado na premissa de que comunicação em rede é a essência desse teste, pode-se observar que o ESXi obteve o melhor desempenho geral na utilização de banda de comunicação de rede, mantendo uma média de transferência de saída de 13,58 Megabits por segundo. E o desempenho de disco rígido foi demasiado explorado pela plataforma de virtualização XenServer, mantendo uma média elevada de leitura e escrita no disco rígido.

**Tabela 15 Fluxo de dados LDAP Windows Server**

Hipervisor	Leitura de disco	Escrita de disco	Tráfego de entrada rede 0	Tráfego de entrada rede 1	Tráfego de saída rede 0	Tráfego de saída rede 1
Xenserver	645,69 kBps	1,59 mBps	115,27 kbps	7,19 kbps	120,96 kbps	7,52 kbps
ESXi	79,36 kBps	1,99 mBps	54,84 kbps	127,27 kbps	49,22 kbps	118,54 kbps
KVM	61,74 kBps	301,89 kBps	19,56 kbps	39,8 kbps	18,56 kbps	53,23 kbps
Hyper -V	61,74 kBps	30,89 kBps	44,49 kbps	66,36 kbps	23,13 kbps	37,26 kbps

A Tabela 15 representa os testes de protocolo LDAP, estes por sua vez, não possuem valores que diferem muito no transporte de informações pela estrutura de rede, o que demonstra um certo equilíbrio de valores entre o ESXi e o Xenserver. O Desempenho de disco rígido foi melhor alcançado pela plataforma Xenserver na Leitura de disco, e o ESXi se sobressaiu com folga na escrita de dados.

**Tabela 16 Fluxo de dados JDBC Windows Server**

Hipervisor	Leitura de disco	Escrita de disco	Tráfego de entrada rede 0	Tráfego de entrada rede 1	Tráfego de saída rede 0	Tráfego de saída rede 1
Xenserver	830,24 kBps	0 B	415 kbps	658,13 bps	51,94 mbps	37,87 bps
ESXi	215,37 Bps	1,43 kBps	813,42 kbps	141,82 bps	71,04 mbps	124,36 bps
KVM	1,07 kBps	92,91 bps	1,51 mbps	1,37 kbps	73,55 mbps	188,36 bps
Hyper -V	977 kBps	17,81 kBps	791,64 kbps	1,61 kbps	65,11 mbps	177,33 bps

Fonte: Elaboração Própria

Na Tabela 16 observamos que os testes de banco de dados realizado pela API JDBC, demanda grandes volumes de dados pela rede, logo temos valores de transferência elevados. O hipervisor KVM supera os demais visivelmente nesse aspecto,

nas transferências de entrada e saída, em ambos os adaptadores de rede. O desempenho de disco rígido foi melhor alcançado pelo hipervisor Hyper-V.

**Tabela 17 Fluxo de dados FTP CentOS**

Hipervisor	Leitura de disco	Escrita de disco	Tráfego de entrada AR 0	Tráfego de entrada AR 1	Tráfego de saída AR 0	Tráfego de saída AR 1
XenServer	3,1 kBps	32,52kBp	268,64 kbps	218 bps	31,27mbps	0 b
ESXi	3,36 kBps	38,42 kBps	172 bps	267,52 kbps	0 b	31,45 mbps
KVM	3,56 kBps	37,85 kBps	128 bps	0 b	360,61 kbps	25,79 mbps
Hyper –V	3,88 kbps	30,29 kBps	7,16 kbps	361,11 kbps	4,87 kbps	24,86 mbps

Fonte: Elaboração Própria

Como demonstrados na Tabela 17, os testes de fluxo de dados FTP no ambiente CentOS, tiveram a melhor eficiência de rede utilizando o Xenserver, com boas transferências de entrada e saída pelo adaptador de rede 0, o fator taxa de transferências em disco rígido não teve grandes diferenciais mensuráveis entre os hipervisores.

**Tabela 18 Fluxo de dados HTTP CentOS**

Hipervisor	Leitura de disco	Escrita de disco	Tráfego de entrada AR 0	Tráfego de entrada AR 1	Tráfego de saída AR 0	Tráfego de saída AR 1
XenServer	17,23 kbps	3,33 kbps	325,81 kbps	437,05 bps	7,3 mbps	0 b
ESXi	18,45 kBps	1,82 kBps	85,65 bps	494.13 kbps	0 b	8,17 mbps
KVM	17,72 kBps	1,31 kBps	136,47 kbps	0 b	492,95 kbps	13,93 mbps
Hyper -V	19,57 kBps	0 b	319,06 bps	396,07 kbps	0 b	8,17 mbps

Fonte: Elaboração Própria

Na Tabela 18, o fluxo de dados HTTP no ambiente CentOS demonstra que o hipervisor mais eficiente em termos de equidade de valores e quantidade de itens testados que sobressaíram os outros hyperviores, fora o Xenserver.

**Tabela 19 Fluxo de dados LDAP CentOS**

Hipervisor	Leitura de disco	Escrita de disco	Tráfego de entrada rede 0	Tráfego de entrada rede 1	Tráfego de saída rede 0	Tráfego de saída rede 1
XenServer	41,94 kBps	1,85 kBps	10,67 kbps	407,3 bps	7,89 kbps	0 b
ESXi	46,35 kBps	0 B	5,73 kbps	162,55 bps	4,63 kbps	0 b
KVM	102,78 kBps	0 B	434,94 bps	0 b	27,24 kbps	22,88 kbps
Hyper – V	196,11 kBps	0 B	1,13 kbps	51,33 kbps	0 b	59,89 kbps

Fonte: Elaboração Própria

A Tabela 19 representa os testes de protocolo LDAP em ambiente Linux, estes por sua vez, não possuem valores que diferem muito no transporte de informações pela estrutura de rede. O Desempenho de disco rígido foi melhor alcançado pela plataforma Hyper – V na taxa de leitura, e Xenserver na taxa de escrita de disco, já que os demais não apresentaram taxa de leitura. Embora não obteve grandes diferenças entre os hipervisores Xenserver, KVM e Hyper-V na comunicação de rede, a plataforma nativa do Linux demonstrou bons resultados nos tráfegos de entrada e saída de rede.

**Tabela 20 Fluxo de dados JDBC CentOS**

Hipervisor	Leitura de disco	Escrita de disco	Tráfego de entrada rede 0	Tráfego de entrada rede 1	Tráfego de saída rede 0	Tráfego de saída rede 1
XenServer	14,44 kBps	3,96 kbps	569,52 kbps	671,27 bps	71 mbps	0 b
ESXi	11,56 kBps	2,17 kBps	51,64 bps	771,93 kbps	0 b	71,33mbps
KVM	14,86 kBps	0 B	572,67 kbps	0 b	751 kbps	99,13 mbps
Hyper – V	16.01 kBps	34,13 kBps	7,58 kbps	7,58 kbps	753,36 kbps	65,12 mbps

Fonte: Elaboração Própria

Os dados expostos na Tabela 20, demonstram que as aplicações de bancos de dados JDBC obtiveram melhor êxito na utilização de disco durante os testes com a plataforma Hyper -V, tanto de leitura quanto de escrita. Quanto à taxa de transferência em rede, o KVM apresentou taxas mais elevadas na entrada e saída de dados.

### **3.3.3 Desempenho geral dos ambientes**

Cada teste teve o intuito de demonstrar como cada ambiente interage com os recursos da máquina, durante as cargas de trabalho, daí então estabelecer um comparativo geral do cenário. Cada técnica, com suas peculiaridades obteve resultados diferentes, em situações igualmente propostas a todas as plataformas montadas. Podemos traçar um comparativo identificando em que situação, determinada técnica se apresenta superior a outra.

O XenServer se mostrou muito versátil, consumindo pouco recurso da máquina, separando bem o desempenho de máquina virtual, e do hipervisor e equilibrando consumo computacional com velocidade de execução da tarefa. Apresentou desempenho melhor nas operações de transferência de arquivo FTP e operações de banco de dados, uma ótima combinação com o Windows Server, no que se refere a transferência de dados em disco, como fora observado no teste de HTTP.

O ESXi é caracterizado por equilibrar bem os recursos diante das tarefas, e se houver necessidade de alto desempenho, apresenta escalabilidade utilizando dinamicamente a carga de processamento provida pelo hypervisor, como pode ser observado no teste de processamento HTTP. Durante o teste de LDAP alcançou uma taxa de transferência em disco elevado junto ao Windows, e demonstrou balanceamento entre recurso e tempo de execução durante o teste FTP.

O sistema Paravirtualizado do KVM mostrou grandes possibilidades na aplicação de transporte de rede, utilizando bastante o recurso provido pelo adaptador. A

combinação com o sistema operacional convidado de kernel semelhante, trouxe ótimos resultados na utilização de dispositivos de comunicação.

Durante o teste de LDAP o Hyper –V se destacou utilizando a aplicação *Active Directory*, no sistema convidado Windows Server, entretanto durante os outros testes, apresentou consumo elevado de recursos em tarefas simples, o próprio gerenciador, faz com que haja pequenos picos de consumo de processador durante o estado de espera.

**Tabela 21 Percentual de superioridade dos hipervisores em cada item.**

	Windows				Linux			
	Xenserver	ESXi	KVM	Hyper-V	Xenserver	ESXi	KVM	Hyper-V
CPU VM	75%	0%	25%	0%	50%	50%	0%	0%
Memória VM	0%	25%	25%	50%	25%	25%	50%	0%
CPU HV	100%	0%	0%	0%	75%	0%	0%	25%
Memória HV	100%	0%	0%	0%	100%	0%	0%	0%
Leitura de Disco	50%	0%	25%	25%	0%	0%	0%	100%
Escrita de Disco	25%	25%	25%	25%	50%	25%	0%	25%
Tráfego de entrada A.R. 0	50%	0%	50%	0%	50%	0%	50%	0%
Tráfego de Saída A.R. 0	50%	25%	25%	0%	50%	0%	50%	0%
Tráfego de Entrada A.R. 1	0%	25%	50%	25%	25%	50%	0%	25%
Tráfego de Saída A.R. 1	0%	25%	50%	25%	0%	25%	50%	25%
Duração do teste	25%	50%	0%	25%	50%	0%	25%	25%
Nº de Amostras Completadas	50%	50%	0%	0%	50%	0%	0%	50%

Fonte: Elaboração Própria

A Tabela 21 demonstra o percentual de superioridade dos hipervisores em cada item, separados por ambiente. Cada protocolo, FTP, HTTP, JDBC e LDAP, representa um quarto do total de testes em cada ambiente, logo cada um representa 25% dos testes em cada item.

Pode-se observar que o Xenserver teve desempenho superior em maior número de testes, sendo que em três itens, se mostrou superior, levando vantagem em todos os protocolos testados, sendo que em 2 itens ele teve 75 por cento de vantagem em relação aos outros. Nos protocolos testados e em dez itens possui superioridade em pelo menos 50% dos resultados, sendo que em 5, está empatado com outros hipervisores. Então chegou-se ao número total de 10 itens no qual o Xenserver é superior em número de protocolos testados, de um total de 24 itens testados nos dois ambientes, 12 Linux e 12 Windows, demonstrando que a técnica de virtualização *Bare-Metal*, utilizando o hipervisor Xenserver de uma maneira geral se saiu melhor nos testes, e por essa razão, conclui-se que o seu custo computacional por benefício é mais equilibrado em aplicações diversas.



## **4 CONSIDERAÇÕES FINAIS E RECOMENDAÇÕES**

Nesse capítulo encontram-se as considerações finais desse estudo, dividido em duas partes para melhor exposição de ideias.

### **4.1 CONSIDERAÇÕES FINAIS**

Os testes foram realizados com o intuito de demonstrar como cada técnica e cada hipervisor reagiriam a diferentes situações, apesar de utilizar uma máquina desktop, e não está de fato em um ambiente em produção, pôde se estabelecer um comparativo e verificar qual melhor solução para determinada necessidade.

O objetivo da aplicação é um fator chave na hora de decidir qual técnica a se utilizar e qual software para fazê-lo, levando em considerações os recursos que serão disponibilizados para a montagem da estrutura de TI. É de suma importância, que antes de montar uma estrutura de aplicação, mesmo que de pequeno porte, se faça um estudo para saber qual solução melhor se adequa à aplicação necessária, então o objetivo desse estudo foi demonstrar de maneira simples e objetiva, quais os ganhos e perdas de cada ferramenta utilizada, ferramentas essas que são bastante consolidadas no mercado. Os testes atingiram os resultados esperados, provendo toda a informação necessária para efetuar o comparativo.

## 4.2 RECOMENDAÇÕES

Uma gama de possibilidades pode ser ampliadas a partir desse trabalho, o próprio J meter, a ferramenta base para o teste, possui uma grande variedade de testes disponíveis, que podem ser aplicados para combinar um leque maior de sistemas operacionais, métricas de avaliações diferentes e ferramentas de *benchmark*, até mesmo aplica-los em um ambiente em produção. A utilização de outros hipervisores também constitui outra possibilidade para trabalhos futuros, enriquecendo a pesquisa na área.

## REFERÊNCIAS

- BOSING, Angela; KAUFMANN, Evelacio Roque. **Virtualização de Servidores e Desktops**. Unoesc & Ciência – Acet, Joaçaba, v. 1, n. 3, p.47-64, jul. 2012. Disponível em: <<http://editora.unoesc.edu.br/index.php/acet/article/viewFile/1483/pdf>>. Acesso em: 23 out. 2016.
- MEDIAWIKI DEFAULT. **KVM**. 2016. Disponível em: <<http://www.linux-kvm.org/page/Documents>>. Acesso em: 26 maio 2017.
- CITRIX SYSTEMS. **XenServer 7.0**. 2017. Disponível em: <<https://docs.citrix.com/en-us/xenserver/xenserver-7-0.html>>. Acesso em: 26 maio 2017.
- COMER, Douglas E. **Redes De Computadores E Internet**. 4. ed. Grenoble: Bookman, 2007.
- COELHO, Fabio de Azeredo; CALZAVARA, Gustavo Santos; LUCIA, Ricardo di. **Virtualização - VMWare e Xen**. 2008. Disponível em: <[https://www.gta.ufrj.br/grad/09\\_1/versao-final/virtualizacao/index.html](https://www.gta.ufrj.br/grad/09_1/versao-final/virtualizacao/index.html)>. Acesso em: 01 jun. 2008.
- APACHE SOFTWARE FOUNDATION. **User's Manual**. 2017. Disponível em: <<http://jmeter.apache.org/usermanual>>. Acesso em: 19 jun. 2018.
- JONES, M. Tim. **Virtualização de aplicativos, passado e futuro**. 2011. Disponível em: <[http://www.ibm.com/developerworks/br/library/l-virtual-machine-architectures/#application\\_vm](http://www.ibm.com/developerworks/br/library/l-virtual-machine-architectures/#application_vm)>. Acesso em: 31 ago. 2016.
- LAUREANO, Marcos. **Máquinas Virtuais e Emuladores: Conceitos, Técnicas e Aplicações**. São Paulo: Novatec editora Ltda, 2006.
- MATTOS, Diogo Menezes Ferrazani. **Virtualização: VMWare e Xen**. 2008. Disponível em: <[https://www.gta.ufrj.br/grad/08\\_1/virtual/index.html](https://www.gta.ufrj.br/grad/08_1/virtual/index.html)>. Acesso em: 17 maio 2017.
- MICROSOFT. **Hyper-V**. 2017. Disponível em: <[https://msdn.microsoft.com/pt-br/library/mt169373\(v=ws.11\).aspx](https://msdn.microsoft.com/pt-br/library/mt169373(v=ws.11).aspx)>. Acesso em: 26 maio 2017.
- POLLON, Vanderlei. **Virtualização de servidores em ambiente heterogêneos e distribuídos - estudo de caso**. 2008. Disponível em:

<<http://www.lume.ufrgs.br/bitstream/handle/10183/15988/000695318.pdf>>. Acesso em: 22 jun. 2016.

SILVA, Edna Lúcia da; MENEZES, Estera Muszkat. **Metodologia da Pesquisa e Elaboração de Dissertação**. 3. ed. Florianópolis: Universidade Federal de Santa Catarina, 2001.

TANENBAUM, A. S., **Sistemas Operacionais Modernos 3ª Edição**; tradução Ronaldo A. L. Gonçalves, Luís A. Consularo, Luciana do Amara Teixeira; revisão Técnica Raphael Y. de Camargo; São Paulo: Pearson Prentice Hall, 2009. p352-357

RODRIGUES, André. **Virtualização completa e paravirtualização**. 2017. Disponível em: <<https://www.portalgsti.com.br/2016/11/virtualizacao-completa-e-paravirtualizacao.html>>. Acesso em: 17 maio 2017.

WAZLAWICK, Raul. **Metodologia de pesquisa para ciência da computação**. 2a Edição ed. Rio de Janeiro, Rio de Janeiro: Elsevier Brasil, 2009. 157 p. .85-352- 6643-7.

VMWARE. **VMware vSphere 6.5 Documentation**. 2017. Disponível em: <<https://pubs.vmware.com/vsphere-65/index.jsp>>. Acesso em: 26 maio 2017.

VERAS, Manoel. **Virtualização**. Brasport, 2011.

ZARPELÃO, B. B. **Deteção de Anomalias e Geração de Alarmes em Redes de Computadores**, Trabalho de Conclusão de Curso de Ciência da Computação, Universidade Estadual de Londrina, 2004

ZABBIX SIA. **Zabbix Documentation 3.0**. 2017. Disponível em: <<https://www.zabbix.com/documentation/3.0/pt/manual>>. Acesso em: 26 maio 2017.

Parkuts, Eliésio. DUFECH, Saulo Muzzolon.. Orlovski, Regiane **VIRTUALIZAÇÃO – CONCEITOS E APLICAÇÕES**. Revista Científica Semana Acadêmica. Fortaleza, ano MMXIV, N°. 000054, 10/04/2014. Disponível em: <<https://semanaacademica.org.br/artigo/virtualizacao-conceitos-e-aplicacoes-0>> Acesso em: 25de junho 2018.

Arunkundram, Rajiv. **Uma introdução ao Hyper-V no Windows Server 2008**. 2008. Disponível em: <<http://technet.microsoft.com/pt-br/magazine/2008.10.hyperv.aspx>> Acesso em: 25 de junho 2018.

BRUMATE, Alan. **VIRTUALIZAÇÃO DE SERVIDORES: UM COMPARATIVO ENTREVMWARE E XEN**. 2010. Disponível em: <<https://pt.scribd.com/document/47189969/virtualizacao-de-Servidores-Um-Comparativo-entre-VMware-e-Xen>>. Acesso em: 25 jun. 2018.

JONES, M. Tim. **Emulação do Sistema com o QEMU: A máquina dentro da máquina.** 2007. Disponível em: <<https://www.ibm.com/developerworks/br/library/l-qemu/l-qemu-pdf.pdf>>. Acesso em: 25 jun. 2018.