Análise e comparação de Desempenho de Containers em Docker e em Apache Mesos

(Versão Definitiva Após Defesa Pública)

David Miguel Prata Ferreira

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática
(2º ciclo de estudos ou mestrado integrado)

Orientador: Prof. Doutor Mário Marques Freire

Covilhã, Janeiro de 2021

Dissertação elaborada no Instituto de Telecomunicações - Delegação da Covilhã e no Departamento de Informática da Universidade da Beira Interior e submetida à Universidade da Beira Interior para discussão em provas públicas.

Este trabalho foi financiado pela FCT/MCTES através de fundos nacionais e quando aplicável cofinanciado por fundos comunitários no âmbito do projeto UIDB/50008/2020 e foi suportado pela operação Centro-01-0145-FEDER-000019 - C4 - Centro de Competências em Cloud Computing, cofinanciada pelo Fundo Europeu de Desenvolvimento Regional (FEDER) através do Programa Operacional Regional do Centro (Centro 2020), no âmbito do Sistema de Apoio à Investigação Científica e Tecnológica - Programas Integrados de IC&DT.

Cofinanciado por:

Dedicatória

Dedico esta dissertação ao meu avô Mário Prata, por todo o apoio que me facultou durante o meu percurso, como estudante universitário.

Agradecimentos

Aos professores do Departamento de Informática da Faculdade de Engenharia da UBI pelo profissionalismo e sabedoria com que ministram as aulas, apresentando as várias áreas do curso, dando-me uma base sólida de conhecimentos.

Aos meus pais, pelo apoio, pela compreensão, pelo incentivo, permitindo que a minha formação seja um objetivo alcançado.

Ao meu irmão, pela sua amizade e por estar sempre presente nos momentos especiais da vida.

Aos meus avós pelo orgulho e carinho que manifestam por mim e por me incentivarem a prosseguir.

Ao meu orientador, professor Dr. Mário Marques Freire, que me acompanha já há algum tempo, obrigada pelo tempo disponibilizado para a elaboração desta dissertação, pela orientação, pelo apoio e pelas sugestões de melhoria.

Resumo

O conceito de virtualização está na origem do paradigma da computação em nuvem e é amplamente utilizado no mundo da tecnologia. A virtualização tem tido uma forte adesão por parte das empresas e instituições a nível mundial, nos mais diversos domínios pois esta oferece maior flexibilidade, melhor utilização de recursos, maior escalabilidade e adaptabilidade e redução de custos.

A utilização de containers inclui um conjunto de tecnologias que evoluíram a partir do espaço de virtualização e que fornecem flexibilidade na infraestrutura de gestão e nos aplicativos como é o caso do Apache Mesos e do Docker. O Apache Mesos e o Docker são dois dos principais produtos disponíveis no mercado, que fornecem níveis de capacidade para o mundo virtualizado.

Esta dissertação pretende avaliar e comparar o desempenho das infraestruturas de virtualização Apache Mesos e Docker comparando-os em relação a seu desempenho. Alicerçámos a investigação em duas partes distintas e complementares, a primeira parte é composta pela fundamentação teórica e a segunda parte consiste no desenvolvimento na instalação das ferramentas de virtualização e testes de desempenho (CPU, File I/O, memória e MySQL). A fundamentação teórica encontra-se definida no capítulo 2 – “Background da Virtualização” Procurámos, através da fundamentação teórica, demonstrar a pertinência da investigação, transmitir algumas teorias pertinentes e introduzir, sempre que possível, investigações existentes na área. A segunda parte do trabalho foi dedicada, à instalação das ferramentas de virtualização e realização de testes. Evidenciámos os métodos de instalação das ferramentas e apresentámos as suas funcionalidades. Por fim refletimos sobre os resultados, de acordo com o enquadramento teórico efetuado na primeira parte.

Palavras-chave

Virtualização, Apache Mesos, Docker e Sysbench.

Abstract

The concept of virtualization is at the origin of the cloud computing paradigm and is used in the world of technology. Virtualization has been strongly supported by companies and institutions worldwide, in the most diverse domains as it offers greater flexibility, better use of resources, greater scalability and adaptability and cost reduction.

The use of containers includes a set of technologies that have evolved from the virtualization space and that provide flexibility in the management infrastructure and in applications such as Apache Mesos and Docker. Apache Mesos and Docker are two of the main products available on the market, which provide capacity levels for the virtualized world.

This dissertation intends to evaluate and compare the performance of Apache Mesos and Docker virtualization infrastructures comparing them in relation to their performance. We based the investigation in two distinct and complementary parts, the first part is composed of the theoretical foundation and the second part consists of the development of the installation of virtualization tools and performance tests (CPU, File I/O, memory and MySQL). The theoretical foundation is defined in chapter 2 - “Background of Virtualization” We sought, through theoretical foundation, to demonstrate the pertinence of the investigation, to transmit some pertinent theories and to introduce, whenever possible, existing investigations in the area. The second part of the work was dedicated to the installation of virtualization tools and realization of tests. We highlighted the methods of installing the tools and presented their features. Finally, we reflect on the results, according to the theoretical framework carried out in the first part.

Keywords

Virtualization, Apache Mesos, Docker and Sysbench.

Índice

Lista de Figuras

Lista de Tabelas

Lista de Acrónimos

API	Application Programming Interface
CPU	Central Process Unit
CTSS	Compatible Time Sharing System
DAS	Direct Attached Storage
DMA	Direct Memory Access
FAZ	Flash All Storage
HD	High Definition
I/O	Input/output
IAAS	Infrastructure-as-a-Service
IBM	International Business Machines Corporation
JVM	Java Virtual Machine
LPAR	Logical Partitioning
LXC	LinuX Containers
MIT	Massachusetts Institute of Technology
NAS	Network Attached Storage
NAT	Network address translation
PAAS	Platform-as-a-Service
REST	Representational State Transfer
SAN	Storage Area Network
SO	Sistema Operativo
VDI	Virtual Desktop Infraestruture
VLAN	Virtual Local Area Network
VM	Virtual Machine
VMCS	Virtual Machine Control Structure
VMM	Virtual Machine Manager
VPN	Virtual Private Network
UFS	Unix File System

Capítulo 1

Introdução

1.1 Enquadramento da Dissertação

O conceito Virtualização, oferece às aplicações e serviços maior flexibilidade, melhor utilização de recursos, maior escalabilidade e adaptabilidade com redução de custos. [1] [2]. Embora date da década de 60 [3], está na origem do paradigma de computação em nuvem e é aplicado em larga escala por parte das empresas e das instituições a nível mundial, nos mais diversos projetos. Segundo o estudo de State of the Cloud Report da RightScale em 2018, 96% da população da sua amostra utiliza computação em nuvem e 81% possui multi-cloud [2] [3]. Este procedimento utiliza servidores virtuais de fácil instalação, gestão e migração e oferece um melhor desempenho [3]. A virtualização desvincula as aplicações e o sistema operativo dos recursos físicos, utiliza instrumentos lógicos e não físicos, estabiliza o ambiente e torna as aplicações independentes do hardware [4].

O conceito de virtualização está na origem do paradigma da computação em nuvem e é amplamente utilizado no mundo da tecnologia. A virtualização tem tido uma forte adesão por parte das empresas e instituições a nível mundial, nos mais diversos domínios pois esta oferece às aplicações e serviços maior flexibilidade, melhor utilização de recursos, maior escalabilidade e adaptabilidade e redução de custos.

O Docker consiste num conjunto de produtos de plataforma de serviço que utilizam virtualização ao nível do sistema operativo, para entregar software em containers e o Apache Mesos é um projeto Open Source, que gere clusters de computadores.

Esta dissertação foca-se no estudo e comparação do desempenho de infraestruturas de virtualização Docker e Apache Mesos.

1.2 Definição do Problema e Objetivos da Investigação

Nesta dissertação pretende-se investigar qual das tecnologias de containers Docker e Apache Mesos apresenta melhor desempenho.

O objetivo principal desta dissertação consiste em analisar e comparar o desempenho de Containers em Docker e em Apache Mesos, usando o benchmarck multiplataforma SysBench.

Para responder a esta questão, investigámos os conceitos de virtualização, a terminologia e suas propriedades, assim como explicitámos o hypervisor e alguns tipos de virtualização existentes. Abordámos também a virtualização baseada em containers Dockers e Apache Mesos, os overheads e os benchmarkings.

Propusemo-nos a construir um cenário de testes que permite obter a comparação de resultados entre o Apache Mesos e o Docker.

1.3 Estratégia para a Resolução do Problema

De acordo com o que nos propusemos nesta dissertação para a resolução do problema, foram utilizadas duas infraestruturas físicas, cada qual com duas máquinas virtuais, instaladas em VMware, com o sistema operativo CentOS 7 com o Apache Mesos e o Docker, instalados separadamente em cada uma das máquinas virtuais, com o objetivo de se extrair os resultados do desempenho de cada uma delas, recorrendo à linguagem de scripting bash, para automatizar a realização dos testes.

Após a realização de testes às duas infraestruturas implementadas, será elaborado um estudo comparativo do desempenho de ambas as infraestruturas, de modo a colocar em evidência os respetivos méritos.

1.4 Contribuições

A principal contribuição desta dissertação consiste na apresentação do estudo comparativo de desempenho entre estes dois sistemas de virtualização, o Apache Mesos e o Docker.
Para além do estudo de desempenho efetuado, esta dissertação permitiu também investigar conceitos de virtualização, a terminologia e suas propriedades, assim como explicitámos o hypervisor e alguns tipos de virtualização existentes. Abordámos também a virtualização baseada em containers Dockers e Apache Mesos, os Overheads e os benchmarkings.

1.5 Limitações do Trabalho Desenvolvido

Pensamos que o trabalho obtido é positivo e que se enquadra nos objetivos que nos propusemos a atingir. Este trabalho de investigação apresenta algumas limitações, consideramos que o tempo para a sua execução foi escasso. A realização dos testes poderia ter sido alargada através da execução dos testes em mais máquinas físicas com outras características para além do mais podia também beneficiar com a instalação de outras ferramentas de benchmark.

1.6 Organização do Documento

A dissertação encontra-se organizada em cinco capítulos. O assunto e a organização dos capítulos podem ser resumidos da seguinte forma:

- O Capítulo 1, descreve o enquadramento da dissertação, o problema a resolver e os objetivos da investigação subjacentes a esta dissertação, a estratégia para a resolução do problema, as principais contribuições, as limitações do trabalho desenvolvido e a organização da dissertação.

- O Capítulo 2 é dedicado ao *Background* da Virtualização e está estruturado por uma introdução à virtualização, o conceito, a terminologia e propriedades, assim como se explicita o hypervisor e alguns tipos de virtualização existentes (storage, sistemas operativos, desktops, redes, hardware). Ainda neste capítulo é abordada a virtualização

baseada em containers Dockers e Apache Mesos, os Overheads e os Benchmarkings assim como as técnicas de segurança e o trabalho relacionado o *background* teórico necessário para a compreensão da temática da dissertação.

- No Capítulo 3 são apresentados os ambientes experimentais e especificados os recursos de *hardware* e *software* utilizados na componente experimental. São expostos os procedimentos da implementação das infraestruturas, através dos detalhes da implementação do armazenamento de dados e através de todas as configurações necessárias para a implementação das infraestruturas. São ainda descritas as ferramentas utilizadas para realizar a análise de desempenho.

- O Capítulo 4 é dedicado à implementação de testes executados pela ferramenta de benchmark Sysbench onde consta a comparação experimental do desempenho das infraestruturas implementadas através de uma análise detalhada dos dados recolhidos pelas ferramentas de benchmark.

- O capítulo 5 menciona as conclusões relativas aos objetivos delineados na dissertação, determinando qual a tecnologia de virtualização que possui melhor desempenho e em que cada cenário se enquadra melhor cada tecnologia, explicitando o trabalho futuro que pode resultar a partir da elaboração da presente dissertação.

Capítulo 2

Background da Virtualização

2.1 Introdução

A virtualização é uma técnica que combina ou divide recursos para prover um ou mais ambientes operativos de execução [5]. É um conceito que data de 1959 com a publicação do artigo "Time sharing processing in large fast computers" por Christopher Strachey, na Conferência Internacional de Processamento de Informação, em Nova York, onde expõe a multiprogramação partilhada, por máquinas de grande porte. [4] Baseado no trabalho deste autor, o MIT desenvolve o padrão Compatible Time Sharing System (CTSS), com a qual a IBM insere o conceito de multiprocessamento nos mainframes, permitindo a várias CPUs trabalhem como uma só e introduzindo a conceção de memória virtual (virtual storage) como parte do Sistema Operativo (SO). [4] Este procedimento antecipou o conceito de virtualização, com a abstração e o mapeamento da memória real para memória virtual, além da especificação de espaços de endereçamento, utilizados por diferentes programas [4].

A virtualização é utilizada para superar as limitações físicas do hardware e o seu benefício máximo reside na habilidade em simular hardware através de software, parcial ou total, permitindo executar vários serviços, programas, sistemas operativos a partir de um equipamento físico [6]. A combinação de recursos computacionais e vários ambientes operativos, permite uma utilização mais ampla da infraestrutura, sem limitações do hardware de execução [5]. Este conceito impulsionou a tecnologia de *Data Center* com vários sistemas operativos e aplicações instalados dentro dos mesmos equipamentos, com recursos fornecidos aos utilizadores através de máquinas virtuais ágeis e escaláveis de forma eficaz e inteligente [7].

Este capítulo apresenta o conceito de virtualização, as diferentes arquiteturas, as diferentes técnicas de virtualização existentes, as tecnologias de virtualização overheads, ferramentas de benchmarking e comparação ao nível de segurança das arquiteturas de virtualização.

2.2 Conceito de Virtualização

Virtualizar significa abstrair os recursos computacionais para outra localização do computador, ou seja, simular o ambiente em que a aplicação é executada para que outros sistemas, aplicações, e utilizadores possam interagir com estes recursos. [8] A camada de abstração dos recursos computacionais [8] que fornece um hardware virtual para cada sistema, permite excluir as caraterísticas físicas e o modo como os sistemas operativos e as aplicações interagem com os recursos computacionais e partilhar os recursos de um mesmo servidor físico entre diferentes aplicações executadas em contextos de processamento isolados, pertencentes aos servidores virtuais [8]. Este conceito é também denominado de LPAR – Logical Partitioning por permitir a divisão de um servidor em várias partições virtuais independentes [8].

Podemos referir que a virtualização tem a sua origem com a IBM que implementou e desenvolveu a terminologia de máquinas virtuais, mais propriamente o sistema operativo experimental M44/44X, a primeira máquina virtual, que permite que a divisão de um único computador em vários [8]; [9]; [10], suprindo as limitações físicas do hardware. Inicialmente, utilizou-se uma técnica de implementação que permitia que os binários de um processador fossem interpretados e substituídos por código equivalente de outro processador ou seja, através da adoção do conceito de máquina virtual de processo, uma aplicação executava sobre um sistema operativo (SO) A e emulava o comportamento de um sistema operativo (SO) B, as aplicações desenvolvidas para B executavam sobre A (figura 1.1)[4].

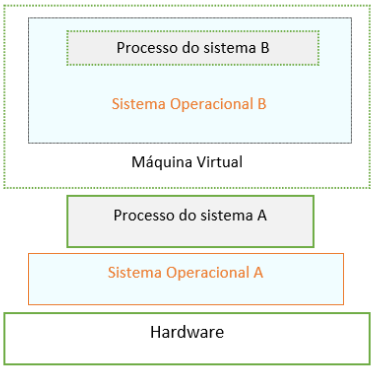

Figura 2.1 – Máquina virtual de processo (figura redesenhada a partir de [4]).

Como esta técnica apresentava desvantagens ao nível do desempenho pela execução em modo de utilizador sendo a tradução de um sistema para o outro, mas também por desperdício de capacidades do hardware físico pois as máquinas virtuais de processo oferecem dispositivos de I/O genéricos e simples [4], surgiram os monitores de máquinas virtuais, os Virtual Monitor Machine (VMM) também conhecidos como hypervisors, implementados como uma camada de software entre o hardware e o Sistema Operativo, oferecendo uma máquina virtual para o Sistema Operativo (figura 1.2).

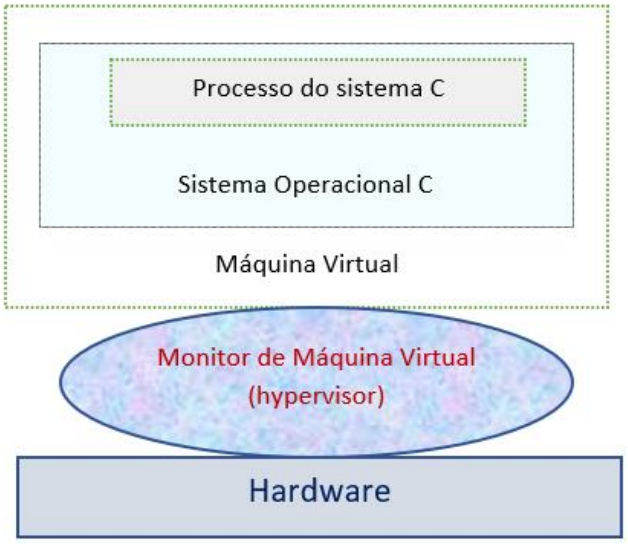

Figura 2.2 – Monitor de máquina virtual – hypervisor (figura redesenhada a partir de [4]).

Em 1980, a virtualização sofre um decréscimo com a popularização dos computadores pessoais e a redução dos sistemas operativos (Unix, Macintosh e Microsoft) [4] para regressar em 1990 e atender à tecnologia Java Virtual Machine (JVM), ao aumento do poder computacional dos processadores, à capacidade de memória dos computadores, à disseminação de sistemas distribuídos e da própria internet e do modelo de serviços baseado numa arquitetura cliente-servidor (caríssimo). Fica então o período de 1998 marcado pelo surgimento da VMware, criada por Diana Greene e por Mendel Rosenblun que desenvolvem o primeiro hypervisor que permite a virtualização de servidores em plataformas x86 e em 2005 a Microsoft lança o Microsoft Virtual Server, o seu primeiro produto com foco na virtualização de servidores. [4]

A investigação destaca benefícios neste processo, múltiplas máquinas virtuais, com sistemas operativos heterogéneos, correm em simultâneo de forma isolada, na mesma máquina física, garantindo a partilha de recursos computacionais por vários utilizadores. [4]; [11]. A emulação do sistema de hardware, desde o processador até à placa de rede, admite que cada máquina virtual partilhe um conjunto de hardware, sem conhecimento mútuo pelas máquinas virtuais, estas apenas observam um conjunto consistente e normalizado de hardware, independentemente do hardware físico [12]. Assim a virtualização admite que os workloads executem eficazmente apenas numa máquina física garantindo isolamento e controlo de recursos, possibilitando aos servidores que sejam alocados numa única máquina, através do uso de máquinas virtuais ou de containers [13].

O termo virtualização surge como forma de destacar os recursos lógicos dos recursos físicos, tal como se utiliza o termo memória virtual na arquitetura dos sistemas operativos já que os programas (*software*) beneficiam de mais recursos do que se estivessem instalados fisicamente, podendo obter maior capacidade de armazenamento em memória principal com significativas vantagens no seu tamanho máximo e no volume de dados que conseguem processar. [14]

O processo de virtualização acarreta vantagens na redução de custos, mas também potencia a agilidade técnica para prover novos servidores, através da criação de processos simplificados de alta disponibilidade e de continuidade, sem causar indisponibilidade do serviço.[15]

2.2.1 Terminologia de Virtualização

Todas as definições apresentadas acerca da temática da virtualização referem uma camada de abstração dos recursos de uma máquina real, que fornece um hardware virtual para cada sistema para abstrair as caraterísticas físicas e o modo como os sistemas operativos e as aplicações interagem com os recursos computacionais, no entanto a virtualização implica o conhecimento de alguns conceitos que fazem parte integrante deste ambiente e que passamos a explicitar.

- A Virtualization Technology é necessária para trabalhar em full virtualization, refere-se ao hardware que suporta a tecnologia de virtualização, como Intel

Virtualization ou AMD Virtualization. Os sistemas Hardware virtualization trabalham em conjunto com o software de virtualização de modo a permitirem a virtualização completa dos dispositivos de hardware, possibilitando a execução de um sistema operativo sem alterações. [14]

- O Standard Computer refere-se a um hardware que não possui suporte especial da tecnologia de virtualização e não consegue executar sistemas operativos que requeiram o modo full virtualization. [14]

- O Host Operating System e Guest Operating System em que o Sistema operativo host (Host Operating System) se refere ao sistema operativo nativo da máquina na qual ocorre a virtualização, é o sistema operativo que é executado diretamente sobre o hardware físico e o sistema operativo Guest (Guest Operating System), ao sistema operativo que é executado sobre o hardware virtualizado, isto é, o sistema operativo que é executado na máquina virtual. Uma máquina na qual é feita a virtualização pode contar com apenas um SO host, mas podem ser executados diversos SOs guest simultaneamente.

- A Virtual Machine Monitor (VMM), ou seja, Monitor de Máquina Virtual, também conhecido por Hypervisor [16] pode ser considerado como um sistema operativo, responsável por controlar os recursos de hardware subjacente e torná-los disponíveis para cada máquina virtual (VM). Esta máquina virtual pode executar qualquer tipo de *software* como um servidor, um cliente ou um *desktop*; é um componente de software que hospeda as máquinas virtuais [6], é a parte central de qualquer solução de virtualização, implementado sobre software ou diretamente sobre o hardware e permite que diversos sistemas operativos possam executar numa única máquina física. O VMM é responsável pela virtualização e controle dos recursos partilhados pelas máquinas virtuais, tais como, processadores, dispositivos de entrada e saída, memória, armazenagem. Também é função do VMM escalonar a máquina virtual que vai executar a cada momento, semelhante ao escalonador de processos ao nível do Sistema operativo [4]. O VMM é executado no modo de supervisor, no entanto as máquinas virtuais são executadas em modo de utilizador. Como as máquinas virtuais são executadas em modo de utilizador, quando estas tentam executar uma instrução privilegiada, é gerada uma interrupção e o VMM encarrega-se de emular a execução desta instrução. O Virtual Machine Server (VMServer) refere-se ao hardware físico e ao software de virtualização que, quando conciliados, formam o virtual host. É o principal componente da virtualização,

pois é o responsável por criar e distribuir os recursos necessários para a criação de várias máquinas virtuais. Para além de host, é também designada por dom0, virtual0 ou domínio privilegiado. Em relação à tecnologia de virtualização (Virtualization Technology) esta refere-se ao hardware que suporta a tecnologia (Intel Virtualization ou AMD Virtualization). Os sistemas Hardware Virtuaization trabalham em conjunto com o software de virtualização de modo a permitirem a virtualização completa dos dispositivos de hardware, possibilitando a execução de um sistema operativo sem alterações. [14]

2.2.2 Propriedades da Virtualização

As propriedades da virtualização são consideradas de modo distinto de acordo com os aspetos mais relevantes do sistema que realçam. Alguns autores defendem [17] que as propriedades da virtualização são o isolamento, o controlo, a interposição, a eficiência, a gestão e a compatibilidade de software, outros [18] além das referidas, acrescentam o encapsulamento e o desempenho. Alguns, no entanto [19] apresentam apenas três propriedades, o particionamento; o isolamento e o encapsulamento. Apresentamos a descrição de cada uma das propriedades, tal como indicado pelo autor:

- Particionamento – partilha do *hardware* físico implementada através do hypervisor. O hypervisor permite a criação de máquinas virtuais com o seu respetivo *hardware* sobre o *hardware* físico.

- Isolamento – Quando um processo é executado numa máquina virtual, não é possível a interferência com outra máquina virtual. Caso existe alguma quebra de segurança ou falha as restantes máquinas virtuais e as físicas, restam isoladas e protegidas e garantem uma execução normal.

- Encapsulamento – O encapsulamento refere-se aos arquivos, à forma virtual do hardware, ao sistema operativo e às aplicações da máquina virtual, permitindo o seu transporte de forma simples através de discos externos, pendrives, CD's ou DVD's.

- Desempenho – a virtualização implica a inserção de uma nova camada de *software*, que pode afetar, de certa forma, o desempenho do SO e não existem resultado que permitam medir o desempenho dos ambientes virtualizados.

- Gestão – a gestão das máquinas virtuais é efetuada de forma independente;

- Compatibilidade de software – o software escrito para uma determinada plataforma é executado numa VM que virtualiza essa mesma plataforma;

- Eficiência – as instruções que não comprometam o host são executadas diretamente no hardware;

- Controlar – quando se procede à virtualização o VMM controla e tem acesso a todas as informações sobre os processos em execução nas suas VM;

- Interposição – existe a possibilidade do VMM poder intercalar ou acrescentar instruções em determinadas operações nas máquinas virtuais.

2.3 Virtualização Baseada em Hypervisors

A virtualização é um termo muito abrangente que otimiza o investimento para além de se adequar de forma mais eficaz às necessidades dos utilizadores [20]. A sua implementação implica vantagens inúmeras das quais fazem parte a libertação de espaço físico, um menor consumo de energia já que existe um menor número de servidores e desktops físicos, uma maior facilidade de gestão por parte das VMM´s, mas também maior desempenho e aproveitamento dos recursos da máquina hospedeira, consolidação de aplicações, de servidores ou até mesmo migração entre ambientes, sejam executados de forma segura sem provocar qualquer tipo de risco para o *host* e segurança já que podem ser criados servidores virtuais para cada tipo de aplicação, assegurando que os restantes computadores não são afetados, caso um servidor apresente algum tipo de problema. Para além das inúmeras vantagens apresentadas engloba diversas tecnologias.

A virtualização permite uma utilização mais eficaz ao nível do hardware, disponibilizando recursos de forma mais flexível. Diversos autores dividem em várias categorias, as tecnologias da virtualização [21], [22], [23].

O investigador Waters [21] aponta três categorias básicas, a virtualização de aplicações, a virtualização dos meios de armazenamento (storage) e a virtualização de sistemas operativos (virtualização de servidores). Já Murphy [22] divide as tecnologias de virtualização em oito categorias, a virtualização de sistemas operativos, a virtualização de servidor de aplicação, a virtualização de aplicação, a virtualização de gestão, a virtualização de rede, a virtualização de hardware, a virtualização de armazenagem

(storage) e a virtualização do serviço. O autor Kusnetzky [23], refere que a virtualização pode ser dividida nas seguintes camadas de virtualização, Acesso (Access virtualization), Aplicativo (Application virtualization), Processamento (Processing Virtualization), Armazenamento (Storage virtualization) e Rede (Network Virtualization). Mário Monginho, M. [14], exibe quatro camadas de virtualização, a virtualização de aplicações, a virtualização de storage, a virtualização de sistemas operativos e a virtualização de hardware. Por fim Veras, M., [4], exibe cinco camadas de virtualização, a virtualização de servidores, a virtualização de desktops, a virtualização do armazenamento, a virtualização das aplicações e a virtualização de redes.

2.3.1 Definição e Tipos de Hypervisors

O Hypervisor (Virtual Machine Monitor - VMM) refere-se a um ambiente de máquina virtual que é criado por um monitor de máquinas virtuais, também denominado "sistema operativo para sistemas operativos" [16]. É uma camada de software que se encontra entre o host e as VMs e que isola o sistema operativo e as aplicações dos seus recursos físicos. O Hypervisor tem o seu próprio kernel, corre diretamente sobre o hardware do host e cria a ilusão de que cada sistema guest tem um hardware exclusivo para ele. Os hypervisors *têm como* característica ser o mais minimalista e leves possível para alcançar bons níveis de eficiência e de segurança, potenciar isolamento entre as VMs e gerir os recursos de hardware. Para fornecer isolamento e minimizar o erro no software o VMM recorre ao Memory Management Unit (MMU) bem como a outras unidades de hardware. O hypervisor tem de gerir o balanceamento de carga de CPU, mapear os endereços físicos para endereços de memória lógicos, migrar VMs entre sistemas físicos, enquanto protege a integridade de cada VM e protege a estabilidade de todo o sistema. Como os recursos de *hardware* são diretamente controlados pelo monitor de máquina virtual e não pelo sistema operativo, cada sistema tem a ilusão de executar independentemente no computador quando na realidade está a partilhar os recursos [4]. Assim é possível executar múltiplos sistemas operativos em paralelo no mesmo hardware [4], [24], [25].

Os hypervisors são classificados em dois tipos [4]:

- Tipo I (bare metal, nativo ou supervisor) que executa diretamente no hardware do servidor. Controla o hardware e o acesso do Sistema Operativo convidado (guest OS). O papel do hypervisor nativo é compartilhar os recursos de hardware entre as máquinas virtuais, de forma que cada uma delas imagina ter recursos exclusivos. O primeiro executa a máquina virtual no próprio hardware como se fosse um sistema operativo (Exemplos de softwares: Oracle OVM para SPARC, ESXi, Hyper-V e KVM) [3].

- Tipo II (host ou hosted hypervisor) uma aplicação que fornece um ambiente de execução para outras aplicações. Executa sob um Sistema Operativo nativo como se fosse um processo deste. A camada de virtualização é composta por um SO hóspede e um hardware virtual, que são criados sobre os recursos de hardware oferecidos por meio do SO nativo [4]. O segundo é executado em um sistema operativo convencional (SO), assim como outros programas de computador (Exemplos de softwares: VMware Fusion, Oracle Virtual Box, Oracle VM para x86, Solaris Zones, Parallels e VMware Workstation) [3].

O Hypervisor é o software capaz de criar o ambiente de virtual, através dele define-se o recurso do hardware utilizado para a virtualização, a máquina virtual.

A virtualização trabalha sobre os seguintes conceitos: Host, Hypervisor, VM (máquina virtual) e Guest. O Host consiste no hardware físico como hard disk, memória RAM, processamento ou placas gráficas que serão utilizados para se estabelecer o ambiente virtualizado. O Guest é o sistema operativo (Windows, Linux ou outro) que será executado na máquina virtual criada pelo host de virtualização.

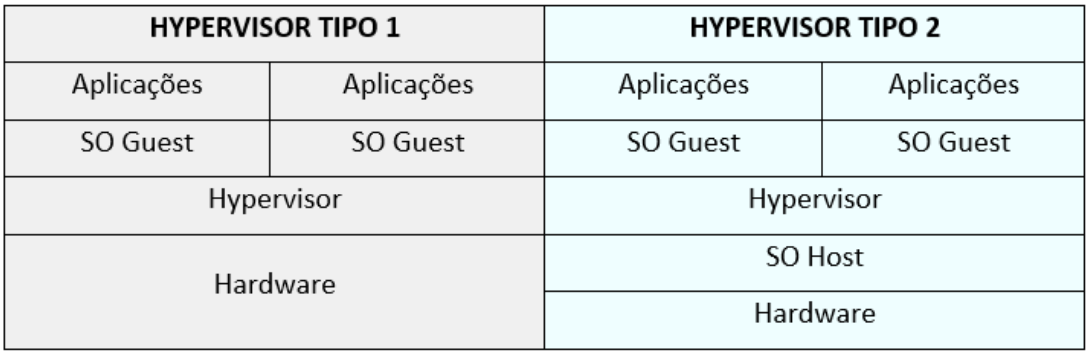

Figura 2.3 – Tipos de Hypervisor (figura redesenhada a partir de [4]).

2.3.2 Virtualização de Aplicações

A virtualização de aplicações permite que as aplicações sejam executadas num ambiente no qual não foram desenvolvidas para funcionar nativamente, sem necessidade de instalação. A virtualização de aplicações consiste na emulação do sistema operativo, bibliotecas e drivers por meio de um software instalado entre o realmente utilizado na máquina e a aplicação virtualizada. A camada de virtualização possui apenas os recursos necessários para a aplicação funcionar, substituindo parte do ambiente de tempo de execução fornecidos pelo sistema operativo. Esta virtualização pode ser feita através do container que mantém todas as bibliotecas necessárias para a execução da aplicação, permitindo assim que possa ser executada em diferentes máquinas, evitando que seja necessária uma nova máquina, na qual seria instalado todo o ambiente necessário para o funcionamento da aplicação, incorrendo em mais custos e manutenção. Assim a virtualização de aplicações é disponibilizada criando um ambiente virtual onde são disponibilizados os recursos necessários à execução da aplicação. [26]

O isolamento provido é denominado de sandboxing. A virtualização de aplicações abrange a tecnologia de software, permitindo execução de aplicações em distintos sistemas operativos e distintas plataformas de hardware. Permite disponibilizar aplicações sem necessidade de as instalar diretamente, graças a um processo denominado sequenciamento de aplicativo, que permite que cada aplicativo execute o seu próprio ambiente virtual de forma independente. Os aplicativos sequenciados são isolados uns dos outros, eliminando conflitos entre aplicativos [23] [4]. Na virtualização de aplicações destaca-se a Virtual Desktop Infraestruture (VDI) em que a VM armazena as aplicações e os sistemas operativos, permitindo ao utilizador aceder a partir somente de um monitor e de um sistema onde possa ligar o teclado e o rato.

A virtualização de aplicações apresenta vários benefícios dos quais referimos a garantia de compatibilidade pois não é necessária a instalação do aplicativo, porque ele estará empacotado com todos os arquivos necessários ao seu funcionamento, podendo ser executado num ambiente comum, existe uma redução da necessidade de espaço físico, uma vez que todos os aplicativos podem ser aproveitados sem a necessidade de instalação em qualquer máquina da empresa, existirão menos ambientes a serem tratados, melhorando assim o controle sobre a segurança da informação, uma redução significativa na necessidade de manutenção de ambientes e custo de novos

equipamentos, visto que não são necessários grandes investimentos em hardware sendo possível redirecionar recursos computacionais para outra atividade a qualquer momento com alta disponibilidade (High Avaliability), mas também de servidores e dispositivos, como a Citrix que possui ferramentas de virtualização como a XenApp e a XenDesktop, a Microsoft que dispensa apresentações, a Azure, que é um datacenter poderoso com foco totalmente voltado para soluções virtualizadas e cloud computing e a VMware que se destaca no setor de soluções para aplicativos virtualizados. [26]

2.3.3 Virtualização de Storage

A virtualização de storage é efetuada através da introdução de um componente (appliance) que permite às unidades heterogéneas de armazenamento (discos físicos) serem vistas como um conjunto homogéneo de recursos. [4]

A virtualização do armazenamento é um conjunto de opções de armazenamento de diferentes dimensões, de distintos fabricantes e modelos agrupados pela virtualização. A virtualização do armazenamento (storage) consiste em aumentar a capacidade de armazenamento do hardware, através da aquisição de storage, como um recurso abstraído que pode ser agrupado e partilhado. A virtualização nos meios de armazenamento de dados possibilita que a informação gravada em diferentes storages possa ser partilhada, replicada, movida e gerida, de uma forma fiável e segura.

A virtualização de storage deriva da necessidade de capacidade de armazenamento de dados cada vez maior, possibilitando o agrupamento de vários dispositivos de armazenamento de rede, num dispositivo de armazenamento gerenciado, a partir de um console central. Os benefícios da virtualização de storage, refletem-se em taxas de adoção cada vez maiores [27].

O storage pode ser implementado como servidor de arquivos, fazer backup ou ser uma área para centralizar, processar ou compartilhar dados. Podem ser classificados como DAS, NAS, SAN e FAZ e diferenciam-se pela forma de conexão com seu computador host ou servidor com diferentes finalidades e recursos:

- As Direct Attached Storage ou DAS são unidades de armazenamento conectadas diretamente a computadores, sejam servidores, notebooks ou estações de trabalho. Ao ser instalado, em que o equipamento se torna um disco adicional de seu host, pen-

drives ou hard disks baseados em portas USB até grandes sistemas baseados em portas minis SAS ou Fibre channel que equipam datacenters [27].

- O Network Attached Storage ou NAS é um sistema de armazenamento conectado diretamente na rede local, funcionando como um hard disk de rede que centraliza e armazena os dados de forma organizada. Com este sistema é possível partilhar e gerir as informações armazenadas por todos os utilizadores de forma racional, utilizando serviços de segurança como acesso protegido por login e senha e registo de atividade.

- A Storage Area Network (SAN) é uma rede de armazenamento dedicada composta por servidores e storages, interligados através de conexões IP (iSCSI) ou Fibre Channel (FC), centra e melhora a gestão das informações, com segurança e velocidade no acesso aos dados [27].

- A Flash All Storage ou FAZ é a solução de armazenamento de maior performance disponível atualmente com o armazenamento total em memória flash e SSD. São storages desenvolvidos para trabalhar diretamente com SSD sem o uso de HDD convencionais com discos rotacionando e partes mecânicas. São direcionados para ambientes que exigem alto número de acesso simultâneo, como banco de dados e máquinas virtuais [27].

2.3.4 Virtualização de Sistemas Operativos

A virtualização de servidores possibilita a execução de vários sistemas operativos diferentes como convidados dentro de um único host de servidor físico. São as denominadas máquinas virtuais (VMs) que executam uma imitação virtual do hardware de servidor. Essa tecnologia representa uma das maneiras mais eficiente de reduzir os custos de infraestrutura pois pode ser aplicada em servidores, em redes, em aplicações e em Data Centers. [4]

A virtualização de sistemas operativos (servidores) permite que numa máquina física exista uma única VM ou mais do que uma. Uma VM é, por norma, um ambiente criado por *software*, dentro do sistema operativo *host*.

A virtualização de sistemas operativos pode ser efetuada por emulação/simulação, virtualização total, paravirtualização, virtualização ao nível do sistema operativo,

No que diz respeito à emulação ou simulação, alguns autores defendem que a emulação não pode ser considerada virtualização [17], já que se recorre a um sistema operativo host. Nesta técnica de virtualização, o software é denominado VMM e é visto como uma aplicação pelo host. Um emulador permite que o computador opere como se fosse um tipo diferente de computador [28], uma forma precisa de simulação que imita exatamente o comportamento ou as circunstâncias que se estão simulando. Tem como função simular todas as operações de acesso ao *hardware* que o *host* controla, possibilitando ao sistema operativo que corra num processador central completamente distinto do *hardware* nativo como o O Qemu que corre sobre o Linux ou o Virtual PC sobre o Windows.

A figura 1.4 demonstra esta técnica, que se serve de um sistema operativo *host* que reconhece o VMM como sendo uma aplicação.

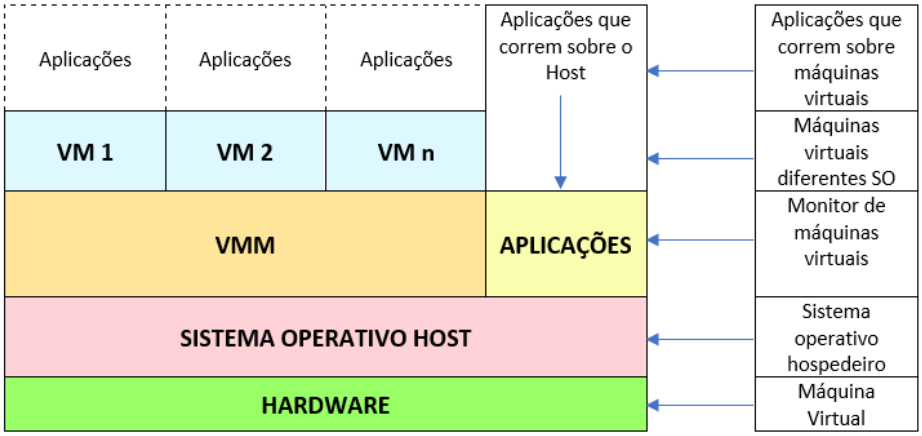

Figura 2.4 – Virtualização de servidores ou desktops utilizando emulação (figura redesenhada a partir de [9]).

- Virtualização nativa ou total

São exemplos de virtualização nativa/completa os aplicativos VMware *Workstation*, IBM VM e Parallels.

A virtualização completa é um modelo de software hypervisor (virtual machine manager) responsável por prover os ambientes virtuais de execução, emulando para os

sistemas operativos das máquinas virtuais hóspedes (VMs) os dispositivos de hardware físicos. A virtualização completa fornece uma réplica virtual do

hardware subjacente de forma a que o sistema operativo e as aplicações do sistema operativo hóspede possam executar diretamente sobre o hardware original [17].

Os sistemas operativos das VMs desconhecem que as suas instruções não são executadas diretamente no hardware físico, pois, antes da execução, as requisições de instruções privilegiadas são intercetadas e traduzidas pelo *hypervisor* [15]. *As* Instruções não privilegiadas, executadas ao nível do utilizador, são enviadas diretamente para processamento no hardware físico [29].

Este tipo de virtualização proporciona um ambiente isolado e tem como vantagem não ser necessária a alteração ao nível do sistema operativo hóspede para ser executada sobre a VMM, as máquinas virtuais podem executar em diferentes arquiteturas de processador, desde que o hypervisor ofereça este suporte. Existem, inconvenientes como a incompatibilidade e a queda no desempenho, que podem resultar em perda de desempenho em razão da tradução das instruções efetuadas pelo hypervisor antes da execução no processador físico. [15] [4]

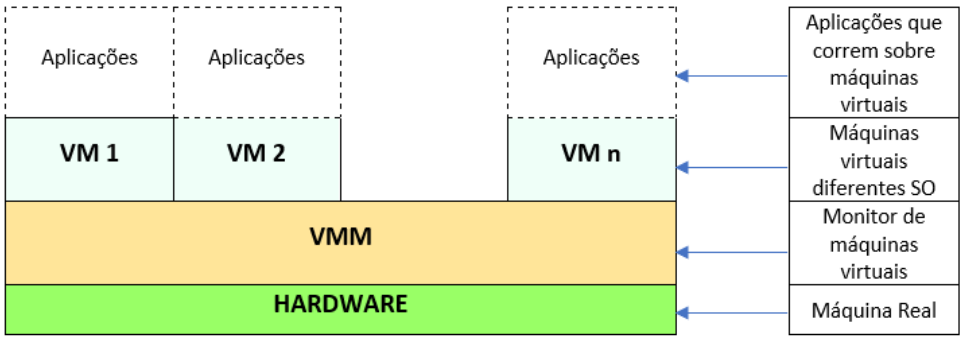

Figura 2.5 – Virtualização total ou nativa (figura redesenhada a partir de [9]).

- Paravirtualização

A VMware ESX Server e a Citrix Xen utilizam essa tecnologia.

Da mesma forma que no modelo completamente virtualizado por tradução binária, na paravirtualização um software hypervisor (VMM) é responsável por prover os ambientes virtuais de execução, emulando para os sistemas operativos

das máquinas virtuais hóspedes (VMs) os dispositivos de hardware físicos [17] Porém a paravirtualização envolve a modificação ao nível do sistema operativo convidado para substituir as chamadas de instruções privilegiadas do kernel por hypercalls (chamadas de hypervisor) o que reduz a sobrecarga causada pela tradução de instruções do modelo de virtualização completa por tradução binária, mas torna o sistema operativo das VMs fortemente acoplado ao hypervisor. [15]. As instruções não privilegiadas, são enviadas diretamente para processamento no hardware [29]. É um tipo de virtualização que possui uma camada mais leve de software entre o hardware host e o sistema operativo convidado cujo desempenho é mais próximo de um hardware nativo, neste modelo o primeiro servidor virtual definido possui acesso privilegiado ao hardware e é responsável por gerir os outros servidores virtuais e os *drivers* dos dispositivos do computador.[15]

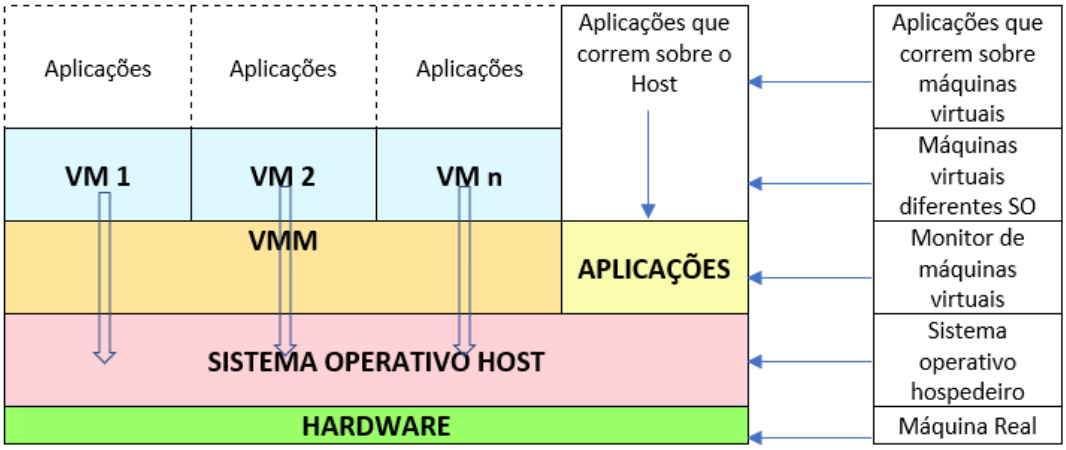

Figura 2.6 – Primeira forma de paravirtualização de servidores (figura redesenhada a partir de [9]).

Na segunda forma de paravirtualização, são efetuadas alterações no sistema operativo host e no VMM, o hardware é acedido diretamente pela VM, como é demonstrado na figura 1.7.

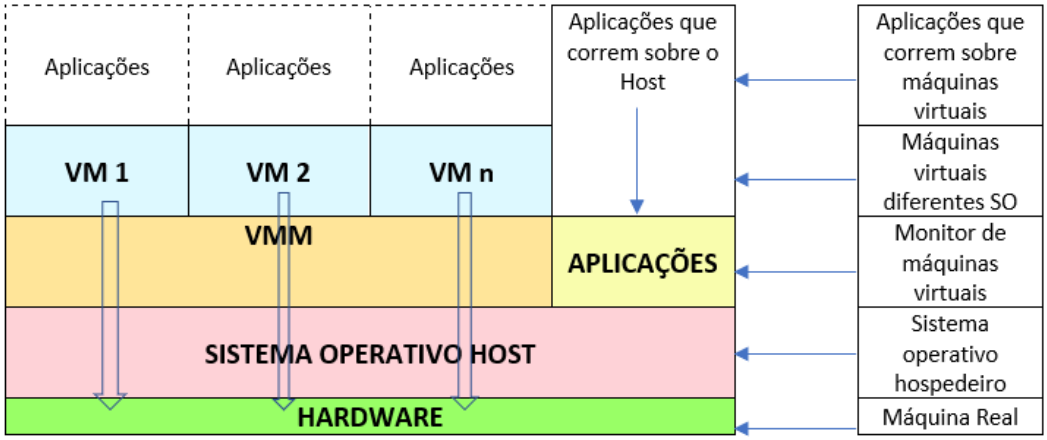

Figura 2.7 – Segunda forma de paravirtualização de servidores (figura redesenhada a partir de [9]).

Na terceira forma de paravirtualização, o acesso é facilitado através da instalação de um *drive* de dispositivo específico no sistema operativo host e o hardware é acedido diretamente pelo VMM.

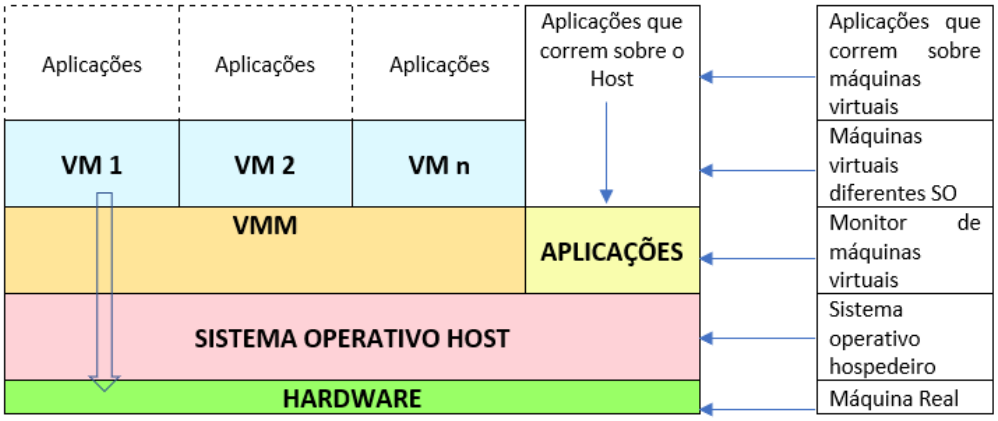

Figura 2.8 - Terceira forma de paravirtualização de servidores (figura redesenhada a partir de [9]).

2.3.5 Virtualização de Desktops

A virtualização de desktops é um processo em que o sistema do utilizador é migrado para um ambiente virtual que é responsável por executar todas as características de uma interface de rede, um processador, um conjunto de memórias RAM e outros elementos que fazem parte de uma máquina real, simulando o funcionamento de um computador real. As aplicações de desktop também podem executar num datacenter, sob a forma de máquinas virtuais, oferecendo maior confiabilidade e otimização do uso de espaço em disco com a consolidação do armazenamento e flexibilidade na escolha do sistema operativo [4].

Na virtualização de desktops, o desktop com sistema operativo é apresentado como tendo vários recursos através de várias máquinas virtuais individuais. [14]

Trata-se da configuração dos desktops dos utilizadores finais numa infraestrutura centralizada virtual em que as aplicações de desktop passam a executar num datacenter, sob a forma de máquinas virtuais. O conceito de Virtual Desktop Infrastructure (VDI), permite a montagem dinâmica de desktops e oferece maior confiabilidade e otimização do uso de espaço em disco com a consolidação do armazenamento e a flexibilidade na escolha do Sistema Operativo. [4]

2.3.6 Virtualização de Redes

A virtualização de redes proporciona a criação de redes lógicas sobre uma única infraestrutura partilhada de rede física. Os ambientes lógicos são criados sobre uma única infraestrutura compartilhada de rede e cada rede lógica fornece ao grupo de utilizadores correspondente serviços de rede, semelhantes aos utilizados por uma rede tradicional não virtualizada. A experiência da perspetiva do utilizador final é a de ter acesso a uma rede própria, com recursos dedicados e políticas de segurança independentes. Como as diversas redes lógicas compartilham uma infraestrutura comum, centralizada e com um

conjunto de equipamentos e servidores, os grupos de utilizadores podem colaborar com maior flexibilidade e capacidade de gerenciamento.

Existem várias formas de virtualização, entre as quais a Network Virtualization, através de Virtual Local Area Network (VLAN) com o objetivo de segmentar a rede e o tráfico e a Virtual Private Networks (VPN) que estabelece ligação entre a empresa e os utilizadores, as entidades e/ou fornecedores de Cloud, permitindo que as aplicações operem em modo seguro. [4]

2.3.7 Virtualização de Hardware

Na virtualização de hardware, o hardware da máquina host fornece os recursos, aproveita a capacidade física dos servidores e dos desktops atuais e torná-os mais rentáveis, executando vários sistemas operativos, onde cada um dispõe das suas aplicações utilizando um único desktop [17]. Ao fornecer várias VM's de uma vez, permite que sejam executados em simultâneo vários sistemas operativos numa única máquina virtual, tal como O VMware, o Virtual PC ou o Virtual Box.

A virtualização de hardware, a máquina host necessita de ser adaptada, as adaptações podem ser a alocação de processadores, de memória, de discos e de interface para uma determinada partição. Esta técnica tem a capacidade de simular o hardware de um desktop dentro doutro desktop, e desta forma dá suporte para que outros sistemas operativos, sejam instalados neste desktop. Ao nível da rede e, principalmente de aplicativos, as VM's são reconhecidas como desktop reais e independentes, mesmo partilhando o mesmo hardware base.

A virtualização por hardware permite a execução de múltiplos sistemas operativos paralelamente num mesmo processador físico, sem necessitar dos processos criados nos modelos de virtualização completa por tradução binária ou de paravirtualização.

São as tecnologias VT-x da Intel (arquiteturas IA-32 e Intel 64) e AMD-V da AMD que criam uma camada de abstração para controlar o acesso privilegiado ao estado dos registos do processador. Para controlar as transições entre os

modos de operação foi criada uma estrutura de memória chamada VMCS (virtual machine control structure) que preserva o contexto de execução de cada VM e VMM (virtual machine manager) no hardware, durante as transições de estado ocorridas entre cada máquina virtual e o hypervisor [30]. É o formato mais comum e possui um nível de isolamento e de desempenho bastante eficaz. É comumente denominado de hypervisor, em que o gerente de máquinas virtuais funciona diretamente sobre o hardware. Apesar de o hypervisor também poder ser considerado um sistema operativo, este executa apenas funções específicas necessárias para o controle de acesso ao hardware pelas máquinas virtuais, eliminando grande parte da sobrecarga de um sistema operativo completo. Além dos recursos de virtualização do processador, os fabricantes de hardware desenvolveram tecnologias de virtualização de I/O, para aumentar o desempenho e a segurança das máquinas virtuais, reduzindo a sobrecarga dos hypervisor em emular os dispositivos de I/O e dispensando o uso de paravirtualização. Deste modo as máquinas virtuais fazem acesso aos dispositivos de I/O, por meio do periférico de DMA (Direct Memory Access), sem a intervenção do hypervisor. Esta tecnologia também pode ser operada diretamente por um SO [30].

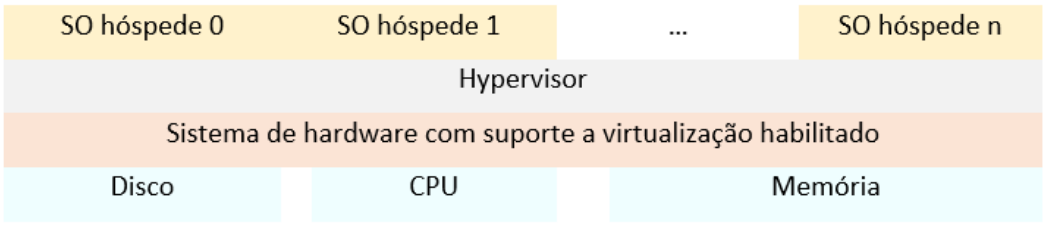

Figura 2.9 – Virtualização nativa ou auxiliada por hardware (figura redesenhada a partir de [15]).

2.4 Virtualização Baseada em Containers

2.4.1 Virtualização ao Nível do Sistema Operativo

A virtualização ao nível do sistema operativo é um modelo utilizado por plataformas Platformas-a-Service (PaaS) e Infrastructure-as-a-Service (IaaS) de forma a

providenciar um ambiente isolado, seguro e escalável para os utilizadores proporcionado por duas abordagens principais: container e hypervisor [31].

A virtualização ao nível de sistema operativo é a base de tecnologia dos sistemas de virtualização Solaris Containers, BSD Jails e Linux Vserver.

A virtualização ao nível do sistema operativo é efetuada através da execução, pelo sistema anfitrião, de um único núcleo do sistema operativo e do controlo de funcionalidade do sistema operativo hóspede, o kernel do sistema operativo permite a criação de VM's isoladas e seguras num mesmo servidor real, sem a utilização de hypervisors, pois é parte do sistema operativo host, que executa todas as funções de um hypervisor de virtualização [32].

A virtualização ao nível do sistema operativo obriga a que o núcleo dos sistemas operativos das VM's e do sistema host sejam idênticos. Assim sendo, não é possível a existência de um servidor Linux e Windows no mesmo servidor físico.

Para a utilização desta técnica, é instalada a camada de software de virtualização em cima do SO e todos os sistemas host correm sobre esta camada, utilizando o mesmo SO que o anfitrião, mas tendo cada sistema host os seus próprios recursos e funcionando independente e de forma isolada dos restantes.

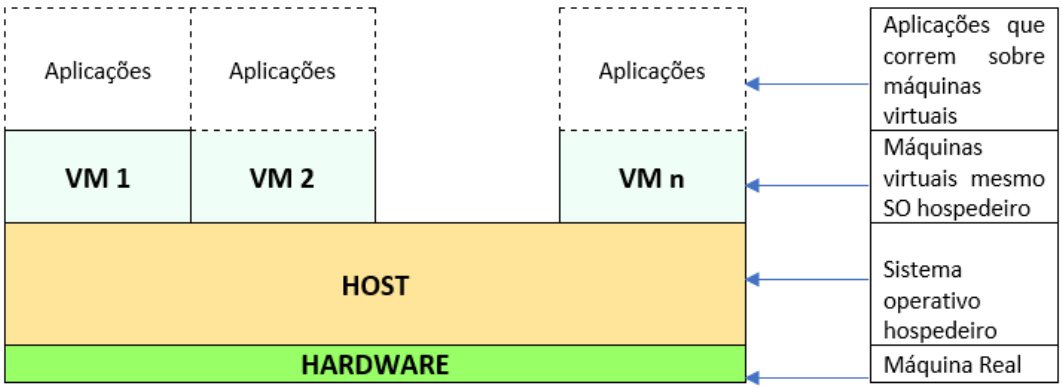

Figura 2.10 – Virtualização de servidores ao nível do sistema operativo (figura redesenhada a partir de [9]).

A virtualização ao nível do sistema operativo permite a criação de múltiplos ambientes aplicacionais isolados, chamados containers, que referenciam um mesmo kernel. Os containers são capazes de um desempenho quase nativo e podem apresentar melhor desempenho que os sistemas baseados em virtual machine em quase todos os aspetos

[33], evitando a sobrecarga de camadas de abstração. A abordagem container obteve uma considerável adoção [34] pois a sua utilização combinada com máquinas virtuais proporciona um grau de isolamento considerável em aplicações na nuvem. Devido às suas características de encapsulamento e isolamento a virtualização é a base para o paradigma da cloud computing. O princípio da virtualização baseada em containers baseia-se na utilização dos recursos do núcleo para criar um ambiente isolado para processos, sem abstrair o hardware do host [35]. Por fazerem parte do próprio sistema operativo não é possível instanciar containers de outros sistemas operativos no mesmo ambiente de execução [15]. Neste modelo de virtualização o isolamento entre os containers é obtido por meio de features do kernel conhecidas como namespaces e control groups (cgroups) [36]

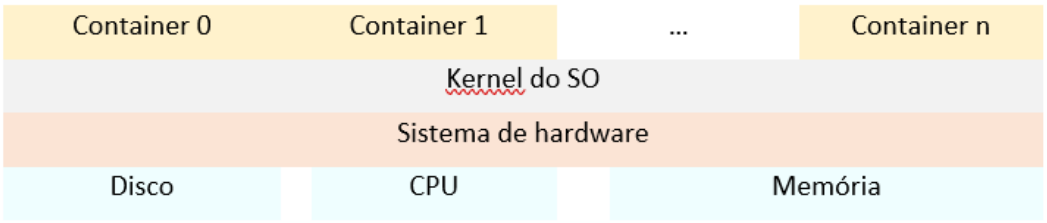

Figura 2.11 – Virtualização ao nível do sistema operativo (figura redesenhada a partir de [15]).

2.5 – Virtualização Baseada em Containers Docker

As tecnologias de containers como o Docker, são uma extensão para o mundo de virtualização do sistema, reduzindo o overhead computacional necessário para executar as aplicações nos diferentes ambientes. A virtualização baseada em containers, fornece gestão e isolamento dos recursos em ambientes Linux e vem modificar o processo de desenvolvimento e de implantação de aplicativos através de uma plataforma que constrói e mantém ambientes para a execução de sistemas distribuídos [18].

O conceito containers apresenta uma solução rápida aos profissionais de sistema para desenvolverem, embarcarem, integrarem e executarem as suas aplicações, com otimização do desenvolvimento, performance, agilidade, entrega e principalmente partilhar os recursos sejam eles físicos ou lógicos [34].

Este capítulo apresenta o conceito de uma arquitetura de provisionamento de plataforma, as características, o funcionamento e as vantagens e por fim um comparativo entre o container, as máquinas virtuais e o Docker.

A plataforma containers Docker tem como objetivo proporcionar múltiplos ambientes isolados dentro de um mesmo servidor, é acessível externamente, via tradução de portas [34]. É uma plataforma de código aberto baseada no kernel do Linux, que automatiza o desenvolvimento de aplicações, por meio de containers portáveis e autossuficientes, a partir de um sistema de arquivos "empilháveis", os containers isolam o SO Base (host) e toda a pilha de dependências da aplicação, bibliotecas, servidores e outros. A sua funcionalidade simplifica o uso dos LXC (LinuX Containers) que, tal como na virtualização, consiste numa forma de isolamento de processo e sistemas, porém mais integrada no Sistema Operativo.

Assim, o Docker possibilita o empacotamento de uma aplicação ou de todo o ambiente dentro de um container, tornando-se portável para qualquer sistema que contenha o Docker instalado, oferece um conjunto completo de ferramentas que transporta a aplicação, quer sejam os sistemas ou as máquinas virtuais ou físicas, permite também executar em qualquer sistema operativo, baseado em Linux dentro de um container com maior flexibilidade. Independente do hardware, da linguagem e da estrutura de desenvolvimento, gere a criação de containers que executam as aplicações de forma isolada [36], através de uma imagem (template). É um projeto de código aberto que permite a criação de containers, a partir de imagens, leves e portáteis para diversas aplicações.

2.5.1 Arquitetura da Plataforma de Containers Docker

O conceito Container é utilizado pelo Docker para representar um ambiente em execução que executa quase todos os softwares e se adapta a todas as versões, de acordo com o código fonte da aplicação. O container utiliza as funcionalidades do Kernel como namepspaces, cgroups, chroot e outros, para construir uma área isolada para a aplicação e tem como diretrizes definir recursos tal como a memória, a rede, o sistema operativo, a aplicação ou o serviço, o ambiente de produção; controlar recursos como a CPU, a memória e o HD através dos parâmetros de configuração, para iniciar ou executar o container e realizar os testes de forma a executar desenvolvimento e estudos [34]. Cada container iniciado no Docker é associado a uma rede especifica; a rede *Bridge quando a rede* não foi explicitamente especificada pelo container ou *a rede* None que isola o container para comunicações externas. A sua única interface de rede

IP será a localhost e Host que entrega para o container todas as interfaces existentes no docker host [33].

A arquitetura Docker foi desenhada num processo cliente/servidor, já que o cliente pode aceder ao servidor Docker de maneira local ou remota e a comunicação pode ser estabelecida por meio de sockets ou RESTful APIs (APIs compatíveis com as restrições do estilo arquitetural REST).

Os Containers Docker utilizam file systems de atualização tipo copy-on-write, que permitem o uso de uma imagem de file system como camada base para a execução de múltiplos containers. As alterações efetuadas em arquivos ou diretórios dentro de um container são isoladas e não podem ser vistas fora dele [34].

O Docker utiliza a linguagem "Go" e as funcionalidades do kernel Linux 2 para a criação de containers de sistema operativo. Os servidores Docker são instalados, em máquinas virtuais ou físicas, num sistema operativo Linux3.

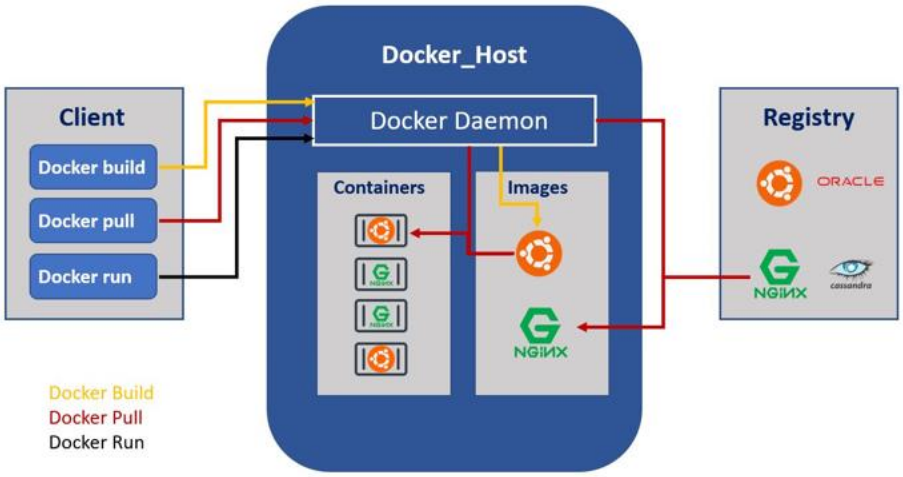

Figura 2.12 – Arquitetura de plataforma de containers Docker (figura adaptada a partir de [34]).

A plataforma containers Docker utiliza os componentes abaixo descritos:

- O Serviço Docker (daemon), que é executado no servidor Docker e é responsável por receber as conexões e comandos dos clientes para construir, empacotar e executar os Containers Docker.

- O Client Docker, que reside na Interface primária de interação do utilizador com o daemon do servidor Docker. É por meio do client Docker que os utilizadores executam os comandos no servidor Docker.

- As Imagens Docker, são um processo que simula o ambiente de execução como um todo, ou seja, o container é a aproximação da imagem e por esse motivo, não pode ser iniciado sem elas, pois estas são o seu ponto de partida. Vários autores [34] definem a imagem como o template que explicita a estrutura, através de uma imagem iniciada por um container é possível gerar novas imagens, basta aplicar um "commit" a cada mudança realizada, as imagens são portanto os recipientes e os containers podem ser transformados em novas imagens otimizadas. Normalmente são imagens do host, mas que podem ser personalizadas para conter um SO básico. Geralmente faz uso do UFS (Sistema de Arquivos Unix), e pode ser criada através do arquivo de configuração denominado "Dockerfile" [34]. As imagens representam um template de leitura constituído por uma imagem de sistema operativo, bibliotecas e binários das aplicações pretendidas. Os utilizadores podem efetuar donwload das imagens, modificar as suas configurações, instalar novas aplicações e salvá-las como novas imagens. Quando uma nova imagem é criada a partir de uma imagem existente, uma nova camada de file system é adicionada ao sistema de arquivos base para o armazenamento das alterações.

- Os Registos Docker ou repositórios externos ou internos de armazenamento das imagens Docker. O Docker Hub é um serviço de repositório SaaS acessível pela internet onde os fabricantes de software disponibilizam suas imagens oficiais Docker para download. Os utilizadores podem efetuar donwload, modificar as imagens padrão, criar as suas próprias imagens e armazená-las de volta no Docker Hub ou num repositório interno. No Docker Hub é possível gravar imagens em áreas públicas ou privadas em que as imagens públicas são abertas para consultas e download para o público em geral e as imagens privadas são acessíveis apenas para utilizadores a quem foram concedidas permissões de acesso.

- Os Containers Docker encapsulam todos os recursos necessários para a execução das aplicações e são executados a partir do sistema operativo do host que os hospedam.

Podem ser iniciados, executados, parados, movidos e apagados. Através dos containers as aplicações são executadas em processos e área de memória isoladas.

- Os Docker files, arquivos com instruções para gerar uma nova imagem Docker. As instruções descrevem o conjunto de passos para a criação da nova imagem, a execução de comandos, a inclusão de novos arquivos ou diretórios, a criação de variáveis de ambiente e a definição de que processos devem ser executados quando um novo container for iniciado por meio da nova imagem criada [34]

2.6 Virtualização Apache Mesos

A utilização de containers inclui um conjunto de tecnologias que evoluíram a partir do espaço de virtualização e que fornecem flexibilidade na infraestrutura de gestão e nos aplicativos. Os dois principais produtos no mercado para essas tecnologias são o Apache Mesos para a gestão dinâmica de infraestrutura e o Docker para plataformas dinâmicas de aplicativos. As duas soluções estão fornecendo novos níveis de capacidade para o mundo virtualizado [37].

A tecnologia Apache Mesos foi projetada na Universidade de Berkeley e é considerada o kernel de sistemas distribuídos. Esta tecnologia introduziu uma arquitetura modular, com uma abordagem de desenvolvimento de código aberto, pensada para ser completamente independente da infraestrutura [37].

Apache Mesos foi adotado pelo Twitter, Uber e muitas empresas de tecnologia para suportar a orientação a microsserviços, a grande volume de dados, análise em tempo real e dimensionamento elástico.

Apache Mesos foi criado com os mesmos princípios do kernel Linux, mas noutro nível de abstração. O Mesos executa em todos os nós de cluster e promove a frameworks como Apache Spark, Hadoop, Cassandra, Kafka, Elastic Search e outros, uma API para alocar recursos e executarem as suas tasks através de todo o cluster ou cloud. No Mesos cluster, CPU, memória, disco e outros recursos computacionais de todas as máquinas, virtuais ou físicas, são abstraídos num único pool de recursos, facilitando a criação e otimizando a utilização de sistemas distribuídos.

O objetivo do Mesos é prover um core confiável e escalável para que diversas aplicações consigam partilhar um mesmo cluster. Para o efeito fornece uma API mínima que permite partilhar recursos de forma eficiente entre os frameworks, atribuindo-lhes o controle para programarem e executarem as suas tasks.

A sua arquitetura é composta por três componente principais: Mesos master, Mesos agent e os frameworks:

- O Mesos master é o processo que gere todo o cluster, definição de recursos para cada framework, registo dos agents, atualização do status das tasks em execução.

- O Mesos agent é o processo daemon que funciona em todos os nós do cluster. A sua função é alocar os recursos definidos para cada task enviada pelo Mesos master e executá-la (através do executor).

- Os frameworks são as aplicações que atuam no Mesos cluster, cada uma delas possui um scheduler e um executor. Sendo o scheduler a aplicação que se regista no master para que possa alocar recursos e enviar tasks para execução e o executor o processo executado nos nós do cluster para executar as tarefas do framework.

- O Apache ZooKeeper é um software que é utilizado para coordenar os nós mestres.

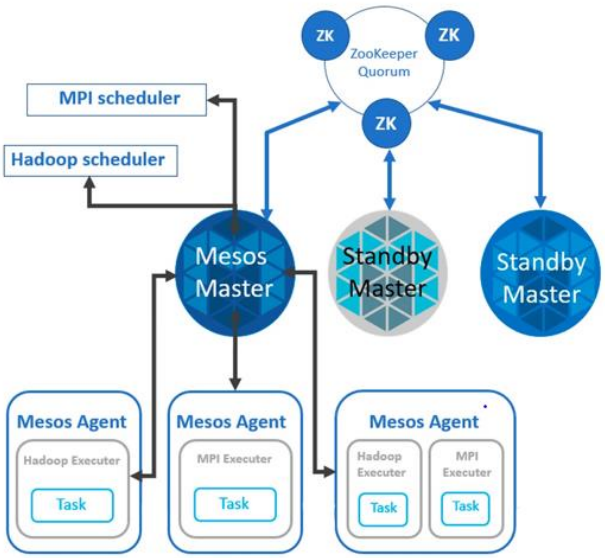

Figura 2.13 - Arquitetura de plataforma Apache Mesos (figura redesenhada a partir de [38]).

O Apache Mesos foi criado de forma a poder simplificar a alocação de recursos, proporcionando uma aplicação coerente e experiência operativo através de nuvens privadas ou públicas, manter diversas tarefas de projeto na mesma infraestrutura analytics, microsserviços, monitorização, dados distribuídos e aplicações, automatizar o dia a dia da operação como desde a implantação, a autorrecuperação, o dimensionamento e os upgrades e fornecer uma infraestrutura tolerante a falha, de alta disponibilidade e extensibilidade aos aplicativos sem necessidade de recodificar. Mesos pode também gerir individualmente um conjunto diversificado de tarefas, incluindo aplicativos tradicionais em Java. Os utilizadores do Apache Mesos concentram-se em big data, containers [38].

2.7 Overheads de Virtualização

O processo de virtualização está associado a overheads de recursos necessários para suportar a virtualização.

Tabela 2.1 Overheads de Virtualização.

Sobrecarga de Virtualização	**Componente do sistema**			
	CPU	Memória	Rede	Disco
Fonte de Sobrecarga	Escalonamento duplo Equidade de escalonamento CPU's assimétricas Interrupções	Recuperação de memória Duplicação de memória	Processamento de pacotes NAT Instabilidade da rede	Escalonamento do I/O de disco Sistemas de ficheiros em camadas

Ao nível da CPU, a virtualização causa sobrecarga de desempenho devido ao Double Scheduling (escalonamento duplo) e de equidade, às assimetrias e às interrupções:

- Double Scheduling (escalonamento duplo) – No processo de virtualização baseado em hypervisor, existem dois níveis de escalonamento: o escalonamento do sistema operativo guest para as CPUs virtuais implementadas no hypervisor como threads e o

escalonamento das threads das CPUs virtuais para a CPU física realizado pelo hypervisor [1], [13]

- Scheduling Fairness (equidade de escalonamento) – A equidade no escalonamento para máquinas virtuais multiprocessadoras simétricas, em que cada uma possui várias CPUs virtuais é difícil de atingir no escalonamento duplo no hypervisor e no sistema operativo guest. Pode existir um desfasamento entre a utilização das máquinas virtuais e das CPU's.

- Asymmetric CPUs (CPUs assimétricas) – embora o algoritmo utilizado no escalonamento procure realizar uma distribuição por todas as CPUs para otimizar a eficiência do sistema, pode existir subutilização ou sobre utilização das CPUs.

- Interrupts - Os interrupts na CPU da máquina física, são retificados pelo mediador de interrupts, nas máquinas virtuais o mediador apenas executa após o escalonamento do CPU virtual para execução do hypervisor, o que induz delays.

- Ao nível da Memory, apresentamos a Memory Reclamation e a Memory Duplication:

- A Memory Reclamation (recuperação de memória) acontece quando o limite da memória física é ultrapassado, o hypervisor procura recuperar memória a partir das máquinas virtuais e desalojar páginas, através do envio destas para o armazenamento físico, no entanto, o hypervisor não possui informação acerca das páginas guest mais qualificadas à expulsão da memória, já que estas são geridas pelos sistemas operativos guest.

- Memory Duplication (duplicação de memória) - Os sistemas operativos guests são constituídos por uma imagem do kernel, bibliotecas partilhadas e serviços do sistema operativo, que causam overhead na pegada da memória. As máquinas virtuais possuem um ambiente isolado e cada máquina tem uma cópia de sistema operativo, logo podem ter a mesma versão do sistema operativo, existindo duplicação da memória.

- Relativamente ao Networking referimo-nos ao Packet Processing, ao protocolo Network Address Translatore (NAT) e ao Unstable Network:

- Packet Processing (processamento de pacotes) - O desempenho ao nível do I/O de rede do hypervisor traduz-se na rapidez do output de TCP/IP sem ter em atenção a

latência e o overhead de cada pacote, o que se torna problemático na execução de aplicações com grandes transferências de pacotes.

- O protocolo NAT introduz um overhead para suportar a tradução de endereços da CPU, que faz aumentar a latência na transmissão de pacotes e o número de conexões necessárias.

• O Unstable Network (instabilidade da rede) refere-se ao output instável de TCP e UDP, causado pela partilha das CPUs, uma variância anormal que causa no output de pacotes um delay que afeta o desempenho da rede.

• Ao nível do Disk destacamos o I/O Scheduling e o Layered Filesystems:

- O I/O Scheduling (escalonamento do I/O de disco) provoca redução de desempenho. Na abordagem de camadas do hypervisor como as máquinas virtuais partilham o mesmo disco físico e existem múltiplas políticas paralelas de escalonamento do I/O de disco em execução, ao nível do hypervisor ou do sistema operativo host e outra ao nível do sistema operativo guest de cada máquina virtual pode exixtir atrito causado pelos escalonadores de I/O de disco que podem ser incompatíveis e haver necessidade de reordenar os pedidos de I/O.

- O Layered Filesystems, (sistemas de ficheiros em camadas) está presente nas tecnologias de containers para reduzir o espaço utilizado para armazenamento e simplificar a gestão de sistemas de ficheiros pois permite ao utilizador empilhar sistemas de ficheiros uns no topo de outros e reutilizar as camadas de containers, mas introduz um overhead significativo no armazenamento porque as operações destes sistemas têm de movimentar-se entre múltiplas camadas.

2.8 Benchmarking

O Benchmarking é uma ferramenta de avaliação cujas propriedades podem incluir a velocidade, a eficiência, a taxa de transferência ou outras [39]. Quando estes têm como foco o desempenho o seu objetivo reside na mais alta eficiência, independentemente do custo, quando está focado nos custos tem como objetivo obter o custo mais baixo independentemente do desempenho do sistema [39]. A sua utilização serve como

propósito fornecer sistemas de aplicações próximos da realidade, para a obtenção de resultados aproximados do cenário real [39], [1], [40], [41].

O SysBench é uma ferramenta de benchmark modular, multiplataforma e multi-threaded que avalia os parâmetros ao nível do sistema operativo, (CPU, memória, File I/O e MySQL).

O SysBench tem um design simples, executa um número especificado de threads e todos eles executam solicitações em paralelo. A carga de trabalho real produzida pelas solicitações depende do modo de teste especificado, é possível limitar o total número de solicitações ou o tempo total para o benchmark, ou ambos. Os modos de teste disponíveis são implementados por módulos compilados e o SysBench foi projetado para tornar a adição de novos modos de teste.

Os testes medem os seguintes parâmetros: performance do CPU, performance de File I/O, alocação de memória e velocidade de transferência da memória, e performance de uma base de dados.

O SysBench tem um design simples, executa um número especificado de threads e todos eles executam solicitações em paralelo. A carga de trabalho real produzida pelas solicitações depende do modo de teste especificado e é possível limitar o total número de solicitações ou o tempo total para o benchmark, ou ambos. Os modos de teste disponíveis são implementados por módulos compilados e o SysBench foi projetado para tornar a adição de novos modos de teste.

Os recursos do SysBench permitem testar o desempenho de I/O, a alocação de memória e velocidade de transferência, o desempenho do servidor da base de dados e o desempenho do CPU.

2.9 A Segurança em Ambientes Virtualizados

As ameaças à segurança inerentes à virtualização trazem dificuldades ao nível da proteção da infraestrutura virtual. De acordo com Schwartz (2010), num estudo realizado pela IBM em 2010, contabilizou que cerca de 35% dos ataques em ambientes virtualizados são direcionados ao hypervisor por ser o elemento central de todo o sistema de virtualização e por gerir todo o ambiente virtual num host físico, reunindo as principais funcionalidades e portas de acesso às VMs [42].

Nesta sequência serão analisados alguns tipos de ataques/ameaças e indicadas algumas técnicas de mitigação [42].

- O VM Escape (escape to hypervisor), resulta numa falha de segurança em hypervisors VMware ESXi em que um invasor consegue executar códigos maliciosos dentro de uma máquina virtual, podendo também executar comandos noutra VM ou até mesmo no hypervisor [42].

- Ataques de negação de serviço são inerentes a qualquer tipo de sistema ou aplicação. O DoS envia um grande número de pacotes (maliciosos ou não) endereçados a algum sistema ou aplicação, acarretando um alto processamento no alvo que está sendo atacado, causando falhas no hardware ou exaustão de recursos [46]. É vulgar encontrar ataques Dos em ambientes virtuais devido à funcionalidade de compartilhamento de recursos entre as VMs e o host físico. Se várias VMs começarem a consumir muito processamento, memória, disco, banda de rede no host físico, a tendência é o host não conseguir mais operar de forma correta ou até mesmo os seus recursos ficarem inacessíveis, podendo gerar danos às outras máquinas virtuais [42].

- VM-Aware Malware que consiste em ataques de malwares a ambientes virtualizados, através de bots, worms, rootkits, e diversos outros tipos de códigos maliciosos, que são capazes de identificar em que ambientes estão sendo executados (virtuais ou físicos) tendo como base informações de memória, hardware, processos ou outros. A partir desta informação ele irá dificultar a análise de seu comportamento pois o mesmo tende a mascarar seu real funcionamento ao identificar que está sendo analisado.

- Roubo de Máquina Virtual, tendo em conta que as máquinas virtuais são arquivos, um utilizador com acesso físico ao local de armazenamento (storages) ou ao hypervisor, pode copiar todos os arquivos das VMs removê-las posteriormente.

- Injeção de Código/Arquivo Malicioso, os ataques de injeção de código malicioso visam fazer com que o código injetado seja executado com um vírus ou qualquer outro código malicioso através dos seguintes métodos:

· Um invasor pode comprometer uma máquina virtual utilizando o hypervisor para transferir uma VM infetada para outro host via FTP e iniciando-a do outro lado;

· O invasor pode fazer passar-se por um “man-in-the-middle” para escutar a comunicação entre o host físico e a máquina virtual, conseguindo assim modificar a troca de informações e fazer com que uma VM infetada se passe como uma VM segura.

- O invasor faz uma busca na rede pelo local onde a máquina virtual está armazenada e caso não haja controle de acesso efetua uma cópia da mesma para outro local, permitindo-lhe fazer as modificações que quiser.

- Footprinting é uma técnica de invasão baseada para obtenção de informações. Geralmente a obtenção dessas informações dá-se a partir de comandos remotos. O objetivo é traçar um perfil do alvo identificando padrões anômalos de tráfego pela rede. Por exemplo, para identificar se o sistema alvo é uma VM ou não, os invasores analisam o tempo de sincronização entre um host e seu SO e o tempo de sincronização de uma VM com seu S.O, sendo o tempo de sincronização e alocação de um processo, numa VM alto e no host baixo [42].

2.9.1 Técnicas de Segurança em Ambientes Virtualizados

Selecionámos algumas técnicas de segurança em ambiente virtualizado, que considerámos relevantes:

- Protection Rings (Anéis de Proteção) - Os anéis de proteção são fornecidos pela CPUs x86 e funcionam como mecanismos de proteção de dados e funcionalidades contra falhas e ações maliciosas. Esses níveis de proteção são níveis de hierarquia de privilégio dentro de uma arquitetura de computação, do mais privilegiado em que 0 é considerado o de maior nível hierárquico ao menos privilegiado. O anel interage com o hardware físico (CPU e memória) e é utilizado pelo sistema operativo. O anel 3 é utilizado para os processos do utilizador [42].

- Guest OS Isolation (Isolamento da Máquina Virtual) - Como o hypervisor é responsável pela gestão do acesso ao hardware, pelos sistemas operativos das VMs, mesmo quando ocorre a partilha, as VMs mantêm-se isoladas umas das outras, sem possibilidade de acesso de umas para as outras. Este recurso está presente em ferramentas como Hyper-V e VMware ESXi e estes são divididos em lógica e física. A divisão lógica significa que o hypervisor entrega recursos para uma ou várias máquinas virtuais, significando que os recursos (memória e processador) podem ou não ser partilhados com várias VMs, como se fosse um pool de recursos com o hypervisor intermediando o seu acesso. A divisão física propõe limitações à alocação de recursos para uma determinada VM e não partilha com as outras máquinas virtuais.

O hypervisor aloca um recurso fixo para uma determinada VM, e essa máquina virtual utiliza apenas uma parte do seu recurso disponível. Essas características impostas pelo hypervisor possibilitam que em caso de infeção esta não atinja a outra máquina virtual ou arquivo [42].

- Guest OS Monitoring (Monitorização da Máquina Virtual - O hypervisor possui recurso de auditoria para monitorizar cada sistema operativo, a introspeção. A introspeção pode prover uma monitorização completa que pode incluir tráfego de rede, memória, processos e diversas outras funcionalidades de um SO, mesmo quando a segurança deste já foi comprometida.

- Snapshot - Diversas ferramentas de virtualização, como o Hyper-V da Microsoft, disponibilizam a funcionalidade de snapshot, que permite que o sistema grave o estado atual da VM para que a mesma possa ser restaurada para algum ponto anteriormente capturado a qualquer momento. A snapshot oferece também uma outra funcionalidade que é a imagem ou clone, que permite ao utilizador criar uma cópia de VM e utilizá-la como sendo outra VM, acrescentando apenas as funcionalidades necessárias.

- Movimentação de Máquinas Virtuais - Este recurso pode ser utilizado no caso de ameaças de ataques ao host físico, consiste em desligar ou reiniciar a VM.
O hypervisor é capaz de realizar esta movimentação de forma automática dependendo das configurações. Por exemplo, quando a carga de processamento em um dado host estiver muito alta, o hypervisor identifica isso e automaticamente move algumas VMs para outro host. Esta funcionalidade está presente nas ferramentas de virtualização como o vMotion no VMware ESXi.

- Criptografia de Máquina Virtual - Existem diversas formas para criptografar os arquivos de uma máquina virtual, na própria máquina virtual, no hypervisor, num dispositivo de armazenamento NAS – Network-Attached Storage, no storage.

Todas as opções apresentadas podem garantir proteção para uma máquina virtual, porém não garantem flexibilidade para acompanhar o fluxo de trabalho de um ambiente de TI, por conta da rapidez do avanço da tecnologia.

Mediante as ameaças e as defesas destacadas, constatamos que o hypervisor, o principal componente do sistema, tem um papel crucial ao nível destes ataques. É necessário adotar medidas não apenas específicas ao hypervisor, mas sim a todo o ambiente em que está inserido, como storage, máquina física, infraestrutura de rede e desktops [42].

2.10 – Implicações da Virtualização

A virtualização é um processo que rentabiliza recursos e/ou reutiliza equipamento e por esse motivo se encontra em expansão. Esta técnica tem a vantagem de distribuir recursos de acordo com a estratégia e a necessidade reduzindo o tempo de inatividade e tornando as soluções com alta disponibilidade e a recuperação de incidentes mais económica, simples e confiável. Assinalamos várias vantagens na sua utilização [43]:

- Economia de espaço físico graças à redução de máquinas físicas,

- Economia de energia já que um menor número de servidores e de desktops físicos reduz os gastos com o ar condicionado,

- Melhor utilização dos recursos graças à capacidade de processamento e armazenamento de novos computadores servidor. A possibilidade de instalação de vários sistemas operativos virtuais na mesma máquina física, permite um melhor desempenho e um melhor aproveitamento dos recursos da máquina hospedeira,

- Múltiplo ambiente num único hardware com a partilha da mesma máquina física pelos ambientes ou sistemas operativos. Esta característica facilita a consolidação de aplicações, de servidores e a migração entre ambientes de forma segura sem provocar qualquer tipo de risco para o host,

- A segurança por ser possível na escolha de máquinas virtuais definir o ambiente para executar cada serviço, com diferentes exigências de segurança, ferramentas diferentes e sistemas operativos mais adequados a cada serviço. Para além do mais, cada VM é independente e isolada das restantes e a vulnerabilidade de um serviço não afeta os restantes,

- Confiança e disponibilidade pois a falha de um software não prejudica os restantes serviços,

- Redução de custos com a consolidação de pequenos servidores noutros mais poderosos,

- Adaptação às diferentes cargas de trabalho com a utilização de ferramentas de gestão para ajustar os recursos de uma máquina virtual para outra,

- Balanceamento de carga possível com o encapsulamento das VMs no VMM, a troca da VM de plataforma com o intuito de aumentar o seu desempenho, torna-se mais fácil,

- Suporte de aplicações legadas com a possibilidade de manter o sistema operativo antigo e corrê-lo na VM, o que ajuda na diminuição de custos. Também existe a possibilidade de executar em hardware recente e com custos de manutenção reduzidos e maior fiabilidade,

- O backup permite armazenar os dados em datacenter, onde se verifica uma maior segurança, menor probabilidade de perda de informação, mecanismos redundantes e sistemas de recuperação geridos por profissionais,

- A gestão beneficia com a partilha de informação e de dados, mantendo as personalizações referentes a cada colaborador. A metodologia da gestão permite a migração de desktops virtuais sem seja necessário ser formatada a máquina física. Um número menor de servidores e de desktops físicos e boas ferramentas, poupa tempo e facilita o trabalho dos responsáveis das redes.

A virtualização apresenta, no entanto, algumas desvantagens tal como a perda de performance das aplicações e o elevado consumo de memória e processamento das VM's, exigindo maior desempenho e configurações mais sofisticadas com um custo maior. Identificam-se as desvantagens seguintes:

- A segurança pois as máquinas virtuais acabam por apresentar menos segurança que as físicas devido às VMM. Se o sistema operativo host estiver vulnerável a algum tipo de ataque, todas as VM's que estão hospedadas nessa máquina física ficam igualmente vulneráveis visto que o VMM é apenas mais uma camada de software e está sujeito a vulnerabilidades e a ataques,

- A gestão pois todos os ambientes ou sistemas operativos virtualizados necessitam de constante monitorização e configuração,

- O desempenho já que não existem métodos sólidos que permitam medir o desempenho dos ambientes virtualizados e que quando se implementa uma camada de software extra entre o sistema operativo e o hardware, esta acaba por gerar um processamento superior ao que teria antes da implementação da virtualização. Também não conseguimos perceber com exatidão quantas VM's podem ser executadas em simultâneo, sem perda de qualidade de serviço.

2.11 Trabalho Relacionado

Esta secção destina-se a apresentar artigos relacionados com o problema em estudo, resultados e soluções existentes envolvendo virtualização nativa ao nível de hardware ou ao nível do sistema operativo.

Kurmus et al. [44], apresentam um estudo comparativo de duas arquiteturas de virtualização, uma baseada em hardware (hypervisor) e outra baseada em sistema operativo (containers), em que são utilizadas ferramentas de benchmark para avaliar ao nível de segurança qual das arquiteturas possui um melhor desempenho. o estudo revela que ambas as arquiteturas são viáveis ao nível de segurança, mas a arquitetura baseada em hypervisors possui um overhead significativo em comparação à arquitetura baseada em containers, devido a possuir camadas adicionais isoladas.

Graziano [45] desenvolve um trabalho que se foca no estudo de duas tecnologias de virtualização (Xen e KVM), em que são conduzidas avaliações de desempenho ao nível geral, taxa de transferência e eficiência de máquinas virtuais ao nível de isolamento e escalabilidade para uso como host para o Xen World Project. Os resultados concluíram que o Xen possui um desempenho superior ao KVM em cenários relacionados com o I/O de disco, e que o KVM demonstra um desempenho superior na execução de operações computacionalmente intensivas.

Sridharan [46] efetua uma abordagem quantitativa e qualitativa, para comparar o desempenho de três hypervisors, o VMware ESXi, o Zen e o KVM, que usam a ferramenta standard de benchamark SPECvirt_sc2010v1.01 para avaliar o seu desempenho sobre diferentes workloads que simulam situações reais. Obteve como conclusão de que o hypervisor VMware ESXi possui um melhor desempenho global, sendo este seguido pelo Zen e por último o KVM.

Enberg [13] aborda os overheads das duas principais técnicas de virtualização, hypervisor e container, sendo as tecnologias de virtualização KVM e Docker. O uso de múltiplas ferramentas de benchmark (Netperf, Memcached e Mutilate) permitiu chegar a uma conclusão de que a tecnologia de virtualização Docker possui um desempenho mais aproximado ao nativo, sendo superior ao desempenho das tecnologias de hypervisor.

Kavita Agarwal [47] realiza um estudo comparativo entre o hypervisor KVM e a tecnologia de containers Docker. Os principais parâmetros da análise comparativa são

a latência, a densidade e a pegada deixada na memória. Conclui que a arquitetura de máquinas virtuais possui uma pegada na memória maior que a da arquitetura de containers.

Voras et al. [48] apresenta uma comparação entre várias tecnologias de virtualização open-source e comerciais e faz uso de múltiplas ferramentas de benchmark para análise completa dos sistemas. O objetivo do estudo é efetuar uma avaliação igualitária de todas as tecnologias para uma tomada de decisão para fins de investigação. Os resultados do estudo revelam que o desempenho das tecnologias de hypervisors se aproxima do desempenho nativo quando os workloads executados são single-threaded. Contudo as diferenças mais acentuadas no desempenho estão associadas à execução de workloads multi-threaded.

Felter et al. [49], exploram o desempenho de máquinas virtuais, em contraste com containers de Linux. Através do uso de workloads gerados por benchmarks efetuam uma análise de desempenho a todos os componentes do sistema. As tecnologias de virtualização analisadas foram o KVM, que representa a tecnologia de hypervisor, e o Docker que representa a tecnologia de containers. Os resultados da análise demonstram que a tecnologia de containers possui um desempenho igual ou superior à tecnologia de hypervisor.

Vishrutha Adla [50] efetua uma análise comparativa dos hypervisors Hyper-V e VMware ESXi, com o objetivo de verificar qual possui melhor desempenho geral. Os resultados da análise revelam que o VMware ESXi possui um melhor desempenho em geral em comparação com o Hyper-V.

Também Naveed Yaqub [39] no seu trabalho pretendeu avaliar através da comparação entre dois hypervisors, KVM e VMware ESXi, recorrendo a ferramentas de benchmark (IOzone, RAMspeed e UnixBench), o melhor desempenho e a menor quantidade de overheads em cada componente do sistema. Como resultado verificou-se que o VMware ESXi possui um melhor desempenho que o KVM ao nível de I/O de disco e da CPU. O KVM possui um melhor desempenho ao nível de memória.

Jianga et al [51], efetuam uma comparação energética de cinco tecnologias de virtualização instaladas sobre seis tipos de hardware diferente. Das tecnologias de virtualização utilizadas, quatro delas são hypervisors (VMware ESXi, Microsoft Hyper-V, KVM e XenServer) e a restante é uma tecnologia baseada em containers (Docker). Retiraram como conclusão que o consumo de energia das tecnologias de virtualização depende do hardware sobre o qual se encontram a executar.

A VMware [52] realizou um teste comparativo entre o seu hypervisor VMware ESX Server 3.0.1 e o hypervisor Xen 3.0.3. e através do uso de ferramentas de benchmark (ferramentas SPEC, Netperf e Passmark) efetuou uma análise quantitativa e qualitativa de cada hypervisor. As conclusões permitem concluir que o VMware ESX Server possui um desempenho superior ao nível da escalabilidade e prontidão para produção, necessário para uma implementação eficiente de um datacenter.

Reddy e Rajamani [53], avaliaram o desempenho de três hypervisors, VMware ESXi, XenServer e KVM, tendo usado o framework SIGAR para obter informações do sistema e a ferramenta de benchmark Passmark para a criação de workloads em ambientes de cloud privada. Concluíram que o hypervisor VMware ESXi possui um melhor desempenho do que os outros dois ao nível da CPU e da rede, enquanto que o XenServer possui uma eficiência superior ao nível de memória e de I/O de disco relativamente aos outros hypervisors em estudo.

Louro, A. [1] efetuou uma comparação do desempenho de infraestruturas virtualizadas de elevada disponibilidade usando hypervisors nativos ou containers: Microsoft Hyper-V Versus Docker. O estudo dos resultados recolhidos pelas ferramentas de benchmark (Geekbench, IOzone, RAMspeed e iPerf) permitiu concluir que a infraestrutura baseada em containers Docker possui um melhor desempenho ao nível da CPU, memória RAM, sistema de ficheiros, rede e migração de containers em comparação com a infraestrutura baseada no hypervisor Hyper-V e que as máquinas virtuais e containers demonstram um desempenho inferior às máquinas nativas devido à presença dos overheads de virtualização.

2.12 Conclusão

Este capítulo abordou os tipos, as técnicas, as tecnologias e as overheads de virtualização, assim como também os problemas de segurança associados à virtualização. Todo este percurso em conjunto com a análise de trabalhos realizados por vários autores, na temática da dissertação facultam a informação necessária para a implementação dos ambientes experimentais do capítulo 3 e posteriormente para a análise de resultados do capítulo 4. A realização de benchmarking é a melhor metodologia para a realização do estudo comparativo entre infraestruturas, porque permite a recolha de resultados ao nível do desempenho.

Capítulo 3

Implementação do Ambiente Experimental

3.1 Introdução

Neste capítulo são apresentados os ambientes experimentais, os recursos de hardware e software utilizados no ambiente de testes, os métodos de implementação dos ambientes experimentais e as ferramentas de avaliação de desempenho (benchmarks) que realizam os testes de desempenho aos ambientes, de acordo com as métricas definidas. A infraestrutura que fornece software para virtualização do ambiente desktop é a VMware e as infraestruturas implementadas no ambiente de testes são, o Apache Mesos e o Docker. A ferramenta de benchmark usada na avaliação das máquinas virtuais das infraestruturas referidas é o Sysbench.

3.2 Ambiente de Testes Experimental

3.2.1 Caracterização do Ambiente de Testes Experimental

A VMWare é uma infraestrutura para a arquitetura x86 de virtualização e fornece o software para a virtualização aos vários ambientes. Os produtos disponibilizados dividem-se em gestão e automação, infraestrutura virtual e plataformas de virtualização [52].

A VMWare é executada no espaço de aplicação, dentro de um sistema operativo host e fica responsável pela abstração dos dispositivos disponibilizados para o sistema operativo guest. Para um acesso mais rápido aos dispositivos, a VMWare instala um driver que permite contornar o problema da necessidade de suportar um amplo conjunto de dispositivos para a arquitetura x86. [7].

Na VMware a virtualização é efetuada ao nível do processador. As instruções privilegiadas são capturadas e virtualizadas pelo VMM, enquanto as restantes instruções são executadas diretamente no processador host.

A VMware também possibilita a virtualização de hardware, fornecido pelo próprio sistema operativo host. Para que haja acesso aos diversos dispositivos, o VMware instala um driver de dispositivo, o VMDriver.

Utilizámos a versão workstation 15.5.2, a versão comercial do VMware que é utilizada em estações de trabalho por permitir criar registos instantâneos (snapshot) de uma máquina virtual, fazer backup ou testar configurações, existe também a possibilidade de poder criar máquinas virtuais em dispositivos externo como um disco rígido ou um pendrive, através de um produto adicional denominado ACE Option Pack [54].

O VMware Workstation destaca-se pela facilidade de uso no processo de criação de máquinas virtuais, este possui um assistente que ajuda a criar clones de máquinas virtuais, grupos de máquinas virtuais e colocá-las em redes [55].

O ambiente de testes experimental é constituído por duas máquinas virtuais criadas através do VMware em que numa foi instalado o Docker e na outra o Apache Mesos, as duas possuem as mesmas características para garantir que o desempenho dos testes não sofre influências, por diferenças a este nível.

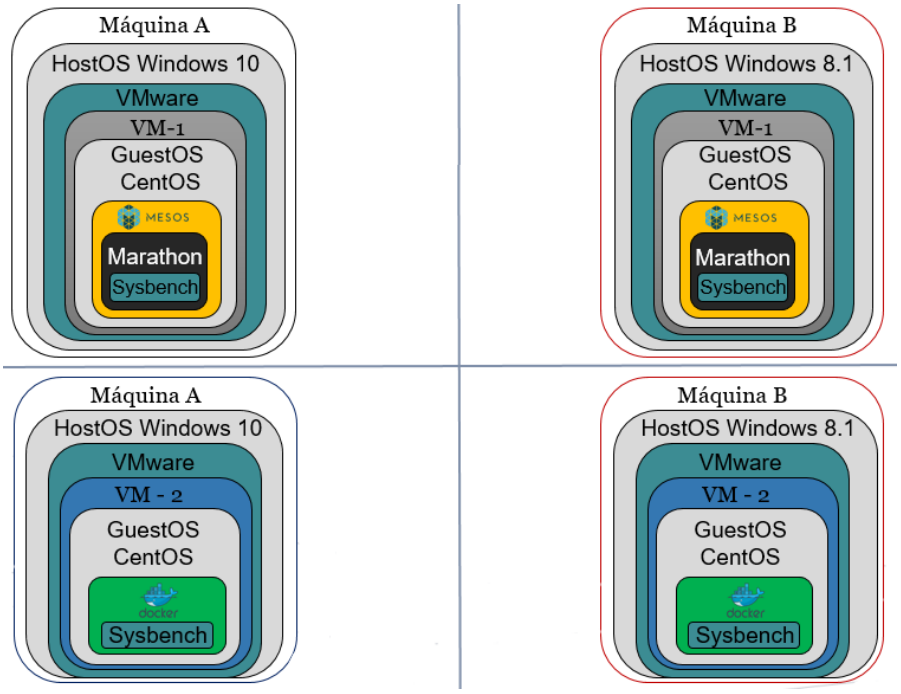

Figura 3.1 – Arquitetura de Ambiente de testes

3.2.2 Especificações de Hardware e Software do Ambiente de Testes

As máquinas criadas têm como sistema operativo o CentOS7, cada uma com 8 GB de memória, quatro processador com um core por cada processador e 50 GB em disco.

No sentido de propiciar maior eficácia e consistência ao ambiente de testes experimental, estes foram realizados em duas máquinas físicas diferentes, como especificado na tabela 1.2.

Tabela 3.1 - Especificações de Hardware e Software do Ambiente de Testes Experimental

	MÁQUINA A	MÁQUINA B
CPU	Intel (R) Core (TM) i7-7700HQ CPU @ 2.80GHZ	Intel(R) Core(TM) i7-5820K CPU @ 3.30GHz
Memória	16GB Ram 2400 MHz	32GB Ram 2400 MHz
Disco	1TB 7200RPM	2TB 7200RPM
Motherboard	MSI MS-17C1	Asus Rampage V Extreme
Sistemas Operativos	Windows 10 Education	Windows 8.1

3.3 Implementação das Infraestruturas Virtualizadas

3.3.1 Instalação do Apache Mesos

Para a instalação do Apache Mesos, o primeiro passo foi adicionar o repositório Mesosphere através do seguinte comando:

```
# sudo rpm -Uvh http://repos.mesosphere.com/el/7/noarch/RPMS/mesosphere-el-repo-7-1.noarch.rpm
```

De seguida foi necessário instalar os pacotes mesos e matathon com o yum:

```
# sudo yum y install mesos marathon
```

De seguida, procedemos à instalação do Zookeeper através do comando:

```
# sudo yum y install mesosphere zookeeper
```

Após a instalação do zookeeper, foi necessário fazer a sua configuração, para isto, foi atribuído um id ao nó mestre dentro do arquivo / var / lib / zookeeper / myid. Esse número deve ser exclusivo e estar entre 1 e 255, caso haja mais do que um nó mestre. De seguida foi anexado o seguinte valor a /etc/zookeeper/conf/zoo.cfg ao nó correspondente, substituindo a tag pelo endereço IP do seu próprio mestre:

```
server.1 = [master1 IP]: 2888: 3888
```

Depois do zookeeper configurado, seguiu-se a sua iniciação:

```
# sudo systemctl enable zookeeper
# sudo systemctl start zookeeper
# sudo journalctl -u zookeeper | tail -10
```

Depois do zookeeper operativo, prosseguiu-se com o Apache Mesos e o marathon. Foi então necessário configurar o nó mestre no arquivo / etc / mesos / zk substituindo a tag pelo IP do nó mestre.

```
zk://[master1 IP]:2181
```

Por fim, foi iniciado o serviço Mesos-master, o Mesos-slave e o marathon

```
# sudo systemctl enable mesos-master
# sudo systemctl start mesos-master
# sudo systemctl enable marathon
# sudo systemctl start marathon
# sudo systemctl enable mesos-slave
# sudo systemctl start mesos-slave
```

Figura 3.2 – Arquitetura de Ambiente Experimental do Apache Mesos (figura adaptada a partir de [56]).

3.3.2 Instalação do Docker

Apresentam-se os diferentes comandos para a instalação da plataforma Docker:

O primeiro passo para a instalação foi atualizar a base de dados dos pacotes, através do comando:

```
# sudo yum check update
```

De seguida adicionámos o repositório oficial do Docker, foi efetuado o download da versão mais recente do Docker e a sua instalação através do seguinte comando:

```
# curl fsSL https://get.docker.com/ | sh
```

Após a conclusão da instalação, foi iniciado o daemon do Docker:

```
# sudo systemctl start Docker
```

Para validar a sua execução foi utilizado o comando:

```
# sudo systemctl status docker
```

Output

```
docker.service - Docker Application Container Engine Loaded: loaded
(/lib/systemd/system/docker.service; enabled; vendor preset: enabled)

Active: active (running) since Sun 2016-05-01 06:53:52 CDT; 1 weeks 3 days ago

Docs: https://docs.docker.com

Main PID: 749 (docker)
```

3.3.3 Instalação das Ferramentas do Benchmark

Depois de o Docker estar instalado, o próximo passo foi criar um novo container para testar o desempenho do Docker através de benchmarks.

```
# docker run it name Sysbench centos
```

Para a criação de um novo container com o nome Sysbench e com a imagem centos, depois de criado foi necessário adicionar o repositório EPEL, para podermos instalar o Sysbench no container.

```
# rpm import /etc/pki/rpm gpg/RPM GPG KEY*
# yum  y install epel release
# yum  y update
```

Com o repositório devidamente adicionado, segue-se então a instalação do benchmark Sysbench.

```
# yum install sysbench
```

Depois de todas as alterações estarem concluídas, é necessário fazer um commit para que as alterações permaneçam.

```
# docker commit 15d31c86c0dd centos/sysbench
```

Ao executar o comando commit, estamos a criar uma imagem baseada na inicial. A nomenclatura utilizada centos/Sysbench é o nome da nova imagem.

```
[root@localhost david]# docker images
REPOSITORY              TAG                 IMAGE ID            CREATED             SIZE
centos/sysbench         latest              40f541beab16        30 minutes ago      408MB
```

Figura 3.3 – Docker Imagens

Com a nova imagem gerada contendo o Sysbench instalado, é possível testar o desempenho do Docker com o desenvolvimento de vários scripts.

No caso do Apache Mesos o procedimento da instalação do Sysbench é semelhante ao do Docker, mas para executar os testes do Sysbench é necessário a instalação do Marathon, que agenda tarefas executadas em nós escravos.

3.4 O Benchmark Sysbench

Para testar o desempenho das diferentes opções de virtualização utilizámos ferramentas e aplicações opensource como o benchmark SysBench [57]. O SysBench é uma ferramenta de benchmark modular, multiplataforma e multi-threaded que avalia os parâmetros ao nível do sistema operativo, (CPU, memória, File I/O e MySQL).

O SysBench tem um design simples, executa um número especificado de threads e todos eles executam solicitações em paralelo. A carga de trabalho real produzida pelas solicitações depende do modo de teste especificado, é possível limitar o total número de solicitações ou o tempo total para o benchmark, ou ambos. Os modos de teste disponíveis são implementados por módulos compilados e o SysBench foi projetado para tornar a adição de novos modos de teste.

Os testes medem os seguintes parâmetros: performance do CPU, performance de File I/O, alocação de memória e velocidade de transferência da memória, e performance de uma base de dados.

3.5 Conclusão

Neste capítulo são descritas as ferramentas utilizadas para a criação do ambiente experimental, são apresentadas as características de hardware das máquinas físicas que vão comportar o ambiente de testes.

São também descritos os métodos de instalação do Docker, do Apache Mesos e da ferramenta de Benchmark e apresentada uma descrição dos testes de Benchmark que são realizados para testar o desempenho das ferramentas de virtualização.

Capítulo 4

Análise dos Resultados Experimentais

4.1 Introdução

Neste capítulo são apresentados os resultados experimentais obtidos através da ferramenta de benchmark Sysbench.

Através da análise dos resultados são obtidas várias conclusões acerca do desempenho dos sistemas de virtualização. As conclusões obtidas dos resultados da ferramenta de benchmark Sysbench permitiram determinar qual das implementações dos sistemas de virtualização possui o melhor desempenho Apache Mesos ou Docker.

4.2 Implementação dos Testes

Foram realizados testes sobre os dois sistemas de virtualização implementados no Capítulo 3, ao nível da CPU, memória, disco e MySQL tendo sido realizados 100 testes por sistema para cada benchmark, para obtenção de resultados normalizados, através da utilização da média. Os testes foram realizados em 2 máquinas o que perfaz um total de 200 testes por benchmark.

Para pôr à prova os dois sistemas de virtualização ao nível do CPU foi utilizado o Sysbench que calcula os números primos até ao valor máximo escolhido, fazendo a divisão padrão do número por todos os números entre 2 e a raiz quadrada do número. Se qualquer número der um resto de 0, o próximo número é calculado e assim o CPU vai ser sobrecarregado. Para a realização do teste de CPU foi criado um script que executa o teste 100 vezes e foi-se aumentando o valor máximo de busca por números primos, usámos os valores: 20000, 50000, 200000, 300000, 500000 e 1000000

verificámos o número de eventos por segundo para cada sistema de virtualização em cada máquina, obtendo assim um total de testes de aproximadamente 2500 testes para o CPU.

A comparação dos valores foi feita pelo número de eventos por segundo em vez do tempo de execução pois, ao executar o teste com vários segmentos, estamos a fazer a soma do tempo de todos os segmentos, o que implica a sobrecarga de acesso à memória partilhada pelas threads. No caso do tempo de execução do evento, o tempo total é a duração do início até ao fim, portanto, não há culminação dos tempos individuais das threads.

Os sistemas de virtualização foram também avaliados pelo desempenho I/O do arquivo e para isso usámos novamente um script que faz a criação de 100 arquivos de teste maiores do que a RAM das máquinas virtuais para evitar que o sistema use a RAM para cache, o que iria interferir nos resultados do benchmark.

Além dos testes ao nível do CPU e dos ficheiros de I/O foi possível avaliar os sistemas de virtualização pelo desempenho da base de dados ao nível de escrita e de leitura em que foi criada uma tabela com 10000 linhas e depois foi avaliado o número total de transações, o total de consultas, incluindo leitura e escrita. Foi também avaliado o desempenho ao nível da memória em que o objetivo foi fazer com que fosse alocado um buffer de memória, para que fosse lido ou escrito nele, até o tamanho total do buffer ser atingido através de leituras e escritas. Para isto foi feito novamente um script com os respetivos comandos que executa os testes 100 vezes.

4.3 Comparação Experimental do Desempenho das Infraestruturas Implementadas

4.3.1 Resultados obtidos com o Sysbench - CPU

A ferramenta Sysbench permitiu testar o desempenho dos sistemas de virtualização Docker e Apache Mesos. Para isso foi necessário executar o script falado na secção 4.2, permitindo obter os resultados que são apresentados no gráfico da figura 4.1.

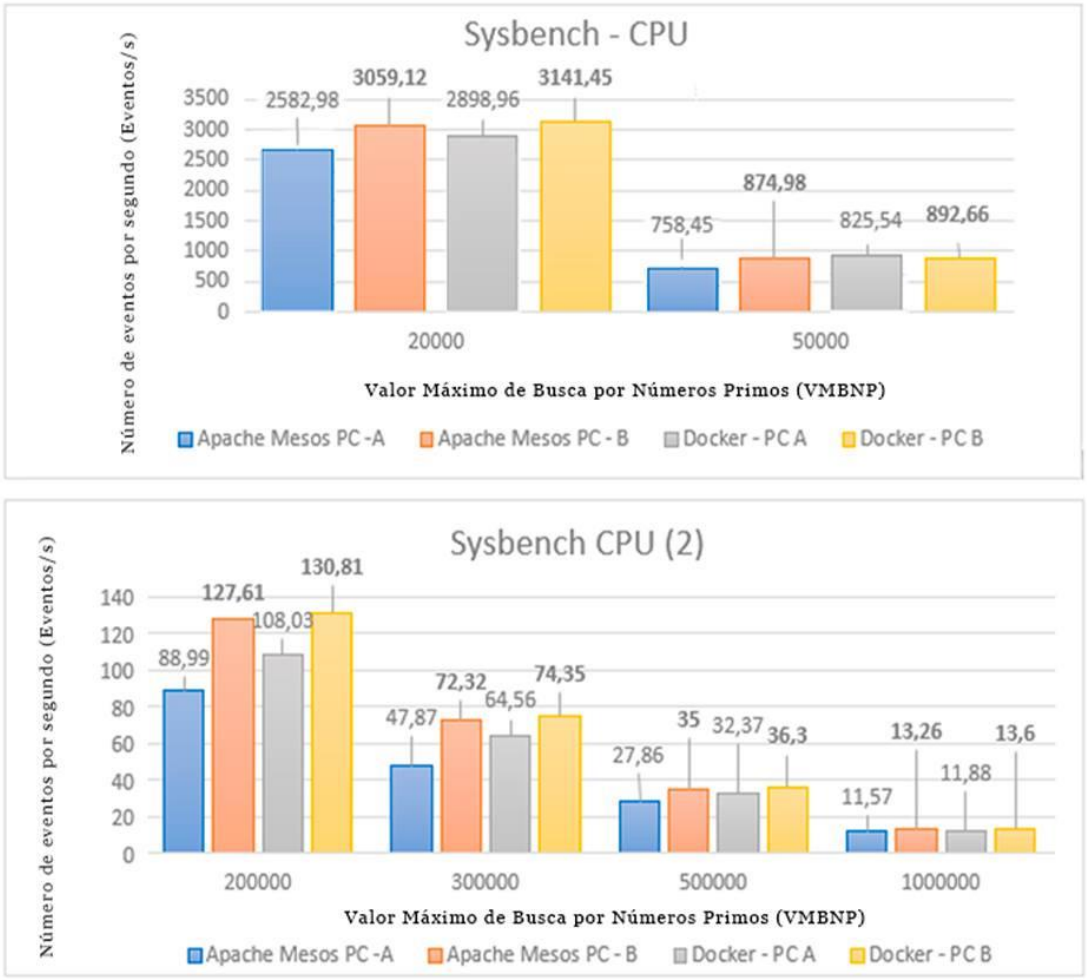

Figura 4.1 – Avaliação de Desempenho do CPU (VMBNP - Valor Máximo de Busca por Números Primos)

Com base nos gráficos anteriores, podemos verificar que a tecnologia de virtualização Docker possui um desempenho superior em comparação ao Apache Mesos ao nível do CPU. Através da análise dos gráficos podemos concluir que o CPU no sistema de virtualização Docker possui uma eficiência superior à do Apache, o número de eventos por segundo é superior no Docker em todos os testes executados, o que significa que consegue executar maior números de eventos em igual tempo.

4.3.2 Resultados obtidos com o Sysbench – FILE I/O

O desempenho dos sistemas de virtualização ao nível dos ficheiros de entrada e saída foram avaliados através da ferramenta de testes Sysbench, através desta ferramenta foi possível analisar qual dos dois sistemas de virtualização teve um melhor desempenho. Foi então executado o script falado na secção 4.2 para obter os resultados mostrados na figura 4.2

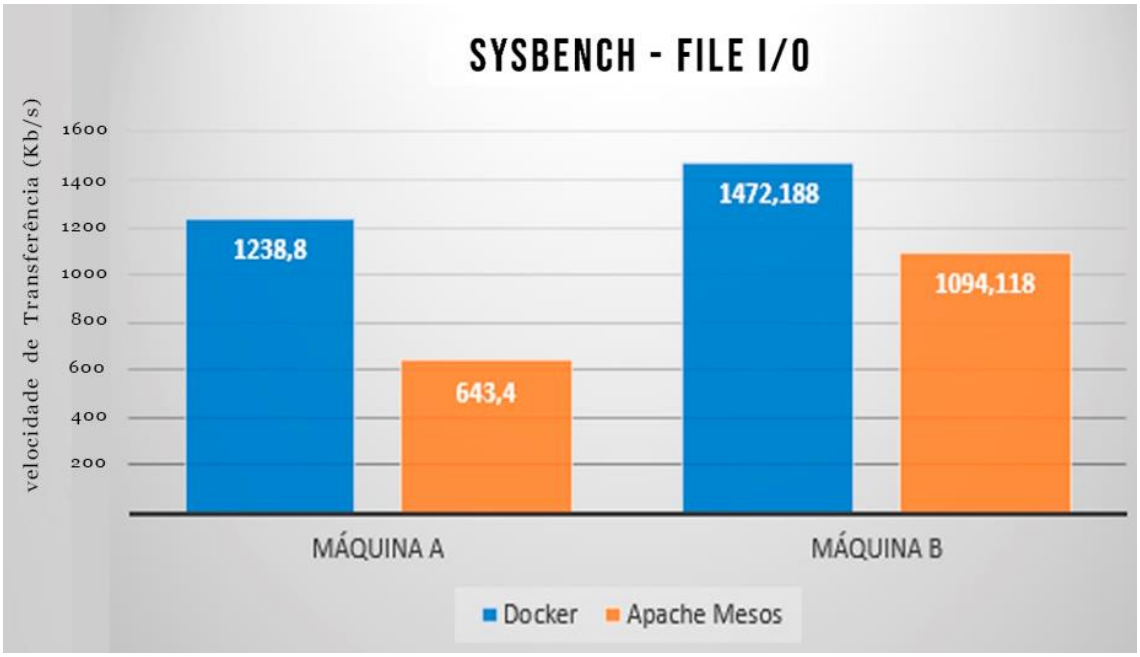

Figura 4.2 – Avaliação de desempenho de File I/O

Com base nos resultados obtidos mostrados pelo gráfico anterior, podemos observar que o Docker tem um desempenho superior Apache Mesos, ou seja, o Docker consegue ler e escrever ficheiros num espaço de tempo menor que o apache Mesos. Assim sendo, é possível afirmar que o Docker tem uma maior velocidade de transferência de ficheiros que o Apache Mesos.

4.3.3 Resultados obtidos com o Sysbench – Memória

Com a ajuda da ferramenta Sysbench foi possível testar o desempenho dos sistemas de virtualização ao nível da memória, conseguimos avaliar qual dos dois sistemas de virtualização tem uma taxa de transferência superior. Com este objetivo em mente, colocámos em prática o script e os procedimentos falados na secção 4.2 permitindo obter os resultados mostrados na figura 4.3

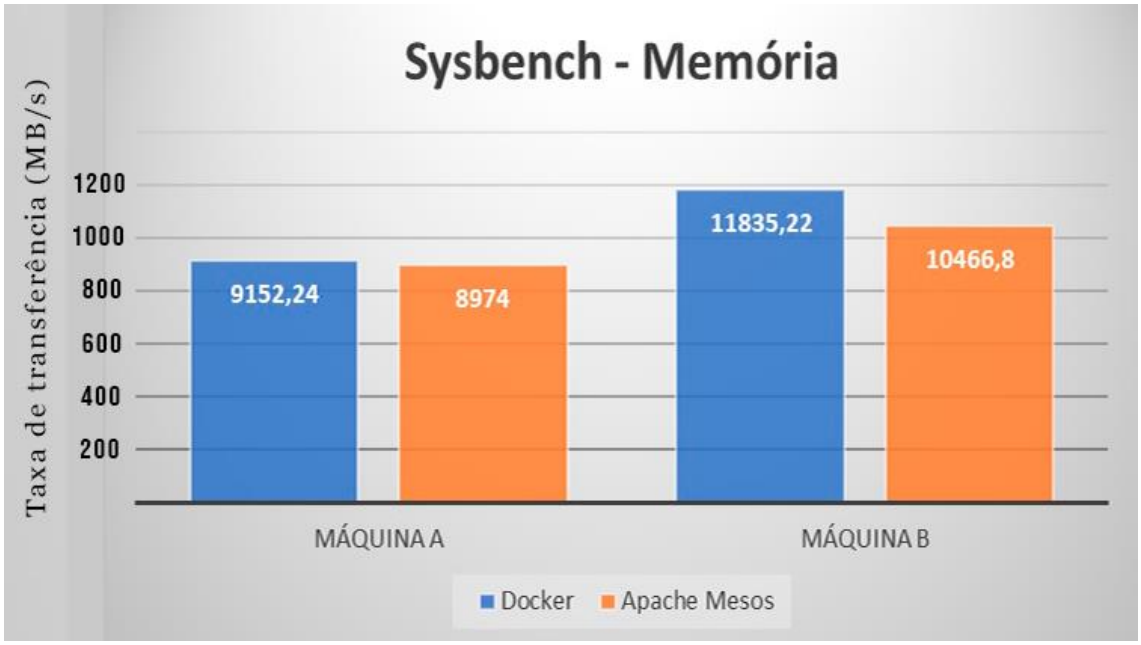

Figura 4.3 – Avaliação de desempenho de Memória

Analisando os resultados mostrados no gráfico anterior, podemos observar que mais uma vez o Docker é superior nas duas máquinas em relação ao Apache Mesos, o que significa que ao nível da memória o Docker apresenta caraterísticas melhores que fazem com que a sua taxa de transferência seja superior à do Apache Mesos.

4.3.4 Resultados obtidos com o Sysbench – MySQL

O desempenho dos sistemas de virtualização foi ainda avaliado ao nível de base de dados em que através da ferramenta Sysbench foi possível observar qual o sistema de virtualização é superior tendo em conta ao número total de transações e ao número total de consultas. Para alcançar estes resultados foi executado o script falado na secção 4.2 e obtivemos os resultados apresentados na figura 4.4.

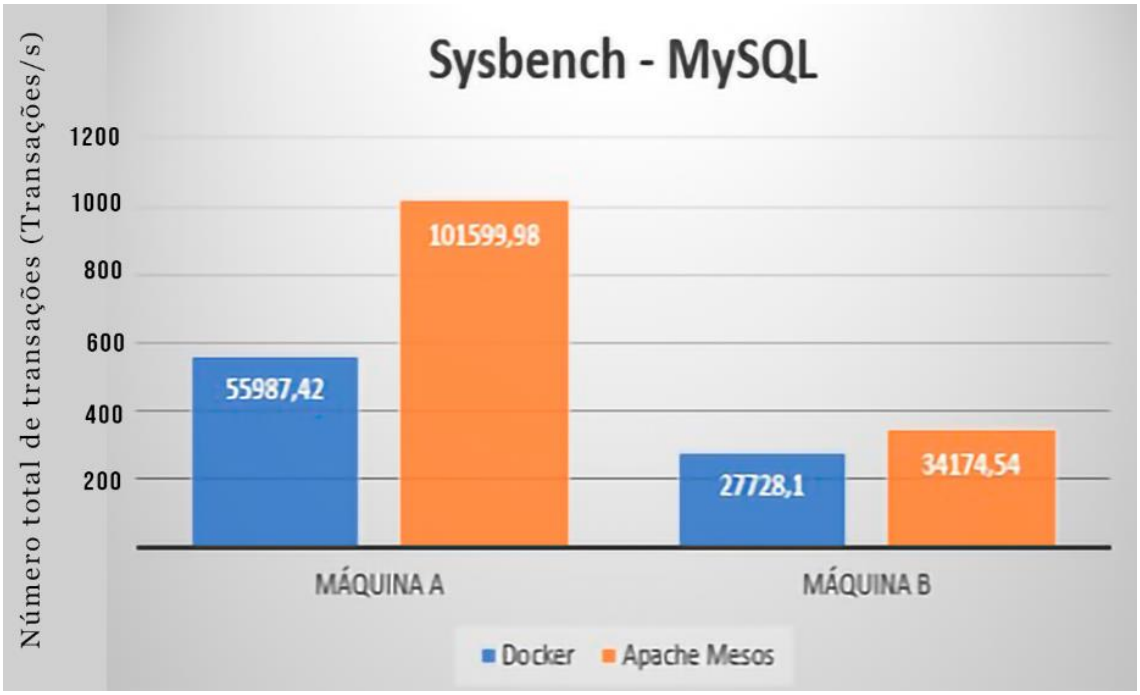

Figura 4.4 – Avaliação de desempenho de MySQL

Analisando os resultados na figura 4.4 foi possível observar que ao contrário do espectável, o Apache Mesos foi superior ao Docker. Com base nos resultados obtidos foi possível concluir que o Apache Mesos se encontra com melhor desempenho ao nível de base de dados relativamente ao número total de consultas e escritas.

4.3.5 Análise e Discussão dos Resultados

Analisando os resultados obtidos através do Sysbench ao nível de CPU e de forma global, podemos observar que o Docker é superior ao Apache Mesos em todos os testes realizados. Numa análise mais aprofundada, podemos observar quanto superior o Docker é, em relação ao Apache Mesos:

- Para o teste feito na máquina A com o valor 20000 (valor máximo de busca por números primos) o docker foi superior em 12.23% em relação ao Apache Mesos, sendo que o Docker obteve uma velocidade de 2898,96 eventos por segundo e o Apache Mesos 2582.98.

- Para o teste feito na máquina B com o valor 20000 (valor máximo de busca por números primos) o docker foi superior em 2.69% em relação ao Apache Mesos, sendo que o Docker obteve uma velocidade de 3141.45 eventos por segundo e o Apache Mesos 3059.12.

- Para o teste feito na máquina A com o valor 50000 (valor máximo de busca por números primos) o docker foi superior em 8.84% em relação ao Apache Mesos, sendo que o Docker obteve uma velocidade de 824.54 eventos por segundo e o Apache Mesos 758.45.

- Para o teste feito na máquina B com o valor 50000 (valor máximo de busca por números primos) o docker foi superior em 2.02% em relação ao Apache Mesos, sendo que o Docker obteve uma velocidade de 892.66 eventos por segundo e o Apache Mesos 874.98.

- Para o teste feito na máquina A com o valor 200000 (valor máximo de busca por números primos) o docker foi superior em 21.39% em relação ao Apache Mesos, sendo que o Docker obteve uma velocidade de 108.03 eventos por segundo e o Apache Mesos 88.99.

- Para o teste feito na máquina B com o valor 200000 (valor máximo de busca por números primos) o docker foi superior em 2.51% em relação ao Apache Mesos, sendo que o Docker obteve uma velocidade de 130.81 eventos por segundo e o Apache Mesos 127.61.

- Para o teste feito na máquina A com o valor 300000 (valor máximo de busca por números primos) o docker foi superior em 34.86% em relação ao Apache Mesos sendo que o Docker obteve uma velocidade de 64.56 eventos por segundo e o Apache Mesos 47.87.

- Para o teste feito na máquina B com o valor 300000 (valor máximo de busca por números primos) o docker foi superior em 2.76% em relação ao Apache Mesos, sendo que o Docker obteve uma velocidade de 74.35 eventos por segundo e o Apache Mesos 72.35.

- Para o teste feito na máquina A com o valor 500000 (valor máximo de busca por números primos) o docker foi superior em 16.18% em relação ao Apache Mesos, sendo que o Docker obteve uma velocidade de 32.37 eventos por segundo e o Apache Mesos 27.86.

- Para o teste feito na máquina B com o valor 500000 (valor máximo de busca por números primos) o docker foi superior em 3.71% em relação ao Apache Mesos, sendo que o Docker obteve uma velocidade de 36.3 eventos por segundo e o Apache Mesos 35.

- Para o teste feito na máquina A com o valor 1000000 (valor máximo de busca por números primos) o docker foi superior em 2.67% em relação ao Apache Mesos, sendo que o Docker obteve uma velocidade de 11.88 eventos por segundo e o Apache Mesos 11.57.

- Para o teste feito na máquina B com o valor 1000000 (valor máximo de busca por números primos) o docker foi superior em 2.56% em relação ao Apache Mesos, sendo que o Docker obteve uma velocidade de 13.6 eventos por segundo e o Apache Mesos 13.26.

Através desta análise de resultados podemos concluir assim que na máquina A o Docker foi em média 16.02% superior ao Apache Mesos, enquanto na máquina B o Docker foi 3.64% superior ao Apache Mesos. Podemos ainda verificar a moda, que é aproximadamente de 2.5% o que significa que nos testes efetuados, o valor com maior frequência 2,5% verificando-se a superioridade do Docker em relação ao Apache Mesos.

Ao nível do file I/O podemos observar que o para a máquina A o Docker foi superior em 92.53% em relação ao Apache Mesos, sendo que o Docker obteve uma velocidade de transferência de 1238.8 Kb/s e o Apache Mesos de 643.4 Kb/s, enquanto que na

máquina B o Docker foi superior em 34.55%, sendo que o Docker obteve uma velocidade de transferência de 1472.188 Kb/s e o Apache Mesos de 1094.118 Kb/s.

Ao nível de memória através dos resultados analisados podemos verificar que para a máquina A o Docker foi superior em apenas 1.98%, sendo que o Docker obteve uma taxa de transferência de 9152.25 MB/s e o Apache Mesos 8974 Mb/s e para a máquina B o Docker foi superior em 13.07% em relação ao Apache Mesos, sendo que as suas taxas de transferências foram 11835.22 MB/s e 10466.8 MB/s respetivamente.

Ao nível do MySQL o Docker foi inferior ao Apache Mesos em 81.46% para a máquina A em que o Docker apresentou 55987.42 transações/segundo, enquanto que o Apache Mesos exibe 101599.98 transações/segundo e em 23.24% para a máquina B, em que para o Docker foi obtido um valor de 27728.1 transações/segundo e para o Apache Mesos 34174.54 transações/segundo.

4.4 Conclusão

Neste capítulo é descrita a implementação dos testes realizados pelas ferramentas de benchmark (Sysbench).

Os resultados obtidos através da elaboração dos testes das ferramentas de benchmark foram alvo de análise e permitiram a obtenção de uma conclusão, sendo esta que o Docker apresenta um desempenho superior para os testes CPU, File I/O e Memória enquanto que o Apache Mesos apresenta melhor desempenho a nível de base de dados.

Capítulo 5

Conclusão e Trabalho Futuro

5.1 Principais Conclusões

Nesta parte final do trabalho não nos vamos alongar, faremos algumas considerações finais que, embora assim designemos podem servir de ponto de partido para trabalhos futuros.

As revisões de literatura permitiram organizar o conhecimento sobre a problemática "Análise do Desempenho de Containers no Docker versus Containers no Apache Mesos" e investigar diferentes opiniões de autores que contribuem para o desenvolvimento da virtualização.

O objetivo principal deste estudo foi implementar e validar testes de benchmark sobre os sistemas de virtualização. As tecnologias de virtualização Docker e Apache Mesos foram implementadas em infraestruturas e permitem a elaboração de testes, através dos quais é possível obter conclusões para o estudo comparativo entre as duas tecnologias. Os resultados obtidos através da elaboração dos testes das ferramentas de *benchmark* permitiram a obtenção de uma conclusão, sendo esta que o Docker apresenta um desempenho superior para os testes CPU, File I/O e Memória enquanto que o Apache Mesos apresenta melhor desempenho a nível de base de dados.

Ao longo deste estudo e no final da implementação e validação do enquadramento o objetivo foi alcançado.

5.2 Sugestões para Trabalho futuro

No trabalho futuro seria interessante dar continuidade ao estudo, introduzindo outras ferramentas de benchmark, outros tipos de hypervisor e de virtualização com um maior número de testes, mas também ampliar as arquiteturas descritas e implementadas ao longo deste trabalho.

Bibliografia

[1] A. Louro, “Comparação do Desempenho de Infraestruturas Virtualizadas de Elevada Disponibilidade Usando Hypervisors Nativos ou Containers: Microsoft Hyper-V Versus Docker.,” Universidade da Beira Interior, 2018.

[2] RightScale Inc., “STATE OF THE CLOUD REPORT : DevOps Trends,” 2016. [Online]. Disponível: https://www.rightscale.com/lp/state-of-the-cloud.

[3] M. Adams, “Introduction to Virtualization,” 2011. https://docs.oracle.com/cd/E20065_01/doc.30/e18549/intro.htm (último acesso: Jun. 15, 2020).

[4] M. Veras and A. Carissimi, *Virtualização de Servidores*. Rio de Janeiro, 2015.

[5] S. Nanda and T. Chiueh, “A Survey on Virtualization Technologies,” *RPE Rep.*, vol. 179, no. Vm, pp. 1–42, 2005, doi: 10.1.1.74.371.

[6] C. Brenton, “The Basics of Virtualization Security,” *Chem. &*, 2012, [Online]. Disponível: https://cloudsecurityalliance.org/wp-content/uploads/2011/11/virtualization-security.pdf.

[7] J. Rhoton, “Cloud Computing Explained: Implementation Handbook for Enterprises,” 2009.

[8] J. Sahoo, S. Mohapatra, and R. Lath, “Virtualization: A survey on concepts, taxonomy and associated security issues,” in *2nd International Conference on Computer and Network Technology, ICCNT 2010*, 2010, pp. 222–226, doi: 10.1109/ICCNT.2010.49.

[9] M. Laureano, “Máquinas Virtuais e Emuladores: Conceitos, Técnicas e Aplicações,” p. 184, 2006.

[10] C. E. Seo, “Virtualização – Problemas e desafios,” *IBM Linux Technol. Cent.*, vol. 1, no. 008278, pp. 1–19, 2009, [Online]. Disponível: http://www.ic.unicamp.br/~ducatte/mo401/1s2009/T2/008278-t2.pdf.

[11] M. A. B. Monginho, “Estudo do impacto da virtualização de hardware num nó de uma organização distribuída: o estudo de caso da Administração Regional de Saúde do Alentejo,” Centro Universitário Ribeirão Preto - São Paulo, 2012.

[12] Sean Campbell and Michael Jeronimo, “An Introduction to Virtualization,” pp. 1–15, 2006.

[13] P. Enberg, “A Performance Evaluation of Hypervisor, Unikernel, and Container Network I/O Virtualization,” pp. 1–68, 2016.

[14] M. A. B. Monginho, “Estudo do impacto da virtualização de hardware num nó de uma organização distribuída: o estudo de caso da Administração Regional de Saúde do Alentejo,” 2012, [Online]. Disponível: http://comum.rcaap.pt/handle/123456789/6743.

[15] T. R. H. B. W. Evi Nemeth Garth Snyder, “UNIX and Linux System Administration Handbook, 4th Edition,” 2010, [Online]. Disponível: http://gen.lib.rus.ec/book/index.php?md5=CB134B9CF1456FC90908BCD163CF15C2.

[16] N. L. Kelem and R. J. Feiertag, “A separation model for virtual machine monitors,” *Proc. Symp. Secur. Priv.*, pp. 78–86, 1991, doi: 10.1109/risp.1991.130776.

[17] D. B. Gonçalves and J. C. V. Junior, “White Paper – Virtualização,” *DigitalAssets*, 2011.

[18] M. ANDRADE, *Usando o Docker*. São Paulo, 2016.

[19] C. Kallas, “Virtualização,” 2006. http://www.cesarkallas.net/arquivos/faculdade/topicos1/Virtualizacao.doc (último acesso: Sep. 09, 2020)

[20] A. Carneiro, “Projeto de Tecnologias e Sistemas de Informação - Estudo de caso no domínio do green computing,” Universidade do Minho, 2017.

[21] J. K. Waters, “Virtualization Definition and Solutions,” pp. 1–5, 2009.

[22] A. Murphy, D. Marketing, O. Pedro, and S. Operacionais, “Virtualização Esclarecida - Oito Diferentes Modos, 2008. https://pt.slideshare.net/fmcosta70/virtualizacao-esclarecidaoitodiferentesmodoswp (último acesso: Sep. 09, 2020)

[23] D. Kusnetzky, “Virtualization is more than Virtual Machine Software,” 2007, [Online]. Disponível: http://download.microsoft.com/download/0/a/c/0ac57003-473c-4f9a-84b0-8adef6ace753/Virtualization_is_more_than_VM.doc.

[24] FAVACHO, Breno I.; MIRANDA, Daniele S.; OLIVEIRA, Luiz Henrique S. "Análise Comparativa do Desempenho da técnica de Virtualização de Servidor," Universidade da Amazônia, 2008.

[25] G. V. N. Oliveira, "Solução de virtualização completa utilizando VMware e software livre: um estudo de caso na CEF," 2007, [Online]. Disponível: http://repositorio.ufla.br/jspui/handle/1/9267.

[26] L. F, "O que é Virtualização de Aplicações?," 2018. https://blog.diferencialti.com.br/virtualizacao-de-aplicacoes/ (último acesso: Sep. 08, 2020).

[27] Controle Net Tecnologia LTDA, "O que é storage? NAS, DAS, SAN ou FAS?" https://www.controle.net/faq/o-que-e-storage (último acesso: Sep. 08, 2020).

[28] A. B. and B. C. Society, *A Glossary of Computing Terms: by the British Computer Society*. 1998.

[29] VMware, "Understanding Full Virtualization, ParaVirtualization, and Hardware Assist," *Memory*, p. 17, 2007, [Online]. Disponível: www.vmware.com.

[30] G. Neiger, "IntelŴVirtualization Technology: Hardware Support for Efficient Processor Virtualization," *Intel Technol. J.*, vol. 10, no. 03, 2006, doi: 10.1535/itj.1003.01.

[31] T. Bui, "Analysis of Docker Security," 2015, [Online]. Disponível: http://arxiv.org/abs/1501.02967.

[32] J. Strickland, "Como funcionam os servidores virtuais." http://informatica.hsw.uol.com.br/servidor-virtual.htm (último acesso: Jun. 28, 2020).

[33] Y. Gao, H. Wang, and X. Huang, "Applying docker swarm cluster into software defined internet of things," *Proc. - 2016 8th Int. Conf. Inf. Technol. Med. Educ. ITME 2016*, pp. 445–449, 2017, doi: 10.1109/ITME.2016.0106.

[34] S. Hykes, "Docker - Build, Ship, and Run Any App, Anywhere," 2013, [Online]. Disponível: https://www.docker.com/.

[35] M. Eder and H. Kinkelin, “Hypervisor- vs. Container-based Virtualization,” *Netw. Archit. Serv.*, no. July, pp. 1–7, 2016, doi: 10.2313/NET-2016-07-1.

[36] A. Otto, M. Ptl, D. Architect, R. C. Peters, and B. E. Whitaker, “Exploring Opportunities: Containers and OpenStack,” [Online]. Disponível: www.openstack.org.

[37] José Guilherme Vanz, “Apache Mesos,” 2016. https://butecotecnologico.com.br/apache-mesos/ (último acesso: Sep. 08, 2020).

[38] A. Mesos, “Mesos Architecture.” http://mesos.apache.org/documentation/latest/architecture/ (último acesso: Sep. 08, 2020).

[39] N. Yaqub, “Comparison of Virtualization Performance: VMWare and KVM,” UNIVERSITY OF OSLO, 2012.

[40] R. M. Hollander and P. V. Bolotoff, “Ramspeed, a cache and memory benchmarking tool,” 2002. http://alasir.com/software/ramspeed/ (último acesso: Sep. 08, 2020).

[41] W. D. Norcott and D. Capps, “Iozone filesystem benchmark,” *URL: www. iozone. org*, 2003.

[42] Plataforma DevMedia, “Hypervisor: Segurança em ambientes virtualizados.” https://www.devmedia.com.br/hypervisor-seguranca-em-ambientes-virtualizados/30993 (último acesso: Sep. 08, 2020).

[43] Gartner Inc., “Gartner Says 60 Percent of Virtualized Servers Will Be Less Secure Than the Physical Servers They Replace Through 2012,” pp. 2012–2014, 2010, [Online]. Disponível: http://www.gartner.com/it/page.jsp?id=1322414.

[44] A. Kurmus, M. Gupta, R. Pletka, C. Cachin, and R. Haas, “A comparison of secure multi-tenancy architectures for filesystem storage clouds,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7049 LNCS, pp. 471–490, 2011, doi: 10.1007/978-3-642-25821-3_24.

[45] C. Graziano, “A performance analysis of Xen and KVM hypervisors for hosting the Xen Worlds Project,” *Master’s Thesis*, 2011, [Online]. Disponível: http://lib.dr.iastate.edu/etd/12215/.

[46] S. Sridharan, “A Performance Comparison of Hypervisors for Cloud Computing,” 2012.

[47] K. Agarwa, “A study of virtualization overheads,” no. August, p. 56, 2015, [Online]. Disponível: https://search.proquest.com/docview/1747117867?accountid=49346.

[48] I. Voras, M. Orlić, and B. Mihaljević, “An early comparison of commercial and open-source cloud platforms for scientific environments,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7327 LNAI, pp. 164–173, 2012, doi: 10.1007/978-3-642-30947-2_20.

[49] W. Felter, A. Ferreira, R. Rajamony, and J. Rubio, “An updated performance comparison of virtual machines and Linux containers,” *ISPASS 2015 - IEEE Int. Symp. Perform. Anal. Syst. Softw.*, pp. 171–172, 2015, doi: 10.1109/ISPASS.2015.7095802.

[50] V. Adla, “Comparing performance of HyperV and VMware considering Network Isolation in virtual machines,” no. September, 2013.

[51] C. Jiang *et al.*, “Energy efficiency comparison of hypervisors,” *Sustain. Comput. Informatics Syst.*, vol. 22, pp. 311–321, 2019, doi: 10.1016/j.suscom.2017.09.005.

[52] Vmware, “A performance comparison of hypervisors,” 2007. https://www.vmware.com/pdf/hypervisor_performance.pdf (último acesso: Sep. 09, 2020).

[53] P. V. V. Reddy and D. L. Rajamani, “Performance Evaluation of Hypervisors in the Private Cloud based on System Information using SIGAR Framework and for System Workloads using Passmark,” *Int. J. Adv. Sci. Technol.*, vol. 70, pp. 17–32, 2014, doi: 10.14257/ijast.2014.70.03.

[54] L. P. da Silva, “Uso da tecnologia de virtualização para melhor aproveitamento de recursos de hardware,” *FaSCi-Tech*, vol. 1, no. 2, 2010, [Online]. Disponível: http://fatecsaocaetano.edu.br/fascitech/index.php/fascitech/article/view/18/17.

[55] VMware, “Workstation Pro.” https://www.vmware.com/products/workstation-pro.html (último acesso: Sep. 09, 2020).

[56] CoopingDevops, “Installing Mesos and marathon on Red Hat 7,” 2015. http://cookingdevops.blogspot.com/2015/11/installing-mesos-and-marathon-on-red.html (último acesso: Sep. 09, 2020).

[57] V. Adla, “Comparing performance of HyperV and VMware considering Network Isolation in virtual machines,” no. September, 2013.