



Edição Nº 6 – 28 de Abril de 2018

ISSN Print: 1646-9976 | ISSN Online: 2184-223X |

DOI: <https://doi.org/10.31112/kriativ-tech-2018-01-01>

<http://www.kriativ-tech.com>

<http://www.kriativ-tech.pt>

## Virtualização: fundamentos

Pedro Ramos Brandão

Professor Coordenador do ISTECS

ISTEC – Departamento de Estudos e Investigação em Tecnologias de Informação e Sociedade

Investigador da Universidade de Évora – CIDEHUS

[pedro.brandao@istec.pt](mailto:pedro.brandao@istec.pt)

ORCID: <https://orcid.org/0000-0001-6351-6272>

**Resumo:** faremos uma breve resenha sobre o contexto histórico do conceito de virtualização. Tentaremos explicar os principais factos que levam os fornecedores de soluções a optarem pela virtualização, enumeraremos os diversos tipos de virtualização, definiremos os seus componentes e analisaremos questões de segurança relacionadas diretamente com a virtualização.

**Palavras-chave:** *Virtualização, Cloud-Computing, Consolidação de Servidores, OVF.*

**Abstract:** *we will briefly review the historical context of the concept of virtualization. We'll try to explain the key facts that drive solution providers to virtualization, enumerate the various types of virtualization, define their components, and discuss security issues that are directly related to virtualization.*

**Keywords:** *Cloud Computing, Security, ISO 27001, ISO 27002, Security Policies.*

## I. Introdução

A era da informação está “explodindo” à nossa volta, dando-nos acesso a quantidades astronómicas de informações em poucos segundos. Mesmo pessoas que não são

alfabetizadas em computação usam o Facebook para conversar com amigos e familiares, usam o Google para pesquisar uma nova opção de restaurante e imprimir instruções para chegar até lá, ou então utilizam o Tweet para publicitar as suas reações ao mundo que as rodeia. A infraestrutura que suporta estes serviços também está a crescer exponencialmente, e a tecnologia que facilita esse rápido crescimento é a virtualização. Contudo e para além de existir uma relação bastante estreita entre virtualização e *cloud computing* muitas vezes confundem-se os dois conceitos e as duas tecnologias. É também objetivo deste artigo explicitar as diferenças através da especificação do que é a virtualização.

## II. História da virtualização

O conceito de memória virtual remonta ao final de 1950, quando um grupo na Universidade de Manchester introduziu a substituição automática de página no sistema Atlas<sup>1</sup>, uma *mainframe*<sup>2</sup> com transístores. O princípio de paginação como um método para armazenar e transmitir dados para cima e para

<sup>1</sup> <http://www.atlassystems.com/> (Verificada disponibilidade online em 11-08-2016).

<sup>2</sup> <http://www.mainframes.com/whatis.html> (Verificada disponibilidade online em 11-08-2016).

baixo na hierarquia de memória não existia no Atlas. Este foi o primeiro a automatizar o processo, proporcionando assim o primeiro protótipo funcional de memória virtual<sup>3</sup>. O termo “máquina virtual”<sup>4</sup> ou “*virtual machine*” (VM) remonta à década de 60 do século XX. Um dos primeiros sistemas de máquinas virtuais foi criado pela IBM. Por volta de 1967, a IBM introduziu o System/360<sup>5</sup> modelo 67, o seu primeiro principal sistema com memória virtual integral. O modelo 67 foi o primeiro a integrar o conceito de autovirtualização, conseguida através de um conjunto de instruções do processador, que foi aperfeiçoado em modelos posteriores [1]. O modelo 67 usava um sistema operativo denominado CP-67, que evoluiu para os sistemas operativos que permitiriam a criação de máquinas virtuais com sistemas operativos diferentes [2]. O objetivo era que a *mainframe* de *hardware* cooperasse com várias instâncias de qualquer sistema operativo, sendo o acesso a cada um deles protegido por um conjunto de instruções. Em meados dos anos 60 do século XX, a IBM também foi pioneira no projeto M44/44X [3], em que explorou o conceito de partilha de tempo emergente. No núcleo da arquitetura do sistema era criado um conjunto de máquinas virtuais, com uma máquina para cada utilizador. A máquina principal para este desenvolvimento foi um IBM 7044<sup>6</sup> e cada máquina virtual era uma imagem [4] do 7044. Este trabalho levou ao desenvolvimento do IBM VM/370, amplamente usado em sistemas de *timesharing*. O conceito de virtualização de *hardware* também surgiu neste período, permitindo que um computador executasse máquinas virtuais num ambiente isolado e protegido.

<sup>3</sup> <http://windows.microsoft.com/pt-pt/windows/what-is-virtual-memory#1TC=windows-7> (Verificada disponibilidade online em 11-08-2016).

<sup>4</sup> <http://www.i-tecnico.pt/o-que-e-uma-maquina-virtual-e-para-que-serve/> (Verificada disponibilidade online em 11-08-2016).

<sup>5</sup> <https://computadoribm360.wordpress.com/> (Verificada disponibilidade online em 11-08-2016).

<sup>6</sup> <http://www.columbia.edu/cu/computinghistory/704.html> (Verificada disponibilidade online em 11-08-2016).

O monitor da máquina virtual era transparente para o *software* em execução, o *software* “pressupunha” que tinha o controlo exclusivo do *hardware*. Este processo foi aperfeiçoado ao longo tempo, dando origem às máquinas virtuais atuais; contudo, naquela altura funcionava com baixo desempenho [5].

Em meados dos anos 70 do século passado, a virtualização foi bem aceite pelos utilizadores de vários sistemas operativos e pelas empresas. O uso da virtualização durante essa década resolveu alguns problemas importantes. Por exemplo, o aparecimento do armazenamento virtual, em larga escala, de sistemas operativos permitiu contornar problemas de falta de memória física que existiriam se os sistemas não fossem virtuais. A virtualização do armazenamento expandiu a capacidade dos sistemas e tornou os programas menos complexos e muito mais produtivos. As máquinas virtuais apresentavam uma forma eficiente de beneficiar ao máximo daquilo que era então um investimento considerável nos centros de dados de uma empresa: o *hardware*. Apesar de a virtualização ser popular, tanto na área da pesquisa como no mercado comercial entre 1960 e 1970, esse interesse praticamente desapareceu durante os anos 80 e 90 [6]. Uma das causas, em geral, foi o aparecimento dos minicomputadores de baixo custo e dos computadores pessoais e a sua massificação ao nível de mercado [7].

A partir dos anos 90, duas empresas desenvolveram sistemas de virtualização que merecem ser referidos, o *Java Virtual Machine* (JVM) da Sun Microsystems (atualmente da Oracle)<sup>7</sup> e o *Common Language Runtime* (CLR) da Microsoft<sup>8</sup>.

A experiência com a plataforma MAC e PC levou a Connectix a criar um produto

<sup>7</sup> <http://docs.oracle.com/javase/specs/jvms/se7/html/> (Verificada disponibilidade online em 11-08-2016).

<sup>8</sup> [https://msdn.microsoft.com/en-us/library/8bs2ecf4\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/8bs2ecf4(v=vs.110).aspx) (Verificada disponibilidade online em 11-08-2016).

chamado Connectix Virtual PC 1.0<sup>9</sup> para MAC. O Virtual PC 1.0 foi um verdadeiro avanço em termos de programação, incorporou um mecanismo de tradução de binários para instruções de um processador *Intel x86* virtual e para um processador *PowerPc* físico, utilizado no MAC, permitindo assim a virtualização de diversos sistemas operativos sob máquinas em que o sistema operativo base era MAC. Este exemplo de emulação levou a Connectix à tecnologia da virtualização.<sup>10</sup>

Fabricantes de *chips* como a Intel e AMD continuaram a introduzir novas tecnologias para dar um melhor suporte para a virtualização na camada de *hardware*. Estas tecnologias incluem processadores multinúcleos, tecnologia de virtualização Intel (originalmente conhecida como Vanderpool e Silverbale)<sup>11</sup>, e AMD-V/SVM (originalmente conhecida como pacífica). Estas tecnologias de virtualização ao nível de *hardware* permitiram às plataformas de virtualização tornarem-se mais eficientes, dando origem aos chamados virtualizadores do Tipo Um, em que as máquinas virtuais podiam comunicar diretamente com o *hardware* de forma mais fácil. As tecnologias Intel-VT e AMD-V incorporavam estas instruções e passaram a controlar o *hypervisor*, para não haver a necessidade de uma camada de *software* complexa, podendo ocasionar problemas de desempenho, originados pela não correta comunicação da VM com o *hardware*. As instruções de virtualização adicionadas nos processadores AMD e Intel têm ajudado a criar novas plataformas de virtualização<sup>12</sup>, como por exemplo a virtualização de armazenamento e de dispositivos intermédios de rede.

<sup>9</sup> <http://en.wikipedia.org/wiki/Connectix> (Verificada disponibilidade online em 11-08-2016).

<sup>10</sup> Ibidem.

<sup>11</sup> <http://www.pcpro.co.uk/news/enterprise/62982/idf-fall-2004-silverbale-to-virtualise-server-processing> (Verificada disponibilidade online em 11-08-2016).

<sup>12</sup> MARSHALL, David, REYNOLDS, Wade A. & MCCRORY, Dave. *Advanced Server Virtualization – VMware and Microsoft Platforms in the Virtual Data Center*. Auerbach Publications, 2006. (Verificada disponibilidade online em 11-08-2016).

### III. Definição de Virtualização

Virtualização é a criação de uma abstração de algo que existe fisicamente, como por exemplo, um sistema operativo, um servidor, dispositivos de alojamento ou dispositivos de rede, etc [8].

A virtualização de um servidor muda as regras do modelo tradicional de funcionamento do sistema físico, em que um servidor físico faz o papel de servidor hospedeiro, onde está instalado um *hypervisor* e onde são instaladas várias máquinas virtuais.

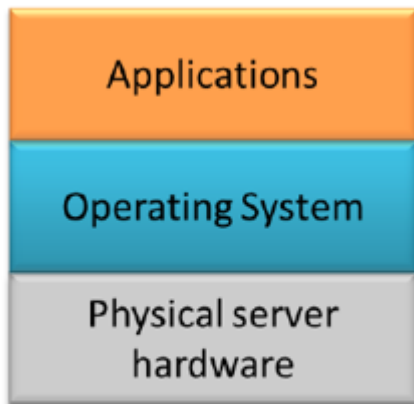
A virtualização permite a criação de vários recursos virtualizados a partir de recursos físicos. Esta forma de virtualização maximiza a utilização dos recursos [9]. A virtualização, como já referido, não é CC, mas facilita substancialmente o estabelecimento e gestão de *Cloud Computing* (CC) [10]. A virtualização pode definir-se como uma camada abstrata, e pode existir como parte ou um todo do *IT Stack*. Por outras palavras, a virtualização pode ser representada como o processo de implementação de um conjunto de tecnologias capazes de camuflar as características físicas dos recursos dos servidores, recursos de redes, e recursos de alojamento, da forma como habitualmente os sistemas, aplicações ou utilizadores finais interagem com esses referidos recursos [11].

### IV. Virtualização em termos técnicos

#### IV – A – Virtualização Servidores

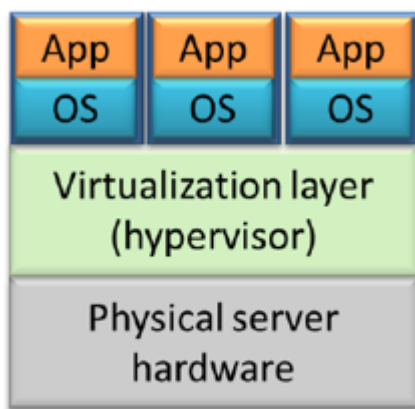
A virtualização do servidor é o principal domínio da virtualização onde um servidor (físico) é convertido num servidor virtual. O termo “servidor físico” é muitas vezes usado por fornecedores de virtualização para descrever a diferença entre um servidor virtual e um servidor físico [12]. O servidor físico refere-se ao *hardware* que faz o processamento de computação real imposta pelo *software*, tal como o sistema operativo e aplicações. Um servidor virtual não pode operar sem um servidor físico. Com a virtualização de servidor, múltiplos servidores físicos podem ser convertidos em

servidores virtuais e serem colocados sobre um servidor físico, a que é chamado o hospedeiro. Os servidores virtuais, por sua vez, são chamados os convidados. Na Imagem 1 e 2 é representada, de modo simples, a diferença entre uma arquitetura de servidor tradicional e uma arquitetura de servidor virtual.



### Traditional Architecture

Imagem 1: arquitetura tradicional.  
Fonte: Microsoft – TechNet (2014)



### Virtual Architecture

Imagem 2: arquitetura virtual.  
Fonte: Microsoft – TechNet (2014)

Tanto a arquitetura tradicional como a virtual mostram a sua arquitetura de servidor. Na camada inferior, ambas as arquiteturas são compostas por um conjunto de computação de *hardware*, como uma unidade central de processamento (CPU), memória, placa de rede (NIC) e um disco rígido local. O *hardware* é colocado dentro de uma “caixa” ou invólucro e é chamado de servidor físico, no qual o *software*, um sistema operativo (SO) e as aplicações, é instalado no topo. No entanto, a instalação do SO ou das aplicações numa arquitetura de servidor tradicional é muito vulnerável a mudanças ou falhas na camada de *hardware*. As alterações a falhas de configuração de *hardware* ou de *hardware* podem imediatamente resultar num mau funcionamento do SO, o que significa que o servidor físico precisa de ser reparado com a mesma configuração de *hardware* ou então requer uma reinstalação do SO e das aplicações.

Com a virtualização de servidores, uma camada de virtualização é colocada acima do *hardware* do servidor físico e sob o SO e as camadas de aplicação. A camada de virtualização torna possível a instalação de um conjunto múltiplo de instâncias de SO e aplicações num servidor físico. Cada conjunto do SO tem aplicações e funções semelhantes à arquitetura de servidor tradicional, com a diferença de executar várias instâncias num servidor físico em vez de apenas uma instância. Além disso, a camada de virtualização isola cada conjunto um do outro, o que faz com que uma instância não seja afetada por falhas ou alterações de outras instâncias ou hardware. Todas as instâncias “julgam” ser a única instância no servidor físico, e desconhecem as outras instâncias virtuais. Estas instâncias são denominadas máquinas virtuais ou servidores virtuais. Como foi mostrado na Imagem 2, a virtualização desacopla o servidor do software (SO e aplicações) a partir de um determinado conjunto de *hardware*, o que faz com que seja independente de uma

configuração específica de *hardware* que é necessária, a fim de funcionar. Desta forma, uma máquina virtual pode operar em servidores físicos que usam diferentes configurações de *hardware* [13].

Outra característica da virtualização de servidor é que um servidor físico tradicional, com uma configuração específica de um SO e aplicações, pode ser convertido num servidor virtual ou máquina virtual. Alguns autores definem máquina virtual como a duplicação eficiente e isolada de uma máquina real. Por máquina real entenda-se uma arquitetura de servidor tradicional, com um único SO e aplicações. No entanto, a utilização de máquinas virtuais com a virtualização de servidores vai além da duplicação de uma máquina real [14].

#### IV – B – Máquina Virtual

Uma máquina virtual é a representação virtual de um servidor físico ou computador, composto por um SO e uma ou mais aplicações. Podemos representar de forma simples uma máquina virtual como ilustra a (Imagem 3).



Imagem 3: Representação Gráfica de uma máquina virtual - Fonte: Microsoft - TechNet (2014)

Na (Imagem 3) são mostradas duas representações de uma máquina virtual: uma está ilustrada com uma caixa com o ícone OS (*operating system* ou SO) e aplicação (app); e a outra com a abreviatura VM. Uma máquina virtual é tipicamente composta por um único arquivo ou um grupo de arquivos que podem ser lidos e executados pela camada de virtualização.

Cada máquina virtual constitui um ambiente de funcionamento independente que se comporta como se fosse um computador separado. As máquinas virtuais diferentes não estão, pois, “cientes” umas das outras. São construídas de tal maneira que ficam isoladas, o que significa que não têm conhecimento acerca de outras máquinas virtuais estarem presentes no mesmo servidor físico [15]. Uma máquina virtual usa a emulação para imitar um conjunto completo de *hardware*, como CPU, memória, placa de rede, etc. Isto é feito por meio de um conjunto de *drivers*, que são compatíveis com diferentes tipos de *hardware*. Os *drivers* são construídos numa máquina virtual, de forma a poderem ser usados em diferentes configurações de *hardware*. Com este tipo de *drivers* uma máquina virtual gera uma versão virtual do *hardware* físico e cria um CPU virtual, uma memória virtual, uma placa virtual de interface de rede (NIC), um disco rígido virtual e outros tipos de *hardware* que podem ser necessários. Quando uma máquina virtual é iniciada, uma certa quantidade de capacidade do processador CPU, memória e espaço em disco são atribuídos automaticamente pela camada de virtualização ou *hypervisor*. Para implementar uma máquina virtual, é adicionada uma camada de virtualização (*hypervisor*) a um servidor físico para suportar a arquitetura desejada. Ao fazermos isso, a máquina virtual pode contornar a compatibilidade de *hardware* e limitações de recursos [16].

#### IV – B – Máquina Virtual

Um *hypervisor*, também conhecido como o monitor de máquina virtual (em inglês, Virtual Machine Manager – VMM), é a camada de série de software que permite que várias máquinas virtuais ou sistemas operativos (SO) possam operar num único servidor físico [17]. Existem dois tipos de *hypervisors*: um denominado "tipo 1" e outro "tipo 2". O tipo 1 é um *hypervisor* que é instalado diretamente no hardware e que também é chamado de *hypervisor* "bare-metal". Os *hypervisors* do tipo 1 estão posicionados entre o hardware e as máquinas virtuais. O *hypervisor* representado na Imagem 4 é um *hypervisor* do tipo "bare metal". Os *hypervisors* do tipo 1 são predominantemente utilizados no

mercado de servidores [18]. O tipo 2 é um *hypervisor* que está instalado no topo de um SO e é também designado por *hypervisor* "hospedeiro". Ao contrário do tipo 1, o *hypervisor* é colocado por "cima" do SO e não abaixo dele. Isto permite ter um SO adicional, a ser executado num ambiente virtual em "cima" de um SO existente. Os *hypervisors* hospedeiros podem ser utilizados para executar um tipo diferente de SO em "cima" do outro SO, como por exemplo permitem executar um sistema Linux num sistema Windows e vice-versa [19]. Na Imagem 4 são apresentados os dois tipos de *hypervisors*.



Imagem 4: *Bare metal hypervisor* –  
Fonte: Microsoft - TechNet (2014)

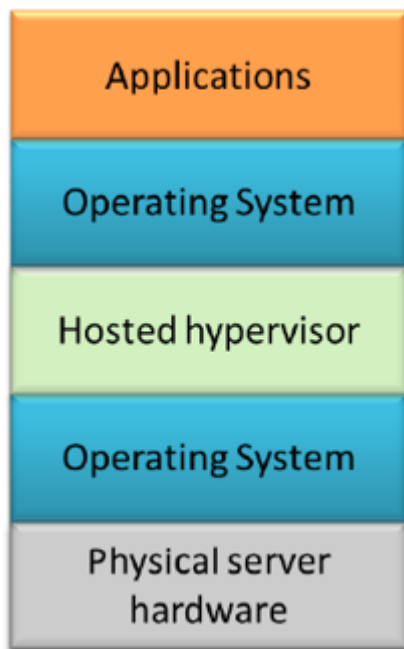


Imagem 5: *Hosted hypervisor* –  
Fonte: Microsoft - TechNet (2014)

Os *hypervisors* são diretamente responsáveis pela hospedagem e administração das máquinas virtuais no host ou servidor físico. Anfitrião é outro nome para o servidor físico e *hypervisor*. As máquinas virtuais que são executadas no hospedeiro são chamadas máquinas virtuais convidadas ou SO convidados [20]. Além disso, um *hypervisor* fornece uma visão uniforme do hardware subjacente, o que significa que ele pode operar em hardware de diferentes fornecedores. Por isso, as máquinas virtuais podem ser executadas em todos os computadores disponíveis e que suportem a virtualização; na prática, o *hypervisor* isola o software do hardware [21]. Os administradores de sistema, que mantêm e operam computadores numa rede, também são capazes de visualizar o seu hardware como um conjunto de recursos que permite novas funcionalidades.

Os *hypervisors* estão equipados com diversas tecnologias diferentes, que variam dependendo do fornecedor de virtualização. No entanto, existem algumas tecnologias comuns que são amplamente conhecidas e utilizadas pelos diferentes fornecedores de virtualização e que trazem as características e benefícios da virtualização de servidores [22]. As características comuns dos *hypervisors* são: *High Availability* (HA) ou alta disponibilidade, "tolerância a falhas", *Live Migration* ("migração ao vivo"), *Distributed Resource Scheduler* (DRS) e *Distributed Power Management* (DPM) (VMware, DRS and DPM, s.d.). Os dois últimos atributos são especificamente utilizados pela VMware, o primeiro industrial a desenvolver estas duas tecnologias. A alta disponibilidade é uma tecnologia que monitoriza continuamente todas as máquinas virtuais em execução na *pool* de recursos virtuais, procurando uma falha de hardware (VMware, *High availability*, s.d.) A *pool* de recursos virtuais, por seu turno, é um conjunto de recursos ou de servidores físicos que executam máquinas virtuais.

#### IV – C – Virtualização de aplicações

A virtualização de aplicações é composta por tecnologias que isolam as mesmas do SO.

Com a virtualização de aplicações, uma aplicação é empacotada num único executável ou num conjunto de arquivos que podem ser distribuídos de forma independente do SO. Existem diferentes tipos de virtualização de aplicações, sendo os mais comuns a aplicação *sandbox* e *streaming*. As aplicações do tipo *sandbox* são completamente isoladas no que é vulgarmente chamado de "bolha", que é encapsulada em relação ao SO subjacente. Nenhuma instalação ou instalação de driver adicional é necessária, o que elimina conflitos de dependência. Todos os recursos do SO necessários para a execução da aplicação já estão incorporados no arquivo executável. Não há dados da aplicação ou arquivos armazenados no SO. Sempre que o utilizador inicia a aplicação, recebe uma cópia "limpa" da mesma, e o arquivo executável pode ser visto como uma imagem da aplicação. Dependendo dos direitos do utilizador, podem ser construídas alterações no executável da aplicação. As aplicações *sandbox* podem ser facilmente distribuídas através de vários meios. Por exemplo, através de um servidor ou numa *pen* USB [23]. No caso de aplicações do tipo *streaming* estas são divididas em vários pacotes. Com o *streaming*, a aplicação é armazenada num servidor central que transmite a mesma para a localização do utilizador. Apenas os dados da aplicação que é necessária serão transmitidos para o utilizador [24].

Por exemplo, quando um utilizador necessita de usar um programa do Office como o *Word*, o servidor não irá transmitir toda a suite do Office. Apenas o pacote da aplicação *Word* lhe será transmitido. Na (Imagem 6), visualizamos exemplos de virtualização de aplicações. As aplicações estão numa área restrita e são armazenadas num servidor central. As caixas cinzentas representam servidores físicos nos quais as máquinas virtuais podem ser instaladas. Os servidores estão conectados a um sistema de armazenamento a partir de onde os pacotes de aplicações podem ser transmitidos para os utilizadores. As bolhas em torno das aplicações *sandbox* indicam que elas estão isoladas doutras aplicações, bem como do SO [25].

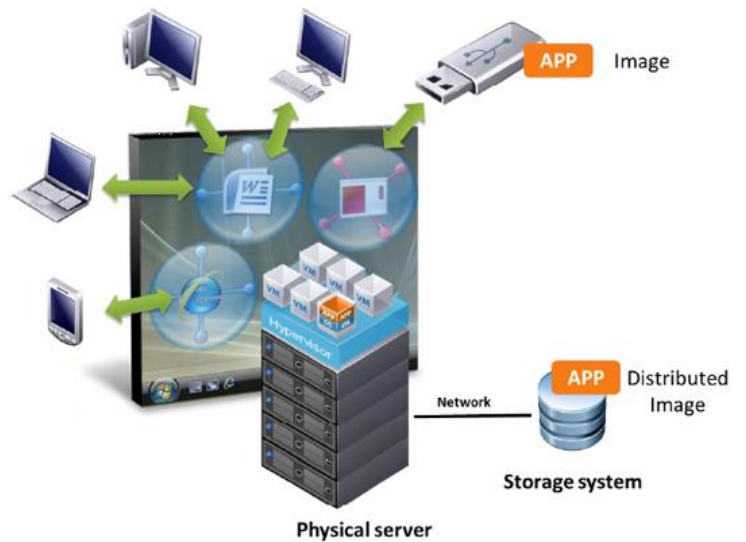


Imagem 6: Virtualização de aplicações –  
Fonte: VMware

## V. Normas padrão

O *Open Virtualization Format* (OVF) é um modelo-padrão. É um dos poucos desenvolvidos especificamente para a virtualização e, provavelmente, o mais importante dos modelos-padrão aplicados a esta tecnologia. Foi desenvolvido pela DMTF, tendo sido também aceite e aprovado pela ISO/IEC. A primeira versão deste modelo-padrão foi lançada em 2010, a aprovação pela ISO ocorreu em 2011, e a norma ISO é a 172703:2011.

O modelo OVF torna explícita a virtualização e aponta para esta e para os serviços de empacotamento virtuais. O OVF descreve em *open source*, de forma segura, portátil, eficiente e extensível o formato da embalagem e distribuição de (coleções) de máquinas virtuais. As principais propriedades deste formato são as seguintes:

- Distribuição otimizada: suporta a verificação de conteúdo e a verificação de integridade baseada no padrão da indústria, com infraestrutura de chave pública, e fornece um esquema básico para gestão de licenças de software;

- Uma experiência de utilizador otimizada para ser simples e automatizada: suporta validação de todo o pacote de medidas de cada máquina virtual ou metadados componentes do OVF, durante as fases de instalação da máquina virtual, e do processo de gestão do ciclo. Ele também empacota com legibilidade informações relevantes para o utilizador, com descrições que podem ser utilizadas por uma plataforma de virtualização para otimizar a experiência de instalação;
- Suporte para configurações de uma ou de várias máquinas virtuais em simultâneo: contém possibilidades de configuração de uma simples máquina virtual, mas também contém a possibilidade de configurações complexas de múltiplas máquinas virtuais com interdependências;
- Pacotes portáteis de máquinas virtuais: é uma plataforma de virtualização neutra, que suporta uma gama completa de formatos de discos rígidos virtuais, usados para máquinas virtuais de hoje, e é extensível para lidar com futuros formatos que possam surgir. As propriedades das máquinas virtuais são capturadas de forma concisa e precisa. As máquinas virtuais podem ser exportadas para outras plataformas;
- Independência das plataformas dos fornecedores: não se baseia no uso de um host específico, ou plataforma de virtualização específica, nem de um SO específico na função *guest*;
- Extensibilidade: é imediatamente útil e extensível. Suporta e permite a codificação de metadados para apoiar mercados verticais específicos;
- Open Standard: o OVF surgiu da colaboração entre os principais fornecedores do setor, e foi desenvolvido como um futuro padrão para equipamentos portáteis e máquinas virtuais.

Este modelo interage com modelos como o CIMI e OCCI, e estes modelos-padrão, através das suas interfaces, simplificam a construção e implementação de plataformas para pacotes com base no OVF e disponibilizam um padrão de gestão de interface para os serviços instalados a partir do referido protocolo.

A base fundamental do OVF é um conjunto de regras de máquinas virtuais estruturadas como uma coleção hierárquica. A

criação de um pacote OVF define a instalação de regras para máquinas virtuais e outras estruturas que irão suportar o serviço. O OVF encaixa-se neste padrão e oferece certa flexibilidade como suporte para implementações alternativas e para as opções dos consumidores no momento de implantação.

## VI. Cloud Computing

Em linguagem comum Cloud Computing (CC) é um conjunto de recursos computacionais disponibilizados a um conjunto de utilizadores, de forma remota, tomando estes a forma de serviços. E como qualquer serviço nos dias de hoje, eletricidade, água, gás, etc., estão sempre disponíveis quer para uso individual quer para uso empresarial. Isto é possível devido ao facto de a indústria tecnológica ter ao longo da segunda metade do século XX adotado um conjunto de modelos padrão oriundos de diferentes fontes e plataformas tecnológicas. O CC funciona como um serviço universal, pago, mas sempre disponível para os utilizadores/clientes que dele necessitem, exatamente da mesma forma de dispõem do serviço de eletricidade.

O conceito de CC surge na sequência de outro conceito que se denominava *time-sharing*, e que implicava uma partilha por parte de várias entidades, em tempos diferentes, o mesmo equipamento de computação. Contudo, hoje em dia, existem substanciais diferenças entre o conceito de *time-sharing* computacional e CC. Na altura do *time-sharing* computacional, os serviços ou máquinas que os detinham, só podiam ser utilizados por um operador de cada vez, ou seja, eram repartidos por porções de tempo e não por porções de espaço de alojamento; hoje a questão da partilha no tempo não existe, paga-se em função do serviço que se obtém ou do espaço que a informação ocupa, mas não em função do tempo em que se está a utilizar esse recurso. Mas a ideia base é muito idêntica, é disponibilizar um serviço de computação de forma remota e não localmente. Utilizando por analogia conceitos simples e diários, podemos dizer que tanto o sistema de *time-sharing* computacional como o de CC,

podem ser vistos como um serviço que é disponibilizado em moldes idênticos à eletricidade ou à água, ou seja, estão disponíveis quando deles necessitamos, pagamos por essa disponibilidade e pela sua utilização, por exemplo, se utilizarmos o Office 365<sup>13</sup> pagamos pela sua utilização, como quando consumimos eletricidade, mas também podemos pagar pelo alojamento dos documentos produzidos pelo mesmo, como pagamos se quisermos ter um sistema de armazenamento de energia de reserva.

Inicialmente, anos 60 do século XX, os sistemas computacionais estavam em grandes Salas com poderosos e dispendiosos sistemas de arrefecimento e com um elevado consumo de energia elétrica. Para, além disto, só podiam ser utilizados localmente. O passo seguinte foi alterar esta situação através de um sistema que permitisse acesso remoto a um computador central, é a partir daqui que podemos falar em *time-sharing*.

As organizações podiam comprar tempo de utilização de um sistema computacional, sem terem que o ter nas suas instalações, e sem terem de se preocupar com a sua manutenção e com a sua administração.

Os microcomputadores começaram a aparecer quando o conceito de *time-sharing* já era popular. Os microcomputadores eram mais pequenos que os mainframes e eram excepcionalmente mais baratos. Rapidamente os microcomputadores tornaram-se mais sofisticados que as antigas *mainframes*, mas mais importante é que os microcomputadores tinham capacidades do tipo multitarefa, permitindo ainda o *time-sharing*.

Rapidamente o conceito de *time-sharing* e os microcomputadores levaram à criação de um novo conceito, o de sistemas distribuídos.

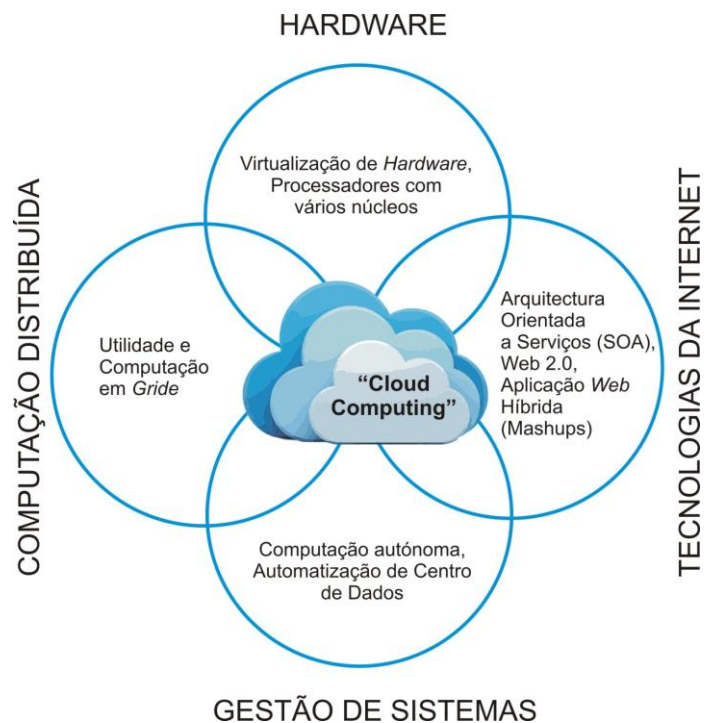


Imagem 7: Convergência de diversas tecnologias que dão funcionalidade tecnológica à Cloud Computing.

A evolução da tecnologia *Cloud Computing* esteve sempre relacionada com a evolução desta lista de diferentes tecnologias. Este modelo pode trazer consideráveis benefícios, quer em termos de custos, quer na possibilidade de os utilizadores dos diversos serviços adaptarem os componentes do modelo às suas necessidades.

## VII. Conclusão

Por um lado, a virtualização nada mais é do a utilização cada vez mais eficiente dos recursos existentes, que proporciona uma enorme economia de custos num curto período de tempo. Por outro lado, oferece às organizações novos modelos de implantação de aplicações para maiores períodos de tempo de disponibilidade a fim de atender às expectativas dos utilizadores, caracterizada por pacotes modulares para fornecer novos serviços em minutos em vez de semanas e recursos avançados que proporcionam balanceamento de carga automático, escalabilidade sem tempo de inatividade, autocorreção, armazenamento de serviços e muitos outros recursos para suportar aplicações

<sup>13</sup> [http://news.cnet.com/8301-10805\\_3-20072888-75/microsoft-launching-office-365-on-june-28/](http://news.cnet.com/8301-10805_3-20072888-75/microsoft-launching-office-365-on-june-28/)

essenciais aos negócios que melhoram a arquitetura tradicional.

O principal princípio que suporta esta tecnologia e este modelo é a oferta de computação, armazenamento e *software* como um serviço. “Cloud é um sistema distribuído e paralelo de computação, consistindo numa coleção de computadores virtualizados e interligados que são apresentados como um conjunto de recursos dinâmicos e unificados, baseados em *Service Level Agreement (SLA)*<sup>14</sup>, estabelecido entre o prestador do serviço e o cliente final”<sup>15</sup>.

Assim podemos concluir que a virtualização é uma abstração computacional que implementada tecnologicamente permite emular sofisticadamente o hardware. Enquanto que *Cloud Computing* é uma tecnologia baseada em computação distribuída permitindo o uso especializado dos sistemas que efetivamente necessitamos sem perdas de recursos e oferecendo um sistema de alta redundância que se designa por geo-redundância.

Contudo, hoje em dia, todos os sistemas de *Cloud Computing* utilizam a virtualização.

---

<sup>14</sup> Um Acordo de Nível de Serviço (ANS ou SLA, do inglês *Service Level Agreement*) é um acordo firmado entre a área de TI e o seu cliente interno, que descreve o serviço de TI, as suas metas de nível de serviço, além dos papéis e responsabilidades das partes envolvidas no acordo. Na atual versão da biblioteca ITIL (versão 3), o ANS insere-se no contexto dos processos de Desenho de Serviço (*Service Design*), especificamente no processo SLM (*Service Level Management*, ou *Gestão de Nível de Serviço*). O SLM envolve a negociação, o acordo e a apropriada documentação de níveis de serviço que atendam as necessidades do negócio, permitindo a entrega de serviços de TI com a qualidade esperada. É exatamente nesse contexto que o ANS/SLA tem fundamental importância, pois é nele que estarão definidos, aceites e formalizados os níveis de serviço esperados pelo cliente de TI.

<sup>15</sup> BUYYA, Rajkumar; Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, *Future Generation Computer Systems*, 25:599-616, 2009

## XIV. Referências

- [1] G. MELL, “The NIST Definition of Cloud Computing version 15,” National Institute of Standards and technology, Information Technology Laboratory, USA, 2009.
- [2] V. WINKLER, *Securing The Cloud*, USA: Syngress, 2011.
- [3] H. SWANSON, “NIST Special Publication 800-18 - Guideline for Developing Security Plans for Federal Information Systems,” US Department of Commerce, USA, 2006.
- [4] E. SIRON, *Hyper-V Security*, USA: PACKT, 2014.
- [5] ISO-27001, “ISO 27001,” [Online]. Available: <https://www.27001.pt>.
- [6] ISO-27002, “ISO 27002,” [Online]. Available: <http://www.iso27001security.com/html/27002.html>.
- [7] N. POOL, “NTP POOL PROJECT,” NTP Portugal, 2018. [Online]. Available: <http://www.pool.ntp.org/zone/pt>. [Acedido em 2018].
- [8] R. ROSS, “Recommended Security Controls for Federal Information Systems,” NYST - Computer Security Division, USA, 2007.
- [9] O. Security, “Open Security Architecture,” 2011. [Online]. Available: <http://www.opensecurityarchitecture.org>. [Acedido em 20 Março 2018].
- [10] G. HARRISON, *Inside a Cloud*, USA: Capitol Records, 2002.
- [11] C. GIBSON, “Time-Sharing in the IBM System/360: Model 67,” em *Sping Joint Computer Conference*, 1966.
- [12] V. TRAVASSOS, “Virtualization Trends Trace Their Origins Back to the Mainframe,” Agosto 2012. [Online]. Available: [http://www.ibmssystemsmag.com/mainframe/administrator/Virtualization/history\\_virtualization/](http://www.ibmssystemsmag.com/mainframe/administrator/Virtualization/history_virtualization/).
- [13] A. SINGH, “An Introduction to Virtualization,” Janeiro 2004. [Online]. Available: <http://www.kernelthread.com/publications/virtualization/>.
- [14] IBM, “IBM SearchManager/370 Client for Windows,” [Online]. Available: [http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?infotype=dd&subtype=sm&apiname=S hopzSeries&htmlfid=897/ENUS5695-070#Header\\_6](http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?infotype=dd&subtype=sm&apiname=S hopzSeries&htmlfid=897/ENUS5695-070#Header_6).
- [15] H. CANDIDO, “História da Virtualização,” 19 Maio 2007. [Online]. Available: <https://caiocandido.wordpress.com/2007/05/19/conceitos-de-virtualizacao/>.

- [16] S. CONROY, “History of Virtualization,” 2010. [Online]. Available: <http://www.everythingvm.com/content/history-virtualization>.
- [17] V. JOSYULA, “Virtualization,” em *Cloud Computing*, Indianapolis, Cisco Press, 2012.
- [18] G. PAGE, “Cloud Computing,” em *Cloud Computing*, Indianapolis, Cisco Press, 2012.
- [19] S. SCHUMATE, “Implications of Virtualization - Technical Report,” DELL, 2014.
- [20] G. POPEK, “Formal Requirements for Virtualization Third Generation Architectures,” ACM, 1974.
- [21] J. SMITH, “Architectures of Virtual Machines,” *IEEE Computer*, vol. 38, Maio 2005.
- [22] J. S. REUBEN, “A Survey on Virtual Machine,,” Helsinki University of Technology, Helsinki, 2007.
- [23] B. MADDEN, “Hypervisors,” 9 Março 2010. [Online]. Available: <http://blog.sharevm.com/2010/03/09/type-1-and-type-2-client-hypervisors>.
- [24] J. A. RIZO, “Emulator,” [Online]. Available: Listing os Virtualization and Remote Control Solutions: Running Windows or Linux Software on a Macintosh and Running Mac OS on other Platforms.
- [25] R. P. GOLDBERG, “Architectural Principles for Virtual Computer Systems,,” Harvard University Press, 1973.
- [26] P. COTTON, “Sun x VM,,” 2008. [Online]. Available: [http://www.sun.com/bigadmin/sundocs/articles/xvm\\_hvsrovw.jsp](http://www.sun.com/bigadmin/sundocs/articles/xvm_hvsrovw.jsp).
- [27] N. COCHRANE, “Stacking up the hypervisors,” 2010. [Online]. Available: <http://www.crn.com.au/Feature/173728,stacking-up-the-hypervisors.adpx>.
- [28] VMware, “DRS and DPM,,” [Online]. Available: <http://www.vmware.com/products/drs>.
- [29] VMware, “High availability,,” [Online]. Available: <http://www.vmware.com/products/high-availability/>.
- [30] J. SANTOS, Artist, *Security Challenges with Virtualization*. [Art]. FCUL.
- [31] LANDESK, “Application Virtualization,,” 2007. [Online]. Available: <http://www.landesk.com/WorkArea/downloadasset.aspx?id=2393>.