A importância da virtualização

A computação em nuvem é resultado da evolução natural e da união de várias outras tecnologias da área de TI, sendo que a definição da computação em nuvem tem como foco principal a transformação da rotina tradicional de como empresas e usuários finais utilizam e adquirem os recursos da tecnologia da informação (TI). Isto é, toda a infraestrutura de TI (hardware, software e gestão de dados e informação), até então tratada como um ativo das empresas, passa a ser administrada pelos provedores da computação em nuvem e acessada pelas empresas e usuários por meio da internet.

Esse acesso remoto é possível de ser realizado de qualquer tipo de equipamento – celulares, notebooks, tablets, computadores etc. Dessa maneira, os provedores de nuvem passam a prover a infraestrutura e os serviços capacitados para atender a essa demanda. Trata-se de um modelo eficiente para utilizar softwares, acessar, armazenar e processar dados por meio de diferentes dispositivos e tecnologias web.

O formato proposto pela computação em nuvem reúne conceitos, tendências e recursos trabalhados na área de TI, como virtualização, conteinerização, computação sem servidor (serverless computing), application service provider (ASP), grid computing, utility computing e software como serviço. Essas tecnologias habilitam a computação em nuvem, tornando-se parte fundamental da arquitetura dos provedores de nuvem.

Virtualização

O termo virtualização tem origem no conceito “virtual”, ou seja, algo abstrato que simula as características de algo real. Esse conceito surgiu na década de 1960, sendo mais divulgado na década posterior. Porém, as limitações tecnológicas da época impediram maiores avanços dessa inovadora tecnologia para a época. Após a chegada da internet, a possibilidade de processar informações e executar operações com o acesso remoto impulsionaram a virtualização e seus recursos.

A virtualização é a tecnologia que permite que diversas aplicações e sistemas operacionais sejam processados em uma mesma máquina.

Dessa maneira, foi possível o datacenter das empresas trabalhar com inúmeras plataformas de sistemas operacionais, sem a necessidade do aumento no número de servidores físicos. Ou seja, a virtualização permite um alto nível de flexibilidade e portabilidade.

Esse tipo de tecnologia permite compartilhamento dos recursos de hardware como processador, memória, interface de rede, disco rígido, entre outros, da máquina física com as máquinas virtuais ali presentes. Todo o gerenciamento e a alocação de recursos de hardware de uma máquina virtual é feito pelo hypervisor ou monitor de máquina virtual (virtual machine monitor – VMM). O hypervisor é uma camada de software localizada entre a camada de hardware e o sistema operacional.

Exemplo

Um usuário utiliza o sistema operacional Windows em seu computador, mas deseja utilizar um software que está disponível apenas para o Linux. Com a virtualização, esse usuário pode executar uma versão de qualquer sistema operacional e seus aplicativos em seu próprio computador, sem necessariamente ter que o instalar fisicamente.

Geralmente, existem dois tipos de hypervisores.

- Hypervisores de tipo 1, chamados de "bare metal" ou “Stand Alone” que são executados diretamente no hardware da máquina física. Este tipo de hipervisor é mais empregado para a virtualização de servidores.
- Hypervisores de tipo 2, conhecidos como “hosted” ou "hospedados", são executados como uma camada de software, um aplicativo instalado no sistema operacional, como outros programas de computador. Este tipo de hipervisor é mais empregado em soluções para desktop, como o VirtualBox.

O tipo hosted possui uma camada a mais de aplicação junto com a camada do hypervisor, e ambas acima do sistema operacional da máquina física.

ambiente virtual e também permite que os usuários possam executar aplicações tais como web browsers e clientes de e-mail paralelamente ao ambiente virtualizado. Isto não é possível no tipo bare-metal.

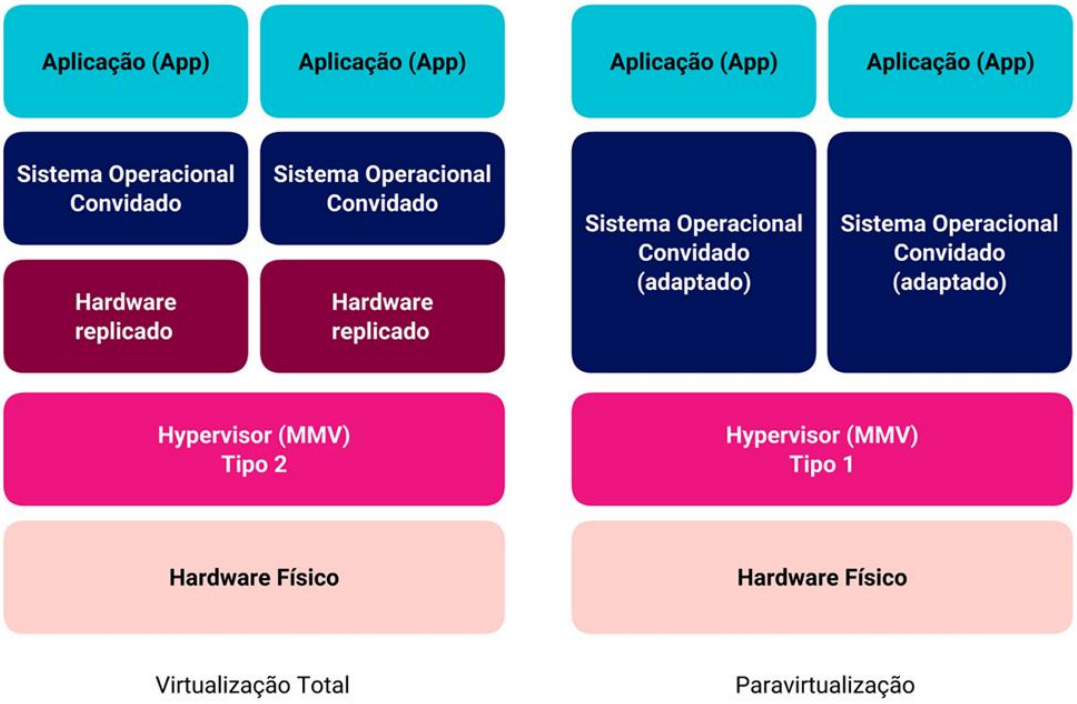
A maioria dos hypervisores oferecem recursos adicionais de hardware, que vão desde controladores USB até o emprego de direct memory access (DMA), visando melhorar o desempenho de controladores de storage, no que diz respeito ao acesso a disco, e placas de rede.

Visto isso, a decisão de usar hypervisor bare-metal ou hosted vai além de “ter ou não ter sistema operacional no host”.

A tipo 1, por exemplo, por estar situado diretamente sobre o hardware, consegue prover um número maior de opções de acesso de entrada e saída (I/O access), disponibilizando mais desempenho para aqueles que optam por essa arquitetura. Importante notar que ao usar o hypervisor bare-metal o sistema operacional a ser utilizado na máquina virtual precisa ser adaptado para esse tipo de solução.

Já o tipo 2 consegue prover maior compatibilidade de hardware, o que permite executar o software de virtualização em uma gama mais ampla de configurações de hardware, diferentemente do modo bare-metal.

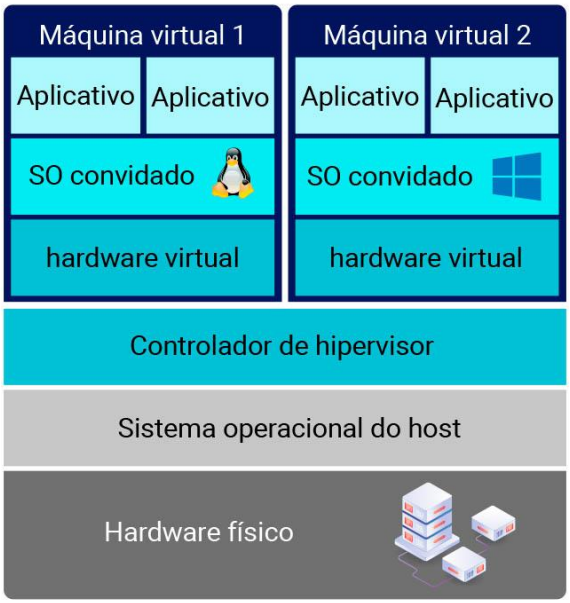
Os hypervisores tipo 1 e tipo 2 irão definir dois tipos de vitualização: paravirtualização e virtualização completa.



Comparação entre virtualização total e paravirtualização.

Na completa, o hypervisor emula todo o hardware da máquina física para as máquinas virtuais. Nesse caso, o sistema operacional executa como se não estivesse em um ambiente virtual. A paravirtualização entrega para as máquinas virtuais um hardware igual ao real, com isso, o sistema a ser virtualizado pode sofrer alterações no decorrer do tempo. Essa funcionalidade não é permitida na virtualização completa, pois nela o hardware é entregue de forma virtual.

Em resumo, a virtualização é um tipo de tecnologia que permite que diversas aplicações e sistemas operacionais sejam processados em uma mesma máquina física, conforme ilustrado na imagem a seguir.

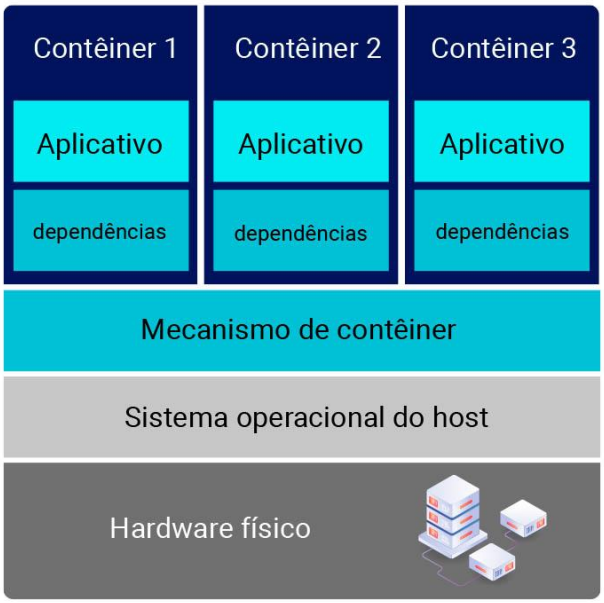


Máquinas virtuais

Virtualização baseada em contêineres

Conteinerização

A conteinerização, também conhecida como virtualização baseada em contêineres, é um método utilizado na implantação e execução de aplicativos distribuídos sem a necessidade de configuração de uma virtual machine (VM) completa para cada um deles. Em vez disso, vários sistemas isolados, chamados de contêineres, são executados em um único host de controle, acessando um único kernel, conforme ilustrado a seguir.



Contêineres

A tecnologia de conteinerização permite a entrega de uma determinada aplicação dentro de uma estrutura virtual (contêiner) que se assemelha a uma VM, consumindo menos recursos e com uma estrutura de portabilidade mais simples entre diferentes ambientes tanto físicos como virtuais. Isto é, um contêiner é a versão enxuta de uma VM padrão (que necessita de um hypervisor para ser executada).

Os contêineres são virtualizados no nível do sistema operacional, com vários contêineres sendo executados diretamente acima do kernel do sistema operacional. Isso significa que são muito mais leves: compartilham o kernel do sistema operacional, iniciam muito mais rápido e usam apenas uma pequena parte da memória, em comparação com a inicialização de um sistema operacional completo.

Exemplo

Vamos imaginar um navio cargueiro com vários contêineres. Se um desses danificar, não afetará os outros ou o navio, pois cada um está isolado e protegido.

Ao contrário do que muita gente pensa, essa tecnologia não é assim tão nova. Vejamos:

Anos 70

Durante o desenvolvimento do Unix V7, foi introduzido o system call ou chamada de sistema, também conhecido como chroot, alterando o diretório raiz de um processo e seus filhos para um novo local no sistema de arquivos, recurso muito utilizado até hoje para eventuais manutenções. Essa evolução trouxe o conceito de isolamento do processo, segregando o acesso a arquivos para cada processo.

Anos 2000

O FreeBSD Jails dá forma à tecnologia de contêineres ao permitir que os administradores de sistema (sysadmins) dividam o sistema operacional FreeBSD em vários sistemas menores e independentes conhecido como jails, inclusive com a capacidade de atribuir IP e hostname para cada sistema e configuração. No mesmo período, é lançado o Linux-VServer "old-school"; a primeira versão do Solaris Contêiner; além do lançamento do projeto OpenVZ (Open Virtuozzo) e do "Process Container", pelo Google.

2008

Surge a primeira e mais completa implementação do gerenciador de contêiner do Linux, o projeto LXC (Linux Contêiner), que serviu de base para outras tecnologias como o Warden, em 2011, e o Docker, em 2013, que levou a tecnologia de contêiner a um novo patamar.

O Docker é uma plataforma open source escrito em Go, uma linguagem de programação de alto desempenho desenvolvida dentro do Google, que facilita a criação e administração de ambientes isolados. Isto é, o Docker é uma implementação de virtualização de contêineres que vêm conquistando cada vez mais espaço devido à computação em nuvem.

E assim, surge o conceito de cloud containers ou contêineres na nuvem, isto é, virtualização baseada em contêiner, um modelo de virtualização na nuvem em nível de sistema operacional, com o objetivo de implantar e executar aplicativos distribuídos. Dessa forma, vários sistemas isolados são acionados em um único host, acessando um único kernel.

Apesar do uso do termo virtualização baseada em contêiner, não podemos confundir com virtualização em si, pois, nesta última, o servidor é configurado para atuar como se fosse uma máquina física, com sistema operacional próprio, garantindo um ambiente funcional. Essencialmente, um conjunto de SO é instalado em um único equipamento físico. E no caso do cloud containers, não

há uso de sistemas operacionais. Os contêineres são independentes e realizam a execução da aplicação, sendo só ela a instalada, facilitando o processo.

Serverless computing e ASP

Computação sem servidor (serverless computing)

Inicialmente, nossas aplicações estavam hospedadas em servidores físicos. Com a evolução da tecnologia, surgiram as máquinas virtuais — e as soluções Platform as a service (PaaS). As soluções PaaS virtualizavam a entrega de servidores, mas a preocupação em manter os sistemas operacionais virtuais do servidor ainda persistia. O próximo passo foi a chegada da tecnologia dos contêineres. Contudo, ainda era necessário conservar os contêineres por pessoas especializadas nessa solução. Com o objetivo de retirar essa carga de trabalho do profissional de desenvolvimento de software, surgiu a arquitetura serverless.

Computação sem servidor ou serverless computing é a tecnologia que permite hospedarmos funções (Plataforma de Função como Serviço – FaaS) sem a preocupação de configuração do servidor, pois todo o ambiente (hardware e software) já está pronto para execução da função desenvolvida.

Por volta de 2006, foi lançada uma plataforma com o objetivo de fazer todo o trabalho rotineiro para o desenvolvimento e a implantação de uma aplicação Javascript cobrando apenas pelo código que fosse executado. Assim nascia a plataforma Zimki, que na época não teve aceitação, mas representa o nascimento de um novo conceito de serviço de computação em nuvem, Function as a Service (FaaS), que seria uma plataforma de função como serviço e, consequentemente, um novo modelo de arquitetura, o serverless computing.

O serverless é orientado a eventos e se diferencia das outras tecnologias de servidores físicos, virtuais e contêineres por sua infraestrutura. Esse modelo de infraestrutura é concentrado na entrada, execução e saída de uma função. Essa solução permite ao desenvolvedor criar e executar suas aplicações e serviços sem se preocupar com os servidores. Uma aplicação serverless não exige qualquer tipo de gerenciamento de servidor.

Entre os diferenciais dessa solução, podemos destacar:

Baixo custo

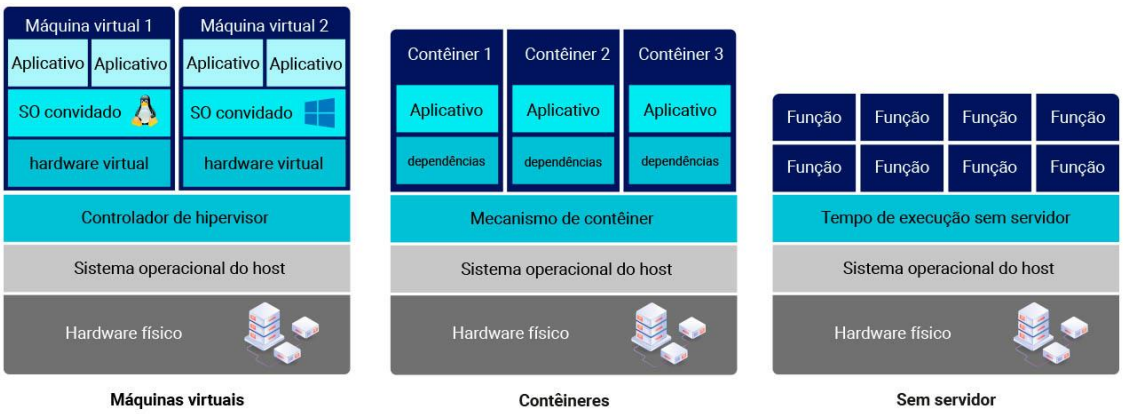
São cobradas por uso: isso significa que você só vai pagar aquilo que realmente está utilizando.

Redução de código

É menos complexa, sem a necessidade de ter um sistema back-end de várias camadas. Além de ser escalável e flexível, não são necessárias configurações adicionais para aproveitar a escalabilidade da arquitetura.

Atualmente, existem três principais fornecedores de soluções serverless. São eles: Amazon AWS, Microsoft Azure e Google Cloud. Apesar do conceito serverless ter tomado maior visibilidade atualmente devido às funções e à capacidade de executar código sem um servidor, já há algum tempo estamos consumindo diferentes serviços que também abordam o conceito.

Confira a diferença entre as três tecnologias estudadas até o momento, máquinas virtuais, contêineres e computação sem servidor.



Comparação entre máquinas virtuais, contêineres e computação sem servidor.

Provedor de serviços de aplicação (ASP)

Provedor de serviços de aplicação ou application service provider (no inglês, ASP) é um formato de terceirização que fornece software e aplicações por meio da internet para usuários finais, pequenas e médias empresas ou até grandes organizações. Em vez de as organizações arcarem com os encargos financeiros, os requisitos de hardware e os conhecimentos técnicos necessários para possuir o software, eles alugam esses aplicativos de terceiros. Nesse modelo, os provedores alugam aplicações e serviços de acordo com a necessidade dos

clientes. E os clientes pagam um valor para usufruir desse serviço como uma assinatura.

Exemplo

Os webmails, como o correio do Yahoo, o correio do Google, o armazenamento de documentos e planilhas no Google Docs são serviços ASP gratuitos.

Por meio de sistema de identificação e autenticação é possível acessar os documentos, planilhas, vídeos, ou seja, acessar os arquivos armazenados remotamente no servidor do provedor.

Algumas características dessa solução são os baixos custos em relação a ter acesso a recursos tecnológicos de ponta, eliminando a necessidade de realizar investimentos em uma infraestrutura própria ou mesmo melhorias nos sistemas já existentes. Os ASPs fornecem configuração e instalação rápidas, pois não é necessário na implementação de um software fazer estudos de viabilidades, demonstrações, testes; o aplicativo já está operacional para o uso.

Computação em grade e utility computing

Grid computing

A grid computing ou computação em grade é a tecnologia que agrupa servidores com o objetivo de trabalhar em conjunto formando uma grande infraestrutura. Esse modelo requer o uso de softwares responsáveis em dividir e distribuir partes de um programa como uma imagem grande do sistema para milhares de computadores.

O termo grid foi usado inicialmente nos anos 1990, no meio acadêmico. Foi originalmente proposto para denotar um sistema de computação distribuída que provia serviços computacionais sob demanda, da mesma forma que os fornecedores de energia elétrica e de água.

Assim, podemos definir grid computing como um tipo de sistema paralelo e distribuído que permite o compartilhamento, seleção e agregação de recursos geograficamente distribuídos dinamicamente e em tempo de execução dependendo da sua disponibilidade, capacidade, performance, custo e requerimentos dos usuários.

extension

Exemplo

Vamos imaginar duas empresas localizadas em países geograficamente distantes e com fusos horários diferentes: essas empresas poderiam formar um grid ao combinar seus servidores, dessa maneira cada empresa utiliza os ciclos de processamento ociosos da outra em seus horários de pico, já que com horários diferentes, os picos de acessos aos servidores de cada empresa ocorrerão em horários diversos.

Entre as características dessa solução, podemos citar a **possibilidade de explorar recursos** subutilizados e recursos adicionais, como ciclos de CPU, espaço em disco, conexões de rede, equipamentos científicos.

Também se destaca pela **capacidade de processamento paralelo**, pois uma aplicação utilizando algoritmos e técnicas de programação paralela pode ser dividida em partes menores e essas partes podem ser separadas e processadas independentemente. Cada uma dessas partes de código pode ser executada em uma máquina distinta no grid, melhorando a performance. Com essa tecnologia, os recursos e as máquinas são agrupados, formando uma organização virtual.

Por último, a **confiabilidade** é uma característica dessa abordagem baseada em máquinas espalhadas por diversos lugares diferentes, quando uma falha e atinge uma parte do grid, as demais podem continuar sua operação, normalmente.

Geralmente, as tecnologias grid e clusters se confundem, porém, existe uma diferença na maneira como os recursos são gerenciados.

Cluster

Há um gerenciador de recursos centralizado e responsável pela alocação de todos os recursos dos clusters e, dessa maneira, todos os nós trabalham em conjunto.

Grid

Cada nó possui seu próprio gerenciador de recursos e não existe a responsabilidade de prover a visão de que faça parte de um só sistema.

Em resumo, o grid forma um ambiente fundamentalmente cooperativo ao compartilhar os ciclos ociosos de processamento em seus sistemas em troca de poder utilizar parte do tempo de processamento do grid.

Utility computing

Utility computing ou computação de utilidade pública é um modelo classificado como computação sob demanda, pois o usuário pode contratar software, hardware e serviços conforme sua necessidade de utilização e em função de fatores como picos, quedas e conforme o período de uso. Dessa maneira, podemos fazer um comparativo com os serviços de fornecimento de água, luz ou telefone, conforme a demanda do cliente.

O termo utility computing vem das chamadas utilities, que em inglês são as empresas públicas com modelos de negócios com a cobrança pelo que é consumido. Ao permitir a aquisição de capacidade temporária de processamento e armazenamento de dados, essa tecnologia potencializa a otimização da infraestrutura de hardware, software e serviços com redução dos custos fixos por capacidade não utilizada.

Algumas características importantes da utility computing são:

Escalabilidade

É uma métrica importante que deve ser garantida na computação de utilidade para fornecer recursos de TI disponíveis a qualquer momento. Se a demanda for estendida, o tempo de resposta e a qualidade não deverão ser afetados.

Preço sob demanda

É programado de forma eficaz, pagando de acordo com o uso do hardware, software e serviços contratados.

Serviços padronizados

O catálogo é produzido com serviços padronizados com diferentes contratos de nível de serviço para os clientes. Os serviços web e outros recursos são compartilhados pelo provedor, usando tecnologias de automação e virtualização.