

Virtualização

Paulo Pires

Maj Cav Para

Por opção do autor, este artigo está redigido segundo os instrumentos ortográficos ao Acordo Ortográfico da Língua Portuguesa de 1990.

Os primeiros passos da virtualização datam de aproximadamente meio século onde preocupações como heterogeneidade, portabilidade, estabilidade e utilização mais eficiente de recursos começaram a dar os primeiros passos. A virtualização de desktops tem um historial da década de 60, já a virtualização de aplicações tem apenas alguns anos.

Tudo começou com a IBM (*International Business Machines Corporation*)(1) e a sua longa lista de variedades de sistemas com as suas sucessivas gerações criando alguns conflitos nos clientes. Diversos problemas foram surgindo, sempre associados à execução sequencial de processos, um verdadeiro transtorno para processamento de múltiplas tarefas, deste modo, surgiu o mainframe S/360, figura 1, um sistema de acesso único para manter compatibilidades e responder à necessidade de execução de múltiplas tarefas (*Batch Jobs*).

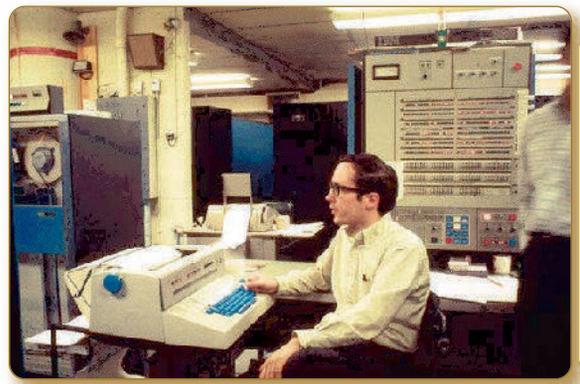


Figura 1 - IBM System/360 modelo 67-2 (duplex) da Universidade de Michigan

O conceito de acesso único alterou-se em julho de 1963 com o projeto MAC (*Multiple Access Computers*) do Massachusetts Institute of Technology (MIT)(2), um projeto para matemática e computação. Mais tarde o nome foi alterado para computador de acesso múltiplo, pelo facto

(1) [Online]. Available: <http://en.wikipedia.org/wiki/IBM>

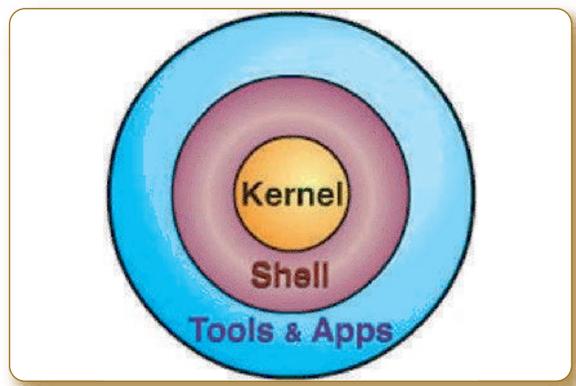
(2) [Online]. Available: http://en.wikipedia.org/wiki/Project_MAC

de permitir o acesso de mais de um utilizador. Também os Laboratórios Bell (*Bell Telephone Laboratories - AT&T*)(3) começaram a dar os seus passos.

A IBM respondeu com o seu CP-40 e posteriormente o CP-67 em 1968 já com suporte à virtualização e com um SO (Sistema Operativo) CP/CMS, um Control Program a correr no *mainframe* para criação de VM's (*Virtual Machines*) e um *Console Monitor System* como SO onde as VM's eram executadas e com interação dos utilizadores.

Na continuação do projeto MAC, o MIT apresentou o MultiCS (*Multiplexed Information and Computing Service*)(4), um exemplo de um sistema já com preocupações como divisão de memória e outros recursos pelos diferentes utilizadores.

Posteriormente desenvolvimentos deste sistema nos laboratórios Bell levou a uma evolução hoje conhecida como *Unix*(5), figura 2.



■ Figura 2 - Os três níveis do sistema UNIX: kernel, shell, e ferramentas e aplicações.

Unix é um sistema escrito em C (contrariamente aos anteriores em assembly) onde vários utilizadores partilham processador, memória, disco e uma pool de recursos para cada um, mas separados uns dos outros, cada um tem o seu

próprio perfil dentro do mesmo sistema. Assim, considera-se o Unix como o sistema pioneiro de múltiplos SO's e dos primeiros passos na virtualização de aplicações.

A virtualização de aplicações deu um grande salto em 1990 com um projeto denominado "*Stealth*" da *Sun Microsystems*, uma linguagem de programação que em 1995 foi renomeada para java e que veio dar um grande avanço na *World Wide Web (WWW)*. Foi publicado em janeiro de 1996 com o nome *Java Development Kit (JDK)*, permitindo que os programadores desenvolvessem aplicações para esta plataforma e pudessem ser executadas independentemente do SO, desde que o ambiente de execução estivesse instalado, o *JRE (Java Runtime Environment)*.

O caminho para a virtualização estava traçado, começando a surgir sistemas como *Virtual PC* da *Apple* em 1997, sistema vendido pela *Connectix*, posteriormente adquirida pela *Microsoft*, *VMware Workstation* da *VMware* em 1998, *ESX Server* também da *VMware* surgiu em 2001, em 2003 começou a ser implementado o *XenSource* comprado posteriormente pela *Citrix Inc* em 2007 e renomeado posteriormente para *XenServer*, sistemas abordados em posteriores publicações do boletim do IGeoE.

I.2. Contexto

A implementação desta tecnologia, tão bem aceite e cada vez mais em uso nas organizações, demonstra a mais-valia das propriedades que caracterizam o seu funcionamento:

Isolamento: Os processos das diferentes máquinas virtuais não podem interferir entre si, ou seja, um processo de uma VM não pode interferir noutra VM, nem sequer no próprio sistema hospedeiro (host) que suporta toda a virtualização;

Inspecção: O gestor de máquinas virtuais (*Virtual Machine Manager - VMM*) ou hipervi-

(3) [Online]. Available: http://en.wikipedia.org/wiki/Bell_Labs

(4) [Online]. Available: <http://web.mit.edu/multics-history/>

5 [Online]. Available: <http://www.bell-labs.com/history/posix/>

sor, deve ter acesso e controlo sobre todas as VMs, e até sobre todos os seus processos;

Interposição: O VMM deve poder interagir com as VMs, controlar as suas operações, ter acesso ao fluxo de instruções máquina que estas executam e remover, adicionar ou alterar essas instruções se for necessário;

Eficiência: As instruções não privilegiadas devem poder ser executadas directamente no hardware, sem interposição do VMM;

Gestão: O VMM deve não só permitir manipular uma VM isoladamente, mas também executar uma operação, de uma só vez, sobre múltiplas VMs;

Compatibilidade de software: A virtualização simula uma determinada plataforma (real) e, desse modo, todo o *software* disponível para essa plataforma deve poder ser executado "tal como está (original)" numa VM;

Desempenho: A sobrecarga (overhead) da camada extra de *software*, pode sacrificar o desempenho do sistema hospedado (*guest*) mas a relação custo/benefícios do uso da virtualização deve ser francamente positiva.

A virtualização tem vindo a crescer e a ganhar terreno principalmente nas médias e grandes empresas. Tem vindo a destacar-se na gestão das TIs, uma mudança completa e um aliviar de trabalho para administradores e gestores.

I.3. Vantagens da tecnologia

Um mundo onde passa a ser possível executar *software* incompatível com o SO host, executar SO's distintos num mesmo conjunto de hardware, isoladamente, sem interferências, etc, uma mudança drástica na tradicional visão de um SO para um conjunto de hardware, apenas.

Segurança: Dentro de um servidor podemos ter vários VMs isoladas e independentes do SO host. Tal garante que qualquer interferência no SO host (erros/ataques acidentais ou intencionais) não afecte as VMs nele instaladas;

Isolamento: Qualquer processo de uma VMs não pode interferir nos processos de outra VMs, este isolamento, intimamente ligado à segurança vem de encontro a um dos princípios básicos aquando da implementação desta tecnologia. Esta vantagem garante-nos confidencialidade entre VMs, assim como integridade e contenção de erros entre VMs (segurança);

Custos: A quantidade de diferentes servidores existentes para diferentes SO's e aplicações é agora compactada num mesmo hardware onde existe a possibilidade de termos várias VMs com diferentes SO's com as mais variadas aplicações. Temos um tudo-em-um que vai reduzir os custos de uma organização na altura do upgrade, bastando para isso o upgrade de um único servidor que suporta todos os serviços da organização;

Energia: Propriedade intimamente ligada aos custos. Com a compactação do hardware são dispensando todos os demais equipamentos, quer de sustentação de toda uma infra-estrutura, quer os próprios equipamentos de apoio. É assim conseguida uma economia de energia bastante significativa para a organização;

Espaço Físico: A compactação é inerente à diminuição do espaço ocupado. Um *Data Center* tem sempre o problema de "falta de espaço" e por isso, a compactação de hardware associada à virtualização vem resolver não só este problema de espaço como também a própria refrigeração e circulação de ar dentro do *Data Center*;

Hardware: Está provado que cada servidor, em média, gasta cerca de 30% do seu poder de



processamento para executar a função para a qual foi implementado. Assim sendo, estamos na presença de um subaproveitamento deste poder computacional. A ideia anterior á virtualização de adquirir um novo servidor para uma nova aplicação, cai agora por terra com o objetivo de aproveitar ao máximo o poder computacional de cada servidor, aproveitando melhor todo esse *hardware* disponível;

Implementação de SO's: A associação de um SO por cada máquina fica agora também em desuso uma vez que a virtualização possibilita-nos a instalação de vários SO's em cada máquina, melhor, além de vários SO's, estes também podem ser diferentes entre si (*Windows*, *linux*, etc). Esta vantagem permite assim instalar SO's cujo *hardware* original (host) nem sequer os suportam, como o caso do SO MAC (*Macintosh*, *Apple*) em *hardware* x86 ou SO *Windows* em *notebooks* (*Apple*);

Migração: A camada que suporta a virtualização permite-nos esta mais-valia. A migração não só do ambiente em si como do próprio *hardware*, bastando para isso uma simples cópia entre discos virtuais;

Testes de software e hardware: A virtualização simula tanto um software como um *hardware*, logo, testes em software, firmware e *hardware* podem ser realizados nestas VMs construídas especificamente para esse fim. Estes primeiros testes podem ser efectuados sem a necessidade de um SO instalado de raiz ou mesmo a presença física do *hardware* específico;

Redução de administradores e gestores: Esta compactação associada à virtualização, como já referido, reduz espaço ocupado, reduz *hardware*, reduz configurações e gestão individual, abrindo caminho à centralização. Assim, a redução da equipa responsável pela administração e gestão é evidente.

I.4. Desvantagens da tecnologia

São notórias as vantagens da virtualização mas problemas como incompatibilidades entre aplicações e perda de desempenho pelo simples adicionar de mais uma camada à arquitectura terão que ser consideradas aquando da escolha.

Segurança: A segurança, neste caso, prende-se à própria camada adicionada à arquitectura (VMM ou hypervisor). Se o próprio SO host tiver algum problema de raiz, implementado ou adquirido, também o próprio VMM contemplará este erro assim como todas as VMs instaladas nesse SO host;

Gestão: O principal problema da implementação desta tecnologia é a gestão da própria virtualização, os ambientes virtualizados precisam de ser instanciados, geridos, configurados, etc. É neste ponto que se prende a preocupação das organizações pelos seus investimentos nesta tecnologia, trata-se de uma opção que, por norma, é licenciada;

Desempenho: Mais uma camada de software (VMM ou hypervisor) implica mais processamento; Incerteza da quantidade de VMs/processador sem que seja afectado a QOS (*Quality Of Service*) e a garantia dessa emulação em termos da disponibilização máxima que o host pode dispensar como é o caso do processamento e memória gráfica.