ARTIGO REF: 6604

iCBD: UMA INFRAESTRUTURA BASEADA NOS CLIENTES PARA EXECUÇÃO DE DESKTOPS VIRTUAIS

Paulo Afonso Lopes[1(*)], Nuno Preguiça[1], Pedro Medeiros[1], Miguel Menezes Martins[2]
[1]Universidade Nova de Lisboa, Depart. Engª Informática, NOVA-LINCS, Monte de Caparica, Portugal
[2]Reditus, S.A., Alfragide, Portugal
[(*)]*Email:* poral@fct.unl.pt

RESUMO

Hoje em dia o uso de hipervisores está largamente disseminado, sendo que é grande a sua utilização nos Centros de Dados: na consolidação de servidores (poupança de energia, espaço, redução no fardo de administração), na rápida instanciação (deployment) e remoção (retirement) de máquinas virtuais (VMs), na facilidade com que se retoma sua execução em caso de faltas (faults - crash/"avaria" de um servidor físico ou virtual), entre outras.

Este artigo relata a investigação e desenvolvimento, por uma equipa mista da Reditus, S.A., e do Centro de Investigação NOVA LINCS da FCT/NOVA, no âmbito de um projecto Portugal 2020, de uma Virtual Desktop Infrastructure (VDI) de um tipo particular: uma client-based VDI, na qual a execução dos desktops (virtuais) é efectuada nos postos físicos, e não em grandes servidores. A vantagem desta abordagem é fundamentalmente ao nível dos custos de infraestrutura, que são muito inferiores aos de uma VDI baseada em servidores, como as propostas pela Citrix, com o XenDesktop, ou pela VMware, com o View.

INTRODUÇÃO

Hoje em dia o uso de hipervisores (Agensen, 2010 e Barham, 2003) está largamente disseminado, sendo que é grande a sua utilização nos Centros de Dados: na consolidação de servidores (poupança de energia, espaço, redução no fardo de administração), na rápida instanciação (deployment) e remoção (retirement) de máquinas virtuais (VMs), na facilidade com que se retoma sua execução em caso de faltas (faults - crash/"avaria" de um servidor físico ou virtual), entre outras.

Embora o seu impacto inicial das tecnologias de virtualização seja fundamentalmente naquilo que se convencionou designar por virtualização de servidores, estas acabaram por "invadir" outras áreas das TIs, e em particular serem utilizadas naquilo que se designa por infraestruturas de desktops virtuais (VDI - *Virtual Desktop Infrastructure*). As VDI são, tal como o nome indica, infraestruturas de desktops (por exemplo, PCs ou laptops) que têm uma característica fundamental – foram virtualizadas, i.e., são VMs.

No momento em que se decide que o suporte para um posto de trabalho (o desktop) não é mais uma máquina física mas sim uma máquina virtual, adquirem-se imediatamente vários "graus de liberdade" que de outra forma dificilmente seriam possíveis de atingir, e as diferentes realizações disponíveis no mercado e/ou investigadas com maior ou menor sucesso correspondem, portanto, a escolhas nessa palete de possibilidades. É assim possível: a) executar uma VM, "imagem" virtualizada de um posto de trabalho, num servidor e aceder ao desktop (virtualizado nessa imagem) através de um protocolo como o RDP ou outro similar,

ou optar por b) executar essa mesma VM num PC, *laptop* ou *tablet* de utilização indiferenciada da organização (i.e., que não seja especificamente pertencente a um utilizador, mas de uso genérico para qualquer elemento da organização).

Este artigo relata a investigação e desenvolvimento, por uma equipa mista da Reditus, S.A., e do Centro de Investigação NOVA LINCS da FCT/NOVA, no âmbito de um projecto Portugal 2020, de uma VDI de um tipo particular: uma *client-based* VDI, na qual a execução dos desktops (virtuais) é efectuada nos postos físicos, e não em grandes servidores, ver Figura 1. A vantagem desta abordagem é fundamentalmente ao nível dos custos de infraestrutura, que são muito inferiores aos de uma VDI baseada em servidores, como as propostas pela Citrix, com o XenDesktop, ou pela VMware, com o View (Dasilva, 2012).

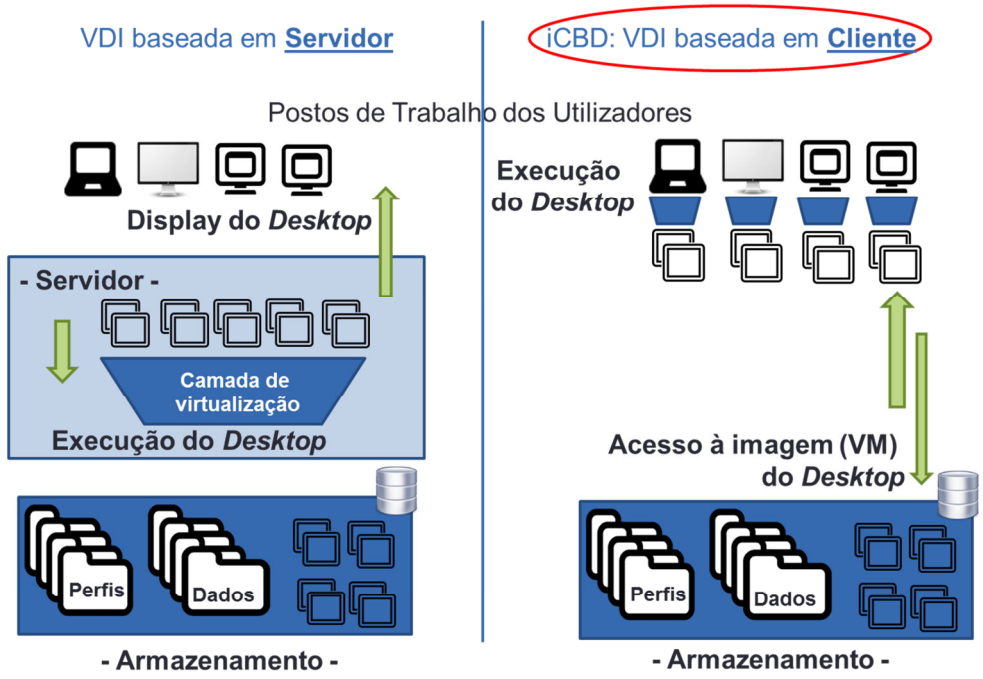


Fig. 1 - iCBD vs. VDI "clássica": execução no posto do utilizador e não no servidor (adaptado de Alves, 2016).

A primeira razão pela qual as VDIs baseadas em servidores são onerosas reside no total desaproveitamento das capacidades de computação do posto de trabalho, que é usado apenas como interface ao desktop virtual que corre no servidor. Ora a maior parte das organizações tem sempre um investimento razoável em PCs e *laptops*, e a não utilização dos recursos computacionais destes dispositivos não faz, do ponto de vista económico, qualquer sentido.

A segunda razão para os custos elevados dessas VDIs decorre da primeira: sendo todo o trabalho computacional realizado nos servidores, e sendo elevado o custo computacional de desenhar um ecrã, e exibir *streams* multimédia, para que a experiência de utilização de uma VDI baseada em servidores não seja inferior à que é usual num posto de trabalho dedicado, os servidores são dotados de co-processadores do tipo GP-GPU (Dowty, 2009); ora tais aceleradores gráficos têm por vezes custos idênticos aos de um servidor comum, sendo que nesses ambientes é comum um servidor ter vários GPUs, o que eleva significativamente os custos de uma VDI "clássica" – i.e., baseada em servidores.

O presente documento está assim organizado: na próxima secção, faz-se uma breve introdução à arquitectura da iCBD, destacando os serviços mais importantes; na secção 3, aborda-se o tema do subsistema de armazenamento e aspectos de investigação com este

relacionados; em seguida, na secção 4, abordamos a forma como se desenrola o processo de arranque de um posto de trabalho (isto é, um cliente) iCBD – desde os momentos iniciais até ao instante em que o desktop está visível no ecrã, permitindo ao utilizador fazer *login*; a secção 5 trata dos aspectos de administração da plataforma; e finalmente, na secção 6 apresentam-se as conclusões e o trabalho futuro.

iCBD: INFRASTRUCTURE FOR CLIENT-BASED (VIRTUAL) DESKTOPS

A iCBD (*Infrastructure for Client-Based virtual Desktops*) caracteriza-se por uma abordagem inovadora: ao contrário de outras soluções (já retiradas do mercado) que também usavam os postos de trabalho, como o Citix XenClient, e que na instalação destruíam (formatavam) o conteúdo do disco do posto de trabalho do utilizador, a iCBD preserva-o totalmente, porque executa, no posto de trabalho, o hipervisor por arranque remoto, mantendo apenas uma imagem em memória da VM. Passadas que foram as provas de conceito, a iCBD é já um protótipo funcional, suportado num *site* único, disponibilizando VMs Windows (7 e 10, mas em geral qualquer versão suportada pelo hipervisor), além de múltiplas distribuições de Linux.

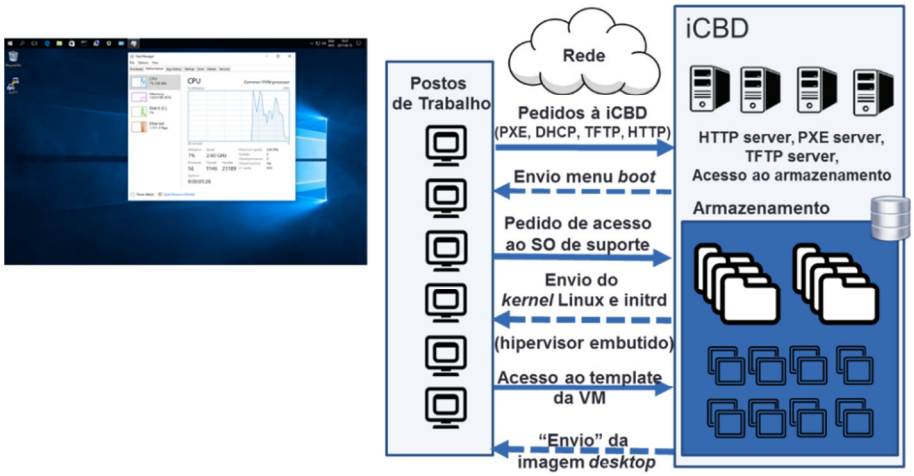


Fig. 2 - iCBD: Arquitectura e serviços (caso de execução de uma VM, adaptado de Alves, 2016).

A Figura 2 ilustra o caso mais vulgar, no qual um utilizador está a correr no seu PC uma VM Windows; contudo, o iCBD permite, exactamente pelo mesmo processo que são executadas VMs (contendo um qualquer sistema de operação *guest* suportado pelo hipervisor), executar também, nativamente no posto do utilizador, imagens Linux – novamente sem intrusão no conteúdo do disco local. Nesta primeira versão disponibilizam-se VMs não-persistentes; isto é, o utilizador corre no seu posto uma VM que, terminada a sessão de trabalho (ou em caso de falha), é destruída. Os dados do utilizador são guardados em sistemas de armazenamento em rede, geralmente do tipo NAS (Network Attached Storage) suportados em protocolos NFS ou CIFS (este último caso muito usado em organizações, onde é vulgarmente designado por "*drive* de rede" ou Microsoft *Share*).

ARMAZENAMENTO DE TEMPLATES E CRIAÇÃO DE INSTÂNCIAS DE VMS

Cada VM em execução (*desktop instance* – omitiremos a palavra virtual) num posto tem, como ponto de partida, uma imagem especial – o *template*, ou *golden image* – que é clonada para criar a instância. Sendo a imagem de uma VM tipicamente armazenada sob a forma de

um ficheiro que representa um disco (contendo não só um SO mas também um conjunto de aplicações, a imagem pode ter dimensões que facilmente chegam, no caso dos sistemas Windows, às várias dezenas de GB. Assim, a clonagem de um *template* para criar a instância privada (VM) que vai executada num posto tem de ser uma operação muito rápida – o que naturalmente não é possível se se recorrer à cópia integral do *template*. Nasce assim a ideia de usar cópias diferenciais, ou *linked-clones* (ou *snapshots*).

A criação de cópias diferenciais pode ser efectuada, ao nível de volumes, por dispositivos muito caros, como os armários de discos (*disk arrays*) da EMC, IBM, HP, etc., ao nível de ficheiros, pelos próprios hipervisores (Citrix, Microsoft, VMware, etc.) e até, muito recentemente, por *software* do "sistema de ficheiros". Assim, no âmbito do projecto realizaram-se dois trabalhos de investigação, já concluídos, que se debruçaram sobre o sistema de armazenamento, tendo sido exploradas duas alternativas: uma usando o Ceph RADOS (Martins, 2016), um sistema de armazenamento de objectos (OBS - *Object-Based Storage*), e outra um sistema de ficheiros, o Btrfs (Alves, 2016). As duas linhas de investigação tiveram um desígnio comum: substituir os mecanismos de *snapshot* (Vaghani, 2010) dos hipervisores, que estão na base do suporte aos *thin-clones*, pelos mecanismos nativos de *snapshoting* do sistema de armazenamento; e avaliar os resultados.

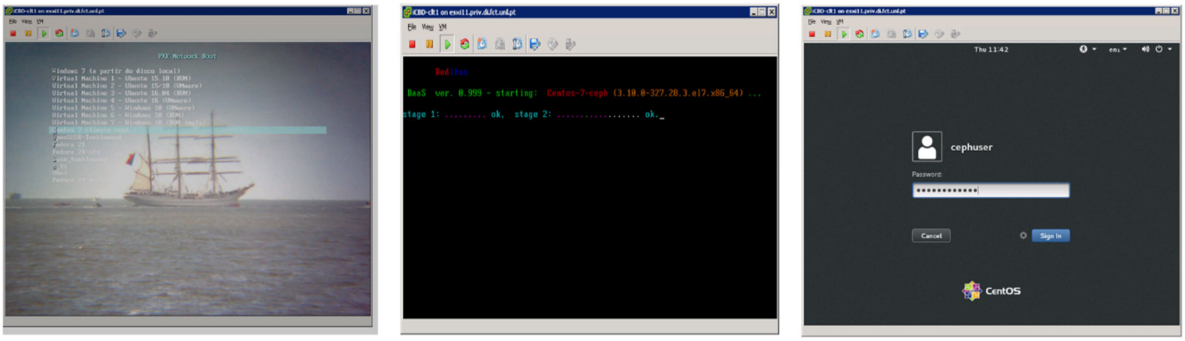
Uma das traves mestras da plataforma iCBD é a integração de tecnologias existentes, *hardware* e *software*, de uso comum (COTS – *Common-Off-The-Shelf*) com nenhumas ou com um mínimo de modificações. Como resultado dos dois trabalhos provou-se que a criação de clones (de *templates*) baseada em mecanismos do subsistema de armazenamento (baseado em objectos ou em sistemas de ficheiros com suporte nativo para *snapshots*) para além de serem muito rápida, ocupa também muito pouco espaço, já que o clone inicial é uma estrutura de dados que, usando uma técnica *copy-on-write*, guarda apenas os blocos alterados na instância e, nos outros casos, aponta os originais. Em ambos os casos, Ceph e Btrfs, a integração da funcionalidade de snapshoting com o hipervisor é muito simples (Ceph) ou transparente a este (Btrfs).

POSTO DE TRABALHO iCBD: DO ARRANQUE AO LOGIN

Ao nível do processo de arranque, e até ao momento em que o utilizador começa a usar o seu *desktop* virtual, mais uma vez se conseguiu fazer o processo usando tecnologias com provas dadas há muitos anos, a saber: PXE (*Pre-boot eXecution Environment*), DHCP (*Dynamic Host Configuration Protocol*), TFTP (*Trivial File Transfer Protocol*) e HTTP (*HyperText Transfer Protocol*). A Figura 3 apresenta uma súmula dos ecrãs mais importantes que o utilizador pode ver até ao momento em que o ecrã de login é apresentado.

Quando um utilizador usa um posto (registado na iCBD e preparado para efectuar o arranque via rede) e o liga, o PXE efectua um broadcast, o servidor DHCP responde com o endereço IP do posto, que carrega o ecrã de opções que se pode ver na Figura 3, em (a). Este ecrã permite ao utilizador escolher com que "imagem" (Windows, Linux nativo, Linux em VM) quer trabalhar.

Depois da escolha do utilizador o processo segue com a transferência por TFTP de um núcleo (*kernel*) Linux e respectiva imagem `initrd`, arranque do núcleo e execução de scripts, etc.. Este processo desenrola-se em várias fases, das quais algumas apresentam informação no ecrã, como a que se pode ver na Figura 3, em (b). Note-se que no caso geral é este sistema que vai suportar a execução de uma VM, pelo que o "disco de sistema" (a imagem) já tem o hipervisor. O iCBD suporta actualmente dois hipervisores: o KVM (software livre) e o VMware Player (software gratuito).



(a) (b) (c)

Fig. 3 - iCBD: Arquitectura e serviços (caso de execução de uma VM, adaptado de Alves, 2016).

Depois de ter terminado o arranque do sistema Linux que vai hospedar a VM o hipervisor embutido na imagem desencadeia a execução da VM. Mas, a imagem que é fornecida ao hipervisor é um *template*, logo (conceptualmente) é apenas acessível em leitura (*read-only*). Assim, de facto, não é essa a imagem que é fornecida ao hipervisor; esta é um *snapshot* (ou *linked clone*) do template, e esse snapshot é, naturalmente, fornecido sob a forma de um volume *read-write*. Finalmente, terminado o arranque da VM, esta apresenta-se como na Figura 3, (c).

iCBD: ADMINISTRAÇÃO DA PLATAFORMA

Na concepção da iCBD houve uma grande preocupação para conseguir que não só a utilização, mas também a administração fosse muito simples. Ao nível da administração, conseguiu-se essa simplicidade pois há apenas uma única entidade a administrar: o *template*. De facto, quer se trate de uma imagem para execução nativa ou virtual, de uma imagem com ou sem hipervisor embutido, a administração é efectuada sempre da mesma forma, e tendo por alvo um *template*. Repare-se que aqui o *template* é mais um conceito – imagem que não pode (deve) ser executada a não ser para administração da mesma – do que propriamente um objecto diferente (como é na VMware) que tem de ser convertido para VM para ser administrado (e depois novamente convertido em *template*).

Quando um administrador actualiza um *template*, por exemplo, fazendo actualizações (*patches*) ou instalando novo *software*, as novas sessões usam imediatamente o novo *template*, enquanto as sessões activas mantêm-se "ligadas" ao *template* anterior. O administrador pode, se o desejar, estabelecer "associações" entre postos de trabalho (ou utilizadores) e *templates* antigos, como forma de resolver problemas específicos de um produto software, questões de licenciamento, etc..

CONCLUSÕES E TRABALHO FUTURO

Neste momento, o protótipo iCBD está completamente funcional. No centro de investigação NOVA LINCS toda a infraestrutura de servidores representada na Figura 2 está realizada como uma única VM (`iCBDmainSrv`) que corre sobre um *cluster* de 3 servidores (dois HP e um Supermicro) suportados em VMware ESXi; na mesma infraestrutura foram criadas VMs que simulam os postos de trabalho físicos – foi dessas VMs que se obtiveram as imagens (*screenshots*) apresentadas. Do ponto de vista dos subsistemas de armazenamento, o volume Btrfs faz parte da `iCBDmainSrv`, ainda que o armazenamento seja externo e efectuado sobre

uma infraestrutura EMC ScaleIO acessível por Gigabit Ethernet. Já o armazenamento Ceph é todo ele realizado sobre VMs (com três Ceph I/O servers).

Ainda que não seja significativo fazer testes de desempenho numa infraestrutura virtualizada e na qual outros trabalhos estavam a correr, (Martins, 2016) obteve resultados que variaram entre um mínimo de 59s e um máximo de 1 min 20s para o arranque de um posto de trabalho (virtualizado em ESXi) sobre o qual corria uma VM CentOS 7 (virtualizada com KVM "em cima de" ESXi) – ou seja, em *nested-virtualization* (virtualização dupla).

Do ponto de vista da investigação há actualmente um trabalho em curso que explora a possibilidade de disponibilizar tolerância a faltas intra e inter-*sites*, usando replicação e *caching*, mas mantendo a necessária consistência. No que ao desenvolvimento se refere, estão para breve início os trabalhos de criação de interfaces de administração amigáveis (*user-friendly*) baseadas em *browser* e/ou aplicações gráficas.

Finalmente, espera-se ansiosamente a entrega de *hardware* dedicado, que permitirá fazer testes de desempenho e comparação com outras soluções, nomeadamente Citrix XenDesktop e, se possível, VMware View.

AGRADECIMENTOS

Os autores agradecem o suporte financeiro concedido pelo POCI – P2020, através do financiamento plurianual do Projecto 11467 (iCBD: Infrastructure for Client-Based Desktops).

REFERÊNCIAS

[1]-Agesen, Ole, et al. "The evolution of an x86 virtual machine monitor." ACM SIGOPS Operating Systems Review 44.4 (2010): 3-18.

[2]-Barham, Paul, et al. "Xen and the art of virtualization." ACM SIGOPS Operating Systems Review. Vol. 37. No. 5. ACM, 2003.

[3]-D. A. Dasilva, L. Liu, N. Bessis and Y. Zhan, "Enabling Green IT through Building a Virtual Desktop Infrastructure," 2012 Eighth International Conference on Semantics, Knowledge and Grids, Beijing, 2012, pp. 32-38. DOI:10.1109/SKG.2012.29.

[4]-Micah Dowty and Jeremy Sugerman. 2009. GPU virtualization on VMware's hosted I/O architecture. ACM SIGOPS Operating Systems Review, Vol. 43. No. 3 (July 2009): 73-82. DOI:10.1145/1618525.1618534.

[5]-Alves, Nuno. "Linked clones baseados em funcionalidades de snapshot do sistema de ficheiros". Dissertação submetida para obtenção do Grau de Mestre em Engenharia Informática, FCT/NOVA, Dezembro 2016.

[6]-Martins, Eduardo. "Object-Based Storage for the support of Linked-Clone Virtual Machines". Dissertação submetida para obtenção do Grau de Mestre em Engenharia Informática, FCT/NOVA, Dezembro 2016.

[7]-Vaghani, Satyam B. "Virtual Machine File System." ACM SIGOPS Operating Systems Review 44.4 (2010): 57-70.