A S T E R O I D E

Architecture **S**ystème des stations de travail de **T**ELECOM Lille 1 pour l **'E**nseignement **R**éseau, l' **O**bservation, l' **I**nnovation, le **D**éveloppement et l' **E**xpérimentation.

-----oOo-----

Jacques Landru ; Tovohérizo Rakotonavalona ; Martine Sion
Institut TELECOM / Université Lille 1 Sciences et Technologies
TELECOM Lille 1
Cité scientifique, rue G. Marconi BP 20145 59653 Villeneuve d'Ascq Cedex

Mots clefs : virtualisation, KVM, SPICE, VDI (*Virtual Desktop Infrastructure*), déploiement des stations en salle de TP, clonage de machines virtuelles, *appliance* plate-forme de TP, cloud privé.

Résumé : ASTEROIDE est un travail exploratoire d'intégration système visant à proposer un cadre architectural pour les postes de travail déployés dans les salles de TP. L'émergence de la virtualisation du poste de travail (VDI Virtual Desktop Infrastructure) se présente sous deux formes distinctes : l'une centralisée (*hosted desktop virtualization*), où les postes virtualisés sont hébergés sur les hyperviseurs du *cloud*; la seconde permettant de cloisonner dans des *appliances* virtuelles distinctes les différentes activités sur le poste individuel (*local desktop virtualization ou bare metal client virtualization*). C'est cette seconde forme de VDI que nous envisageons de déployer sur les postes de salles de TP, en remplaçant le système *multi-boot* actuel, par un socle de virtualisation composé de KVM (*Kernel based Virtual Machine*) et du protocole SPICE (*Simple Protocol for Independent Computing Environment*) sur lequel s'exécuteront les *appliances* virtuelles dédiées aux plate-formes de TP. L'objectif principal est d'améliorer la flexibilité de gestion des postes de travail. Au delà de cette architecture de base nous entre-voyons des perspectives nouvelles d'utilisations mutualisées des ressources des salles de TP que nous présenterons en seconde partie de ce document. A terme nous explorerons les nouvelles possibilités d'agrégation de ressources offertes par ce changement d'architecture.

1 Introduction

Les postes de travail des salles de TP, en accès libre, de notre établissement offrent un socle applicatif commun bureautique et web, ainsi que les environnements logiciels relatifs aux enseignements informatique et réseau, l'expérimentation, les projets de développement, les labos de langue, ... Elles s'appuient sur une installation système en *multi-boot*. Tous les postes d'une même salle ont une configuration matérielle identique. A l'exception de quelques outils sous licence, les salles disposent de configurations systèmes et logicielles identiques. L'homogénéisation des configurations facilite le déroulement de la majorité des séances de TP dans n'importe quelle salle banalisée. La gestion des réservations en est ainsi facilitée. L'objectif est de limiter le nombre de salles spécialisées, uniquement utilisées lors des UV de spécialisation de fin de cursus.

L'introduction du *multi-boot* et la réplication des disques en mode bloc a facilité la banalisation des postes polyvalents « multi-TP ». Cependant le nombre de systèmes distincts déployés par le *multi-boot* reste limité à l'OS sous licence propriétaire livré avec le poste (Microsoft Windows), auquel s'est ajouté une ou deux déclinaisons d'OS libres. Afin de protéger l'intégrité des disques de ces systèmes, le démarrage des stations de ces salles de TP banalisées depuis un périphérique portable (CDrom, DVD ou clé USB) est verrouillé. De fait, les différentes déclinaisons de la plate-forme de TP réseau VIMINAL (*Virtual Model for IP Network Architecture* Lab) [1], MEDI6 [2], MOBIDIK [2] et VODKA [2] disponibles sous forme de *LiveDVD* continuent à être « jouées » sur les postes d'une salle informatique dédiée, à accès restreint, où le démarrage de poste sur ce type de support amovible reste accessible.

La gestion de ces stations de travail s'appuie sur une machine de référence sur laquelle sont installés, testés, validés et mis à jour les différents systèmes du poste. Les configurations sont ensuite « poussées » à travers le réseau local sur les machines en salle de TP à l'aide d'outils de réplication de disques en mode bloc (tels que *ghost* ou *fog*). La re-configuration d'une machine ou d'une salle est relativement aisée. Toutefois ce mode de fonctionnement nécessite de disposer d'un poste de référence supplémentaire pour chaque type de configuration matérielle relatif au poste des salles de TP.

Avec ASTEROIDE (Architecture Système des stations de travail de TELECOM Lille 1 pour l 'Enseignement Réseau, l' Observation, l' Innovation, le Développement et l' Expérimentation) il s'agit d'aller plus loin en explorant les possibilités nouvelles offertes par la virtualisation du poste de travail et le déploiement des plate-formes de TP sous forme d'*appliances* virtuelles. La virtualisation permet de faire abstraction de la plate-forme matérielle. Le poste de référence d'une plate-forme de TP n'est alors plus lié à la configuration physique de la machine qui l'héberge, il devient générique.

2 Objectifs

La virtualisation, rupture technologique des architectures systèmes, s'est banalisée ces dernières années tout d'abord au niveau des salles machines. Elle s'est traduite par une consolidation des serveurs sur les machines dans le but d'améliorer le taux d'usage des matériels modernes, dont la puissance continue à progresser selon la loi de Moore. Elle contribue à l'émergence du phénomène *cloud computing*, quand bien même ce dernier ne se résume pas uniquement à la virtualisation. Après s'être imposée dans les centres serveur, la virtualisation gagne le poste de travail. Les éditeurs dominants de la virtualisation mettent en avant leurs nouvelles offres dites VDI (*Virtual Desktop Infrastructure*) : Citrix et son offre XenDesktop [3], VMWARE avec View [4] et Microsoft VDI Suite [5], pour ne citer que les plus médiatiquement connus. Le monde du libre n'est pas en reste, Red Hat, avec Red Hat Enterprise Virtualization for desktop (RHEV) [6] développe activement une solution basée sur *Kernel based Virtual Machine* (KVM) [7] et le *Simple Protocol for Independent Computing Environment* (SPICE) [8].

La virtualisation du poste de travail peut prendre deux formes distinctes :

- l'une centralisée (*hosted desktop virtualization*), où les postes de travail virtualisés sont hébergés sur les hyperviseurs du *cloud* type *Infrastructure as a Service (IaaS*). Elle se positionne dans la continuité de la consolidation des serveurs observée dans la première phase du déploiement de la virtualisation. En virtualisant les postes de travail sur les ressources des *data centers* on rationalise les parcs machines, on espère des gains substantiels sur la disponibilité des postes, leur gestion et leur administration.
- la seconde forme du *VDI* dite (*local desktop virtualization* ou *bare metal client virtualization*) vise à cloisonner dans des *appliances* virtuelles distinctes les différentes activités du poste individuel. Les postes, notamment ceux des utilisateurs nomades, peuvent ainsi faire cohabiter le système de l'utilisateur avec ses outils personnels et familiers de communication (messagerie, navigateurs, agenda, gadgets, réseaux sociaux...) et les *appliances* professionnelles portant les applicatifs métiers dans des machines virtuelles distinctes. L'isolation et l'étanchéité des différentes machines virtuelles contribuent au renforcement de la sécurisation des différentes activités hébergées sur le poste.

En s'appuyant sur cette seconde forme de VDI, le projet ASTEROIDE vise à explorer la flexibilité nouvelle de ce type d'architecture en transposant les différents environnements de TP dans des *appliances* virtuelles.

3 Socle de base du poste de travail

3.1 Capitalisation d'expérience

Nous capitalisons sur l'expérience acquise avec l'environnement de virtualisation *Kernel based Virtual Machine* (KVM) utilisé pour nos environnements serveurs [9]. KVM n'est certes pas l'environnement le plus répandu mais il a fait preuve de son efficacité, il est libre et évite donc de retomber dans la dépendance à l'éditeur dominant. Il s'agit d'un environnement de virtualisation initialement développé par la société Qumranet. Il est nativement intégré au noyau GNU/Linux depuis la version 2.6.20. L'éditeur Red Hat, qui a racheté Qumranet en 2008, l'a adopté comme principal mécanisme de virtualisation afin de se positionner sur le marché concurrentiel de la virtualisation face aux solutions d'autres éditeurs tels VMWare (VMWare ESXi), Microsoft (Hyper-V), Citrix (XEN) et Oracle (OracleVM et Virtualbox)... pour ne citer que les plus médiatiquement connus, parmi la cinquantaine d'environnements de virtualisation recensés par Wikipedia [10].

3.2 Eco-système KVM

En se basant sur les technologies d'assistance matérielle à la virtualisation (*Hardware Virtual Machine* : HVM) des processeurs modernes, les principaux hyperviseurs ou systèmes de virtualisation offrent aujourd'hui des performances proches de celles des systèmes natifs. Plus que les performances pures, ce qui distingue, aujourd'hui, les hyperviseurs des principaux éditeurs est essentiellement la richesse fonctionnelle des outils de gestion de l'éco système de virtualisation. HVM se traduit par l'extension du jeu d'instructions des processeurs fournissant une assistance matérielle à la virtualisation, extensions nommées *Virtual Machine eXtensions* (VMX) chez Intel, *Secure Virtual Machine* (SVM) chez AMD. KVM est un système de virtualisation complète (*full virtualization*) s'appuyant sur ces mécanismes de virtualisation matérielle. Bien qu'ils soient spécifiques à chacun de ces fondeurs, KVM en présente une interface unifiée au niveau du noyau du système de la machine hôte (la machine physique). Cela se concrétise sous la forme de deux modules du noyau GNU-Linux que l'on charge lors du démarrage de la machine hôte associés à un contrôleur en espace utilisateur basé sur qemu [11] (*qemu-kvm*). Un simple test du jeu d'instructions du processeur de la machine hôte permet de sélectionner le module noyau *kvm-intel* ou *kvm-amd* à charger. En l'absence du module, soit parce que le processeur de l'hôte est ancien et ne dispose pas des extensions HVM, soit celles ci sont présentes dans le processeur mais inactives, car verrouillées par le BIOS de la machine, la virtualisation KVM fonctionnera avec des performances moindres.

3.3 Socle de virtualisation

Le poste de travail, station hôte (*host*), qui va héberger localement les *appliances* virtuelles de TP (machines virtuelles, VM), devra donc être configuré conformément à KVM. Il s'agit d'activer un certain nombre de paramètres au sein du noyau GNU/Linux. La recompilation de celui ci générera les modules, *kvm-intel* et *kvm-amd* qui pourront être chargés sélectivement au démarrage du poste.

La seconde partie de l'environnement KVM, telle qu'elle est schématisée sur le figure 1, se compose d'éléments en espace utilisateur qui permettent la gestion par l'administrateur de l'éco sytème de virtualisation :

- L'élément principal, est une version adaptée KVM du célèbre émulateur libre QEMU [11], souvent dénommé *qemu-kvm*. Celui-ci accepte tout un ensemble de paramètres qui permettent d'implanter l'ensemble de la configuration virtuelle (caractéristiques mémoire, cpu, entrées/sorties, type d'écran,...) de la VM. Dans l'architecture ASTEROIDE, nous devons intégrer une version récente de *qemu-kvm* (>= 0.14), comportant le protocole SPICE.

- Le switch réseau virtuel : L'éco-système KVM dispose de deux méthode d'accès réseau :

    - Le mode dit « utilisateur » dans lequel le processus *qemu-kvm*, fait office de routeur *FireWall* (serveur DHCP, translation NAT/PT, routeur par défaut) et assure le cloisonnement de la machine virtuelle qu'il contrôle. Dans ce mode, la machine virtuelle est complètement isolée du réseau de l'établissement. Elle peut initier des communications à travers la fonction NAT/PT du processus *qemu-kvm* mais ne peut être atteinte de l'extérieur (comportement analogue à une machine connectée derrière une box d'opérateur).

    - Le second mode, dit « ponté » (*bridged*), permet l'intégration des machines virtuelles au sein d'un domaine de diffusion (VLAN) du réseau d'établissement, au même titre que n'importe quel poste de l'architecture. C'est ce mode que nous mettrons en œuvre pour ASTEROIDE. Il nécessite la mise en place d'un *switch* ethernet logiciel sur la machine hôte, (paquetage dénommé *bridge-utils* dans la plupart des distributions GNU/Linux). La carte ethernet (eth0) de l'hôte est connectée à ce *bridge* virtuel, pour assurer la liaison avec le réseau d'infrastructure. Une interface logicielle (*tun/tap*) est affectée au raccordement de chaque machine virtuelle au *bridge*.

- La librairie *libvirt* [12] et ses outils associés (*virsh*, *virt-install*, ...). La librairie assure une couche d'abstraction de virtualisation commune à toute une panoplie d'hyperviseurs et d'environnements de virtualisation (KVM, Xen, LXC, UserMode Linux, virtuoso, virtualbox, VMWARE, ...). Elle fournit un format XML de description des machines virtuelles, sur lequel s'appuient les outils associés (*virsh* : *virtualization shell*, *virt-install*, ...) pour piloter les machines virtuelles. Cette librairie fournit un socle générique de gestion sur lequel s'appuie tout un ensemble d'outils d'administration de haut niveau (*virt-manager, ovirt, abicloud, archipel*, ...) qui permettent l'exploitation d'un parc de VM. Dans le cadre d'ASTEROIDE, c'est essentiellement la description unifiée de machines virtuelles et les commandes du *shell* de virtualisation (*virsh*) qui seront utilisées. A terme un outil avancé de gestion de *cloud* pourra être déployé pour une utilisation mutualisée des ressources des salles de TP.

- L'accès distant en mode commande pour le pilotage de chaque hôte est classiquement assuré par la pile d'outils SSH, éventuellement complété pour un outil de pilotage de parc tel que « *fabric* » [13],
- Sur la machine hôte doit également être installé un environnement graphique minimal (serveur Xorg) support de l'écran d'accueil pour l'affichage graphique avancée des différentes *appliances* disponibles sur le poste, ainsi que le support du protocole SPICE et du client SPICE spicy (tel qu'ils sont décrits au paragraphe 5).
- Dans le cadre d'une utilisation ultérieure en mode *private cloud* des postes de salle de TP (cf paragraphe 7), l'intégration d'outils d'hibernation système et de réveil type *Wake On Lan* (WOL) permettra la prise en compte d'une gestion raisonnée de l'énergie électrique (*green-IT*).

Ce socle doit rester compact et le plus générique possible, il s'agit de limiter la dépendance aux spécificités de la configuration matérielle des postes de salle de TP. Il est identique pour tous les postes de salle de TP. Il est déployé selon la méthodologie actuelle, à savoir : mise en oeuvre et tests sur un poste de référence. La configuration est ensuite poussée à travers le réseau local sur les postes des salles au moyen d'un outil de réplication en mode bloc.

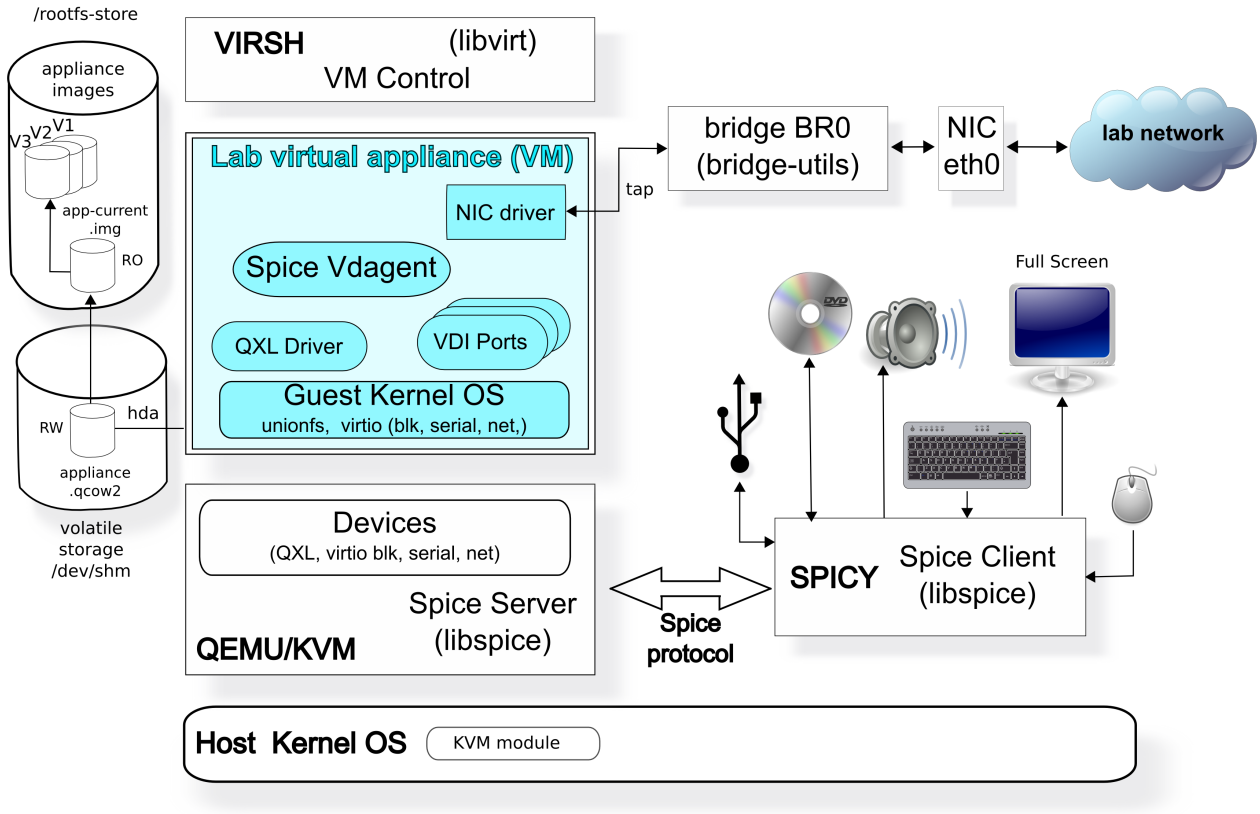

Figure 1 - Intégration système du poste de travail

4 *Appliances* Virtuelles

4.1 Notion d'*appliance* virtuelle

La notion de *virtual appliance*, dépasse le concept de paquetage applicatif. Celui ci permet de distribuer un logiciel applicatif actualisé pour une installation automatisée sur un système. Les formats les plus courants étant RPM pour les distributions des familles Red Hat ou Mandriva, DEB pour Debian et Ubuntu, ebuild pour Gentoo, ... Toutefois certains environnements logiciels nécessitent des configurations et/ou des dépendances aux bibliothèques logicielles système pouvant être lourdes ou complexes.

Avec la banalisation de la virtualisation on assiste à l'émergence d'environnements systèmes complètement assemblés ; intégrant le système d'exploitation de base, les applicatifs et leurs dépendances correctement pré-configurées. L'utilisateur est déchargé de la phase d'installation et d'intégration. S'il souhaite par exemple un environnement de type LAMP (Linux, Apache, MySQL, PHP), il sélectionne l'*appliance* correspondante qu'il n'a plus qu'à personnaliser. Les *appliances* virtuelles sont à la virtualisation ce que les liveCD sont pour la découverte ou le déploiement des systèmes libres. Les fichiers iso des liveCD sont d'ailleurs souvent utilisés sous forme d'*appliances* sur des machines virtuelles. Des sites de distribution de telles *appliances*, « clefs en main », pour un panel vaste d'outils et d'environnements ont fait leur apparition [14].

4.2 Diversité des appliances de TP

Avec ASTEROIDE les environnements de travail dédiés à chaque plate-forme de TP deviennent des *appliances* virtuelles dédiées (cf sous ensemble coloré de la figure 1). Chaque *appliance* est configurée avec les environnements logiciels propres à chaque usage (poste bureautique internet, (poste type windows, poste type GNU/Linux), plate-forme de développement Eclipse et langages associés (C, java,...), poste de labo de langue, poste de TP réseau sous forme des fichiers *iso* des différentes déclinaisons des *lab* VIMINAL [1], (mobidik, medi6, routing lab, vodka...) [2]. Ces *appliances* seront définies et configurées selon les besoins de chaque enseignant de TP. La souplesse de l'environnement QEMU-KVM permet de « durcir » ces *appliances*. En forçant les mode « RO » (*Read Only*) lecture-seule sur le système de fichiers (*rootfs*) de la machine virtuelle et en déportant les écritures sur un espace de débordement dédié et volatile (fichier COW *Copy On Write*) on rend « inaltérable » (à la manière des *LiveCD*) le système. En re-initialisant à chaque redémarrage l'espace de débordement COW on a l'assurance de redémarrer l'*appliance* dans son état primitif d'origine. Voire pour certains systèmes on évite ainsi la fragmentation de son espace disque, qui à la longue finit par pénaliser les performances du poste. Pour les postes sous GNU/Linux, la personnalisation ou la spécialisation de la configuration pourra être déportée sur un système de fichiers unifié tel que celui décrit dans GUSTAV (Gestion Unifiée des Systèmes de fichiers Transposée aux Appareillages Virtuels) [9], qui a fait l'objet d'un article au JRES 2009.

5 Le protocole SPICE

5.1 Des épices pour KVM

La virtualisation du poste de travail nécessite des protocoles de déport des ressources plus aboutis que les traditionnels protocoles de déport d'écran-clavier-souris. Le vénérable protocole X11 assure depuis des lustres l'accès en mode graphique aux applications locales ou distantes. Les entrées/sorties clavier, souris, écran vidéo (KVM *Keyboard, Video, Mouse*, à ne pas confondre avec *Kernel based Virtual Machine* technique de virtualisation choisie dans notre architecture) sont véhiculées par un protocole applicatif dédié visant à rendre transparent la localisation de l'applicatif. De nombreux protocoles coexistent aujourd'hui tant chez les éditeurs propriétaires (RDP chez Microsoft, ICA chez Citrix) que dans le monde du logiciel libre (le fameux X11, mais aussi le protocole RFB (*Remote Frame Buffer* qui a récemment fait l'objet d'une publication officielle dans le RFC6143 [15]) de VNC (*Virtual Network Computing*) pour ne citer que les plus connus.

Dans le cadre du VDI les ressources multimédia du poste virtualisé (audio, video, connectique USB, graphique avancé (2D, 3D), écrans multiples,...) doivent être accessibles sur le poste client quelque soit la localisation de celui ci (locale ou distante). Les principaux éditeurs en compétition sur le segment VDI, proposent aujourd'hui des protocoles avancés (PCoIP pour VMWare [16], Remote Desktop Services alias RDS pour Microsoft [17], *High Definition user eXperience* HDX pour Citrix/XenDesktop [18]) qui permettent de déléguer aux ressources matérielles disponibles sur le poste client les fonctions multimédia du poste virtualisé. Ainsi par exemple dans le cadre de la visualisation d'une vidéo, celle ci est effectivement décodée par le lecteur vidéo s'exécutant sur la machine virtuelle mais affichée sur l'écran du client dans la résolution optimale de ce dernier. De même le branchement d'une clé USB ou le chargement d'un DVD sur le poste client sera pris en compte par le système virtualisé s'exécutant sur les machines du *data center* ou déporté sur le *cloud*. Il ne s'agit donc plus de ne prendre en compte sur le poste distant uniquement les entrées-sorties écran, clavier, souris mais également celles relatives aux périphériques multimédia (lecteur DVD, clé USB, 3D, audio, multi-écrans, ...).

La startup israélienne Qumranet à l'origine du développement de KVM et de son intégration officielle dans le noyau Linux a également développé son propre éco système de VDI : le protocole SPICE [8] et le système SolidIce, sur lequel elle avait bâti son modèle économique. L'éditeur Red Hat qui a racheté Qumranet, a entrepris l'ouverture et la publication sous la licence ouverte et libre du protocole SPICE de déport des ressource audio, video, USB, ... Le développement du protocole et son client/serveur est coordonné par Red Hat. Le « *consortium* » Xorg assure, quant à lui, le développement des pilotes para-virtualisés (*drivers*) permettant l'intégration des fonctionnalités avancées de SPICE pour les machines virtuelles de type Unix, disposant d'un environnement graphique X11. Les drivers vidéo pour les environnements MS Windows sont fournis par Red Hat. Munies de ces

éléments les machines virtuelles disposent des capacités multimédia optimales disponibles sur le client SPICE (ajustement optimale de la résolution d'affichage, affichage multi-écrans, son, lecteur de DVD ,...). Dès lors qu'elle soit locale ou distante la machine virtuelle offre les mêmes facilités et la même qualité que les postes de bureau modernes.

ASTEROIDE nécessite l'intégration du protocole SPICE et du logiciel client (*spicy* du paquetage spice-GTK ou *spicec*) sur le socle de virtualisation de l'hôte, ainsi que des pilotes para-virtualisés d'affichage (*drivers* QXL et *daemon vdagentd*) dans les *appliances* virtuelles.

Le protocole SPICE fonctionne en mode client/serveur. Les fonctions serveur sont localisées au niveau du « virtualisateur » contrôlant la VM (le processus *qemu-kvm* dans notre cas). Elles assurent le contrôle des connexions clientes et le relayage des primitives multimédia générées par les applicatifs de la machine virtuelle vers le client distant. Le client, logiciel d'affichage dédié résidant sur le poste utilisateur (local ou distant), assure le décodage des primitives multimédia du protocole SPICE et l'exécution en s'appuyant sur les composants matériels locaux (carte vidéo, carte audio, ...). Au niveau de la machine virtuelle, le périphérique virtuel d'affichage doit être en mode para-virtualisé et non pas en mode émulation. C'est à dire qu'il doit « avoir conscience qu'il est virtualisé » afin de déléguer les primitives d'affichage au serveur. La para-virtualisation offre des meilleures performances comparativement à l'émulation matérielle, mais elle nécessite une adaptation à la virtualisation. Pour la pseudo carte video QXL, cela se traduit par un pilote spécifique (d*river* QXL) qui doit être décliné pour les différents OS des machines virtuelles.

Un élément facultatif de contrôle supplémentaire, dénommé *vdagent*, peut également être installé sur la VM, sous la forme d'un *daemon* (Unix) ou d'un service (MS Windows). Tournant en tâche de fond sur l'OS invité, il permet de modifier, à la volée, la configuration ou le comportement de la pseudo carte vidéo QXL. Des services de confort sont alors disponibles, tels que le couper/coller entre l'hôte et la VM, mais également l'adaptation dynamique de la résolution de l'affichage de la VM en fonction de la taille de la fenêtre ou de l'écran du client. Ainsi lorsque ce dernier est lancé en mode plein écran, l'affichage de la VM est réglé automatiquement à l'optimum de la résolution de la carte vidéo du client. Dès lors, à l'affichage, il devient quasiment impossible de distinguer un machine virtuelle d'une machine réelle. C'est ce comportement étonnant que le marketing anglo-saxon dénomme *User eXperience* (UX) qui est à l'origine du projet ASTREROIDE. L'usage de ce composant *vdagent*, rendra indistinctes les plate-formes de TP antérieures et celles qui auront été converties en *appliance*.

5.2 Des *appliances* sauce aux épices

Les *appliances* type ASTEROIDE sont des machines virtuelles classiques. Elles se composent d'un système de fichers dit *rootfs* contenant l'OS et les applicatifs d'une plate-forme de TP. Celle ci sera donc installée, configurée et testée en une version de référence qui pourra ensuite être clonée à volonté sur les postes des salles de TP ou sur les serveurs du *cloud*.

Le disque virtuel, (*rootfs*) embarque l'OS de base classique (GNU/Linux ou MS Windows), ainsi qu'un ensemble de pseudo composants matériels para-virtualisés. Outre la carte vidéo QXL évoquée précédemment, SPICE ne fonctionne de façon optimale que si les entrées/sorties série (canal de commandes) et réseau sont également para-virtualisées. Les versions para-virtualisées des pilotes KVM sont nativement intégrées dans le noyau Linux (paramètres VIRTIO_XXXX de la configuration du noyau) et mis à disposition par Red Hat sous forme de pilotes spécifiques pour les machines virtuelles Windows.

Le *daemon* UNIX *vdagentd* est disponible sous forme de paquetage pour la plupart des distributions, il peut également être compilé directement à partir des code sources. Le service équivalent pour MS Windows est mis à disposition par Red Hat.

Par rapport à l'installation classique de base du système, il s'agit donc d'ajouter

- Pour les *appliances* Linux :
    - les *drivers virtio* du noyau : ensemble de pilotes para-virtualisés, spécifiquement adaptés à la virtualisation KVM, pour des performances quasi natives,
    - *driver* vidéo para-virtualisé *xf86-video-qxl* du projet Xorg,
    - *unionfs* ou *aufs* pour conformité GUSTAV [9],
    - *spice-vdagent*, pour l'adaptation dynamique de la résolution de l'écran virtuel, (*daemon vdagentd*).
- Pour les *appliances* Microsoft Windows
    - *driver* Windows (*Windows qxl driver*, *Windows virtio-serial driver*) fournis par Red Hat,

- le service *spice-vdagent*, pour l'adaptation dynamique de la résolution de l'écran virtuel, fourni par Red Hat.

6 Implantation, le diable se cache dans les détails...

Un laborieux travail d'intégration a permis d'assembler les différents composants de base pour aboutir à une maquette de poste de travail. L'architecture système d'ASTEROIDE est donc validée. Cependant le déploiement n'est pour le moment pas effectif, car le poste prototype manque de stabilité du fait de la jeunesse de certains composants. SPICE et les drivers para-virtualisés sont encore en développement, leur intégration nécessite de s'appuyer sur les versions développeur, éventuellement *patchées* manuellement.

Sur la base de ce prototype nous pouvons évaluer les points qui nécessiteront un effort supplémentaire d'intégration avant d'envisager de migrer une salle de TP en pré-production. Nous résumons dans ce paragraphe les différents points d'achoppement.

L'interface d'accueil du poste, devrait dans l'idéal être graphique et afficher l'ensemble des *appliances* disponibles. Fonctionnellement elle doit se substituer au menu du *bootloader* des stations *multi-boot* actuelles. Dans le cadre du prototype, il s'agit pour le moment d'une simple page web, affichant les logos des différents systèmes. Un simple clic permet de lancer la machine virtuelle correspondante. La difficulté qu'il reste à lever, est que l'on ne souhaite qu'une seule *appliance* active à la fois. En effet, il n'est pas souhaitable que l'utilisateur puisse démarrer parallèlement plusieurs systèmes. Les ressources du poste, bien que confortables, restent limitées. A terme, il s'agit donc de développer une interface graphique spécifique, capable d'afficher l'ensemble des *appliances* disponibles sur le poste et d'assurer l'exclusivité des ressources multimédia (écran, lecteur DVD, port USB, ...) à la seule *appliance* en avant plan.

L'affichage en mode plein écran, à la résolution native du client SPICE est particulièrement étonnante : du point de vue de l'utilisateur, il devient quasiment impossible de distinguer à l'affichage une machine virtuelle d'un système natif. Cependant, cela n'est réalisable que lorsque la machine virtuelle a basculé en mode graphique. La phase de démarrage de la machine virtuelle, quelque soit le système (Linux ou Windows) demeure en mode texte et n'est pas pris en charge par SPICE. Les drivers para-virtualisés QXL ne sont pour le moment pas disponibles pour les consoles graphiques (mode *frame-buffer*). De plus les versions actuelles des clients SPICE (*spicy* ou *spicec*) ne permettent pas de verrouiller le mode plein écran au démarrage de la machine virtuelle. L'utilisateur peut, à sa guise, basculer du mode plein écran au mode fenêtre. Or on ne souhaite pas qu'il puisse accéder au système sous-jacent, quand bien même l'utilisateur sous lequel tourne celui ci ne dispose pas de droits avancés.

Pour les systèmes non libres, tels que Windows, la gestion des clés d'activation de licence reste problématique. L'image du système de fichiers de référence de l'*appliance* doit être clonée sur l'ensemble des postes. Les services de déploiement de parc de l'éditeur Microsoft prennent en compte la gestion des licences. Mais ce service a un coût non négligeable. De plus il s'avère à l'usage qu'un changement de version de l'hyperviseur KVM est détecté par l'OS de la machine virtuelle comme un changement de configuration matérielle qui nécessite une revalidation de la clé. De plus cette vérification de validité du système semble s'être durcie lors du passage de Windows XP à Windows7. Bien que le parc des postes des salles de TP soit encore sous la version XP, la question se posera à terme. La gestion centralisée des clés de licence devra donc être évaluée. Il faudra notamment évaluer et contrôler le nombre de clés à activer.

Le déploiement de l'architecture ASTEROIDE devra donc se faire en plusieurs phases. L'architecture cible reste le remplacement du *multi-boot* par l'hyperviseur « épicé » KVM et sa collection complète d'*appliances*. En attendant la stabilisation des développements des drivers QXL, du client SPICY et l'apprentissage de la distribution automatique et maîtrisée des clés de licence Windows, on peut démarrer une première phase conservant le *multi-boot* actuel en ajoutant un système supplémentaire ASTEROIDE ne comportant que la collection d'*appliances* VIMINAL (medi6, mobidik, vodka) et une *appliance* Linux équivalente du Linux natif. Par cette approche :

- On ne remet pas en cause les usages de la plateforme actuelle (Linux, Windows) ce qui garantit la continuité du service de la salle de TP dans les conditions habituelles,
- on peut enfin réaliser les séances de TP VIMINAL dans les salles banalisées. L'installation prochaine d'une nouvelle salle de TP réseau permettra de déployer cette approche et servira de phase de test sur une salle complète,
- On valide, ainsi, par l'usage l'utilisation de l'*appliance* Linux pour les TP sous cet environnement.

7 Evolutions et perspectives

Cette architecture offre des perspectives nouvelles d'utilisations mutualisées des ressources des salles de TP. Chaque poste d'une salle de TP, toutes proportions gardées, se comporte comme n'importe quel hyperviseur d'une machine de centre de calcul. En agrégeant les capacités de l'ensemble des postes des salles de TP sous forme de *cluster* on disposera d'un *cloud* privé d'une centaine d'hyperviseurs sur lequel peuvent migrer une grande variété de machines virtuelles. Les performances d'un tel *cluster* sont modestes au regard de la puissance des grands centres de données, mais à l'échelle de l'école elles peuvent néanmoins être mises à disposition des activités d'ingénierie, de recherche ou pour déployer des maquettes de réseaux. Les outils de type WOL (*Wake On Lan*) intégrés au niveau de chaque poste, permettront de mobiliser ponctuellement les ressources sur les périodes d'inactivité des salles, de 22H00-6H00 par exemple.

Pour ce type d'usage des ressources des salles de TP on distingue deux types de machines virtuelles :

- Les machines virtuelles de premier plan (les *appliances* de TP), affichées sur l'écran local du poste au travers du client SPICE (*spicy* ou *spicec*)
- les machines virtuelles d'arrière plan qui ne sollicitent pas les capacités interactives d'affichage locales (clavier/écran). dénommées machines virtuelles « fantômes ».

Les machines virtuelles fantômes peuvent être mutualisées pour constituer un « *ghost cluster* ». Il ne s'agit pas d'utiliser ces machines fantômes pour faire du calcul haute performance, mais de les solliciter pour partager une partie des ressources du poste de la salle de TP. Une première idée serait d'agréger une partie de l'espace disque de chaque poste, sous forme d'une partition dédiée, pour déployer un système de fichiers distribués (GFS, Ceph, Hadoop,...).

-----oOo-----

8 Bibliographie

[1] Jacques Landru, Jean Philippe Vendeborre, [1] VIMINAL (VIrtual Model for Ip Network Architecture Lab) Dans *Actes du congrès JRES2005* http://2005.jres.org/paper/49.pdf

[2] VIMINAL collection : Medi6, Mobidik, Vodka :
http://www.telecom-lille1.eu/people/landru/viminal

[3] Citrix XenDesktop : http://www.citrix.fr/Produits_et_Solutions/Produits/XenDesktop/

[4] VMWare View : http://www.vmware.com/products/view/overview.html

[5] Microsoft VDI : http://www.microsoft.com/uk/windows/enterprise/solutions/virtualization/improve-flexibility.aspx
http://www.microsoft.com/france/virtualisation/produits/poste-de-travail/default.mspx

[6] Red Hat Enterpise Virtualization for desktop (RHEV) :
http://www.redhat.com/virtualization/rhev/desktop/
http://www.redhat.com/f/pdf/rhev/final2.2/DOC250_RHEV_forDesktops_2819417_0610_cw_web.pdf
http://www.redhat.com/f/pdf/rhev/final2.2/DOC255_RH_WP_RHEV_D_2832287_0610_ma_web.pdf

[7] Kernel based Virtual Machine (KVM) :
http://www.linux-kvm.org/

[8] Simple Protocol for Independent Computing Environment (SPICE) : http://spice-space.org/

[9] Jacques Landru, Tovoherizo Rakotonavalona, GUSTAV Gestion Unifiée des Systèmes de fichiers Transposée aux Appareillages Virtuels. Dans *Actes du congrès JRES2009*
https://2009.jres.org/planning_files/article/pdf/14.pdf

[10] Wikipedia, Comparison of platform virtual machines :
http://en.wikipedia.org/wiki/Comparison_of_platform_virtual_machines

[11] QEMU : http://wiki.qemu.org/Main_Page

[12] La librairie Libvirt : http://libvirt.org/

[13] Fabric : http://docs.fabfile.org/en/0.9.0/index.html

[14] TurnKey Linux : http://www.turnkeylinux.org/all

[15] Stéphane Bortzmeyer, Résumé libre et personnel du RFC 6143 :
http://www.bortzmeyer.org/6143.html

[16] Vmware View 4 with PCoIP :
http://www.vmware.com/files/pdf/VMware-View-4-PCoIP-DS-EN.pdf

[17] Microsoft Remote Desktop Serivices (RDS) :
http://www.microsoft.com/en-us/server-cloud/windows-server/remote-desktop-services.aspx
http://download.microsoft.com/download/D/4/C/D4C2C737-F4E0-43C3-BE7C-5E3873AEAD79/Datasheet%20WS08R2SP1_RDS.pdf

[18] Citrix High Definition user eXperience (HDX) :
http://hdx.citrix.com/

[19] Gestionaires d'infrastructures virtuelles (cloud) :
http://libvirt.org/apps.html
http://www.linux-kvm.org/page/Management_Tools