

# Les Infrastructures Virtuelles à la demande pour un usage flexible de l'Internet

Fabienne Anhalt\*, Guilherme Koslovski\*, Marcelo Pasin<sup>†</sup>, Jean-Patrick Gelas<sup>‡</sup> and Pascale Vicat-Blanc Primet\*

\*INRIA - ENS Lyon, Email : {fabienne.anhalt, guilherme.koslovski, pascale.primet}@ens-lyon.fr

<sup>†</sup>Universidade de Lisboa, Email : pasin@di.fc.ul.pt

<sup>‡</sup>UCBL - ENS Lyon, Email : jean-patrick.gelas@ens-lyon.fr

**Résumé**—L'usage de l'Internet évolue d'une telle façon qu'il repose non seulement un service de communication, mais aujourd'hui également des services de stockage et de calcul. Dans ce contexte, le concept de "serviced" infrastructure devient un paradigme pour supporter des besoins de communication et de calcul divers. Des solutions pour créer des infrastructures personnalisées à la demande sur Internet sont requises. Le but de cet article est d'explorer comment le paradigme de virtualisation utilisé classiquement pour résoudre des problèmes d'isolation, de sécurité et de consolidation peut être appliqué sur tous les éléments du réseau. Nous étudions le concept d'*infrastructure virtuelle* qui combine des ressources virtuelles de calcul, des liens et des routeurs virtuels. Cette approche nous permet de faire cohabiter plusieurs infrastructures virtuelles homogènes de types, de propriétaires et de protocoles différents à l'intérieur d'un réseau physique d'interconnexion complexe. Pour décrire de telles infrastructures virtuelles personnalisées, nous proposons un langage extensible. Nous implémentons un modèle de routeur virtuel avec Xen, puis nous analysons la performance du routeur virtuel. Enfin nous montrons comment la relocation dynamique de routeur virtuel permet de réorganiser des infrastructures virtuelles au dessus du réseau physique.

## I. INTRODUCTION

Les techniques de virtualisation, introduites par IBM dès 1973 [1] sont devenues très populaires avec l'arrivée des systèmes tels que Xen [2] et VMware [3]. Elles sont en particulier très utiles pour accroître l'isolation, la mobilité, la reconfiguration dynamique et la tolérance aux pannes des systèmes distribués. La virtualisation est employée par exemple pour héberger plusieurs serveurs sur une même machine physique. Cette méthode permet aux organisations et aux entreprises de mieux exploiter les ressources physiques d'une machine en termes de CPU et d'énergie. De plus, la virtualisation des serveurs permet d'accroître leur sécurité en les isolant les uns des autres et un serveur virtuel peut être migré sur une autre machine physique en cas de panne ou d'opération de maintenance.

Pour introduire ces avantages à l'intérieur même des réseaux, une idée émergente est de s'appuyer sur les techniques de virtualisation non seulement dans les serveurs en bordure du réseau d'interconnexion mais aussi sur les équipements réseau comme les routeurs. Cette technique permet à des réseaux de types, de propriétaires et de protocoles différents de cohabiter dans un même réseau physique. De tels réseaux IP virtuels constituent une approche émergente pour mettre à disposition de l'utilisateur un service d'infrastructures réseau virtuelles,

lui permettant de créer et d'administrer un réseau individuel avec un routage IP personnalisé au dessus d'une infrastructure partagée. De telles infrastructures virtuelles peuvent être utilisées pour la recherche (implémentation et test de nouveaux protocoles), puis pour découpler service et infrastructure dans l'Internet réel [4], et représentent une approche pour une nouvelle architecture de dorsale.

Si la virtualisation permettait de résoudre des questions clés de l'Internet actuel (sécurité, mobilité, fiabilité, configurabilité), elle introduirait néanmoins un surcoût dû aux couches supplémentaires s'interposant entre les machines virtuelles et le matériel. Par ailleurs, à la problématique du routage et de l'allocation de bande passante dévolue aux routeurs traditionnels s'ajoute le problème du partage des ressources locales par différents réseaux virtuels. Ce partage de ressources comme les interfaces réseau, les processeurs et la mémoire constitue un défi pour obtenir une performance prévisible, stable et optimale.

L'objectif de ce travail est d'explorer cette approche d'infrastructures réseau virtuelles composables à la demande. Nous proposons un modèle de composant "routeur virtuel" pour ces infrastructures et nous analysons ses performances dans une implantation avec la technologie Xen [2]. Pour organiser en réseau et configurer à la demande de tels routeurs, nous introduisons un langage que nous avons spécifié pour la modélisation d'infrastructures virtuelles. Il permet aux utilisateurs et aux composants d'intergiciels de décrire les ressources virtuelles et physiques pour définir un modèle de configuration et d'organisation (topologie réseau). Ce langage appelé VXDL permet de définir l'organisation des routeurs virtuels et le contrôle de bande passante, permettant aux utilisateurs de spécifier un modèle et l'organisation d'infrastructure virtuelle. Dans la section II nous définissons les concepts principaux des Infrastructures réseau virtuelles (VXI) avec leur modélisation en utilisant VXDL et le modèle de routeur virtuel en détaillant son architecture et implémentation. Puis, en section III, nous décrivons des expériences relatives à la performance globale d'un routeur virtuel et la capacité d'adaptation d'une VXI aux besoins des utilisateurs par réorganisation dynamique. Enfin, la section V conclut cet article et propose quelques perspectives.

## II. INFRASTRUCTURE RÉSEAU VIRTUELLE (VXI)

Une infrastructure réseau virtuelle (VXI) est composée d'un ensemble de nœuds virtuels interconnectés par des liens vir-

tuels, à travers lesquels la communication peut être contrôlée par des routeurs virtuels. Une telle infrastructure peut être adaptée aux besoins de l'application que l'utilisateur a spécifiés. Les ressources et liens virtuels peuvent être provisionnés à la demande, afin que la VXI respecte les besoins réels des applications.

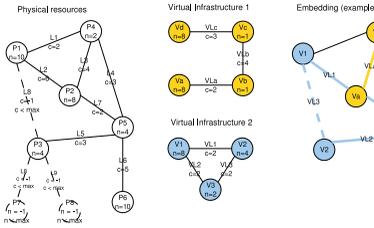


FIG. 1. Composition de VXI.

La figure 1 montre un exemple de composition de VXI, utilisant un graphe noté  $G(V, E)$  où  $V$  sont les sommets qui représentent les ressources de calcul, de stockage ou de routage et  $E$  les arêtes qui correspondent aux liens. Cette notation est utilisée pour les ressources physiques ainsi que pour les ressources virtuelles. Pour les ressources virtuelles, les sommets peuvent correspondre à des ressources (calcul, stockage), routeurs ou nuages et les arêtes sont des liens virtuels qui interconnectent ces ressources dans la topologie requise par la VXI. Chaque composant (ressource ou lien) présente un nom, une capacité ( $c$ ) et un nombre de ressources ( $n$ ). Pour les ressources physiques,  $c$  représente les ressources libres qui peuvent être allouées ; pour les ressources virtuelles, ce paramètre représente la configuration désirée. Par exemple, le nœud physique  $P1$  a pour définition  $n = 10$  ce qui peut représenter un cluster composé de 10 processeurs.  $P1$  est connecté au nœud physique  $P4$  par un lien physique (arête  $L1$ ) dont la capacité est définie par  $c = 2$ . Dans ce cas,  $c$  représente une configuration de lien comme la bande passante, la latence ou une combinaison des deux.

Le premier graphe (côté gauche) représente un ensemble de ressources physiques (substrat de réseau) qui va accueillir des ressources virtuelles. Les sommets et arêtes en pointillés représentent respectivement des ressources et liens provisionnés, qui ne sont pas disponibles pour être alloués dans un premier temps, mais qui peuvent être alloués plus tard à la demande. Les graphes au milieu et à droite correspondent respectivement à des infrastructures virtuelles et un emboîtement possible sur un substrat physique.  $VXI 1$  et  $VXI 2$  sont deux infrastructures virtuelles différentes qui doivent partager les ressources physiques. Dans cet exemple de solution de *matching*, les deux infrastructures virtuelles peuvent être allouées dans les ressources physiques, tout en satisfaisant leurs demandes en capacité.

Une approche clé dans l'allocation des VXIs est la possibilité de reconfiguration dynamique. A la base, cette situation

peut se produire dans plusieurs cas : I) une VXI nécessite plus de ressources (à la demande) ; II) une nouvelle requête de VXI arrive ; ou III) une ressource physique n'est plus disponible. Dans le cas I, des composants provisionnés (ressources et liens) peuvent être sollicités et alloués pour une VXI existante. Le cas II correspond à une nouvelle requête de VXI qui peut être allouée sur le même substrat physique qu'une VXI déjà existante, mais nécessite donc une réorganisation de celle-ci pour répondre à la spécification de l'utilisateur. Dans le cas III, nous considérons le facteur temps, c'est-à-dire que les ressources du substrat ne peuvent être disponibles que pour une certaine période de temps (en accord avec la réservation), et quand cette période se termine, chaque VXI correspondante doit être réallouée. Grâce à l'isolation offerte par les techniques de virtualisation, un composant (nœud ou routeur virtuel) peut être migré d'une ressource physique vers une autre sans interruption d'exécution [5]. Ce concept est la base pour le repositionnement de VXIs d'une manière transparente pour les applications utilisateurs.

A. Le composant routeur virtuel

Cette section décrit l'architecture et l'implémentation avec Xen du composant "routeur virtuel" d'une VXI.

1) Architecture: Le principe d'un routeur virtuel est qu'un seul routeur physique est divisé en plusieurs routeurs virtuels qui résident à l'intérieur de machines virtuelles.

L'idée est de faire tourner un routeur logiciel à l'intérieur

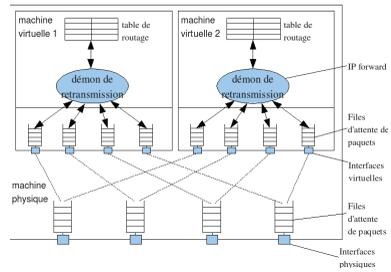


FIG. 2. Machine hébergeant deux routeurs virtuels.

de telles machines virtuelles. L'architecture du modèle de routeur résultant est représentée sur la figure 2. Le principe de fonctionnement à l'intérieur d'une machine virtuelle est le même que celui d'un routeur logiciel classique, sauf que les machines virtuelles n'ont pas accès directement aux interfaces physiques. Les paquets sont obligés de passer par des interfaces réseau virtuelles pour atteindre l'interface physique. Un mécanisme de multiplexage et de démultiplexage implémenté dans une couche intermédiaire située entre le matériel et les machines virtuelles permet d'acheminer les paquets entre les interfaces virtuelles et les interfaces physiques correspondantes.

2) *Implémentation avec Xen*: L'accès au réseau par des machines virtuelles dans Xen comporte une couche supplémentaire de virtualisation. Dans Xen, chaque machine virtuelle réside dans un *domaine invité*. Parmi les domaines invités, seulement un domaine privilégié appelé *domaine 0* a l'accès direct au matériel. Les autres domaines sont appelés *domaines U* (pour "unprivileged"). Chaque domaine U possède une interface virtuelle pour chaque interface physique de carte réseau de la machine physique. Cette interface virtuelle est accessible via un *split device driver* d'interface virtuelle qui est constitué de deux parties, le *frontend driver* en domaine U et le *backend driver* en domaine 0 [6]. La figure 3 illustre

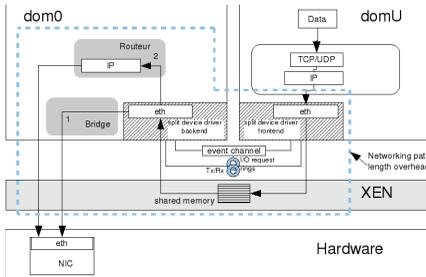


FIG. 3. Parcours d'un paquet réseau avec Xen, en émission depuis un domaine U jusqu'à la carte réseau.

le chemin suivi par les paquets réseau en émission depuis une machine virtuelle résidant dans un domaine U jusqu'au lien physique. Les paquets envoyés sur le réseau traversent l'interface virtuelle du domaine U, puis sont copiés dans un segment de mémoire partagée par l'hyperviseur Xen et ensuite transmis dans le domaine 0. Deux buffers circulaires à jetons contiennent les requêtes d'envoi (Tx) et de réception (Rx) de paquets. Un canal d'événements permet de signaler la présence d'un paquet à envoyer ou à recevoir. A l'intérieur du domaine 0, les paquets en provenance des différentes machines virtuelles sont multiplexés et acheminés vers l'interface physique de la carte réseau. L'acheminement se fait par défaut par un pont au niveau de la couche liaison (chemin 1 sur la figure 3). L'alternative est de mettre en place un routeur en domaine 0. Dans ce cas les paquets sont remontés dans la pile réseau jusqu'à la couche réseau pour être acheminés (chemin 2 sur la figure 3).

La réception de paquets sur une machine virtuelle suit le même mécanisme mais dans le sens inverse.

Le chemin supplémentaire dû à la virtualisation qu'un paquet doit parcourir pour atteindre la carte réseau depuis un domaine U ou inversement est encadré en pointillé sur la figure 3. Un surcoût de CPU non négligeable peut-être engendré par la copie supplémentaire entre domaine U et domaine 0 dans la mémoire partagée.

B. Spécification de VXI avec VXDL

Pour donner aux utilisateurs la possibilité de choisir les ressources nécessaires, leur topologie et leur organisation pour une application, nous proposons VXDL, un langage de spécification pour l'interconnexion de ressources virtuelles en réseaux virtuels [7]. Le premier but de VXDL est d'intégrer l'interconnexion réseau, les contraintes de virtualisation et le bail à la description classique des ressources. Avec ces concepts, les utilisateurs peuvent modéliser une VXI en accord avec leurs besoins, en explorant des points clés existants dans des descriptions virtuelles, comme : I) le lieu de l'ancrage, II) la topologie virtuelle et III) la spécification de bail.

Une option d'ancrage permet à l'utilisateur de spécifier l'endroit physique où un ou plusieurs composants doivent être alloués. Ce paramètre sert à modéliser des environnements où il y a une dépendance locale pour l'exécution. Pouvant détailler la topologie réseau, des utilisateurs peuvent adapter leur infrastructure virtuelle pour des applications sensibles à la communication, c'est-à-dire des applications qui ont besoin de transférer un grand volume de données ou qui nécessitent une latence réseau faible. La spécification peut détailler des paramètres internes en groupes (nuages) ou entre groupes, appliquant des paramètres comme la latence et la bande passante à tous les liens. A travers l'identification de la source et de la destination, le même composant peut avoir différents canaux de communication. La spécification d'un bail virtuel sert à identifier la période de temps où les ressources sont nécessaires, à travers une identification nominale de variables de temps, ce qui définit les composants nécessaires et la durée d'exécution (transfert de données ou temps de calcul).

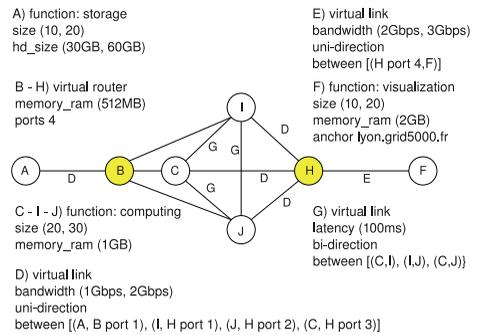


FIG. 4. Exemple d'une spécification de VXI.

La figure 4 montre un exemple de spécification d'une VXI par un graphe qui peut être décrit entièrement avec VXDL. Les sommets B et H identifient deux routeurs virtuels ayant chacun 4 ports, et les sommets A, C, I, J et F représentent une ressource ou un groupe de ressources qui ont des configurations et besoins différents. Sur cette figure, certaines arêtes ont des paramètres de configuration définis comme la latence, la

bande passante, la direction et le port destination. Cette VXI peut être utilisée dans différents contextes, avec des périodes d'exécution prédéfinies ou non. Dans une première période (I), les données stockées dans le composant *A* sont à transférer aux composants *C*, *I* et *J* en utilisant une configuration de lien entre eux et un routeur virtuel *B*. Au cours de la période II, les composants *C*, *I* et *J* peuvent traiter les données transférées en accord avec les capacités des liens avec la définition *G*. Dans une période finale III, les composants *C*, *I* et *J* peuvent effectuer un transfert de données vers le composant *F*, avec une bande passante contrôlée par le routeur virtuel *H*.

En plus de la définition d'applications utilisateurs, la version actuelle de la grammaire VXDL a été définie en utilisant le format XML et est utilisée dans les projets ANR HIPCAL<sup>1</sup> et CARRIOCAS<sup>2</sup>, pour la communication standard par des composants système, décrivant des ressources physiques et virtuelles.

### III. EXPÉRIMENTATIONS

Dans cette section, nous proposons une évaluation quantitative de la performance du composant routeur virtuel pour une VXI et les avantages de la réorganisation de VXIs.

#### A. Description de la plateforme expérimentale

Toutes les expériences sont effectuées sur la plateforme française de test Grid'5000 [8]. Les machines utilisées sont des IBM eServer 326m équipés de deux interfaces réseau Gigabit Ethernet (pilote tg3), de deux CPU AMD Opteron 246 (2.0GHz/1MB) à un cœur chacun et 2 GB de mémoire. La configuration logicielle utilisée est l'hyperviseur XEN 3.2.1 avec le noyau linux modifié 2.6.18.8-xen en domaine 0 et le noyau 2.6.18.8 en domaine U.

Les machines intervenant dans chaque expérience sont interconnectées par un lien Gigabit Ethernet 1Gb/s et sont interconnectées via le même commutateur.

Les expériences de référence sont menées sur une distribution GNU/Linux Debian (Sid) avec un noyau 2.6.18.

#### B. Transmission sur des routeurs virtuels

a) *Débit et CPU*: Pour déterminer les performances de routeurs virtuels, nous générons du trafic UDP avec des paquets de 1500 octets puis des paquets de 64 octets et les envoyons à travers un ou plusieurs routeurs virtuels (allant de 1 jusqu'à 8) partageant une machine physique. Tous les flux sont envoyés à un débit maximal depuis des machines physiques distinctes pour éviter des biais. Puis nous évaluons le débit TCP de bout-en-bout. La figure 5 montre le débit individuel par routeur virtuel ( $R_i$ ) et sa variation et le débit agrégé ( $R_{agg}$ ) avec des paquets de 1500 octets en UDP et TCP. Le tableau I détaille le taux de paquets UDP et le taux de pertes par domaine U avec des paquets de petite et grande taille. Avec UDP, le taux de pertes avec des paquets de grande taille sur chaque machine virtuelle correspond à  $1 - R_{theorique} / (N * R_{theorique})$  où  $R_{theorique}$  est le *goodput*

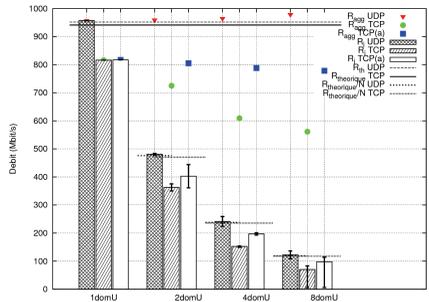


FIG. 5. Débit côté récepteur sur 1, 2, 4 ou 8 routeurs virtuels simultanément.

	Grande taille (1500 octets)		Petite taille (64 octets)	
	paquets/VM	pertes/VM	paquets/VM	pertes/VM
1 VM	81284	0 %	109685	60 %
2 VM	40841	50 %	12052	96 %
4 VM	20486	75 %	/	/
8 VM	10393	87 %	/	/
linux	81277	0.00 %	356494	0.06 %

TAB. I  
TAUX DE TRANSMISSION UDP MOYEN ET TAUX DE PERTES  
PAR DOMAINE U.

théorique en UDP. La bande passante est partagée équitablement entre les routeurs virtuels. Les résultats montrent l'efficacité des routeurs ; le débit (Figure 5) correspond à la valeur obtenue sur un routeur linux classique (957 Mb/s). Les dépassements sont dus à l'augmentation de débit UDP en fonction du nombre de domaines U et peuvent s'expliquer par un démarrage non synchrone des différents flux. Le partage des ressources est équitable et donc la performance du modèle est prédictible. Avec des paquets de petite taille sur 4 ou 8 routeurs virtuels, le domaine 0 est saturé et n'est donc plus capable de transmettre pour tous les routeurs virtuels. Avec TCP, le débit diminue, en particulier avec un nombre de routeurs virtuels qui augmente. Ceci peut être lié au fait que la latence augmente comme décrit dans la section suivante. En donnant une part de CPU plus grande au domaine 0 (Figure 5, expérience Xen TCP(a)), le débit TCP agrégé augmente. La part de CPU nécessaire pour le domaine 0 est beaucoup plus importante que la part requise par les domaines U, dû au fait que le domaine 0 est en charge d'acheminer tous les paquets entre les interfaces physiques et les interfaces virtuelles (p. ex. 16 dans le cas de 8 routeurs virtuels). En augmentant la part de CPU du domaine 0, celui-ci est capable de traiter plus de paquets pendant chaque période d'ordonnancement et permet donc aux domaines U d'acheminer plus de paquets. L'utilisation globale de CPU et le débit TCP augmentent.

b) *Latence*: Dans cette expérience, nous mesurons la latence sur un routeur virtuel, pendant que des routeurs virtuels concurrents (1, 3 ou 7) qui partagent la même machine phy-

<sup>1</sup>http://hipcal.lri.fr

<sup>2</sup>http://www.carriocas.org

sique sont inactifs ou stressés par la transmission de flux TCP à un débit maximal. Le tableau II représente les résultats dans les deux cas. La latence à travers un routeur virtuel partageant une

Latence(ms)	linux	1 VM	2 VM	4 VM	8 VM
inactif	0.084	0.147	0.150	0.147	0.154
stressé			0.888	1.376	3.8515

TAB. II

LATENCE INTRODUITE PAR UN ROUTEUR EXÉCUTANT 1, 2, 4 OU 8 ROUTEURS VIRTUELS. LES ROUTEURS VIRTUELS SONT INACTIFS OU STRESSÉS PAR LA TRANSMISSION DE FLUX TCP.

machines physique avec d'autres routeurs virtuels inactifs est d'environ 0.15 ms quel que soit le nombre de routeurs virtuels, ce qui est presque le double de la latence sur un routeur linux classique (0.084 ms). Dans le cas d'un système stressé, la latence sur le routeur virtuel considéré augmente avec le nombre de routeurs concurrents transmettant des flux TCP à un débit maximal. La latence moyenne atteint presque 4 ms sur un routeur virtuel qui partage une machine physique avec 7 routeurs virtuels qui transmettent des flux TCP. La latence sur des routeurs virtuels est très dépendante de l'activité des machines virtuelles concurrentes.

C. Migration d'un routeur virtuel

Dans cette expérience, nous considérons deux VXIs partageant un substrat physique. Une VXI nécessite plus de débit que ce que la configuration actuelle lui permet. Nous effectuons donc une migration et observons le changement d'assignement de ressources physiques et de performance. Dans les deux VXIs considérées, chacune étant composée de 8 hôtes virtuels organisés en deux sous réseaux virtuels interconnectés par un routeur virtuel par VXI. Chaque sous-réseau (4 hôtes virtuels) réside dans une seule machine physique (Figure 6). Les deux routeurs virtuels des deux VXIs partagent

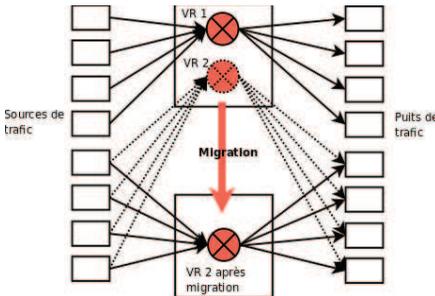


FIG. 6. Architecture de l'expérience de migration d'un routeur virtuel (VR).

au début la même machine physique, puis un des deux routeurs virtuels est migré sur une autre machines physique libre de façon à ce que chaque routeur virtuel dispose d'une machine physique à lui qui n'héberge pas d'autres routeurs virtuels. Ainsi chaque routeur pourra transmettre à un débit deux fois

plus élevé.

La figure 7 représente les résultats obtenus en termes de débit dans le cas de TCP et dans le cas d'UDP côté récepteur au cours d'une période de 240 secondes. La migration du routeur est déclenchée au bout de 20 secondes. Dans les deux cas

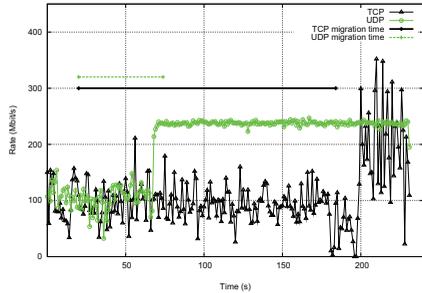


FIG. 7. Evolution du débit sur un routeur au cours de sa migration.

(TCP ou UDP), on remarque que le débit jusqu'à la fin de la migration varie autour du débit équitable maximal (environ 100 Mbit/s) qu'on peut obtenir sur un routeur (cf. résultats précédents), sachant que la machine comporte deux routeurs virtuels, chacun en train de traiter quatre flux. A la fin de la période de migration, le débit double, ce qui s'explique par le doublement des ressources physique : chaque routeur virtuel peut utiliser la totalité des ressources d'une machine physique au lieu de la partager. En TCP, à la fin de la période de migration, le débit chute soudainement pendant quelques secondes car le routeur virtuel est arrêté dans la machine physique d'origine, puis réactivé dans la machine physique de destination. Cette dégradation de débit est liée à une augmentation importante de la latence. Celle-ci atteint plus que 50 ms. Ce pic de latence s'explique par un ARP reply que Xen utilise pour signaler que l'adresse IP de la machine virtuelle change de machine physique. En UDP, on observe que la latence augmente pendant la phase de migration, atteignant par moments le double de la valeur initiale, soit environ 20 ms.

IV. TRAVAUX RELATIFS

L'approche des infrastructures réseau virtuelles est une idée émergente et traitée dans différents contextes, comme CABO (Concurrent Architectures are Better than One) [4] qui explore la virtualisation et décrit un concept pour permettre à des fournisseurs de service Internet de proposer différents services de bout-en-bout à travers des réseaux hétérogènes appartenant à des fournisseurs d'infrastructure distincts. Comme dans notre proposition, l'idée est de découpler le service du support physique. Peu d'implémentations de réseaux virtuels ont été proposées. VINI(Virtual Network Infrastructures) [9] permet également de partager le réseau physique en plusieurs réseaux

virtuels. Différents protocoles de routage peuvent être implémentés. VINI virtualise le plan de contrôle et propose d'utiliser différents protocoles de routage. Dans notre solution, la virtualisation complète permet d'utiliser un système d'exploitation totalement personnalisé et donc de créer des infrastructures virtuelles entièrement configurables.

Plusieurs langages pour la description et sélection de ressources comme ClassAd [10], vgDL [11] ou SWORD [12] ont été proposés. Ces langages se distinguent par leur grammaire, les paramètres proposés et les spécificités d'implémentation. Mais le réseau d'interconnexion qui joue un rôle central dans des infrastructures de calcul distribués ne peut pas être finement modélisé par ces langages.

Les performances de routeurs virtuels ont été étudiées dans les versions précédentes de Xen. Avec Xen 2.0, le débit mesuré sur un domaine U était très faible comparé au débit théorique du lien, presque quatre fois inférieur [13]. Les auteurs ont apporté des optimisations majeures pour augmenter la performance de façon notable en émission. La transmission sur des routeurs virtuels avec Xen 3.0 a été traitée [14] et montre que le débit agrégé de transmission sur des domaines U atteint moins de 25% du débit théorique pour des paquets de taille minimale. Dans un article récent [15], les auteurs ont montré que la mémoire est le goulot d'étranglement majeur dans Xen.

La migration de machines virtuelles a été proposée pour Xen [5], puis un concept très récent dédié à la migration de routeurs virtuels [16] propose de migrer séparément le plan de contrôle et le plan de données.

## V. CONCLUSION ET PERSPECTIVES

Cet article explore le nouveau concept de réseaux virtuels comme infrastructures de service. Cette approche peut pallier à la rigidité de l'Internet actuel. Nous avons proposé un modèle d'infrastructure réseau virtuelle composée de routeurs virtuels totalement configurables et un langage de spécification pour permettre à l'utilisateur de décrire l'infrastructure virtuelle adaptée à son application pour se créer un réseau personnalisé. Dans nos expériences, nous avons montré que la performance réseau des routeurs virtuels est prometteuse. Nous avons également montré que la migration de routeurs peut permettre de changer à la demande l'allocation des ressources pour obtenir une meilleure composition du substrat physique afin d'offrir la performance requise. Notre objectif actuel est d'intégrer le service d'allocation d'infrastructures virtuelles à la demande dans le logiciel HIPerNET<sup>3</sup>.

## REMERCIEMENTS

Nous remercions Olivier Mornard (IE INRIA) pour sa participation. Ce travail a été financé par l'INRIA et le Ministère de l'Éducation et de la Recherche Française via le projet ANR HIPCAL et par le projet CARRIOCAS pôle System@tic. Les expériences présentées dans cet article ont été exécutées sur la plateforme expérimentale Grid'5000, une

initiative du Ministère de Recherche Français à travers l'action ACI GRID, INRIA, CNRS et RENATER et d'autre partenaires (<https://www.grid5000.fr>).

## RÉFÉRENCES

- [1] G. J. Popek and R. P. Goldberg, "Formal requirements for virtualizable third generation architectures," in *SOSP '73 : Proceedings of the fourth ACM symposium on Operating system principles*, (New York, NY, USA), p. 121, ACM, 1973.
- [2] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in *SOSP '03 : Proceedings of the nineteenth ACM symposium on Operating systems principles*, (New York, NY, USA), pp. 164–177, ACM, 2003.
- [3] J. Sugerman, G. Venkitchalam, and B.-H. Lim, "Virtualizing I/O devices on VMware workstation's hosted virtual machine monitor," in *Proc. 2001 Usenix Annual Technical Conference*, pp. 1–14, Usenix Assoc., 2001.
- [4] N. Feamster, L. Gao, and J. Rexford, "How to lease the internet in your spare time," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 1, pp. 61–64, 2007.
- [5] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in *Proc. 2nd Symposium on Networked Systems Design and Implementation (NSDI '05)*, (Boston, USA), Usenix, May 2005.
- [6] D. Chisnall, *The Definitive Guide to the Xen Hypervisor*. Prentice Hall, 2007.
- [7] G. P. Koslovski, P. Vicat-Blanc Primet, and A. S. Charão, "VXDL : Virtual Resources and Interconnection Networks Description Language," in *GridNets 2008*, Oct. 2008.
- [8] F. Cappello, P. Primet et al., "Grid'5000 : A large scale and highly reconfigurable grid experimental testbed," in *GRID '05 : Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing*, pp. 99–106, IEEE Computer Society, 2005.
- [9] A. Bavier, N. Feamster, M. Huang, L. Peterson, and J. Rexford, "In vini veritas : realistic and controlled network experimentation," in *SIGCOMM '06 : Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 3–14, ACM, 2006.
- [10] R. Raman, M. Livny, and M. H. Solomon, "Matchmaking : Distributed Resource Management for High Throughput Computing," in *HPDC*, pp. 140–, 1998.
- [11] A. Chien, H. Casanova, Y. suk Kee, and R. Huang, "The Virtual Grid Description Language : vgDL," Tech. Rep. TR095, VGrADS Project, 2004.
- [12] D. Oppenheimer, J. Albrecht, D. Patterson, and A. Vahdat, "Design and implementation tradeoffs for wide-area resource discovery," 2005.
- [13] A. Menon, A. L. Cox, and W. Zwaenepoel, "Optimizing network virtualization in xen," in *ATEC '06 : Proceedings of the annual conference on USENIX '06 Annual Technical Conference*, pp. 2–2, USENIX Association, 2006.
- [14] N. Egi et al., "Evaluating Xen for Router Virtualization," in *ICCCN*, pp. 1256–1261, 2007.
- [15] N. Egi, A. Greenhalgh, M. Handley, M. Hoerdet, F. Huici, and L. Mathy, "Fairness issues in software virtual routers," in *PRESTO '08*, pp. 33–38, ACM, 2008.
- [16] Y. Wang, E. Keller, B. Biskeborn, J. van der Merwe, and J. Rexford, "Virtual routers on the move : live router migration as a network-management primitive," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 231–242, 2008.

<sup>3</sup><http://hipcal.lri.fr>