Remplacer un routeur par un serveur Linux : retour d'expérience des passerelles d'accès à Grid'5000

Dimitri Delabroye, Simon Delamare, David Loup, Lucas Nussbaum

Remplacer un routeur par un serveur Linux

Retour d'expérience des passerelles d'accès à Grid'5000

Dimitri Delabroye
INRIA Lille Nord Europe

Simon Delamare
CNRS, Laboratoire de l'Informatique du Parallélisme (LIP) , Lyon

David Loup
INRIA Rhône-Alpes, Laboratoire de l'Informatique du Parallélisme (LIP), Lyon

Lucas Nussbaum
Université de Lorraine, LORIA

Résumé

Grid'5000 est une infrastructure pour la recherche en informatique distribuée (Clouds, réseau, *HPC etc.). La plateforme est composée d'environ 700 nœuds, mis à disposition des chercheurs, répartis sur 8 sites reliés entre eux par un réseau 10 Gbits/s fourni par* Renater. *L'interconnexion de* Grid'5000 *avec Internet est réalisée par 2 routeurs.*

Jusqu'en 2018, des équipements réseaux traditionnels étaient utilisés pour cette interconnexion avec Internet, mais les performances de ces routeurs n'étaient pas suffisantes pour répondre aux besoins des chercheurs, qui téléchargent des données de plus en plus volumineuses depuis Internet (images Docker*, grands jeux de données…)*

Les débits plafonnaient en effet à 150 Mbits/s, bien loin du Gigabit/s qu'ils sont supposés atteindre. Les devis réalisés pour les remplacer par des équipements plus performants ayant un coût trop élevé, nous avons fait le pari de changer à moindre coût les équipements actuels pour des serveurs Linux classiques, fonctionnant sous Debian.

Pari réussi : les performances sont convaincantes, nous saturons sans problème le lien Gigabit et depuis, avons même basculé notre accès internet à 10 Gbit/s avec succès. De plus, l'utilisation d'un système Linux *simplifie la configuration de l'équipement : gestion via l'outil d'orchestration* Puppet*, mise en place d'un* proxy cache*, développement de nouveaux services réseaux pour les chercheurs …*

Nous vous proposons donc un retour d'expérience sur cette migration vers des serveurs Linux*. Nous présenterons la configuration matérielle et logicielle utilisée, comment sont effectuées les fonctions de routage et de filtrage, les services mis en place qui n'auraient pas pu l'être sur des équipements traditionnels, ainsi que les performances obtenues.*

Mots-clefs

Réseau, *Routage*, Linux, FRRouting, Quagga, OSPF, BGP, DMZ, *Plate-forme expérimentale*

1 Introduction

Grid'5000 est une infrastructure pour la recherche en informatique distribuée (*Clouds*, réseau, *HPC*, *etc*.). La plateforme permet aux chercheurs de réaliser des expériences en leur mettant à disposition plus de 700 serveurs entièrement reconfigurables, ainsi que de nombreux outils logiciels. La plateforme est répartie sur 8 sites en France et au Luxembourg, reliés entre eux par un réseau 10 Gbits/s fourni par *Renater*. L'accès à internet depuis les nœuds de *Grid'5000* s'effectue via 2 routeurs à Lille et Sophia-Antipolis.

Dans cet article, nous proposons un retour d'expérience sur la migration de ces routeurs d'accès à *Grid'5000* vers des serveurs *Linux* classiques. Ces équipements sont essentiels au fonctionnement de *Grid'5000* : Ils doivent être suffisamment performants pour permettre aux utilisateurs de réaliser leurs expériences confortablement (transfert de volume de données important entre Internet et *Grid'5000*, téléchargement d'images de machines virtuelles, etc.). Ils jouent également un rôle important pour la sécurité, puisque ce sont les principaux points d'accès à la plateforme. Les premières réflexions en lien avec cette migration ont commencé en juillet 2017 et le projet a abouti fin 2018.

Après une introduction à l'architecture réseau de *Grid'5000*, les limitations de l'ancienne solution basée sur des équipements réseaux « classiques » sera détaillée. Les arguments en faveur d'une nouvelle solution basée sur des serveurs *Linux* seront expliqués. Les aspects techniques pour la mise en place de ces serveurs seront abordés avant de détailler les performances obtenues ainsi que les gains fonctionnels .

2 Le réseau de *Grid'5000*

2.1 Topologie générale

La Figure 1 montre le cœur de réseau *Grid'5000* dit « *backbone* », constitué de liens dédiés fournis par *Renater*. Il connecte les différents sites de *Grid'5000*. Chaque site utilise un routeur appelé « *gw.site* » qui est connecté au *backbone* et au « réseau de site » auquel appartiennent les différents serveurs utilisés pour les expériences, ainsi que des machines de service (déployant des services tels que *DNS*, *DHCP*…).

Pour l'accès à Internet, 2 routeurs appelés « *gw-north* » et « *gw-south* », situés respectivement physiquement à Lille et Sophia-Antipolis, interconnectent Internet, le réseau de *backbone* et le réseau « *DMZ* » qui héberge les services de *Grid'5000* accessibles depuis l'extérieur (site Web, machine d'accès *SSH* etc.).

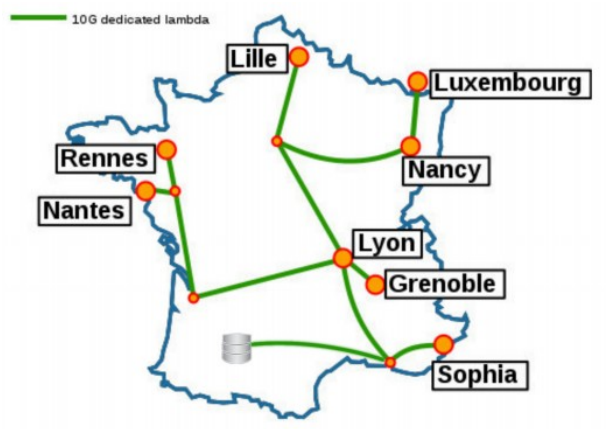

Figure 1: Cœur de réseau 10 Gbit/s de Grid'5000

La topologie de niveau 2 du réseau de *Grid'5000* peut être représenté comme sur le schéma de la figure 2.

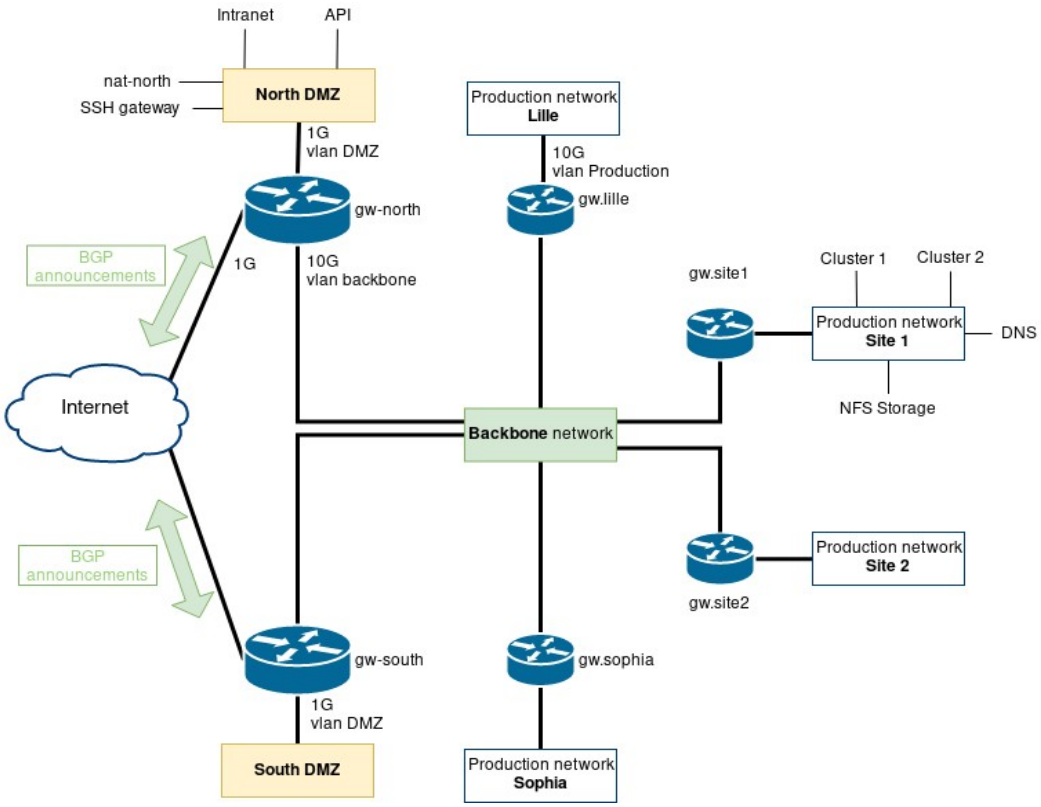

Figure 2: Les différents réseaux d'interconnexion de Grid'5000

Ainsi, la table de routage de gw.site1 contient, entre autres, les routes suivantes:

```
<Réseau Site 1>        directly connected
<Réseau Backbone>      directly connected
<Réseau Site 2>        via gw.site2
<South DMZ>            via gw-south
<North DMZ>            via gw-north
<Internet>             via gw-north
```

2.2 La *DMZ* et l'accès à Internet

Grid'5000 dispose d'une plage de 62 adresses IPv4 publiques. Cette plage est essentiellement utilisée par les services de la *DMZ*. Le protocole *BGP* est utilisé pour annoncer la plage d'adresses de la DMZ au point de présence *Renater*. Pour des raisons de sécurité, il est nécessaire de filtrer le trafic entre Internet et *Grid'5000* et en particulier de restreindre au maximum les communications entrantes et sortantes vers la *DMZ*.

À l'intérieur de *Grid'5000*, les machines utilisent un adressage privé : un nœud peut communiquer avec d'autres nœuds du même site ou d'un autre site *Grid'5000* en utilisant ces adresses privées. En revanche, pour sortir sur internet, les mécanismes classiques de traduction d'adresse (*NAT*) sont utilisés pour permettre la communication avec internet.

3 Solution précédemment en place

3.1 Routeurs *gw-north* et *gw-south*

Avant la migration, sujet de cet article, des équipements Cisco 2911 étaient utilisés pour les routeurs d'accès « *gw-north* » et « *gw-south* ».

Ces équipements réalisaient les annonces de la plage d'IP publiques par *BGP*, ainsi que le filtrage à l'aide d'un ensemble de règles de contrôle d'accès (*ACL*) sans états.

3.2 Machines virtuelles pour le *NAT*

Comme vu précédemment, un *NAT* est nécessaire pour permettre aux nœuds de *Grid'5000* d'accéder à Internet. De plus, pour des raisons légales, l'historique de ces accès doit être conservé durant 1 an. Ces fonctionnalités étaient réalisées par des machines virtuelles, « *nat-north* » et « *nat-south* », placées dans la *DMZ*.

Comme illustré dans la Figure 3 pour la partie nord, le trafic à destination d'Internet doit passer une première fois par *gw-north*, qui redirige le trafic vers *nat-north* qui réalise la

traduction d'adresse *NAT*, enregistre la connexion dans le système de stockage des traces et envoie de nouveau le trafic vers *gw*-north, qui finalement route le paquet vers Internet.

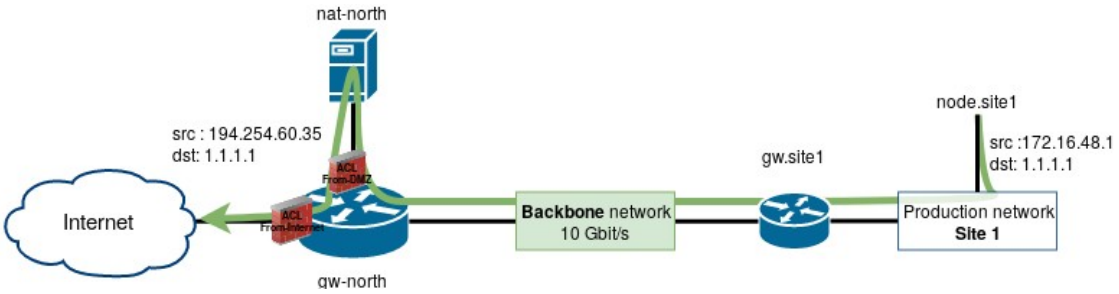

Figure 3: Accès à Internet via le NAT de la précédente solution

La redirection vers la machine de *NAT* ne doit être réalisée que lorsque nécessaire, c'est-à-dire pour les paquets à destination d'Internet et originaires du réseau *Grid'5000*. Ceci est permis par une fonctionnalité de « *source routing* » disponible sur les routeurs d'accès. Voici l'extrait de la configuration de *gw-north* qui permet d'activer cette fonction :

```
ip access-list standard From-G5K
 permit 172.16.0.0 0.15.255.255
 permit 10.0.0.0 0.255.255.255
 permit 131.254.202.0 0.0.1.255
route-map NAT-From-G5K permit 10
 match ip address From-G5K
 set ip default next-hop 192.168.66.35
```

La route par défaut de *gw-north* est celle donnée par son voisin *BGP* pour accéder à Internet. La configuration ci-dessus permet, pour le trafic provenant d'une IP privée *Grid'5000*, de changer ce « *default next-hop* » vers *nat-north* (192.168.66.35).

3.3 Inconvénients

3.3.1 Une administration manuelle

L'ensemble des configurations des serveurs *Grid'5000* sont versionnées avec *Git* et déployées avec *Puppet.* Ce n'est pas le cas des équipements réseaux, pour lesquels les configurations sont appliquées directement par l'administrateur. Cette situation augmente les risques d'erreurs : par exemple, pour ajouter une règle de filtrage, il était nécessaire de se connecter aux deux routeurs *gw-north* et *gw-south* pour y appliquer deux fois la même modification.

De plus, le suivi des changements des configurations des équipements était très rudimentaire et les configurations sauvegardées ne correspondaient pas toujours à la configuration réellement en place sur les routeurs, car cette opération de synchronisation devait être effectuée manuellement et n'était pas faite systématiquement.

3.3.2 Un routage complexe

Comme nous l'avons vu dans la figure 3, le trafic sortant emprunte le chemin suivant :

serveur.site1 → *gw.site1* → *gw-north* → *nat-north* → *gw-north* → *Internet*

Ce « double passage » par *gw-north* et la nécessité de prendre en compte l'IP source pour faire le routage complexifie l'architecture réseau et a l'inconvénient de faire passer tout le trafic sortant par une machine virtuelle, ce qui est susceptible d'avoir un impact sur les performances.

3.3.3 Problème de performance

Le principal problème rencontré avec cette solution étaient ses faibles performances. En effet, le débit réseau maximum depuis Internet vers *Grid'5000* était limité à 150 Mbits/s , alors que le lien fournit par *Renater* possédait une capacité de 1 Gbit/s.

Ce faible débit devenait problématique pour les chercheurs qui manipulent des volumes de données de plus en plus importants. Une enquête réalisée auprès des 15 plus gros utilisateurs de la plateforme a montré qu'un meilleur accès à Internet était une des principales évolutions souhaitées.

Une mesure des performances des différentes parties du réseau *Grid'5000* a permis d'identifier le goulot d'étranglement causant ces faibles performances : le problème venait des routeurs d'accès *Cisco 2911*, *gw-north* et *gw-south.*

Avant d'envisager un changement d'équipement, différents tests ont été réalisés pour identifier la raison des faibles performances obtenus sur ces équipements : Il a été constaté que le CPU des routeurs était très sollicité et que sa charge était liée au nombre d'*ACL*.

Voici les résultats obtenus entre un serveur de la *DMZ* nord à Lille et un nœud du site de Rennes avec le logiciel *Iperf* :

	Débit obtenu avec *Iperf*	Utilisation CPU sur *gw-north*
200 règles *ACL* configurées (configuration utilisée dans *Grid'5000*)	120 Mbits/s	95 %
50 règles *ACL* configurées	271 Mbits/s	50 %

Bien qu'une réduction du nombre d'*ACL* permet d'alléger la charge du CPU et d'obtenir des débits plus important, ces derniers restent insuffisants.

Ces routeurs étaient donc incapables de répondre aux nouveaux besoins de la plateforme. En juillet 2017, il est décidé d'étudier une nouvelle solution pour remplacer ces équipements.

4 Choix d'une nouvelle solution

4.1 Critères de choix

Les nouveaux équipements devaient supporter les protocoles et fonctionnalités nécessaires à l'interconnexion de *Grid'5000* et Internet présentés précédemment (*BGP*, *NAT*, filtrage et enregistrement des connexions…).

Ce sont des fonctionnalités assez classiques que la plupart des équipements réseaux actuels supportent. Le rapport coût/performance de l'équipement, les exigences de traçabilité, mais surtout la simplicité d'administration furent donc les critères déterminants pour le choix de la nouvelle solution.

4.2 Les équipements réseaux dédiés

Une solution naturelle aurait consisté à remplacer nos équipements réseaux par des routeurs offrant les performances attendues. Cependant, le coût pressenti de ces équipements était élevé. En particulier, la mise à disposition par Renater de liens 10 Gbit/s pour l'accès à Internet était envisagée et le montant des devis qui ont été établis pour des routeurs garantissant d'atteindre un tel débit était de l'ordre de 30 000 € par équipement.

De plus, comme évoqué plus haut, ces équipements sont limités dans leur utilisation : leur administration est plus complexe qu'un serveur *Linux* classique et est spécifique au constructeur de l'équipement. Les fonctionnalités d'enregistrement des connexions sont moins avancées que ce qu'il est possible de réaliser avec *Iptables* et son intégration à notre système de stockage de traces est plus difficile. Il est également impossible d'y déployer un nouveau service qui ne serait pas déjà disponible dans l'équipement. Des fonctionnalités telles que la mise en cache ou la modification dynamique de *VLANs*, décrites dans la suite du document, ne pourraient pas être mises en place directement sur ces équipements.

4.3 Les serveurs Linux généralistes

L'utilisation de serveurs *Linux* généralistes a été envisagée comme alternative aux équipements réseaux classiques. Les systèmes *Linux* disposent de toutes les fonctionnalités nécessaires pour l'interconnexion de *Grid'5000* à Internet en ce qui concerne le routage et le filtrage. De plus, cette solution laisse entrevoir une intégration complète dans notre infrastructure : système d'exploitation identique à celui utilisé pour nos serveurs, gestion des configurations par *Puppet*, intégration complète au système de supervision et de gestion des traces utilisé dans *Grid'5000*, déploiement de services avancés, etc. Enfin, le coût du matériel pour la mise en œuvre de tels systèmes est bien plus faible que celui des équipements réseaux qui ont été envisagés.

En revanche les performances qu'il était possible d'obtenir avec cette solution étaient inconnues. Différentes études[1][2][3] semblent indiquer que des débits de 10 Gbit/s sont atteignables, mais ces études sont parfois anciennes, nécessitent la mise en œuvre de matériel ou de système d'exploitation spécifiques ou de paramétrages avancés du noyau. Nous n'étions pas assurés de disposer d'une solution répondant à la fois aux critères de performance et de facilité d'intégration.

Afin de vérifier la capacité d'un serveur classique à traiter et filtrer des paquets à 10 Gbit/s, nous avons utilisé les ressources de *Grid'5000* pour réaliser un test simple, impliquant 3 nœuds du site de Nancy. La configuration de cette expérimentation est illustrée sur la figure 4.

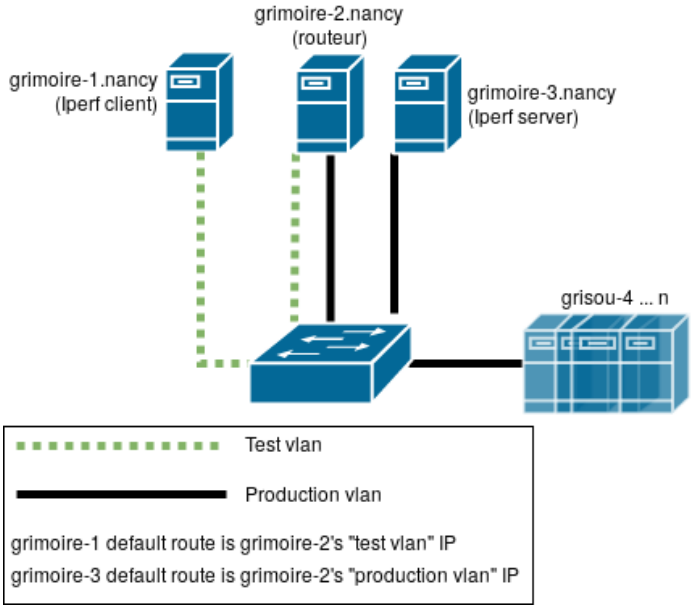

Figure 4: Evaluation des performances de la solution à base de serveur Linux Grid'5000 à l'aide de machines Grid'5000

Un des nœuds est utilisé comme routeur entre les 2 autres nœuds. Il utilise un système Debian 9, comme le reste de nos serveurs utilisés dans l'infrastructure de *Grid'5000*. De manière similaire à ce qui est utilisé dans les équipements *gw-north* et *gw-south,* 20 règles de routage et 200 règles de filtrage ont été mises en places manuellement. L'enregistrement des connexions qui transitent par la machine est également mis en place, avec I*ptables*.

Les mesures de performance, réalisées avec Iperf configuré pour initier plusieurs connexions en parallèle, indiquent qu'un débit à 10 Gbit/s est atteint de manière stable entre les 2 nœuds connectés au nœud routeur. On constate un impact sensible du nombre de règles de filtrage sur la charge CPU, sans que cela n'entraîne une diminution du débit.

Bien que ces tests ne soient pas complètement représentatifs des conditions réelles du travail effectué par les passerelles d'accès (absence de *BGP*/*OSPF*, tests au sein d'un même cluster sur le même site, matériel différent), ils nous permettent de confirmer que le routage à 10 Gbit/s est atteignable sans difficultés majeures, avec du matériel et un système *Linux* généralistes.

La solution utilisant des serveurs *Linux* généralistes a donc été retenue pour remplacer les passerelles d'accès de *Grid'5000* à Internet.

5 Mise en œuvre

Nous allons maintenant décrire la mise en œuvre du replacement de nos équipements réseaux par des routeurs basés sur des serveurs *Linux* généralistes.

5.1 Configuration matérielle

2 serveurs *Dell PowerEdge R640*, appelés « *gwol-north* » (***G**ate**w**ay **O**n **L**inux **N**orth*) et « *gwol-south* », ont été achetés pour un montant unitaire inférieur à 3 000 €. Leur configuration matérielle est la suivante :

- 1 CPU Intel Xeon Gold 5122 3.6Ghz, 4 Cœurs, 8 Threads
- 1 Carte réseau Intel 2P X710/2P I350 + 1 Intel X710
  - Total de 4 Ports 10G SFP+
  - Total de 2 Ports 1G RJ45
- 32Go DDR4
- 2 x SSD 450Go 6Gbits/s

5.2 Architecture réseau

Avec cette configuration, une nouvelle architecture réseau, très similaire à l'ancienne, a été mise en place :

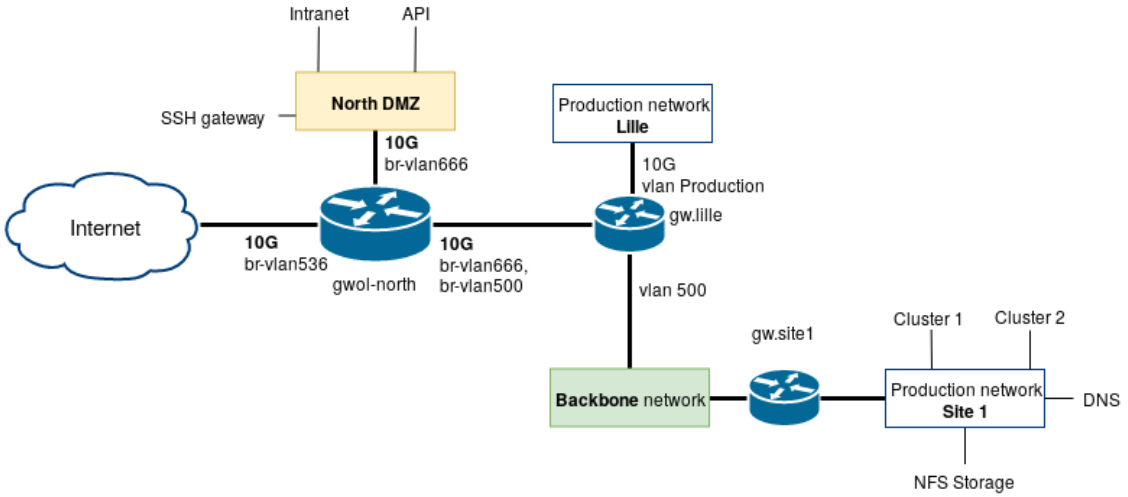

Figure 5: Nouvelle architecture réseau avec les routeurs Linux

Comme on peut le voir sur la figure 5, *gwol-north* remplace l'équipement qui relie la *DMZ*, le réseau de *backbone* et Internet. De plus, la machine virtuelle réalisant le *NAT* n'est plus nécessaire, cette opération pouvant être réalisée directement sur le serveur *gwol-north*. Le lien vers Internet est d'abord resté en 1 Gbit/s, il a été remplacé par un lien 10 Gbit/s dans un second temps, après l'installation et les premiers tests de cette nouvelle solution.

Cette partie de l'article va décrire le serveur *gwol-north* : les logiciels utilisés et leurs configurations, la mise en place des *VLANs*, du filtrage et des autres services. Le serveur *gwol-south* est configuré de manière identique.

5.3 Configuration des interfaces réseaux

Trois interfaces réseaux sont utilisées et reliées aux réseaux Internet, *DMZ* et *backbone*. Les interfaces sont placées dans les *VLANs* correspondants à leur réseau et dans des « *bridges* » de la manière suivante :

```
g5kadmin@gwol-north.grid5000.fr(physical):~$ sudo brctl show
bridge name     bridge id               STP enabled     interfaces
br-vlan500              8000.246e96dd1f56       no              eno1.500
br-vlan536              8000.246e96dd1f58       no              eno2.536
br-vlan666              8000.246e96dd1f56       no              eno1.666
                                                                enp179s0f1.666
```

Les différentes adresses IP nécessaires à l'interconnexion de niveau 3 sont placées sur les *bridges* :

```
g5kadmin@gwol-north.grid5000.fr(physical):~$ ip address show type bridge
9: br-vlan666: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 24:6e:96:dd:1f:56 brd ff:ff:ff:ff:ff:ff
    inet 192.168.66.253/24 brd 192.168.66.255 scope global br-vlan666
       valid_lft forever preferred_lft forever
    inet 194.254.60.35/26 brd 194.254.60.63 scope global br-vlan666:1
       valid_lft forever preferred_lft forever
13: br-vlan500: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 24:6e:96:dd:1f:56 brd ff:ff:ff:ff:ff:ff
    inet 192.168.4.253/24 brd 192.168.4.255 scope global br-vlan500
       valid_lft forever preferred_lft forever
15: br-vlan536: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 24:6e:96:dd:1f:58 brd ff:ff:ff:ff:ff:ff
    inet 193.51.183.177/30 brd 193.51.183.179 scope global br-vlan536
       valid_lft forever preferred_lft forever
```

5.4 Routage dynamique avec *FRR*

FRRouting (*FRR*) est un logiciel implémentant un ensemble de protocoles de routage dynamiques pour *Linux*, en particulier *BGP* et *OSPF* (ce dernier est utilisé pour le routage interne à *Grid'5000*). C'est le successeur de *Quagga*, que nous utilisions déjà dans *Grid'5000*.

La configuration de *FRR* est contenue dans un fichier de configuration dont la syntaxe est similaire à celle des équipements Cisco. Il fournit également une interface en ligne de commande similaire à celle proposée par les équipements de cette marque. Par exemple, l'extrait montre la sortie de la commande *show ip ospf neighbors* :

```
gwol-north# show ip ospf neighbor

Neighbor ID     Pri State           Dead Time Address         Interface            RXmtL RqstL DBsmL
1.0.0.254         1 Full/DR           37.369s 192.168.66.36   br-vlan666:192.168.66.253     0     0     0
1.1.1.12          1 Full/DROther      35.773s 192.168.4.12    br-vlan500:192.168.4.253      0     0     0
1.1.1.13          0 Full/DROther      35.318s 192.168.4.13    br-vlan500:192.168.4.253      0     0     0
1.1.1.14          1 Full/DROther      37.443s 192.168.4.14    br-vlan500:192.168.4.253      0     0     0
1.1.1.15          0 Full/DROther      35.301s 192.168.4.15    br-vlan500:192.168.4.253      0     0     0
1.1.1.18          0 Full/DROther      34.399s 192.168.4.18    br-vlan500:192.168.4.253      0     0     0
1.1.1.19          0 Full/DROther      39.332s 192.168.4.19    br-vlan500:192.168.4.253      0     0     0
1.1.1.21          1 Full/DROther      37.323s 192.168.4.21    br-vlan500:192.168.4.253      0     0     0
1.1.1.22          0 Full/DROther      33.964s 192.168.4.22    br-vlan500:192.168.4.253      0     0     0
1.1.1.60          0 Full/DROther      32.450s 192.168.4.60    br-vlan500:192.168.4.253      0     0     0
255.255.255.255   4 Full/Backup       33.227s 192.168.4.254   br-vlan500:192.168.4.253      0     0     0
```

Puisque le serveur est pleinement intégré au gestionnaire de configuration *Puppet* utilisé dans *Grid'5000*, la configuration de *FRR* est déployée par ce dernier. Il est de plus possible d'utiliser la fonctionnalité de « *template* » *Puppet*, afin que le contenu du fichier de configuration soit généré dynamiquement, en fonction de variables fournies par *Puppet* et extraites de la base de données *Hiera*. Cette base contient les informations communes à l'ensemble de l'infrastructure *Grid'5000*. Voici un extrait du fichier de configuration de *FRR*, les variables étant délimitées par les balises *<%= ... %>*

```
router bgp <%= bgp['local_as'] %>
 bgp log-neighbor-changes
 network <%= bgp['primary'] %> mask <%= bgp['primary_mask']%> route-map Primary-BGP
 network <%= bgp['secondary'] %> mask <%= bgp['secondary_mask'] %> route-map Secondary-BGP
 neighbor <%= bgp['neighbor'] %> remote-as <%= bgp['remote_as'] %>
 neighbor <%= bgp['neighbor'] %> password <%= bgp['pass'] %>
 neighbor <%= bgp['neighbor'] %> send-community
!
```

5.5 Filtrage, *NAT* et enregistrement des connexions avec *Iptables*

La mise en place du filtrage des connexions dans le serveur *Linux* est significativement plus simple qu'elle ne l'était pour les équipements réseaux précédemment utilisés. En effet, les règles de filtrage sont contenues dans un fichier de configuration *Iptables* et, comme pour *FRR*, ce dernier est mis en place par *Puppet* dynamiquement. Ceci amène les mêmes bénéfices : utilisation d'une base d'information *Hiera* commune à toute la plateforme, meilleure lisibilité des règles de filtrage, etc. La fonctionnalité de *NAT* est également mis en œuvre, très simplement, à l'aide d'une règle *Iptables*.

L'enregistrement des connexions est lui aussi réalisé par une règle *Iptables* : à chaque établissement d'une nouvelle connexion, les informations sur l'origine et la destination du trafic sont dirigées vers le système *rsyslog* déployé sur la machine, qui à son tour transfère cette information au système de gestion des traces utilisé par l'infrastructure *Grid'5000*.

5.6 Services avancés

L'utilisation d'un système *Linux* permet la mise en place de services évolués directement sur le routeur et qui n'auraient pas pu être déployés sur un équipement réseau traditionnel.

Par exemple, un système de proxy transparent a été mis en place. Celui-ci, à l'aide du logiciel *Squid*, permet de conserver en cache les fichiers les plus volumineux téléchargés par les machines de *Grid'5000* et de les servir directement aux machines souhaitant télécharger un fichier déjà en cache. Le proxy est transparent, c'est-à-dire qu'il ne nécessite aucune configuration de la machine cliente : une règle *Iptables* appliquée sur le routeur se charge de rediriger vers le proxy *Squid* le trafic concerné.

De plus, un service de connexion à la demande entre des machines *Grid'5000* et des plateformes extérieures, via le réseau *GEANT* et le *Software Defined eXchange* de la fédération européenne *Fed4FIRE*[4], a été mis en place. Celui-ci a été développé par *Grid'5000* à l'aide du framework *Ruby On Rails* et est déployé sur les machines d'accès. Il permet aux utilisateurs de *Grid'5000* d'interconnecter leurs machines aux plateformes extérieures en manipulant une *API* REST. Le service réalise alors dynamiquement les changements de *VLANs* sur le routeur d'accès, de manière à interconnecter les différentes plateformes pour la durée d'une expérience.

6 Bilan de la migration

6.1 Un gain de performances

Les performances des nouveaux routeurs d'accès donnent pleinement satisfaction. Les liens d'accès à Internet ont été mis à jour à 10 Gbit/s par *Renater* peu de temps après la migration vers les systèmes *Linux* et ces débits sont pratiquement atteints (le trafic transitant par *gwol-south* atteint quasiment 10 Gbit/s, celui transitant par *gwol-north* est de 8 Gbits/s, pour une raison encore inconnue et qui n'est probablement pas liée au routeur lui-même). Ceci est illustré dans les figures 6 et 7, qui montrent que la charge sur le serveur est très faible et que ce dernier ne constitue pas un goulot d'étranglement pour les performances.

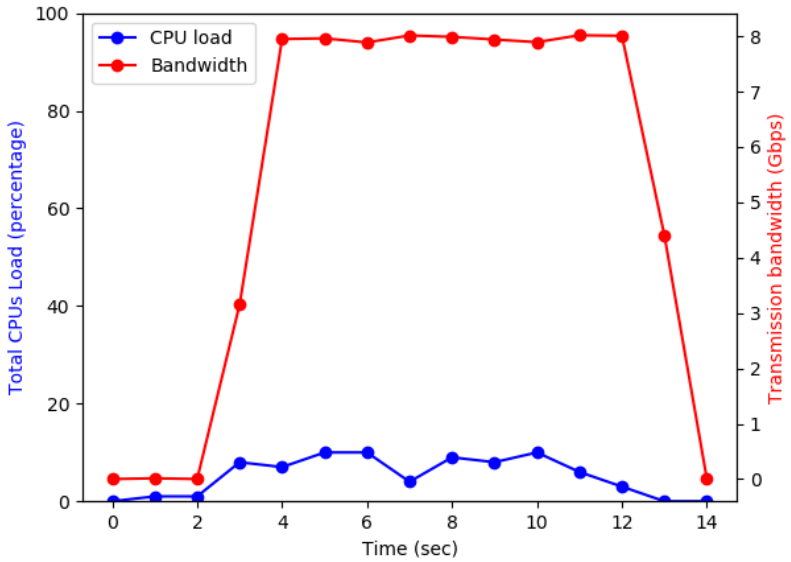

Figure 6: Charge moyenne de l'ensemble des CPUs de gwol-north lorsque le débit réseau maximal est atteint (20 clients générant des connexions TCP vers 4 serveurs différents).

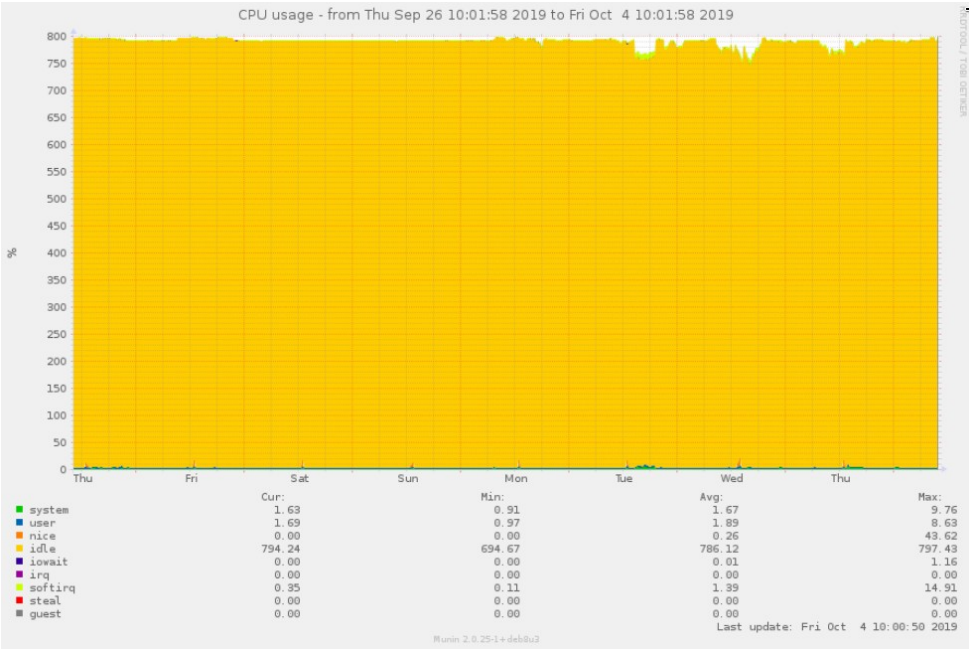

Figure 7: Charge CPU moyenne sur gwol-north durant une semaine, reportée par le logiciel de supervision Munin : la charge du serveur n'est pas significative.

6.2 De nouvelles possibilités

Les services « avancés » qui ont pu être mis en place sur les routeurs *Linux* n'auraient pas pu l'être sur des équipements réseau traditionnels. Disposer d'un système *Linux* complet laisse la possibilité de mettre en place des services spécialisés et répondant aux besoins spécifiques d'une plateforme de recherche telle que *Grid'5000*.

6.3 Une administration simplifiée

Il est bien plus facile pour l'équipe technique *Grid'5000*, composée essentiellement d'ingénieurs systèmes qui ne sont pas spécialistes réseau, d'administrer la nouvelle solution.

Celle-ci est pleinement intégrée au système de monitoring de *Grid'5000* et l'ensemble de ses configurations est géré par *Puppet*, de manière identique aux autres serveurs utilisés pour le fonctionnement de la plateforme.

Ceci permet de stocker et manière centralisée et versionnée les configurations des équipements, ce qui facilite la traçabilité par l'ensemble de l'équipe des modifications effectuées.

De plus, *Puppet* apporte beaucoup de souplesse pour la modification des configurations des équipements, celle-ci s'opérant depuis le dépôt centralisé. Par exemple, pour rediriger l'ensemble du trafic sortant de *Grid'5000* et le faire transiter vers *gwol-south* (au lien de *gwol*-north, qui est utilisé par défaut), une seule ligne de configuration doit

être modifiée dans le dépôt *Puppet*. Avec l'ancienne solution, cette opération nécessitait d'intervenir directement sur les deux routeurs d'accès et d'y appliquer plusieurs modifications.

7 Conclusion

Nous avons présenté dans cet article notre retour d'expérience sur le remplacement de routeurs réseau traditionnels par des serveurs *Linux* généralistes. Nous avons mis en évidence les différents avantages de cette transition : meilleures performances, flexibilité permise par l'utilisation d'un système *Linux*, administration simplifiée grâce à l'intégration à l'infrastructure existante et coûts réduits. Cette solution nous semble particulièrement adaptée pour des infrastructures déployant déjà de nombreux serveurs *Linux* et dont les besoins en connectivité réseau sont modestes.

Les évolutions envisagées, en lien avec ce projet sont notamment liées au déploiement d'IPv6 dans *Grid'5000*. Par ailleurs, il est prévu de déployer un service de pare-feu dynamique, que les utilisateurs puissent manipuler pour ouvrir des accès depuis Internet vers les nœuds de leurs expériences.

Une autre perspective en lien avec ce sujet serait l'utilisation d'équipements de type « *white box switch* » [5] qui mettent à disposition du matériel spécifique réseau muni d'un système d'exploitation *Linux* classique. Cela pourrait apporter les bénéfices liés à l'utilisation des systèmes *Linux* aux équipements de commutation utilisés pour les nœuds *Grid'5000* et dont les performances (débits de 10 Gbit/s à 100 Gbit/s sur des dizaines de ports) sont encore probablement hors de portée d'un serveur généraliste.

Bibliographie

[1] O Hagsand, R.Olsson., B. Gorden. Towards 10Gb/s open source routing. In *Proceedings of the Linux Symposium, Hambur.* 2008

[2] V. Bernat, Performance progression of IPv4 route lookup on Linux, Available online : https://vincent.bernat.ch/en/blog/2017-performance-progression-ipv4-route-lookup-linux

[3] C. Cui, C. Chiu, and L. Xue. Linux Based Router Over 10GE LAN. *Project report from Louisiana State University.* 2008

[4] Fed4FIRE. https://www.fed4fire.eu

[5] R. Jain. OpenFlow, Software Defined Networking (SDN) and Network Function Virtualization (NFV). In : *A half-day tutorial at the IEEE International Conference on Communications (ICC 2014), Sydney, Australia*. 2014.