Plate-forme d'hébergement de services tolérante à la panne, redondée géographiquement

Yann Dupont
DSI de l'Université de Nantes
2, rue de la Houssinière
44322 Nantes Cedex 3

Yoann Juet
DSI de l'Université de Nantes
2, rue de la Houssinière
44322 Nantes Cedex 3

Jean-Philippe Menil
DSI de l'Université de Nantes
2, rue de la Houssinière
44322 Nantes Cedex 3

Résumé

L'université de Nantes s'est dotée dès 2005 d'une infrastructure mono-site de services à haute disponibilité. Bien que stable, l'architecture logicielle souffrait de plusieurs insuffisances dont une incompatibilité IPv6. L'hébergement des nœuds de service sur plusieurs salles serveurs devait pallier aux dysfonctionnements majeurs que sont les coupures électriques, défaillances matérielles. Pourtant, des incidents récents comme une panne prolongée sur le TGBT du campus, ont souligné les limites de la solution. L'aspect mono-site était un facteur aggravant, de surcroît incompatible avec un futur Plan de Continuité d'Activité.

Ces insuffisances nous ont naturellement amené à repenser l'architecture initiale. Multi-site, la nouvelle infrastructure se retrouve aujourd'hui répartie sur trois plaques géographiques hébergeant chacune l'un des nœuds de routage de la dorsale. Les directeurs IPVS assurant la balance de charge du trafic applicatif utilisent désormais un mode routé asymétrique en double pile. L'ensemble des systèmes est virtualisé sous Linux/KVM ou Linux/LXC de façon homogène sur les plaques. Les services exploitent des adresses virtuelles IPv4/IPv6 spécifiques, lesquelles sont annoncées sur tous les directeurs puis redistribuées par OSPF sur la dorsale. Cette implémentation de l'anycast nous garantit une excellente tolérance à la panne et optimise de surcroît les flux grâce à un routage au plus proche de l'utilisateur.

Mots clefs

Haute disponibilité, répartition de charge, IPv6, routage dynamique, anycast, GNU/Linux, KVM, LXC, csync2, quagga, VRRP, IPVS, ldirectord.

1 Introduction

La plate-forme d'hébergement de services à haute disponibilité présentée dans cet article est l'évolution d'une première version démarrée dès l'année 2005, en phase finale de démantèlement. Des changements fondamentaux aussi bien techniques, politiques que fonctionnels, intervenus tout au long de ces six dernières années, nous ont conduit à y apporter une révision conséquente. Cette nouvelle itération, fonctionnelle depuis décembre 2010, a vu les premières mises en production de services trois mois plus tard. Le présent article va décrire les développements réalisés durant la transition en version 2.

1.1 Une première version de la plate-forme de haute disponibilité

La plate-forme initiale s'appuyait sur des logiciels libres, pour des questions de coûts, de pérennité, de compétences et d'agilité. Au niveau logiciel, le choix s'était porté sur ultramonkey [ULTRAMONKEY], fondé sur trois briques : **LVS** et sa partie standard du noyau GNU/Linux, appelée IPVS, associant une adresse IP virtuelle ou VIP à un ensemble d'autres adresses dites réelles via des règles d'association ; **ldirectord**, un script vérifiant la disponibilité des machines hébergeant un service et altérant dynamiquement les règles IPVS de façon accordante. Ces deux briques fonctionnent de concert sur un serveur appelé directeur, balançant la charge réseau. Pour garantir la haute disponibilité de cette fonction critique, deux machines en mode actif/passif sont généralement déployées à l'aide d'une troisième brique : **Heartbeat ou Keepalived[1] [KEEPALIVED]**.

[1]Bien que ne faisant pas partie d'ultramonkey, keepalived est un remplaçant potentiel à heartbeat.

Un directeur transforme les paquets réseaux reçus avant de les distribuer vers des machines hébergeant réellement les applications, communément appelées nœuds du cluster ou nœuds de service. Trois modes opératoires sont possibles : NAT, tunnel ou routage direct. Le mode le plus simple étant *a priori* le NAT, c'est le choix qui avait été fait à l'époque.

Pré-datant de la DSI, la plate-forme initiale fut conçue par le CRI de l'Université de Nantes (1995-2008†) avec ses ressources de l'époque. En particulier l'implantation des machines n'avait pu se faire qu'au sein des salles contrôlées par le CRI. Eu égard au nombre important de serveurs à déployer, et étant déjà alors engagés dans une politique de virtualisation massive, nous avions choisi d'implanter les directeurs sur des machines virtuelles XEN [XEN] à cause de l'exigence de virtualisation de la couche réseau. Les nœuds de service utilisaient Vserver [VSERVER], pour des raisons d'efficacité et de densité de machines. La distribution GNU/Linux Debian a été retenue comme système d'exploitation unique sur nos serveurs.

1.2 Des insuffisances techniques et structurelles

La plate-forme initiale a fonctionné, sans anomalie majeure, avec une bonne fiabilité et des performances satisfaisantes, pendant six années. Ceci prouve qu'une partie des choix initiaux était bons. Des insuffisances sont néanmoins apparues au fil du temps :

- Les implantations géographiques, bien que semblant suffisamment sécurisées, étaient inadaptées. La trop forte proximité des deux salles d'hébergement les rendait vulnérables à un sinistre majeur tel qu'un incendie. De plus, elles dépendaient du même TGBT[2] qui alimente la moitié du campus, sans groupe électrogène disponible. Une panne, survenue récemment, a eu les conséquences redoutées. La coupure générale de plusieurs heures a excédé la capacité de nos onduleurs.

- Nos choix initiaux de simplicité d'architecture se sont révélés limitant. Initialement, IPVS ne supportait pas IPv6. Des évolutions récentes du noyau Linux l'ont permis, à l'exception du mode NAT. De plus, dans ce mode, les nœuds ne peuvent être adressés qu'avec un seul directeur – ils l'utilisent comme routeur – ce qui restreint les stratégies de haute disponibilité. Ensuite, la brique Heartbeat s'avère parfois capricieuse et non triviale à configurer. Quant aux technologies de virtualisation, XEN et VServer n'étaient pas intégrés dans le noyau standard. Les patchs qu'il fallait appliquer aboutissaient à un retard conséquent de version vis-à-vis de la branche stable des noyaux GNU/Linux. De surcroît, la non virtualisation de la couche réseau de ce dernier a abouti à une difficulté de ré-entrance dans le cluster. Ce n'est qu'au prix d'un NAT de sortie sur des machines dédiées et de tables de routages complexes sur les nœuds du cluster que la situation a pu être corrigée.

En somme, les choix initiaux n'ont finalement pas tenu leurs promesses en termes de simplicité. Ils sont devenus un frein à la généralisation de la solution.

1.3 Les objectifs affichés de la nouvelle plate-forme d'hébergement

La pertinence d'évoluer vers des balanceurs de charge commerciaux s'est posée puis a vite été éliminée, car, outre l'attachement fort de l'équipe aux solutions libres, il nous a paru possible de faire progresser la plate-forme rapidement, de façon satisfaisante, tant techniquement que financièrement. De façon évidente, il fallait conserver ce qui avait donné satisfaction et changer ce qui n'allait pas.

La première exigence était de maximiser la fiabilité des services rendus à l'utilisateur, ce qui passait par une redistribution géographique des machines. Le changement politique, conduisant du CRI à la DSI en 2008, a justement élargi les possibilités de zones d'accueil des serveurs. Prises individuellement, ces salles machines, si tant est qu'on puisseles qualifier ainsi, ne sont malheureusement pas du tout fiables électriquement et thermiquement parlant. En attendant d'hypothétiques améliorations sur ce point, la solution est d'admettre la survenue de pannes, tout en faisant en sorte qu'elles n'aient plus d'impact majeur. La difficulté est simplement contournée en multipliant les points de présence géographiques des directeurs et nœuds de service.

La seconde, et non la moindre, était d'avoir une solution aussi simple que possible à déployer et à dépanner. La précédente plate-forme, trop complexe et maîtrisée par peu de personnes, a montré ce qu'il fallait éviter. Les choix techniques sont guidés par cette volonté forte de garder une solution techniquement contrôlable. Vouloir tendre vers 100 % de disponibilité peut s'avérer contre productif. Un taux de disponibilité de 99.9 % représente, sur une année, une perte de service de 8 heures, 45 minutes et 36 secondes ; un taux de 99,99 %, un arrêt réduit à 52 minutes et 33 secondes, et 99,999 %, seulement 5 minutes et 15 secondes. Pour arriver aux taux les plus élevés, il est nécessaire d'échafauder des solutions d'un tel degré de complexité, qu'en cas de panne,

[2]Transformateur Général Basse Tension

il peut s'écouler des heures avant que le problème soit simplement compris, ruinant par là même, l'objectif initial. De façon pragmatique, l'Université de Nantes n'a pas besoin d'une disponibilité à 99,99 %. Même si elle héberge des applications sensibles, critiques à différents moments de l'année, elle n'a pas pour autant les contraintes de disponibilité d'un grand centre de données ou d'un établissement hospitalier où des vies sont potentiellement en jeu. Il y a davantage à gagner en tirant parti de l'automatisation de différentes tâches d'exploitation, en améliorant les outils, qu'à atteindre une hypothétique seconde ou troisième décimale de disponibilité.

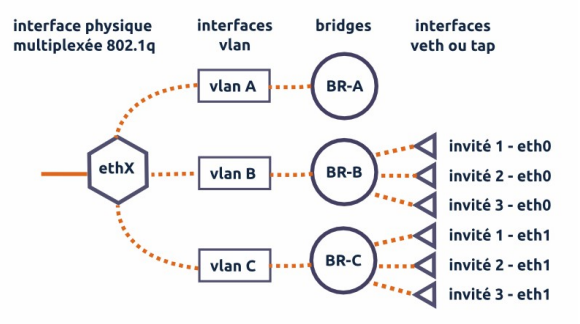
Enfin, une troisième exigence était de conserver une plate-forme évolutive, simple dans la mise à l'échelle des services. Les performances obtenues avec VServer devaient être maintenues, menant à une augmentation nulle ou limitée du nombre moyen des nœuds d'une ferme. Nous ne souhaitions pas que ce chiffre explose du fait des limitations imposées par la technologie de virtualisation.

1.4 Les choix techniques de la nouvelle plate-forme

Au fil des années, des solutions de virtualisation ont été intégrées au cœur même du noyau Linux (LXC[3], KVM[4]), menant à une simplification de la maintenance de notre ferme d'hébergement qui a ainsi progressivement basculé sur ces nouvelles technologies. L'architecture précédente utilisait, pour des raisons d'efficacité, deux types de virtualisation : XEN côté directeurs, VServer pour les nœuds de service. Ce principe de séparation, qui avait donné satisfaction, a naturellement été conservé. Pour autant, nous avons rafraîchi les technologies utilisées pour être en phase avec le reste de nos déploiements. KVM a de cette façon remplacé XEN pendant que LXC [LXC] supplantait VServer.

Le besoin de ré-entrance dans le cluster, source principale de complexité dans la précédente plate-forme, demandait une virtualisation complète de la couche réseau. VServer était de fait écarté des nœuds de service, malgré son appréciable efficacité. Nous souhaitions conserver une forte densité de machines virtuelles sous GNU/Linux, sans faire de compromis sur les performances. Là, KVM présentait quelques lacunes qui l'ont mis à l'écart. Malgré sa relative jeunesse LXC s'est très vite affirmé comme une bonne alternative à VServer. Basé sur le concept de conteneur, les invités LXC tournent sous le même noyau que leur hôte dans un environnement réseau entièrement virtualisé. La performance pure est favorisée, les ressources CPU, mémoire sont partagées de façon optimale. LXC ou plus généralement la technologie des conteneurs est un parfait prétendant à la densification de l'hébergement des nœuds de service.

Du point de vue opérationnel, les couches de virtualisation réseau proposées par KVM et LXC se gèrent de manière comparable. Seule la nature même des interfaces virtuelles change. Les directeurs disposent d'interfaces virtuelles du type "tap". Celles-ci sont connectées à des bridges créés sur l'hôte, s'appuyant eux-mêmes sur des interfaces vlan déclarées sur une interface physique 10Gbps reliant chaque lame au commutateur intégré du châssis blade. Le commutateur en question est lui-même relié à 10Gbps à son routeur de dorsale. Les nœuds de service supportent sur un autre type d'interfaces virtuelles, les 'veth' pour Virtual ETHernet avec un principe de rattachement réseau identique à celui des directeurs.



L'illustration ci-contre représente les relations entre l'interface physique d'un hôte (ethX avec multiplexage 802.1q), les interfaces vlan, les bridges et les interfaces virtuelles des invités (veth ou tap). Un invité, soit-il directeur ou simple nœud de service, gagne une connectivité dans un ou plusieurs réseaux (vlans) grâce à la mise en relation du bridge avec l'interface vlan souhaitée.

Illustration 1 : les interfaces physiques, logiques et virtuelles sur l'hôte

[3] **LinuX C**ontainers : Le noyau Linux intègre depuis mars 2009 une couche de virtualisation de type conteneur dont le principe est simplement d'isoler un groupe de processus utilisateurs dans un contexte d'exécution, et est donc limitée à Linux. LXC est le nom de la suite logicielle qui permet à l'utilisateur de s'en servir.

[4] **K**ernel-based **V**irtual **M**achine : Intégré depuis janvier 2007 au noyau Linux standard, celui-ci est utilisé comme hyperviseur de type 2, en exploitant les instructions de virtualisation matérielles AMD-V et Intel-VT des processeurs récents. KVM virtualise complètement et efficacement une machine Windows, BSD, Linux…

Des pare-feu sont dressés en différents endroits des couches réseaux évoquées plus haut. L'interface logique d'administration hors de la bande de l'hôte est naturellement protégée par tout un ensemble de règles netfilter/iptables. En amont, une grappe de pare-feu protège ce réseau de management des hôtes vis-à-vis du reste de l'université. Le trafic entrant sortant des invités est tout autant contrôlé par un ensemble de règles spécifiques, appliquées cette fois-ci sur les interfaces virtuelles 'tap' et 'veth'. L'infrastructure est entièrement protégée par des pare-feu exactement comme si elle était à l'intérieur d'une zone démilitarisée.

2 Le routage anycast, pierre angulaire de la plate-forme

2.1 Topologie et spécificités

L'Université de Nantes a la chance d'adosser sa dorsale sur tout un réseau de fibres noires, reliant en 1Gbps ou 10Gbps l'ensemble de ses sites. Trois châssis de commutateur/routeur assurent le routage inter-site sur la métropole nantaise. Même si le trafic n'est pas uniformément réparti entre les trois plaques, chacune d'elle héberge une densité de composantes, d'étudiants et personnels importante.

Les lieux d'implantation des serveurs étaient dès lors tout trouvés. Nous avons disposé sur ces trois plaques des fermes d'hébergement virtualisé sur la base de châssis blade, afin d'accueillir, entre autres, directeurs et nœuds de service. Alors que l'ancienne plate-forme se satisfaisait de deux directeurs en actif/passif, la nouvelle demande pas moins de trois directeurs actifs, situation que ne sait pas vraiment gérer heartbeat. D'autre part, il est souhaitable que les nœuds, quel que soit leur emplacement, puissent être adressés indifféremment depuis chaque directeur. La première conséquence est de changer le mode opératoire d'IPVS. C'est, de toute façon, indispensable au support du protocole IPv6. La seconde est de remplacer la brique heartbeat par un autre mécanisme. La solution adoptée fut de faire fonctionner IPVS en mode routage direct, tout en assurant la haute disponibilité des directeurs via un mécanisme réseau particulier, l'adressage anycast [RFC4786]. Il convient de noter que nous n'avons pas recherché l'iso-fonctionnalité avec heartbeat : la défaillance d'un directeur se traduira inévitablement par une brève coupure de service de quelques secondes. Conformément au principe de simplicité énoncé précédemment, nous avons considéré ce comportement comme acceptable. Les efforts nécessaires pour conserver une cohérence de toutes les sessions réseaux ouvertes par l'ensemble des directeurs sont bien trop importants face au faible gain que cela représenterait. En revanche, nous bénéficions d'une redondance géographique bien utile face aux déficiences de nos principales salles d'hébergement.

2.2 Présentation générale de l'anycast

Anycast n'est pas un énième protocole de routage dynamique, plutôt une méthode combinant adressage virtuel et routage dynamique, favorisant le déploiement de services à haute disponibilité. La particularité de l'anycast par rapport aux solutions traditionnelles est sa capacité à fournir des services au plus proche de l'utilisateur, autrement dit, avec le maximum d'efficacité sur le coût du routage. Pour y parvenir, des associations dites « 1 vers N » sont opérées. À une VIP, fut-elle IPv4 ou IPv6[5], sont associés un ou plusieurs serveurs fournisseurs de service, également qualifiés de nœuds de service. Ceux-ci doivent être positionnés en des lieux stratégiques, idéalement au plus proche des utilisateurs finaux. Contrairement aux modes de diffusion broadcast et multicast, un seul nœud se trouve sollicité en anycast ; les raisons tiennent à la fois au routage et au processus de sélection réalisé en ces lieux par des balanceurs de charge. Ils assurent la correspondance entre les VIP et les nœuds de service, priorisant le plus souvent les nœuds locaux sur les distants. Leur rôle ne s'arrête pas là, ils annoncent les VIP sur le réseau de transit. Ainsi, une même VIP se retrouve annoncée en différents points du réseau avec un poids variant selon la priorité que l'on souhaite donner à une plaque par rapport à une autre. Un exemple concret d'implémentation de l'adressage anycast à l'échelle mondiale est donné avec les principaux DNS racine de l'internet. Ceux-ci sont répartis en différents points du globe pour garantir une forte résilience du service critique qu'est la résolution de noms.

[5] Une VIP est rattachée à un service applicatif comme la résolution DNS, la navigation web avec mandataire.

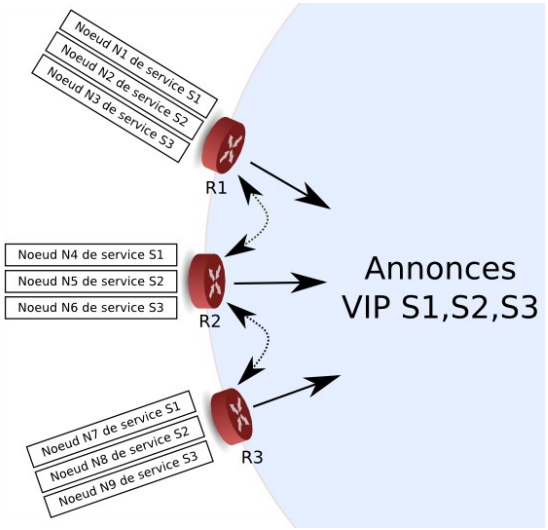


Illustration 2 : l'annonce des services en anycast

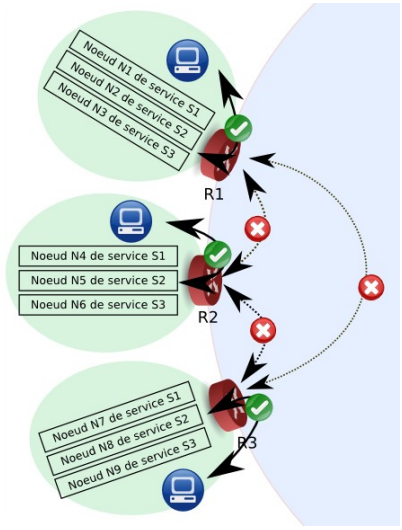


Illustration 3 : localité des échanges sur défaillances réseaux

Dans les illustrations 2 et 3 du fonctionnement de l'anycast, neuf machines ou nœuds, N1 à N9, se répartissent uniformément trois services, S1 à S3, sur trois plaques géographiques routées par les équipements R1 à R3. Les VIP associées à chacun des trois services VIP S1 à VIP S3, sont annoncées sur les routeurs R1 à R3. Ceux-ci les redistribuent ensuite dans l'arbre de routage OSPF, BGP ou autre du réseau de transit. L'isolation complète des plaques entre elles, autrement dit une perte simultanée des liens R1-R2, R2-R3 et R1-R3 n'engendrera pas la moindre perturbation côté utilisateur. Les services S1 à S3 demeurent joignables par les utilisateurs de la plaque, grâce aux nœuds de service locaux. Le schéma 3 illustre la situation extrême d'isolation inter-plaque.

2.3 Intégration du routage anycast à l'Université de Nantes

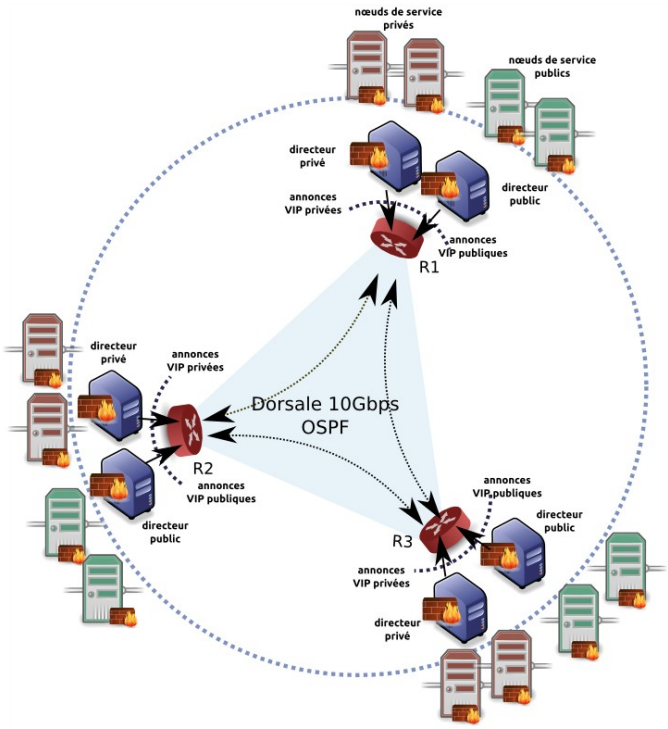


Illustration 4: implémentation de l'anycast à l'Université de Nantes

L'infrastructure système, comprenant les directeurs et les nœuds de service, est doublée par plaque, non pour des raisons de sûreté de fonctionnement, mais plutôt pour des questions de sécurité informatique. Une première infrastructure, exclusivement accessible aux réseaux filaires des composantes de l'Université et, sous conditions, au wi-fi, rend des services dits privés. Une seconde, dite publique, offre des services visibles de l'internet et/ou nécessitant une connectivité directe sur l'internet. Les services concernés sont, à titre d'exemple, les mandataires web, les mandataires inversés.

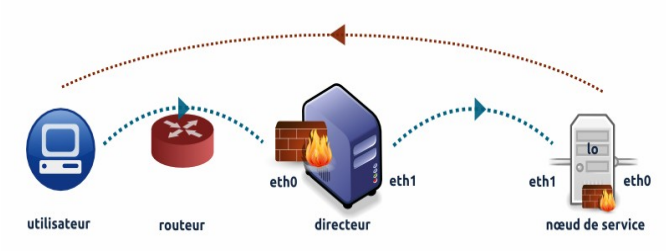
Chaque plaque se retrouve avec un directeur public communiquant avec des nœuds de service spécifiques. En parallèle, un directeur privé échange avec les nœuds de sa propre famille. Au total, ce ne sont pas moins de trois directeurs publics et autant de directeurs privés qui se trouvent uniformément répartis sur les trois plaques de service de l'Université de Nantes.

Deux infrastructures indépendantes dans l'absolu, interdépendantes dans les faits, cohabitent en parallèle : une publique et une privée, chacune ayant sa propre politique de sécurité, ses propres vlans et machines virtuelles. À l'isolation dite logique, nous ajoutons une isolation physique en hébergeant les nœuds publics et privés sur des lames serveurs distinctes.

3 Description de l'infrastructure opérationnelle

Comme nous l'avons évoqué plus haut, l'infrastructure générale de haute disponibilité associe trois éléments principaux :

- Les **routeurs** répartis sur les différentes plaques géographiques dont le rôle est de redistribuer sur la dorsale les routes associées aux services exposés. Il s'agit de routes en /32 pour les VIP4 en IPv4 ou en /128 pour les VIP6 en IPv6 ;
- Les **directeurs** balancent la charge de trafic vers les multiples nœuds de service. Ils sont également responsables des annonces VIP4 et VIP6 vers les routeurs de la dorsale ;
- Les **nœuds de service** hébergent les applications rendant le service attendu.



L'illustration ci-contre donne un premier aperçu de la direction et du sens des échanges entre les trois éléments d'infrastructure. Tous les éléments sont installés en double pile, autrement dit, leurs interfaces sont adressées à la fois en IPv4 et en IPv6.

Illustration 5 : le cheminement des requêtes et réponses au sein de la plate-forme

3.1 Les directeurs

Chaque directeur possède deux interfaces de communication en plus d'une loopback. La première interface, eth0, assure l'interconnexion entre les directeurs de plaque et leur routeur de dorsale. C'est par elle que les requêtes arrivent (sens utilisateur vers service) puis passent à l'algorithme de balance de charge. C'est aussi par son intermédiaire que les directeurs sont administrés et supervisés dans la bande. Eth1 est l'interface de sortie des directeurs par laquelle les nœuds de service sont contactés. Elle sert à la fois à relayer les requêtes de service des utilisateurs mais également à tester la présence des nœuds et des services hébergés. Enfin, une troisième interface, la loopback lo joue un rôle clé. Les VIP4 et VIP6 de services y sont toutes déclarées avant d'être annoncées dans l'arbre OSPF de la dorsale.

Le duo IPVS et ldirectord joue toujours le rôle de balanceur de charge sur les directeurs. Ceux-ci n'étant pas redondés par plaque mais dupliqués géographiquement, nous avons jugé l'intégration des briques heartbeat ou keepalived inutile. Il en résulte qu'en cas de défaillance d'un directeur, les sessions TCP applicatives des utilisateurs devront être rétablies à travers le directeur désigné par le routage dynamique. Le routage anycast est ainsi utilisé comme une sorte de substitut à heartbeat ou keepalived, obligeant les directeurs à annoncer leurs VIP sur la dorsale. Pour se faire, une autre brique logicielle libre est installée, Quagga [QUAGGA]. Quagga annonce respectivement en OSPFv2 (IPv4) et OSPFv3 (IPv6) les VIP4 et VIP6 de service vers le routeur de plaque qui les redistribue dans l'arbre OSPF de la dorsale. Une même VIP4 ou VIP6 se retrouve routée différemment suivant la localité de l'utilisateur. C'est ainsi que les sites universitaires rattachés à l'une des plaques se trouveront irrémédiablement routés vers le directeur local. À moins d'une défaillance matérielle ou logicielle de celui-ci, aucun trafic n'aboutira sur les autres directeurs.

IPVS est configuré dans un mode dit de routage direct, où les requêtes IP sont transmises en l'état, sans altération. Contrairement au mode alternatif NAT évoqué au début de l'article, le routage direct pose deux exigences fortes sur les directeurs et nœuds de service. La première est que les nœuds de service doivent accepter du trafic sur les VIP déclarées côté directeur. La seconde est qu'ils doivent pouvoir répondre directement aux utilisateurs sans passer par les directeurs. Ceux-ci voient transiter, dans un tel mode, une moitié du trafic applicatif, à savoir les requêtes transmises par les utilisateurs vers les VIP de service. L'en-tête IP des requêtes n'étant pas modifié par les directeurs, ceux-ci interviennent sur l'en-tête Ethernet pour diriger les paquets vers un nœud

précis. L'adresse MAC destination se trouve être celle du nœud sélectionné par l'algorithme de balance de charge. Les réponses envoyées par le nœud reprennent naturellement les mêmes informations de niveau 3, inversées. Grâce à l'adressage particulier de sa loopback, et à l'écoute des applications sur ses VIP4 et VIP6, le nœud est capable d'instruire des échanges mettant en jeu des adresses IP virtuelles, fussent-elles IPv4 ou IPv6.

3.2 Les nœuds de service

Les nœuds de service sont d'ordinaire distribués sur les trois plaques géographiques formant la dorsale de l'Université de Nantes. Dans l'absolu, rien n'oblige à suivre ce schéma de répartition : les nœuds sont indépendants et tous joignables par les directeurs. Il est imaginable d'avoir, pour un service peu utilisé, seulement deux nœuds répartis – minimum pour de la haute disponibilité –, et quatre ou cinq nœuds pour des services fortement sollicités. Ceux-ci sont bi-connectés au réseau avec une interface d'entrée, côté directeur (eth1) et une interface de sortie sur la dorsale (eth0). Eth0 est l'interface de sortie par laquelle les nœuds répondent aux requêtes relayées par les directeurs, reçues sur eth1. C'est également leur route par défaut. On remarquera le routage asymétrique des transactions puisque les requêtes (sens utilisateur vers service) arrivent par eth1 tandis que les réponses (sens service vers utilisateur) partent par eth0. C'est par l'entremise de leur eth1 que les nœuds de service reçoivent les requêtes des utilisateurs. C'est également par elle que les directeurs vérifient la présence des nœuds et le bon fonctionnement des services hébergés.

Une nouvelle fois, une troisième interface joue un rôle clé dans la délivrance du service. Le mode de fonctionnement particulier des directeurs conduit à la déclaration, sur l'interface lo, des VIP4 et VIP6. En substance, non seulement l'application cachée derrière le service doit écouter sur eth1 (contrôles applicatifs des directeurs), eth0 (supervision centrale) mais aussi sur lo.

3.3 L'alliance de la commutation de niveau 2 et 3

L'association du mode routage direct avec des services répartis géographiquement pose une dernière exigence forte d'implémentation. Un directeur doit pouvoir joindre en direct, autrement dit sans routage intermédiaire, n'importe lequel des nœuds, indépendamment de l'endroit où ils sont implantés. Un seul et même réseau doit être défini entre la sortie des directeurs et l'entrée des nœuds de service. A l'Université de Nantes, l'exigence se traduit par la propagation, sur la dorsale, du vlan rattaché à l'interface eth1 des directeurs et des nœuds de service.

Un second vlan est diffusé sur la dorsale, celui de l'interface eth0 des nœuds de service. C'est au travers de celle-ci qu'est déclarée leur route par défaut, avec laquelle les nœuds répondent aux requêtes des utilisateurs. Dans une optique de simplification, nous avons fait le choix d'élire par VRRP3 [RFC5798], les routeurs virtuels IPv4 et IPv6 utilisés par les nœuds. Concrètement, un des trois routeurs physiques de la dorsale, défini à l'avance, joue ce rôle pour les nœuds publics et privés. Sur défaillance de celui-ci, un des deux autres routeurs prendra le relais selon une priorité pré-paramétrée. Dans la situation extrême où les routeurs seraient totalement isolés, le protocole VRRP3 aboutirait à l'élection de trois routeurs virtuels, un pour chacune des plaques.

4 Faits marquants, points durs et évolutions prochaines

L'infrastructure actuelle trilocalisée pourrait très bien à l'avenir s'étendre à cinq plaques – incluant Saint-Nazaire et La Roche/Yon – raccordées elles aussi au 10Gbps à la dorsale –, ou bien se réduire à deux dans une optique de simplification. Une plate-forme a minima bilocalisée est un pré-requis du routage anycast ou de tout Plan de Continuité d'Activité ambitieux.

4.1 Le filtrage du trafic utilisateur ou comment se passer du contrôle d'état…

Le mariage de la virtualisation et du routage asymétrique sur les directeurs et nœuds de service est, au début du moins, une source manifeste de difficultés.

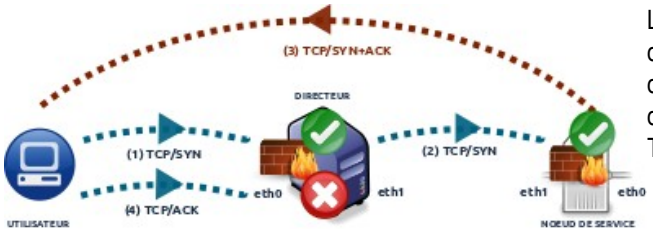


Illustration 6 : l'asymétrie des échanges dans le mode routage direct

Le routage asymétrique est synonyme de perte du contrôle d'état des communications de niveau 4 pour les pare-feu embarqués dans les éléments d'infrastructure. Le schéma ci-contre tente d'éclaircir ce point, en prenant l'exemple d'un simple échange TCP.

Le premier paquet d'établissement TCP/SYN est transmis sans embûche, du poste utilisateur jusqu'au nœud de service. Le directeur réceptionne le paquet (1) et le renvoie (2) au nœud sélectionné par l'algorithme de balance de charge. Le nœud reçoit le paquet sur son eth1 puis envoie l'acquittement TCP/SYN+ACK (3) sur son eth0, sa route par défaut. L'utilisateur y répond par un TCP/ACK (4), c'est le "3-way handshake TCP" traditionnel. L'ennui, c'est que le directeur reçoit un TCP/ACK sans avoir vu passer le TCP/SYN+ACK le précédant chronologiquement dans un établissement TCP ! Il considère, à juste titre, que l'établissement de connexion est inconsistant. Le paquet (4) est rejeté sine die, bloquant l'accès aux services applicatifs à l'utilisateur final.

Une solution radicalement simpliste aurait été de désactiver les pare-feu embarqués sur les directeurs et nœuds de service. Nous ne l'avons naturellement pas retenue pour d'évidentes raisons liées à la sécurité… Nous avons plutôt choisi de désactiver de manière très ciblée le contrôle d'état sur les pare-feu netfilter/iptables. Seules les communications relatives aux VIP4 et VIP6 sont concernées. Le reste du trafic, comprenant l'administration, la supervision des éléments, reste filtré avec contrôle d'état de niveau 4 et plus, comme sur n'importe quel pare-feu. L'intérêt d'une telle solution technique est double :

- la table des sessions, dite conntrack, ne se remplit plus inutilement de nouvelle entrées dans un état transitoire 'timewait', conséquence du routage asymétrique ; cet état pouvant potentiellement perturber les flux à l'expiration des compteurs ;
- les paquets de retour des utilisateurs sont servis normalement, exactement comme dans un mode NAT.

Le corollaire est une certaine dégradation, somme toute contenue, du niveau de sécurité de la solution. Un filtrage, certes sans contrôle d'état, est réalisé sur les services accessibles par les utilisateurs à destination des nœuds de service. Le réel suivi d'état de la session de niveau 7 est laissé aux applications des nœuds, et elles seules.

En réalité, la solution corrective basée sur la cible NOTRACK d'iptables ne se trouve ni appliquée sur les directeurs, ni sur les nœuds de service. Grâce à la virtualisation des couches réseaux, tout se déroule sur les hôtes et non plus sur les invités. Qui plus est, la politique de sécurité des invités est centralisée sur les hôtes. L'avantage d'une telle méthode est qu'en cas de compromission système d'une machine invitée, l'intégrité des règles de filtrage est garantie, sauf exposition de l'hôte lui-même. Les hôtes étant administrés hors de la bande, dans un réseau inconnu des invités, l'opération serait audacieuse.

4.2 Scénarios de panne

Un directeur agit tel un aimant sur le trafic des utilisateurs les plus proches : il attire tout ce qui passe à proximité. C'est à la fois un avantage, puisque des services applicatifs sont rendus aux usagers avoisinants, mais aussi un inconvénient lorsque le directeur commence à défaillir. Dans la situation d'une panne franche – arrêt du directeur, défaillance de l'hôte –, l'impact est réduit. Le trafic est automatiquement re-routé vers un nouveau directeur géographiquement plus éloigné. En revanche, une panne partielle est susceptible d'engendrer des désagréments notables. Un exemple concret est donné avec l'arrêt brutal du démon ldirectord. Pendant que le démon Quagga, lui, continuerait à annoncer les VIP des services déclarées sur la loopback, le directeur serait incapable de satisfaire les nouvelles requêtes des utilisateurs colocalisés. Une défaillance logicielle partielle mènerait à une perte de service franche quoique, fort heureusement, localisée à la plaque. En supervisant par snmp les processus applicatifs actifs, il est possible de limiter la portée qu'aurait une telle défaillance. Une solution proactive, elle, serait de contrôler localement les processus des directeurs avec un outil comme monit. Les démons ldirectord et quagga seraient comme appariés, l'arrêt du premier conduisant à celui du second. La perte de service serait efficacement évitée.

D'autres scénarios de panne sont naturellement envisageables. Le tableau ci-dessous référence les principaux symptômes qu'est susceptible de rencontrer la plate-forme tout au long de sa vie.

Symptôme	***Conséquences***	***Criticité***
Perte des liaisons inter-plaque	Les plaques se retrouvent isolées les unes des autres. Les services clusterisés restent joignables dans un mode dégradé.	Moyenne
Panne d'un nœud de service	Les directeurs constatent le dysfonctionnement au bout de six secondes maximum (checkinterval).	Faible
Panne franche d'un directeur	Seuls les utilisateurs localisés sur la plaque du directeur en panne sont impactés. Il faut attendre un délai de reconvergence OSPF de l'ordre de quatre secondes (dead-interval) avant que le trafic ne soit re-routé vers un autre directeur. Les utilisateurs doivent rétablir leurs sessions applicatives.	Faible
Panne partielle d'un directeur	Dans la situation d'une défaillance seule du démon ldirectord, une perte de service pour les utilisateurs colocalisés est inéluctable. Un monitoring actif basé sur un outil comme monit, engendrerait la même perturbation qu'une panne franche. Elle serait indéniablement plus forte avec une supervision classique, l'incident n'étant réglé qu'à l'intervention des équipes opérationnelles.	Faible à élevée

Symptôme	*Conséquences*	*Criticité*
Panne d'un routeur de la dorsale	La plaque concernée est isolée, tout comme ses directeurs et nœuds de service. Tous les services demeurent inaccessibles aux utilisateurs locaux.	Moyenne à élevée

Plusieurs de ces symptômes se rencontreraient tôt ou tard pendant une mise à jour système et/ou applicative. Pour éviter leur apparition, nous avons mis en place des directeurs et nœuds de tests nous permettant de valider, in situ, toute évolution logicielle ou fonctionnelle avant sa mise en production.

4.3 Évolutions fonctionnelles en cours et futures

L'exploitation d'une ferme de serveurs conduit à des changements de configuration plus ou moins fréquents selon la nature de ses nœuds. Chaque modification apportée à l'un d'eux doit être répercutée, à l'identique, sur le reste de la ferme. Des erreurs de saisie avec une intervention 100 % manuelle sont inéluctables. Il est préférable, de notre point de vue, d'automatiser cette tâche contraignante. Un outil libre existe spécialement pour ce besoin, csync2 [CSYNC2]. Tous les éléments de configuration d'un nœud qualifié de référence sont synchronisés sur les *n* nœuds de la ferme, à la manière d'un rsync. D'éventuels conflits de version sont détectés et nécessitent alors une action corrective manuelle. Cet outil est capable d'exécuter n'importe quelle commande système sur déclenchement d'une synchronisation, comme le rechargement d'un processus. Il procure de l'agilité : des changements sont applicables à l'ensemble d'une ferme, en se connectant seulement à l'un de ses nœuds. Même si la fonctionnalité est alléchante, nous préférons actuellement recharger manuellement les processus concernés sur chaque machine virtuelle. Le but est d'éviter un plantage généralisé des nœuds si une erreur de configuration venait à se glisser sur celui d'où est déclenchée la synchronisation.

Ces mêmes principes de synchronisation sont aujourd'hui répétés sur les directeurs. Les définitions des services à gérer sont consignées dans un unique fichier, ldirectord.cf. Son contenu déclare de façon exhaustive mais basique, tous les nœuds à considérer pour un service donné. Chaque déclinaison de ce service selon différents critères (IPv4, IPv6, TCP, UDP, ports sécurisés), fait l'objet d'une nouvelle déclaration indépendante, ce qui le rend particulièrement volumineux : aujourd'hui, le fichier de configuration des directeurs privés fait près de 700 lignes. Le souhait de privilégier les échanges locaux entre directeur et nœud de services co-localisés est impossible à obtenir en synchronisant cet unique fichier entre les trois directeurs privés ou publics.

Des évolutions techniques sont en cours de validation pour garantir la localité des échanges. L'idée est de décliner, au travers d'un script en cours de développement, une politique de balance de charge selon la localité du directeur. Le directeur d'une plaque pourra alors prioriser les nœuds locaux sur les nœuds distants. Ce script est déjà capable, grâce à une convention de nommage dans nos DNS, des tests de connectivité réseau et d'autres mécanismes, de produire des fichiers ldirectord.cf spécifiques à chaque directeur. Il suffit d'éditer quelques lignes de définitions de services, sans rapport avec la quantité de nœuds associés. Outre une simplification évidente dans la définition des services, l'utilisation d'un script permet également d'assurer la cohérence syntaxique de ce fichier et d'éviter les erreurs humaines.

Dans sa version actuelle, ldirectord sait notifier une station de supervision du retrait ou de l'ajout d'un nœud dans une ferme de service. Malheureusement, l'alerte déclenchée est trop générique. Elle ne contient pas la moindre information sur la ferme – typiquement, quelle est la VIP impactée –, ni sur le nœud concerné. C'est pour le moins préjudiciable à un directeur gérant des dizaines de services et trois à quatre fois plus de nœuds… Une révision du code Perl de ldirectord est inévitable pour étendre ses fonctionnalités de supervision.

5 Bilan d'étape

Depuis la mise en production des premiers nœuds de service en mars 2011, un long chemin a été parcouru. Aujourd'hui, alors que la totalité des services de l'ancienne plate-forme a migré, de nouvelles applications sont intégrées au gré des besoins. Les services déployés profitent nativement d'une double connectivité IPv4/IPv6 alliée à une mise à l'échelle rapide. Résolveurs DNS, mandataires de navigation, mandataires inversés avec WAF[6], Central Authentication Service (CAS), annuaires LDAP, webmails, bases de données MySQL, sont quelques-uns des services déjà en production. En l'espace de neuf mois, les éléments fondateurs d'architecture que sont les routeurs, directeurs, nœuds de service ont emmagasiné une charge de trafic toujours croissante, sans montrer le moindre signe d'essoufflement. Autour de 8000 sessions simultanées sont gérées par les directeurs privés, contre 1500 pour les publics. Une récente panne électrique ayant mis à mal les matériels d'une des trois plaques, a d'ailleurs prouvé le bien-fondé que nous avions à répartir géographiquement les services avec anycast.

[6]Web Application Firewall ou pare-feu applicatif spécialisé dans l'analyse du trafic web.

Qualifier d'élémentaire la nouvelle plate-forme serait une grossière erreur. La complexité est omniprésente, inhérente à ce type d'infrastructure. En revanche, elle s'est clairement déplacée à la fois sur les directeurs et sur les routeurs de la dorsale. Heureusement, les difficultés d'exploitation supplémentaires sont mesurées grâce à l'automatisation et la simplification des tâches et outils. Les nœuds de service, eux, se déploient désormais très facilement, ce qui équilibre au final la facilité d'emploi à l'échelle du cluster. L'objectif initial de conserver une solution contrôlable techniquement est finalement atteint. Le fait de gérer des services adossés à un faible volume de données y a contribué.

Des systèmes travaillant sur une volumétrie importante sont plus délicats à intégrer. Les applications aux architectures N-tiers peuvent recourir à une instance redondée de leur SGBD : nous avons ainsi déployé un service mysql. Pour des besoins d'accès concurrents à des données, un système de fichiers en réseau, tel que NFS, est suffisant – mais se pose alors la question de la disponibilité de cette ressource –. Autre possibilité, une réplication en temps réel des informations sur chaque nœud, envisageable avec des outils comme lsyncd ou drbd. Il est aussi possible de partager des volumes de données via un SAN et un système de fichiers cluster, tel qu'OCFS2. La dernière solution, la plus en phase avec la philosophie de nos déploiements, est de déployer des systèmes de fichiers distribués, tels que glusterfs ou ceph. Malheureusement, ces technologies très intéressantes mais complexes sont à peine matures, et nous ne les avons pas encore qualifiées pour une phase de déploiement.

Des écueils révélés en phase de maquettage de la plate-forme, qui ne lui sont toutefois pas imputables directement, nous ont passablement ralentis. Pertes de paquets à travers les commutateurs Ethernet des châssis blade, limites par défaut des noyaux GNU/Linux, suites logicielles d'administration des invités LXC et KVM perfectibles, sont autant de points que nous avons dû solutionner. La plate-forme est à tout moment évolutive, grâce à la souplesse des briques open-source. Un rapide bilan financier viendrait appuyer nos choix techniques et confirmerait des économies très substantielles : études amont, maquettage, installation matérielles et logicielles, recette, ont monopolisé l'équivalent de 45 Homme/Jour. La mise en production d'un nouveau nœud de service nous occupe quelques minutes (sans compter le temps de paramétrage applicatif)… 0€ de licence sur l'année, 0€ de prestations intellectuelles, d'accompagnement, 2 H/J de maintenance par mois donnent une idée du coût global de la plate-forme. S'engager dans une solution 100 % libre comporte des contraintes évidentes et des avantages certains !

6 Bibliographie

[ULTRAMONKEY] Load Balancing and High Availability Solution, http://www.ultramonkey.org/

[CSYNC2] Cluster SYNchronization tool, 2nd generation, http://oss.linbit.com/csync2/

[QUAGGA] Quagga Routing Software Suite, http://www.quagga.net/

[RFC5798] S. Nadas, « Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6 », Mars 2010.

[RFC4786] J. Abley, K. Lindqvist, « Operation of Anycast Services », RFC 4786, Décembre 2006.

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

[LXC] Linux Containers, http://lxc.sourceforge.net/

[XEN] XEN, http://xen.org/

[VSERVER] Linux-VServer, http://linux-vserver.org

[KEEPALIVED] Keepalived, HealthChecking for LVS & High-Availability, http://www.keepalived.org/