Unification du système de messagerie de l'Université de Nantes

Yann Dupont
CRI de l'Université de Nantes
2, rue de la Houssinière, 44322 Nantes
Yann.Dupont@univ-nantes.fr

Arnaud Abelard
CRI de l'Université de Nantes
2, rue de la Houssinière, 44322 Nantes
Arnaud.Abelard@univ-nantes.fr

Michel Allemand
CRI de l'Université de Nantes
2, rue de la Houssinière, 44322 Nantes
Michel.Allemand@univ-nantes.fr

Jacky Carimalo
CRI de l'Université de Nantes
2, rue de la Houssinière, 44322 Nantes
Jacky.Carimalo@univ-nantes.fr

Résumé

Cet article relate notre expérience de la mutualisation complète de la messagerie électronique sur l'Université de Nantes. Nous mettons l'accent sur les particularités du système mis en œuvre, les difficultés rencontrées, autant techniques qu'humaines.

Notre solution est tournée vers la simplicité et l'évolutivité. Pour parvenir à ces fins, nous avons fait le choix de découper le système en de multiples services élémentaires, hébergés sur des serveurs virtuels. Notre solution est entièrement constituée d'outils open source.

L'ensemble, déployé depuis septembre 2004, gère les comptes du personnel (5000 comptes) et a permis d'échanger depuis quelques millions de messages. Le dispositif donnant satisfaction, l'opération a été reconduite pour les étudiants (40.000 comptes). Malgré la différence d'échelle, les deux installations s'avèrent finalement très peu différentes, les choix faits pour la première opération ayant été quasiment reconduits à l'identique lors de la seconde opération.

Mots clefs

Courrier électronique, Anti SPAM, LDAP, Mutualisation, Virtualisation

1 Introduction

Le service de courrier électronique sur l'Université de Nantes a été historiquement assuré sur un modèle décentralisé, délégué à qui voulait bien s'en charger: laboratoire, équipe ou UFR. Chacun étant uniquement responsable de son propre domaine, indépendamment des autres. L'existant, fin 2003, était constitué d'une cinquantaine de serveurs disséminés sur toute l'Université, avec une grande disparité au niveau des moyens techniques et humains engagés, du nombre d'utilisateurs gérés et de la qualité du service rendu. Cette pléthore de serveurs a fini par poser de nombreux problèmes et a ouvert la voie à une réflexion pour installer un meilleur système.

Cette réflexion a été menée à la fois sur un plan politique et sur un plan technique. Le soutien actif de la direction de l'Université, concrétisé par une inscription dans le projet d'établissement et un vote en conseil d'administration, a aidé à faire rapidement avancer ce projet. La mise en place d'une équipe technique constituée de personnels des différentes composantes de l'Université a permis la prise en compte de tous les besoins ainsi que la réalisation technique du projet dans des délais raisonnables.

Le déploiement préalable d'un annuaire LDAP référençant tout le personnel nous a conduit à imaginer un système global reposant sur cette technologie.

Quand nous a été confiée la mise en place de ce nouveau système, nous avions finalement peu de métrologie sur l'existant à remplacer, que ce soit en terme de mails échangés, de taille des boîtes ou de fréquence des consultations. Nous n'avions de plus aucune vision des modifications de comportement qu'allaient introduire les avantages apportés par la mise en œuvre d'un service mutualisé.

Ces incertitudes, le nombre potentiel de personnels ayant accès au courrier et le nombre de serveurs à remplacer nous ont conduit à être très prudents, et à chercher, dans un premier temps, les solutions les plus simples. En tant que tel, l'échange de courrier électronique est quelque chose de très répandu, et nous n'avons certainement pas réinventé la roue. Nous avons fait le tour de ce qui existait et avons essayé de l'assembler au mieux, et le plus simplement en fonction de nos objectifs, qui sont détaillés ci-après.

Les choix qui se sont dégagés ont été faits pour ne pas être limités si nous avions fait de grosses erreurs d'estimation. Un changement ou une évolution de stratégie devait pouvoir être possible à tout moment, sans impact sensible pour l'utilisateur. Et le tout en restant dans des budgets compatibles avec nos moyens. Les technologies de virtualisation, que nous déployons depuis plusieurs années, nous ont grandement aidé dans cette voie, et ont permis d'atteindre ce but.

Les chapitres suivants détaillent la stratégie adoptée ainsi que la sélection des outils utilisés. Puis, l'installation effective de la solution, les phases de migration nécessaires et les problèmes rencontrés seront examinés. Nous reviendrons ensuite sur les développements spécifiques, nécessaires pour la mise en œuvre du projet.

2 Les objectifs du courrier électronique unifié

Les réflexions politiques ont fixé un certain nombre d'objectifs dont les principaux sont les suivants :

- L'homogénéité : tous les courriers électroniques sont formés sur le modèle « Prenom.Nom@univ-nantes.fr »,

avec possibilité de conserver les anciennes adresses comme alias. Il est aussi possible de créer des adresses fonctionnelles comme « Directeur.Composante@univ-nantes.fr ». Le couple login/mot de passe utilisé pour la messagerie doit être le même que pour les autres services, en particulier l'Intranet et les accès WiFi.

- L'égalité : tous les personnels disposent de la même qualité de service, quelle que soit leur composante de rattachement. Tous les utilisateurs bénéficient de fonctionnalités avancées, comme l'antivirus, l'anti-SPAM et un webmail efficace.
- La robustesse : cette nouvelle messagerie doit être d'une très grande disponibilité, les messages stockés doivent être protégés et sauvegardés.

Les réflexions techniques ont aussi fixé des objectifs dont les principaux sont les suivants :

- L'évolutivité : le projet est conçu pour pouvoir évoluer et bénéficier des dernières nouveautés technologiques. Des plate-formes de tests sont disponibles pour l'équipe technique afin de tester et valider de nouvelles technologies.
- La performance : il aurait été très mal accepté que la nouvelle messagerie soit plus lente que les services « simples » qu'elle remplace. Si cela avait été le cas, elle aurait été ressentie comme une régression.
- La simplicité : toutes les machines clientes sont configurées de la même façon. L'accès à distance est disponible. Tout nouvel arrivant à l'université reçoit automatiquement une adresse électronique. Pour l'équipe qui gère l'ensemble, la maintenance doit aussi pouvoir être simple.

Cette centralisation permet en outre la mise en place de permanences pendant les périodes de fermeture de l'université. Beaucoup d'informaticiens de l'université se trouvent ainsi presqu'entièrement libérés de la gestion des infrastructures du service de messagerie électronique, tout en continuant de gérer les usagers de leur composantes.

2.1 Portée du déploiement

Initialement, seules les boîtes aux lettres du personnel, environ 5.000 comptes, ont été prises en compte. Le système devait être calibré pour cette population, et tenir la charge alors répartie sur cinquante serveurs.

Il a été de choisi de migrer les listes de diffusion plus tard. Techniquement, il n'y avait pas urgence, puisque une liste de diffusion peut être perçue comme un simple client d'une messagerie, et à ce titre, est peu liée avec toute la mécanique mise en place.

La phase pilote a débuté en janvier 2004, la mise en service effective en juin 2004 et les migrations massives ont débuté en septembre 2004 pour s'achever officiellement fin 2004. En réalité, des migrations étaient toujours en cours sur certains sites en mars 2005. La relative lenteur du déploiement est essentiellement due à des problèmes humains. Quelques migrations ont aussi été rendues plus complexes car certains sous-domaines (par exemple polytech.univ-nantes.fr), se trouvaient utilisés autant par le personnel que par les étudiants. Il a fallu traiter ce cas particulier.

Devant le fonctionnement satisfaisant de la solution, elle a été reconduite, quasiment à l'identique, pour les étudiants, ce qui représente 40.000 utilisateurs concernés. La mise en place est effective depuis septembre 2005.

3 Principe général : diviser pour régner

Bien entendu, gérer les messages de 50 ou 5.000 personnes ne se conçoit pas de la même façon. Nous avons décidé de partir d'une feuille blanche. De prime abord, il nous a paru évident de monter un **service** de courrier électronique, pas un serveur de messagerie. Une solution monolithique n'a jamais été envisagée. Au contraire, le choix de diviser au **maximum** les opérations a prévalu.

En simplifiant, la cinématique d'un envoi de message peut être représenté ainsi:

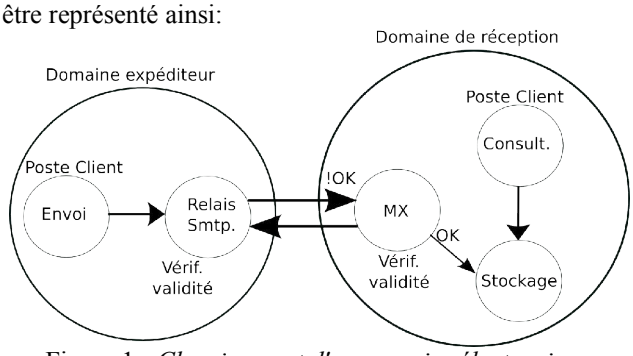

Figure 1 - *Cheminement d'un courrier électronique*

Depuis un poste, l'expéditeur rédige une lettre à destination d'une autre personne : le récepteur. Le message est envoyé du poste local vers le serveur de messagerie, qui en vérifie la validité (pas de virus) , et l'injecte alors sur le réseau via un logiciel[1] utilisant le protocole SMTP. Le domaine de destination réceptionne le courrier, le vérifie à son tour (virus et SPAM), pour ensuite le déposer[2] dans la boîte aux lettres du destinataire. Celle-ci est ensuite consultée par le récepteur, grâce à un logiciel de messagerie[3]. Il lit ce message, puis décide ou non d'y répondre. Et le circuit recommence.

Malgré la séparation de la figure ci-dessus en deux parties, dans la réalité un utilisateur reçoit du courrier et en envoie également. Le schéma ci-dessus est symétrique et nous devons implanter les deux fonctions : émission et réception.

Le choix fondamental a été de traiter chacune de ces opérations **indépendamment**, sur des serveurs **différents** et redondés. Le but est clair : si l'une de ces parties est défaillante (serveur en panne), elle n'empêche pas le reste de l'ensemble de rester fonctionnel.

3.1 Diviser pour fiabiliser

En effet, un des objectifs techniques importants, si ce n'est le plus important était de disposer d'une structuration

[1]Le MTA : Mail Transport Agent

[2]Le MDA : Mail Delivery Agent

[3]Le MUA : Mail User Agent

tolérant les pannes, qu'elles soient matérielles, logicielles, électriques, ou dues à des erreurs humaines. Le choix de diviser le projet en de nombreux sous-services contribue à cette fiabilité, chacun étant rendu par des programmes déployés en plusieurs exemplaires sur des machines différentes. Les chemins de circulation de l'information sont ainsi augmentés.

Un autre choix important a été de complètement dissocier les données des serveurs. Ceux-ci, s'ils traitent ou stockent des données, sont connectés à un SAN qui centralise les informations. Les machines physiques deviennent ainsi relativement standards et il devient aisé de déployer, redéployer ou de multiplier simplement la majorité d'entre elles.

Certains services s'avèrent plus critiques que d'autres. Sans eux, l'ensemble ne peut fonctionner. LDAP en est un bon exemple. Des techniques, détaillées plus loin ont été déployées pour tenter, autant que possible, de rendre le projet tolérant aux pannes.

3.2 Diviser pour augmenter la performance

Étant donnée la charge attendue sur les serveurs, il nous paraît plus efficace de gérer n machines distinctes qu'une seule, munie de n processeurs. Si les opérations sont indépendantes les unes des autres, l'évolution de capacité de traitement peut se faire très simplement, quasiment linéairement, en augmentant le nombre de machines dédiés aux fonctions le nécessitant. Néanmoins, pour certaines opérations (le stockage du courrier), nous verrons que cela n'a rien d'évident.

3.3 Diviser pour simplifier et évoluer

Diviser permet de simplifier non seulement la sécurisation des services mais surtout leur maintenance. Chaque serveur possède son propre firewall et les règles implantées sont courtes et peu complexes, puisque ne concernant que l'unique service hébergé. Ainsi on obtient une sécurisation très simple à vérifier, à maintenir et à faire évoluer. La relative standardisation des machines physiques, qui découle de cette division et de la séparation des données, permet en outre de redéployer ou d'augmenter très simplement certains services.

Les mises à jour des logiciels ou du système d'exploitation sous-jacent sont elles aussi simplifiées puisque les effets de bord sont limités. Il est ainsi facile de migrer des versions de programmes sans remettre en cause la stabilité de l'ensemble. Sans plus d'impact, il est possible de changer de stratégie pour un service donné. De fait, au cours de la vie de ce projet, qui n'est en rien figé, des évolutions ont déjà eu lieu, et d'autres sont à venir rapidement. Nous relaterons ces évolutions au fil du document.

Cependant, cette stratégie de division, augmente de façon très importante le nombre final de serveurs à déployer. Heureusement, dans les faits, il est possible de mettre en place une relative automatisation de la maintenance du parc de serveurs, de nombreux éléments étant communs a toutes les configurations.

4 Stratégie de déploiement

Ce choix d'éclater complètement le projet a donc été pris dès le départ. Par contre, nous manquions d'expérience dans le déploiement d'un projet d'une telle dimension. L'expérience acquise dans des installations plus modestes nous a rapidement conduit à architecturer le projet autour d'une poignée de techniques que nous maîtrisions.

4.1 Annuaire du personnel

Avant même de réfléchir aux technologies à choisir pour échanger des messages, se pose la question de savoir à qui et où les déposer. Pour cela, il faut référencer nos utilisateurs. Un annuaire de tous les personnels a été construit depuis 2001. Il est basé sur un schéma spécifique, qui a été étendu au fur et à mesure pour s'acquitter de nouvelles fonctions, via l'ajout de schémas standards. Début 2004, nous l'avons donc doté des informations nécessaires pour gérer la messagerie électronique. Les données nécessaires à la réception d'un courrier sont contenues dans le schéma qmail-LDAP v3[4][1]:

- adresse email principale.
- adresses email secondaires (alias).
- destination finale du courrier (locale ou distante).
- quota.
- le serveur gérant la boîte.

L'annuaire LDAP n'est pas la source de référence à l'Université, mais uniquement une vue partielle de la base de données globale sur le personnel. Sa génération se fait via un logiciel développé en interne, DYNA, détaillé plus loin dans le document.

DYNA prend en compte les cas particuliers difficiles à gérer, comme les personnels ayant quitté provisoirement ou définitivement l'université. Les comptes sont dotés d'une durée de vie fixée en fonction du contrat et des situations. Par exemple, pour les étudiants, un délai de 6 mois avant fermeture définitive du compte est donné, pour gérer les cas de réinscription tardive.

C'est aussi DYNA qui décide de l'adresse électronique, et gère les homonymies (avec intervention humaine pour le choix définitif). Les alias permettent de régler facilement ces problèmes, au demeurant peu nombreux parmi le personnel.

Utiliser LDAP comme source d'information sur les utilisateurs nous permet aussi d'atteindre le but d'unification des logins et mots de passe avec d'autres services existants.

4.2 Virtualisation

Il reste un problème de taille : en poussant jusqu'au bout la logique d'un serveur par service, et en mettant en place une technique de tolérance aux pannes, le nombre de machines impliquées devient important, et supérieur à ce qu'autorise notre budget. Nous avons donc choisi le chemin de la virtualisation.

Les techniques de virtualisation de serveurs sont utilisées au CRI de l'université depuis fin 2002. Ce point a fortement influencé nos décisions ; devant les bons

[4] OID 7914 assigné par IANA

résultats des premiers services installés via cette solution, nous avons décidé de déployer massivement des serveurs virtuels, afin de répondre à notre volonté de séparer au mieux toutes les étapes de la messagerie. Les techniques utilisées sont décrites plus précisément au chapitre suivant.

4.3 Tolérance à la panne

L'implémentation s'est effectuée en suivant le schéma de principe (figure 1) que nous avons étoffé afin de permettre une migration en douceur et un degré de fiabilité satisfaisant. Il a été décidé de déployer des serveurs virtuels similaires pour diviser la charge de travail de chacun, et de redonder le projet, conformément à nos objectifs.

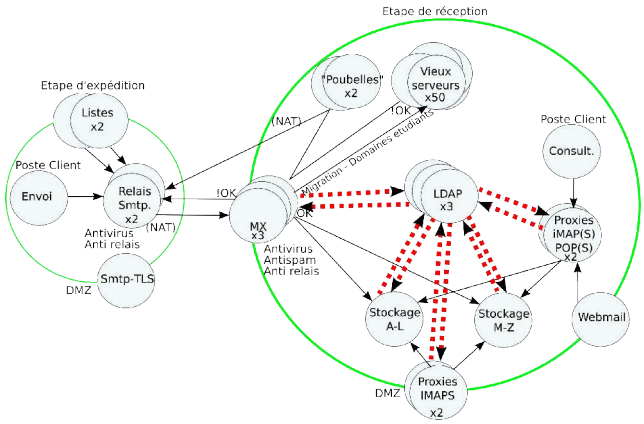

Figure 2 - *Schéma d'implantation*

La figure 2 est une évolution de la première illustration qui rentre dans le détail des opérations. Le coté gauche de la figure représente la partie envoi de message, celle de droite la consultation. Cette séparation est purement logique, dans un but de lisibilité du schéma. Dans les faits, nous implantons les deux fonctions.

En frontière du schéma (zone publique DMZ) se trouvent les fonctions accessibles de l'extérieur de l'université : pour la partie envoi de messages, les deux serveurs de liste de diffusion et celui d'envoi avec authentification. Pour la partie réception, les trois MX de l'université, le Webmail, ainsi que les deux IMAPS.

En zone privée (Intranet), se trouvent le reste des services ; deux serveurs POP/IMAP et deux gérant le stockage des courriers. Tous utilisent les trois LDAP (flux en pointillés). On trouve également en interne les deux serveurs poubelles, deux autres gérant les envois et une cinquantaine d'anciens serveurs de messageries, conservés pour la phase de migration.

Le nombre de serveurs déployé pour chaque service a été calculé en fonction des besoins évalués pour le remplacement de l'existant, plus un supplémentaire, pour raison de disponibilité. La technique de virtualisation nous permet d'augmenter facilement ce nombre si le besoin s'en faisait sentir. Pour un même service, les serveurs virtuels sont déployés sur des machines physiques différentes.

5 Choix des logiciels

Les choix directeurs importants désormais figés, nous avons pris notre temps pour évaluer les logiciels devant s'acquitter de chaque opération élémentaire. Le principe de complètement éclater le système en petits éléments amène en effet de nombreux choix possibles sur chacune de ces briques.

5.1 Virtualisation

Virtualisation des serveurs.

Plusieurs techniques de virtualisation existent. Le but recherché reste toujours le même : faire fonctionner indépendamment, et de façon transparente, plusieurs serveurs sur une unique machine physique. Dans le cas du courrier électronique unique, le logiciel Linux-Vserver[2] est utilisé. Il emploie une technique de virtualisation légère. C'est le noyau du système d'exploitation qui réalise une isolation entre des machines logiques, tout comme il isole déjà les processus entre eux. L'avantage se traduit par une performance native, sans dégradation sensible.

La consommation mémoire demeure légère, elle est mutualisée entre l'hôte et les vservers (la mémoire demandée à l'hôte est celle réellement utilisée par les processus actifs). Déployer de nouveaux vservers est peu coûteux en moyens, qu'il soient financiers ou humains. Des patrons de serveurs virtuels peuvent être construits, puis déployés en quelques minutes.

Outre la possibilité de mutualiser plusieurs serveurs sur une même machine physique, et donc de limiter le nombre de machines réelles à installer, cette technique permet aussi de déployer des services clonés sur plusieurs machines physiques, dans le but d'en augmenter la disponibilité. C'est d'ailleurs plus que conseillé ; en effet la panne d'une machine physique entraîne la panne de tous ses serveurs virtuels hébergés. Il nous paraît donc nécessaire de redonder tous les serveurs virtuels déployés, pour une fiabilité maximale.

Un effort d'administration particulier doit aussi être fait. Par exemple, pour éviter tout débordement et tout effet de bord, il est souhaitable d'utiliser au moins une partition disque dédiée pour chaque vserver. De cette façon, il ne peut saturer le disque du serveur hôte.

Virtualisation du stockage.

La multiplicité des consommateurs d'espace de stockage (les serveurs virtuels), et des fournisseurs d'espace de stockage (disques locaux et LUN SAN) rend la situation complexe. De plus, le dimensionnement initial des volumes assignés à un vserver est parfois difficile à évaluer. La mise en place d'un SAN amène une grande souplesse d'utilisation de ces volumes. Hélas, le schéma traditionnel de partitionnement des disques sur l'architecture PC est très rigide mais peut être améliorée en utilisant une virtualisation de l'espace de stockage, comme EVMS [3].

Ce logiciel open source est développé par IBM. Il est désormais basé sur la couche « device mapper » [4] du noyau Linux, comme LVM2 [5]. Cette couche standard, introduite dans le noyau 2.6, permet de masquer la réalité des disques physiques et de présenter des disques logiques. Les fonctions d'EVMS sont nombreuses. Pour notre projet, les fonctions de migrations de volume disque à chaud et de snapshots de disque, qui simplifient les sauvegardes sont

utilisées. Il est aussi possible d'augmenter un espace de stockage à chaud. Voir la figure 3 pour une illustration.

5.2 Annuaire du personnel

Ce service existait déjà, et OpenLDAP [6] était la seule solution libre à l'époque. Tous les attributs utilisés de façon importante par les fonctions de messagerie sont indexés. La rapidité des accès en lecture est un atout majeur.

Problèmes rencontrés

LDAP est la source unique d'informations utilisée par l'ensemble. Il s'agit de la pierre angulaire du système. Si LDAP connaît des dysfonctionnements, la sanction est immédiate et les messages seront perdus.

Les LDAPs sont redondés, mais cela ne nous place pas à l'abri d'une erreur humaine, ou d'un bug dans DYNA, puisque la génération des LDAPs est automatique.

Le système est stable en consultation, mais a démontré une certaine fragilité lors de grosses campagnes de modification de l'annuaire (période de rentrée des étudiants). Heureusement, sans impact pour l'utilisateur. C'est donc un des aspects les plus sensibles de notre système, et des solutions devront être trouvées pour minimiser les risques, via de nombreux services de contrôle interne de cohérence.

Évolutions futures

Des alternatives libres à OpenLDAP commencent à être disponibles. Elles seront évaluées.

5.3 Transport du courrier

Ce choix n'a pas provoqué beaucoup de discussions. En effet, tout le monde dans l'équipe technique connaissait bien Postfix [7], unanimement apprécié pour sa performance, sa sûreté et sa (relative) simplicité de configuration.

Cela ne signifie pas pour autant que d'autres MTA n'aient pu faire l'affaire. Exim4, par exemple, présente des fonctionnalités LDAP et des possibilités d'extensions très appréciables.

Toutes les étapes SMTP du système sont donc confiées à Postfix.

5.4 Réception des messages

C'est donc ici Postfix qui officie. Ce serveur (appelé MX) réceptionne les messages en provenance de l'extérieur. L'intégralité des sous-domaines de univ-nantes.fr (une cinquantaine) sont acceptés et pointent sur ce service (déclaration des MX dans le DNS).

La configuration de cette étape a demandé le plus grand soin, car le traitement des destinations est particulier. Certains domaines gèrent indifféremment personnels et étudiants. Les serveurs de listes de diffusion ont également continué assez longtemps à être gérées sur les sites d'origine. Aussi deux cas doivent être considérés :

a) Le domaine a entièrement migré, et l'annuaire LDAP fait référence absolue.

b) Le domaine contient des comptes n'ayant ou ne pouvant pas migrer, par exemple ceux des étudiants.

Les domaines de type b) sont listés dans une table statique, consultée par Postfix, avec une table de transport associée, qui indique où délivrer les messages dont on ne sait que faire (adresse IP de l'ancien serveur de messagerie).

Les étapes peuvent se décomposer de la façon suivante:

1. Réception du courrier :
   Le serveur vérifie l'existence du destinataire dans notre annuaire LDAP. S'il est connu, le message est accepté. Sinon et si le domaine de réception est du type a), le message est immédiatement refusé (il s'agit en général de SPAM utilisant des adresses générées).
   Si le domaine est de type b), nous devons accepter le courrier sans savoir si la destination est valide ou non dans ce domaine et le traiter. S'il y a lieu, le refus final du message devra se faire par un autre mécanisme.
2. Traitement antiviral et anti-SPAM :
   Le mail est passé au traitement antivirus, et antispam. Si un virus est détecté, le message est supprimé et une alerte est envoyée, uniquement aux administrateurs.
3. Envoi pour stockage :

   Le courrier doit ensuite être transmis pour assurer son stockage. Une requête LDAP suffit à déterminer la destination finale du courrier. Soit l'interrogation retourne un résultat et le courrier est envoyé au serveur de stockage concerné, soit la requête ne retourne aucun résultat et nous sommes dans le cas d'un message à destination d'un domaine de type b), pour une adresse non répertoriée (un étudiant, ou un ancien serveur de liste de diffusion). La table de transport associée est consultée et le message transporté sur l'ancien serveur.

Évolutions futures

Cette complexité est en train de disparaître avec la mise en place de la messagerie des étudiants. Les listes de diffusion ont, elles aussi, migré en central depuis mi 2005. Il ne reste plus que quelques domaines dans le cas b) en septembre 2005.

5.5 Anti-SPAM et anti-virus

Ce point était le plus faible de tout ce qui avait été déployé jusqu'à présent à l'université. Très peu de serveurs existants avaient un traitement anti SPAM. C'était donc un plus attendu par de nombreux utilisateurs. C'est aussi un domaine où le développement est important et où de nombreuses solutions existent.

Nous avons pris le parti de déployer dans un premier temps des solutions simples, que nous maîtrisions bien, sans pour autant ignorer des technologies plus avancées, afin de pouvoir les installer lorsque le besoin s'en fera sentir. Il nous a paru important de ne pas prendre le moindre risque, afin de ne pas perdre un seul message.

Pour nous, le but du système anti-SPAM n'est pas de bloquer les courriers dans le doute, mais simplement de donner un avis quant à la possibilité qu'un mail soit un SPAM. Nous avons décidé que l'anti-SPAM ne devait jamais refuser de courrier, mais juste marquer le sujet et ajouter un en-tête précisant le « niveau » de SPAM du courrier.

C'est un choix probablement extrêmement conservateur.

Techniquement, nous avons choisi d'utiliser amavisd-new [8]. Ce logiciel, en PERL, fait l'interface entre le MTA et les systèmes Anti-virus et Anti-SPAM. Il s'interface bien avec Postfix et s'adaptait facilement à nos besoins .

Antivirus

Nous avons choisi, par sécurité, d'utiliser deux antivirus en parallèle. Notre choix s'est porté sur McAfee uvscan car nous possédons la licence du Ministère, ainsi que clamav [9], un antivirus open source performant et dont les signatures sont mises à jour de façon très réactive.

Dans les faits, les deux anti-virus détectent quasiment à l'identique les virus. À l'apparition de nouvelles souches ou variantes, le temps de création des signatures peut varier de quelques heures entre les deux logiciels. Le premier à posséder la nouvelle signature protège alors l'intégralité des usagers pendant ce laps de temps.

L'université de Nantes est miroir officiel de clamav, ce qui nous permet d'avoir les signatures dès qu'elles sont disponibles.

Le double anti-virus nous a aussi protégé d'erreurs de récupérations de signatures (déplacement de l'archive chez l'éditeur).

Anti SPAM

Amavisd-new s'interface aussi avec Spamassassin [10]. Dans une première phase, celui-ci examine l'intégralité du message, puis utilise un jeu de règles statiques associées à un score. Une règle peut être applicable au corps du message ou à un en-tête. Nous en avons ajouté plusieurs en plus de celles par défaut, comme par exemple la détection de chaînes aléatoires ou la détection de balises invisibles. Une deuxième étape confronte le message à un filtre à apprentissage Bayesien. Ce filtre nous a semblé perfectible. Nous l'avons donc remplacé par mailfilter.crm, un filtre à apprentissage écrit dans le langage CRM114 [11]. Bien que mailfilter.crm puisse complètement remplacer Spamassassin, nous souhaitions conserver la première partie statique d'examen du courrier. Cela a demandé une petite adaptation spécifique, détaillée plus loin. Comme pour le filtre par défaut, celui-ci est appelé à la suite de spamassassin et lui aussi donnera son avis quant à la probabilité que le mail analysé soit un SPAM ou non. mailfilter.crm a donné des résultats tout à fait remarquables par rapport aux filtres bayesiens habituels.

La base de connaissances sur les SPAM est locale au serveur. Elle n'est pas centralisée[5].

Pour des raisons de confiance limitée, nous utilisons uniquement les RBL[6] au sein de Spamassassin. Si l'expéditeur est listé dans l'une d'elles, le mail n'est pas abandonné, mais son score devient fortement majoré.

Problèmes rencontrés

Nous avions sous-estimé la taille des tables d'expressions de CRM114. Nous avons atteint la limite de la table en quelques mois, ce qui empêchait tout nouvel apprentissage. Il a suffi d'en créer de nouvelles plus grandes, et de réintégrer le contenu des anciennes tables. L'apprentissage a ainsi pu être conservé.

Au niveau de l'anti-SPAM, la solution actuelle n'est qu'une solution parmi d'autres et bien que très efficace, elle n'est pas optimisée :

- CRM114 est relativement lent.
- Nos modifications d'amavis ne sont pas applicables avec Spamassassin 3.

Nous avons aussi constaté une moins bonne précision du traitement du SPAM depuis la fin de l'été 2005. La faute n'en incombe pas toujours directement à mailfilter ;

- Des erreurs humaines de corrections ont été commises : des courriers corrects ont été transmis par erreur comme apprentissage SPAM. CRM114 met alors un certain temps pour reconverger.
- Il y a eu des bugs dans nos scripts de traitement SPAM qui ont parfois oublié les mises à jour de CRM114 sur certains serveurs. Par exemple, 2 des 3 MX n'étaient plus corrigés pendant tout l'été 2005. La précision du jugement a commencé à dévier de façon sensible.
- Des problèmes récents d'empoisonnement du filtre anti-SPAM ont été constatés. Les SPAMS contiennent des mots ayant pour but d'empoisonner le filtre pour qu'un message puisse ne pas être considéré comme SPAM, parce qu'il reconnaît des mots qui lui semblent bons. Une phase d'apprentissage aura pour conséquence la considération, à tord, de ces mots comme SPAM.

Enfin, un effet de bord dû à l'unification des adresses : les adresses unifiées de type Prenom.Nom@univ-nantes.fr permettent de dériver facilement l'adresse Email à partir du nom de la personne, et donc de constituer facilement un fichier SPAM (sans parler de certains sites WWW de l'Université où les adresses électroniques apparaissent simplement en clair).

Évolutions futures

Il n'est pas certain que notre approche simpliste, uniquement basée sur l'étude des messages déjà reçus, soit encore suffisante dans six mois. L'actuelle recrudescence des SPAMS ne fait que remettre ce point en lumière.

De par la modularité du système, il est possible de rajouter assez facilement d'autres armes anti-SPAM sur les MX. Des techniques différentes, telles que le grey listing [12], agissent en amont du traitement de messages reçus. Ainsi, le nombre de SPAMs reçus et traités baisse. Nous allons tester cette solution. Notre choix devra être fait parmi les logiciels de grey listing existant pour Postfix et se portera sur un logiciel qui ne remettra pas en cause la stabilité actuelle du système.

En plus d'utiliser le grey listing, il est aussi possible d'améliorer les outils actuels de filtrage. Parmi les options possibles, on peut citer :

-Basculer sur spamassassin 3, et donc, modifier nos adaptations.
-Changer le poids des scores de mailfiter.crm
-Utiliser des traitements plus avancés de CRM114
-Utilisation d'une autre solution anti-SPAM centralisée (dspam) [13], en remplacement de spamassassin.

[5]Elle pourrait être synchronisée.

[6]Realtime Blackhole List

5.6 Format des boîtes aux lettres

Étant donné le nombre de boîtes aux lettres à gérer, il nous a paru tout à fait impossible de les gérer en format mbox[7], puisque celle-ci est parcourue à chaque consultation, ce qui dégrade très fortement les performances.

Aussi le format Maildir++ [14] a été choisi. Maildir++ est une structure de stockage basée sur Maildir, avec des extensions mineures pour supporter les quotas et les sous-dossiers. Les deux formats de stockage sont compatibles entre eux, ils associent systématiquement un message à un fichier.

5.7 Stockage des messages

Les serveurs de stockage reçoivent les courriers en provenance des MX et les stockent sur le disque dur, dans les boîtes des utilisateurs, au format Maildir++. La communication avec les MX se fait par SMTP et donc par Postfix.

Il agit ici également comme MDA puisqu'il va réellement déposer les emails. Pour complètement satisfaire à nos besoins, il a été nécessaire d'ajouter un patch à Postfix, afin de s'accommoder des quotas de messagerie, stockés dans LDAP. Il s'agit du patch VDA [15], qui permet à Postfix d'être compatible avec le format Maildir++.

Pour partager la charge, une division simple a été réalisée ; les machines sont actuellement au nombre de deux. La distribution des boîtes s'est faite par ordre alphabétique. Le premier serveur gère les utilisateurs dont le nom commence par les lettres de A à L. L'autre gère le reste.

La configuration de postfix est simple :

- n'accepter que les courriers en provenance des MX.
- vérifier dans LDAP la destination du mèl : si celle-ci est un nom de boîte, c'est la destination finale, le message est déposé. Si la destination est une autre adresse mèl, relayer le message vers cette adresse via les serveurs SMTP[8]. La destination peut être à la fois locale et distante, dans ce cas le courrier est dupliqué en local et relayé vers la (ou les) adresse(s) distante(s). Cette technique a servi pendant les migrations, pour assurer une double livraison.

Problèmes rencontrés

La séparation du stockage des messages sur deux machines devait permettre une meilleure répartition de la charge. Mais malheureusement, beaucoup de gros consommateurs se sont retrouvés sur le même serveur, entraînant un déséquilibre, la charge du premier étant le double de celle du second. Plusieurs solutions sont possibles.

Améliorations possibles

La plus simple consiste à rajouter des serveurs, et d'y migrer des utilisateurs. C'est techniquement assez simple, mais nécessite une coupure temporaire de ces serveurs pour redistribuer les données. Des changements sont aussi à faire dans les fiches LDAP des personnes concernées. Suivant le redécoupage choisi, l'opération peut être fastidieuse.

Il est aussi possible d'utiliser un système de fichier utilisable par plusieurs serveurs. Il est alors facile d'augmenter la puissance de traitement, mécaniquement, jusqu'à concurrence de la puissance de traitement des disques, qui constitue alors la limite. Nous expérimentons actuellement cette technique, et espérons la mettre en place rapidement sur la messagerie des étudiants.

Mais la modularité nous permet un changement plus radical du stockage. Il est possible de se passer d'un système traditionnel et de tout entreposer dans une base de données. C'est ce que permet Dbmail [16]. Les versions actuelles n'implémentent pas encore tout ce qui nous est nécessaire[9]. Mais cela reste à surveiller. Notre architecture découpée fait que seuls les serveurs centraux seraient impactés par ce changement. Il n'y aurait rien à changer en périphérie (MX, SMTP). Il est possible que cette solution puisse fonctionner sur plusieurs machines simultanément, et ainsi, résoudre nos problèmes de charge de façon élégante.

5.8 Système de fichiers sur les baies RAID

Le choix actuel de maildir++ implique des millions de petits fichiers stockés sur les baies de disques. Le choix du système de fichiers s'avère donc crucial. Quelques tests ont été menés, afin de sélectionner le plus à même de répondre à cette sollicitation.

Pour des raisons de fiabilité, seuls les systèmes de fichiers journalisés ont été retenus. En janvier 2004, un programme spécifique a été écrit pour tenter de simuler le comportement du serveur selon le profil d'utilisation que nous imaginions. Le noyau 2.4 a alors été choisi pour tester. EXT3 s'est révélé le grand perdant du test. Il a fini bon dernier[10]. JFS a également terminé loin derrière les deux concurrents en tête : XFS et Reiserfs 3.

Reiserfs 3 aurait pu être un bon choix, car il excelle dans la gestion des petits fichiers (temps de création et suppression très faibles) mais semble moins à l'aise pour gérer de multiples flux en parallèle. De plus, il consomme plus de puissance processeur, et c'est un point gênant pour ces machines prévues comme étant très chargées à ce niveau.

XFS est globalement plus homogène et moins gourmand en ce qui concerne le processeur. C'est lui qui a été choisi.

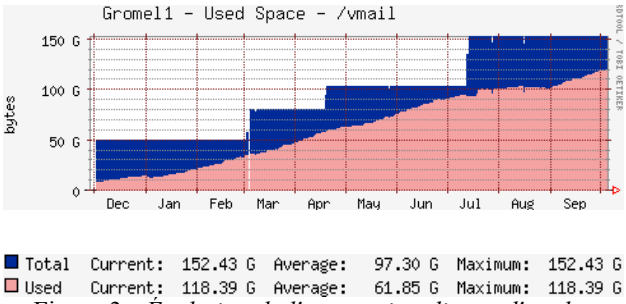

Figure 3 - *Évolution de l'occupation disque d'un des serveurs de stockage*

[7]Un seul fichier contient tous les messages.

[8]Équivalent d'un .forward. Cette fonctionnalité est normalement désactivée, et doit fait l'objet d'une motivation légitime pour être acceptée.

[9]LDAP n'est pas encore supporté dans la version stable.

[10]Dans les noyaux 2.6 récents, EXT3 a bénéficié d'optimisations importantes qui lui permettent d'avoir des performances plus décentes.

Après un an d'exploitation, l'espace consommé nous a surpris. Nous pensions avoir plus de données à gérer, mais il s'avère qu'XFS exploite efficacement son espace de fichiers. La fragmentation reste en dessous de 1% pour 1.6 millions de fichiers. Leur taille moyenne est faible, quelques centaines d'octets, comme prévu. Les temps de réponse sont très bons, les accès aux boîtes pratiquement toujours immédiats, même en période chargée .

Un effet de la virtualisation du stockage peut être vu sur la figure 3 : par exemple, en juillet, un second LUN de 105Go a été rajouté dans le conteneur LVM, après que le premier LUN ait été consommé en trois étapes.

Problèmes rencontrés: performances du noyau Linux 2.6

Suite au basculement du noyau 2.4 au noyau 2.6 en juillet, de meilleures performances ont initialement été observées. Ce point est visible sur la figure 4.

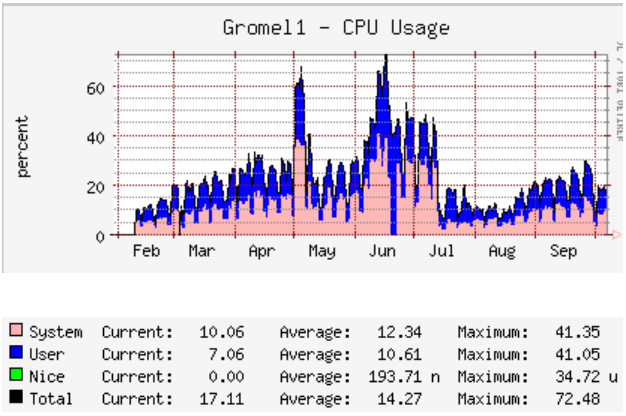

Figure 4 - *Charge d'un des serveurs de stockage*

Il est intéressant d'observer que le passage du noyau 2.4 au 2.6 a surtout réduit le temps consommé par les processus internes au système. Fort logiquement, le temps passé au niveau des processus utilisateurs est resté stable.

Néanmoins, après ce changement, il est apparu des temps de service nettement plus longs, sporadiquement. En fait, il est apparu que la machine ne parvenait plus, par moment, à suivre le rythme d'entrées/sorties demandé, bien qu'elle semblait généralement plus véloce. Le point troublant était que la machine ne paraissait pas spécialement chargée à ces moments. Les graphiques ne montraient rien de particulier, contredisant même nos observations. Après des recherches, il semble que nous ayons rencontré un bug d'un des ordonnanceurs d'entrées/sorties de Linux. Par défaut, le noyau 2.6 utilise l'ordonnanceur « anticipatory » [17], très rapide, mais également complexe. Il semble qu'en cas de très forte charge, il puisse provoquer une perte de performance importante. C'est ce que nous avons observé.

Les noyaux 2.6 pouvant changer d'ordonnanceur à chaud, le scheduler « deadline », similaire à celui utilisé dans les noyaux 2.4 a été testé ; les performances sont immédiatement revenues à la normale.

Ce bug a probablement été corrigé début septembre 2005. Un autre ordonnanceur, CFQ[11] v3 a aussi été intégré récemment et sera testé. Il promet un partage équitable des ressources d'entrées/sorties.

Enfin, des petites optimisations sur le système de fichier XFS[12] ont permis de descendre encore la charge du serveur a des niveaux très satisfaisants.

Évolutions futures : système de fichier cluster

XFS est un système de fichier conçu pour être utilisé sur une seule et unique machine (tout comme ses concurrents EXT3, reiserfs, etc...). C'est la raison initiale de la domiciliation des boîtes sur deux serveurs distincts. Cette stratégie est aisée à mettre en œuvre, mais ne permet pas une évolution simple de la capacité de traitement. Rajouter un serveur est problématique; il faut reloger les boîtes.

NFS permet une utilisation d'un système de fichiers sur plusieurs serveurs. Il a été testé pendant deux mois mais a présenté deux problèmes distincts :

- le problème de redondance se déplace ; si les serveurs de stockage du courrier peuvent alors être redondés, ce n'est pas le cas du serveur NFS qui reste unique.
- Les problèmes de charge sur le serveur NFS seront pratiquement les mêmes que sur celui de stockage[13], et le problème reste entier.
- Les performances ne sont pas adaptées à un nombre important de petites entrées/sorties. Le serveur de stockage (alors client NFS) ne peut suivre le rythme.

La solution NFS a vite été abandonnée. D'autre techniques de haute disponibilité, comme l'utilisation d'heartbeat [18] avec un serveur de secours dédié a été rapidement testé, puis abandonné. En effet, le risque de monter deux fois le système[14] de fichiers est non nul. EVMS dispose d'un plugin heartbeat qui a pour but de prévenir ce genre de problèmes, mais nous n'avons pas poussé plus loin notre expérimentation parce que cela n'apportait pas de réponse à notre problème de performance.

Notre espoir se porte désormais sur les systèmes de fichier cluster. Ceux-ci ne nous avaient pas paru convenir au moment de la mise en place (OpenGFS avait été testé). De nouvelles solutions (OCFS2 [19], GFS2 [20]) arrivent, et sont actuellement testées.

5.9 Mise à disposition des messages

Les serveurs de stockage mettent également les messages à disposition des utilisateurs.

Le CRI préconise fortement l'utilisation d'IMAPS pour plusieurs raisons : avec IMAP, les courriers restent stockés sur notre infrastructure et sont donc sauvegardés toutes les nuits. IMAPS ajoute une sécurisation des échanges, ce qui ne peut être que recommandable. Nous ne pouvons cependant pas forcer ce choix, de nombreux utilisateurs souhaitant ne pas changer leur outil habituel, qui ne gère pas forcément IMAP/S. Aussi, les protocoles POP et IMAP, et leur variante sécurisée, POPS et IMAPS sont proposés aux utilisateurs en interne.

[11] completely fair queueing

[12] le volume est monté ainsi : /dev/evms/GROMEL /vmail xfs defaults, sync, noatime, nodiratime, ihashsize=64433

[13] En simplifiant, en enlevant une partie de la charge utilisateur de la figure 4

[14] et donc, le corrompre de façon sure.

La figure 5 représente les moyennes de connexions d'un des serveurs POP et IMAP. La tendance à migrer vers IMAP est visible. Un temps évalué pour rendre ce service, et précédé d'une bonne réputation, cyrus [21] n'a cependant pas été retenu. En effet son utilisation intensive de Berkeley DB posait des problèmes lors des snapshots pour les sauvegardes, via EVMS. Cyrus est également déconseillé dans une configuration avec disques partagés.

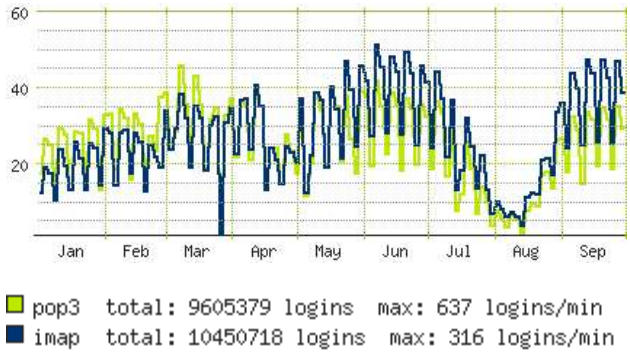

Figure 5 - *Évolution des consultations POP et IMAP*

Aussi le choix s'est porté sur le logiciel « courier » [22], plus simple. Courier ne gère pas d'index, ce qui facilite sa maintenance et sa sauvegarde, mais pourrait ralentir son fonctionnement. Les performances s'avèrent cependant tout à fait satisfaisantes. Courier est capable de s'interfacer avec le moniteur de modification de fichiers (FAM)[15], ce qui lui permet de consommer peu de puissance.

Les boîtes étant réparties sur deux serveurs distincts, se pose le problème de la configuration du client de messagerie. Il est plus simple et cohérent d'avoir tous les postes de l'université configurés de façon similaire.

La solution est d'utiliser un proxy POP et IMAP.

Évolutions futures

Avec un système de fichiers centralisé, les fonctions de stockage et de consultation de message pourraient être gérées indépendamment.

D'autres logiciels, tels Dovecot [23] ou Dbmail, existent et sont actuellement activement développés. Ils ont pour but d'être plus performants que courier et cyrus. Pour l'instant, Dovecot n'implémente pas encore le système de quota dont nous avons besoin, et dbmail ne s'interface pas avec LDAP. Courrier donnant satisfaction, nous n'envisageons pas de changement pour l'instant.

5.10 Proxies Pop et Imap

Le logiciel Perdition [24] est un proxy/démultiplexeur POP et IMAP. Ce programme très léger et très rapide permet d'utiliser une adresse unique pour consulter plusieurs serveurs IMAP. Lorsqu'un client se connecte et s'identifie sur un serveur perdition, celui-ci utilise l'identifiant de l'utilisateur pour faire une interrogation LDAP et pouvoir rediriger la connexion vers le bon serveur IMAP ou POP.

Nous en avons déployé deux en zone privée pour la consultation des boîtes de l'intérieur de l'université en POP, POPS, IMAP et IMAPS et deux sur la zone publique à l'intention des populations nomades. Ici, le seul protocole IMAPS est autorisé, afin d'obtenir un niveau de sécurité satisfaisant. Il est possible d'utiliser ce serveur de l'intérieur de l'université, afin d'éviter aux personnes équipées d'ordinateurs portables de le reconfigurer en permanence.

Les performances et la stabilité de perdition en font un produit hautement recommandable.

Problèmes rencontrés : connexion SSL limitée

Nous n'assurons pas de connexion SSL sur les serveurs de stockage pour ne pas les surcharger. Ce sont les serveurs perdition qui assurent les connexions SSL entre le client et la zone de confiance que constitue le réseau local du CRI. La connexion entre les serveurs perdition et les serveurs POP et IMAP reste donc en clair. Ce problème sera réglé avec la mise en cluster des serveurs de stockage, la charge des serveurs ne devenant plus cruciale.

5.11 Envoi et renvoi de messages

La partie concernant l'expédition du courrier de l'intérieur de l'université se compose de plusieurs sous-services.

Le service SMTP d'envoi principal est uniquement accessible en adressage privé. Pour transmettre le message à l'extérieur, une translation d'adresses en sortie est effectuée, ce qui procure un bon niveau de sécurité.

Ce service a une configuration logicielle basique:

- Postfix
- amavisd-new avec uniquement l'antivirus. L'anti-SPAM est désactivé.

Amavisd-new est configuré pour interdire tout attachement avec une double extension, tous .exe, .bat, .com, .scr, ainsi que les courriers multi-parties. À la différence des MX, lors de l'interception d'une pièce attachée bannie, une alerte explicative et en français est envoyée à l'expéditeur.

Le nombre de personnels de l'Université équipés d'un ordinateur personnel connecté à Internet ne cessant de croître, nous avons trouvé judicieux la mise en place d'un service SMTP sécurisé et authentifié visible et utilisable de l'extérieur de l'université. Ce service repose sur TLS [25].

Enfin, pour bien séparer les différentes tâches afférentes à la gestion du courrier électronique, un autre service SMTP a été mis en place : les serveurs poubelles. Ils traitent les messages en erreur. Ceux-ci proviennent essentiellement des anciens serveurs de messagerie, qui ne peuvent déposer un message que nous avons laissé passer (cf § 5.4).

Le rôle de ces serveurs est simple: éviter que les MX ayant pour unique tâche la réception de courriers n'aient à en renvoyer. Ces messages d'erreurs étant souvent eux même invalides (SPAMS avec adresse forgée), ils engendrent un encombrement des queues d'envoi, et des traitements inutiles sur les MX, par ailleurs bien occupés. Ce service est donc simplement utilisé par les MX pour déléguer le renvoi de tous les courriers d'erreurs vers l'expéditeur.

Problèmes rencontrés : réponses mal dirigées

En mettant en place ce mécanisme, nous n'avions pas pensé que certains anti-SPAM effectuent une vérification de la provenance des mails d'erreurs. Le serveur envoyant le mail d'erreur n'est pas celui annoncé dans l'erreur.

[15]File Alteration Monitor, ou son descendant récent, GAMIN

D'une manière générale, ces serveurs sont utiles tant que la vérification de l'existence d'un utilisateur ne pourra être complète. En effet, incapables d'effectuer cette vérification sur certains domaines, nous acceptons temporairement le message, avant éventuellement de le rejeter plus tard. Malheureusement, si le message possède une adresse source forgée, nous ne sommes plus en connexion SMTP avec le serveur qui l'a réellement envoyé.

Le message est donc renvoyé au champ 'from:' du message original, qui est alors très certainement un faux. Au mieux, ce message n'aboutira pas, et sera supprimé au bout de 5 jours. Au pire, il ira remplir la boîte d'un innocent qui n'en demandait pas tant...

Ces serveurs doivent disparaître dès que possible.

5.12 Webmail

Le logiciel retenu est Squirrelmail [26]. Il avait déjà fait ses preuves dans les différentes UFR avant l'unification et avait été très sollicité par les utilisateurs. Son ergonomie et sa simplicité de déploiement l'ont favorisé par rapport à IMP.

6 Techniques de balance de charge et tolérance à la panne

La plupart des services ont été déployés en multiples exemplaires afin d'assurer une balance de charge, et aussi augmenter la disponibilité du système.

Les MX sont triplés avec le même poids dans le DNS assurant une redondance et disponibilité inhérente au protocole.

Pour le reste des services, nous avons utilisé la classique technique du round-robin DNS [27] consistant à assigner plusieurs adresses IP à un nom pleinement qualifié unique. Les connexions se font alors à tour de rôle sur l'un ou l'autre des serveurs. Sur de nombreuses connexions, la balance de charge est effective.

Nous l'avons utilisée sur les services suivants.

- LDAP : Trois serveurs.
- Perdition : Deux serveurs en DMZ, autant en zone privée. Eux-même assurent une balance de charge **statique** sur les deux serveurs de stockage.
- SMTP et Serveurs poubelles : deux serveurs pour chaque fonction en zone privée.

Cette technique de Round-Robin DNS reste néanmoins naïve et perfectible. Elle n'assure qu'une balance de charge, parfois mise en défaut par des DNS locaux. De plus ce n'est pas vraiment une technique de haute disponibilité. En effet, si un des serveurs tombe en panne, le DNS pointe toujours sur la machine en panne. Il a rapidement été vu que certains clients se sortaient assez bien du problème (si un des serveurs ne répond pas, le client tente l'autre). Mais d'autres clients partent en dépassement de temps, induisant des lenteurs, voire des pertes de service. Enfin, le problème est pire avec le serveur encore en fonctionnement, mais avec son service ne répondant plus ou mal ; Le fonctionnement est alors dégradé ou incorrect. Après plusieurs mois de ce fonctionnement, il est évident que cette technique est insuffisante.

6.1 Éléments non redondants.

La technique ci-dessus n'apporte donc qu'une forme primitive de redondance. De plus, elle n'a pu être déployée partout. Il a été discuté plus haut pourquoi les serveurs de stockage ne sont pas redondés. Le webmail du personnel est également pour l'instant une machine unique, également pour des raisons techniques (stockage des préférences).

Évolutions en cours

Lors du récent développement de la messagerie unifiée pour les étudiants, nous nous sommes permis l'utilisation de solutions un peu plus audacieuses sans pour autant prendre beaucoup de risques, et avons ainsi pu résoudre quelques problèmes précédemment invoqués.

La technique de round robin DNS a été remplacée par des clusters Ultramonkey [28] implémentant vraiment une haute disponibilité. Un cluster Ultramonkey est constitué de trois logiciels fonctionnant ensemble :

- Une balance de charge, gérée directement par les couches réseau du noyau Linux : LVS [29]. Une adresse IP virtuelle est donnée pour un ensemble de vrais serveurs effectuant le travail. Le serveur implantant ce mécanisme est appelé directeur.
- Une haute disponibilité des directeurs : deux machines sont aptes à être directeur, une seule est active. Heartbeat surveille les deux et fait basculer la machine active sur la seconde, en cas de panne détectée.
- Un système de suivi des nœuds du cluster, assuré par ldirectord [30]. Il adapte dynamiquement les règles de LVS.

Ces clusters sont actuellement déployé pour la messagerie des étudiants et gèrent LDAP, Perdition et Webmail.

Intégré en cluster, Squirellmail fonctionne ainsi : au lieu de stocker les préférences de l'utilisateur et l'état des sessions PHP sur le disque local, ces informations sont stockées dans une base de données mysql, permettant ainsi leur centralisation. L'ajout d'un nœud supplémentaire dans le cluster s'en trouve facilité. Le seul problème restant est celui des attachements transférés sur le webmail avant l'envoi. Ceux-ci sont téléchargés et stockés sous la forme de fichiers temporaires. Ils sont locaux à une unique machine. Avec la balance de charge, rien n'indique que c'est la même machine qui réceptionné ces documents et les envoie. Ce problème est actuellement réglé par la gestion d'une persistance de connexion de plusieurs minutes sur le cluster, ce qui nuit à une bonne balance de charge. La meilleure solution semble là aussi un système de fichiers centralisé.

6.2 En cas de panne d'un serveur de stockage

Pour le moment, même en cas de panne sur l'un d'entre eux ou sur les deux, un certain niveau de fiabilité est conservé. La consultation des messages est momentanément impossible mais aucun message n'est perdu, ce qui est le plus critique, puisque le courrier continue à arriver sur l'université via les MX et est stocké temporairement en local sur ces serveurs en attendant de pouvoir être déposé.

6.3 Délocalisation

Un objectif a plus long terme sera de déménager physiquement une partie du système afin de minimiser l'impact d'une catastrophe locale, et donc, augmenter encore la disponibilité du système. Notre découpage permet ce genre d'opération, et fin 2005, le SAN couvrira l'intégralité des pôles de l'université, ce qui augmentera l'étendue géographique des clusters.

7 Phases de migration

Il n'était pas question de changer brutalement de système de courrier électronique. Des phases de migration ont été planifiées, ensemble par ensemble. (L'université de Nantes compte une vingtaine d'UFR, ce sont autant de phases de migration). Elles ont impliqué de nombreux correspondants du CRI dans chaque partie de l'université, et ont représenté un travail important.

Les phases de migration ont été programmées ainsi :

- Validation préalable des comptes de tous les utilisateurs et ajout des alias mails dans les LDAPs du personnel. Cette étape a été la plus pénible pour les correspondants des sites, puisque les fiches LDAP ont souvent du être vérifiées et mises à jour.
- Mise en place d'une double livraison sur le nouveau ET sur l'ancien système pour assurer une migration en douceur, UFR par UFR. Les MX acceptaient alors tout mail pour le domaine en cours de migration, tous les comptes n'étant pas encore présents dans la base LDAP.
- Migration progressive des utilisateurs sur le nouveau système avec désactivation de la double livraison et arrêt de l'ancien serveur.
- Activation, lorsque cela était possible de la vérification des adresses pour ce domaine, cf § 5.4.

Problèmes humains

Certains laboratoires n'avaient pas pris le soin de vérifier les entrées LDAP les concernant, bien que l'annuaire ait été pourtant déployé depuis longtemps. La migration vers la messagerie unifiée les a obligés à faire ce travail, souvent en urgence, ce qui a engendré des tensions.

La communication autour du nouveau système n'a pas bien fonctionné. Il y a eu confusion quant à la validité des anciennes adresses en @ufr.univ-nantes.fr (par ex. sciences.univ-nantes.fr ou iut-nantes.univ-nantes.fr). Des bruits ont couru sur la fin de ces adresses alors que nous maintenions cette compatibilité. Cette annonce a soulevé une certaine réticence de la part des utilisateurs qui ne voulaient pas se défaire de leurs anciennes adresses.

8 Développements spécifiques

Notre solution est constituée d'outils open source mais des développements supplémentaires ont dû être engagés.

8.1 Dyna

Dyna est une application développée à l'université, permettant la création d'une base de données servant à la génération d'annuaires LDAP. Dyna existait avant le projet de courrier unifié. Il a été étendu à cette occasion.

Dyna fonctionne en intégrant une partie des données fournie directement par la base de données de la division des ressources humaines de l'université (Harpège, et maintenant géode pour les étudiants). Une autre partie doit être entrée manuellement par les administrateurs, par exemple, les alias mail, les mots de passe. Le tout est intégré dans une base de données.

Dyna est doté d'une interface web permettant l'ajout de nouveaux comptes. A chaque modification, Dyna resynchronise les serveurs LDAP primaires avec les nouvelles données entrées. Il a effectivement la possibilité de gérer plusieurs modèles d'annuaires à partir d'une base de données unique. SUPANN [31] est, par exemple, un autre annuaire LDAP, d'une structure différente de la nôtre, qui est généré automatiquement et en continu, parallèlement aux autres annuaires déployés. De ce point de vue, les annuaires LDAP ne sont qu'une vue partielle de la base de données globale de l'université.

L'annuaire LDAP maître ainsi généré n'est pas directement utilisé par les clients : il ne fait que se répliquer via les mécanismes LDAP standards, sur trois serveurs qui sont, eux, accessibles aux clients.

8.2 Traitement anti-SPAM et anti-virus

Amavisd-new intègre spamassassin. Pour changer le filtre à apprentissage de spamassassin et le remplacer par mailfilter.crm, il a fallu faire des adaptations.

intégration

mailfilter.crm est un programme CRM114. CRM114 n'est, en tant que tel, pas un filtre de messagerie mais un langage de programmation spécialisé dans la manipulation de chaînes de caractères et la discrimination d'expressions.

Ce langage vient avec quelques scripts déjà écrits, dont mailfilter.crm, qui est un filtre de quelques centaines de lignes, intégrant néanmoins tout ce qui est nécessaire pour le traitement du SPAM. Ce script est court, il s'appuie sur la capacité de discrimination intégrée dans le langage, capable de traiter des chaînes via des filtres markoviens et, plus récemment, par d'autres techniques plus précises.

L'anti-SPAM se fait donc via deux moyens : les règles Spamassassin et mailfilter. Or amavisd-new n'est pas prévu pour y faire appel. Nous avons donc modifié amavis de telle sorte que celui-ci considère mailfilter comme un test supplémentaire de SpamAssassin.

Correction modérée

Afin de rendre le système le plus efficace possible, nous voulions pouvoir laisser aux utilisateurs finaux le soin de corriger les avis du filtre à apprentissage en cas de mauvaise décision, mais uniquement après modération. Il a été développé dans ce but un jeu de scripts PERL permettant une correction de CRM114 en toute simplicité : lorsqu'un mail n'a pas été correctement détecté par le filtre anti-SPAM, l'utilisateur l'ayant reçu peut le soumettre pour correction en l'envoyant en pièce jointe à une adresse électronique spéciale, connectée à un script PERL qui va déterminer s'il est corrigible. Dans ce cas, le mail en pièce

jointe est automatiquement extrait, reconstitué et transféré vers une boîte de modération, accessible à quelques administrateurs. Ceux-ci n'ont alors qu'à vérifier rapidement si la soumission n'est pas une erreur. Ensuite, ce message est redirigé à une autre adresse spéciale, également connectée à un second script PERL qui soumettra la correction à CRM114. Celui-ci n'accepte les courriers que depuis quelques machines (celle des modérateurs). Il est possible de soumettre plus de 100 pièces attachées d'un coup aux scripts de correction, ce qui simplifie son traitement.

globalisation de la correction

Le système mis en place est globalisé, les règles sont les mêmes pour tous. Les soumissions des usagers sont, dans 90% des cas, indiscutables, et profitent donc rapidement à la communauté. Il reste des cas limites, tels que des publicités d'un de nos fournisseurs, ou des publi-informations techniques, émanant de grandes sociétés ou cabinets spécialisés. Ces messages intéressent certaines personnes, bien qu'ils aient été abonnés sans leur demander leur avis. Techniquement, il s'agit donc de SPAM. Certaines personnes se plaignent donc, de façon justifiée. Si l'apprentissage en tant que SPAM est fait, une autre moitié des personnes va se plaindre des messages faussement classifiés comme SPAM. Hormis une technique complémentaire de traitement (filtres personnalisés via l'outil de messagerie, ou outil anti-SPAM à apprentissage personnel, tel celui de mozilla ou thunderbird...), nous ne voyons actuellement pas d'autres solutions simples pour ces cas limites.

Remontée d'alerte en cas de virus local :

Il nous a semblé intéressant que lorsqu'un courrier électronique contenant un virus est arrêté, amavis puisse détecter si le courrier provient d'une machine de l'université (même si l'adresse de la source est une adresse privée) et prévienne l'administrateur du site en question. Un petit script a été développé dans ce sens.

9 Conclusion

Le système est désormais en production depuis un an et malgré quelques heures d'interruption de service principalement liées à des problèmes électriques, plus de 19 millions de courriers électroniques ont été traités au cours de cette année. La progression du nombre de courriers traités se poursuit, quelques chiffres et graphiques représentatifs de l'activité du système peuvent être consultés sur le site du CRIUN [32].

Le système est fiable et répond à nos attentes. L'idée de tout diviser nous a réellement permis de gagner un temps précieux dans la gestion du parc de serveurs. De même, après quelques pannes ou erreurs de configuration, nous apprécions fortement l'indépendance de chaque serveur, qui rend le système robuste. L'impact de ces dysfonctionnements a été très faible. Après un an d'exploitation, nous voyons néanmoins les limitations de cette première solution, essentiellement au niveau de la disponibilité des services. La mise en place de la messagerie unifiée pour les étudiants nous a permis de tester et déployer de nouvelles techniques, comme les clusters de machines, et le déploiement de systèmes de fichiers centralisés. La mise en place semblant probante, ces nouveaux mécanismes seront intégrés progressivement dans le courrier unifié du personnel.

Bibliographie

[1] qmail-LDAP v3 : http://www.qmail-ldap.org/

[2] Le projet Linux-Vserver : http://linux-vserver.org/

[3] EVMS : http://evms.sourceforge.net/

[4] Device mapper : http://sources.redhat.com/dm/

[5] LVM2 : http://sources.redhat.com/lvm2/

[6] OpenLDAP : http://www.openldap.org/

[7] The Postfix home page: http://www.postfix.org/

[8] Amavisd-new : http://www.ijs.si/software/amavisd/

[9] Clam antivirus : http://www.clamav.net/

[10] The Apache SpamAssassin project : http://spamassassin.apache.org/

[11] CRM114, the Controllable Regex Mutilator : http://crm114.sourceforge.net/

[12] http://www.greylisting.org/

[13] DSPAM : http://dspam.nuclearelephant.com/

[14] Maildir++: http://inter7.com/courierimap/README.maildirquota.html

[15] Postfix VDA : http://web.onda.com.br/nadal/

[16] Dbmail : http://www.dbmail.org/

[17] I/O schedulers : Enhancements to Linux I/O scheduling, dans *vol 2, proceedings Linux symposium 2005*, ottawa, p 175 – 191

[18] Heartbeat : http://www.linux-ha.org/

[19] OCFS2 : http://oss.oracle.com/projects/ocfs2/

[20] GFS/GFS2 : http://sources.redhat.com/cluster/gfs/

[21] Project Cyrus : http://asg.web.cmu.edu/cyrus/

[22] Courier mail server : http://www.courier-mta.org/

[23] Dovecot : http://dovecot.org/

[24] Perdition : http://www.vergenet.net/linux/perdition/

[25] T. Dierks, C. Allen : *RFC 2246* : The TLS protocol version 1.0, Janvier 1999

[26] Squirellmail : http://www.squirrelmail.org/

[27] Paul Albitz & Cricket liu : *DNS et BIND, 3ème édition*, 1999, O'reilly, pages 252-253.

[28] Ultramonkey : http://www.ultramonkey.org/

[29] LVS : http://www.linuxvirtualserver.org/

[30] Ldirectord : http://www.vergenet.net/linux/ldirectord/

[31] http://www.cru.fr/ldap/supann/

[32] http://www.cri.univ-nantes.fr/JRES2005