Introduction à qmail

Djalil Chafaï
Club des Utilisateurs de Linux Toulousains

Introduction à qmail

par Djalil Chafaï

1998-12-21 v0.4

Je présente ici la messagerie *qmail*, une réelle alternative à *sendmail*.

qmail est déjà très bien documenté, mais les nouveaux venus sont souvent déconcertés. Ce document est écrit pour eux. Il est centré autour de *Linux* mais demeure valable en grande partie pour la majorité des *Unix*.

Le but est avant tout de faire comprendre le fonctionnement de la messagerie *qmail* plutôt que de donner des recettes toutes faites.

Le texte n'est pas encore au point et certaines sections ne sont pas encore rédigées, aussi, n'hésitez surtout pas à me faire part de vos critiques ou suggestions pouvant le faire évoluer à l'adresse dchafai@mail.dotcom.fr. Vous pourrez ainsi en faire profiter tout le monde.

Table des matières

Chapitre 1. Généralités

Voici comment Daniel J. Bernstein, l'auteur de *qmail*, nous présente son programme :

«qmail is a secure, reliable, efficient, simple message transfer agent. It is meant as a replacement for the entire sendmail-binmail system on typical Internet-connected UNIX hosts».

Le but de ce document est de présenter le fonctionnement du système *qmail*. Cependant, si vous recherchez une configuration «clé en main» pour commencer, vous trouverez peut être votre bonheur dans la *section sur les configuration types*

Ce document traite de *qmail* 1.01. Toutefois, ce qui est dit est encore valable pour *qmail* 1.03, la dernière version à l'heure où j'écrit ces lignes. Certaines nouvelles fonctionnalités ne sont donc pas abordées ici mais le seront je l'espère dans les prochaines versions de ce document. Je pense par exemple au protocole QMQP.

Pourquoi choisir qmail

sendmail et encore moins son petit remplaçant *smail* ne brillent par leurs performances, sans parler des innombrables trous de sécurité (certes moins nombreux pour *smail*) dûs à la fois à leur structuration et à la complexité de leur tâche. Pour couronner le tout, la configuration de *sendmail* n'est pas des plus intuitives.

Pour ne rien vous cacher, les détracteurs de *qmail* lui reprochent d'être moins «tout terrain» que son grand rival *sendmail*, ce qui est vrai à mon avis, en tout les cas pour l'instant. La polyvalence vient souvent avec le temps. *sendmail* est de plus un standard de fait, très répandu et très documenté, ce qui plaide en sa faveur. Certains n'aiment pas du tout le grand nombre de binaires et de fichiers de configuration de *qmail*, mais je pense que cela est avant tout histore de goût et d'habitudes. Je trouve même que celà permet de clarifier les choses.

D'autre part, D.J. Bernstein, l'auteur de *qmail*, est assez critiqué, voir même traité de «con» (sic) par certains. On lui reproche sa façon arrogante de répondre sur les forums et les listes de diffusions ainsi qu'un dénigrement un peu systématique de *sendmail*.

Il n'en demeure pas moins que l'oeuvre est plus importante que son auteur. *qmail* est un exellent logiciel qui vous apportera une grande satisfaction. Ses supporters, dont je fais partie, le trouvent à la fois très sûr, facilement configurable, bien documenté, performant et très orienté utilisateurs alors qu'ils reprochent à *sendmail* ses trous de sécurité, sa lourdeur et le coté abscons de sa configuration. Mais bien entendu, comme tout logiciel, *qmail* a ses limites et est l'oeuvre d'un être humain.

Pour être tout à fait honnète, la concurrence entre MTA sous *Unix* ne se limite pas à un affrontement entre *sendmail* et *qmail*. Le plus modeste *smail* peut s'avérer bien souvent suffisant. On entend également de plus en plus parler de *vmailer* et de *exim*.

Organisation de la messagerie qmail

À la différence de *sendmail*, *qmail* n'est pas monolithique. Le système *qmail* se compose de plusieurs programmes tournant sous des UID/GID différents et presque tous non nuls rendant difficile toute tentative d'intrusion. De plus, la gestion des chaînes de caractères dans *qmail* a été traitée avec beaucoup de soin pour éviter les problèmes de débordement, souvent à l'origine des trous de sécurité. Un groupe d'utilisateurs offre même un prix de 1000$ à toute personne qui trouverait un trou de sécurité dans *qmail* !

Aucun binaire ne porte le nom `qmail`. Par contre, beaucoup de binaires ont un nom qui commence par `qmail-`.

qmail est en fait un ensemble de binaires, tous situés par défaut dans `/var/qmail/bin/`, comprenant une partie MTA avec ses «facettes» SMTP avec *qmail-smtpd* et QMTP avec *qmail-qmtpd*, *qmail-inject* et *mailsubj* pour l'injection dans la file d'attente et une partie MDA *qmail-local* ainsi qu'un petit wrapper pour *sendmail*.

Les binaires du système *qmail* ont tous leur page de manuel. L'aide comprend également des pages de manuel supplémentaires décrivant certains formats de fichiers ou d'entêtes par exemple. En voici quelques unes : *qmail*, *qmail-limits*, *qmail-upgrade*, *qmail-header*, *addresses*, *envelopes*.

Le POP3 est disponible via le triplet *qmail-popup*, *checkpassword* et *qmail-pop3d*. Il existe des patches pour les gestionnaires IMAP et POP3 habituels permettant de les utiliser avec *qmail*, voir la page de *qmail* http://www.qmail.org/ pour plus de détails. Les messages sont placés dans la file d'attente par *qmail-queue*. La gestion de la file d'attente est effectuée par differents programmes dont *qmail-lspawn*, *qmail-clean* et *qmail-rspawn*, tous fils du daemon *qmail-send*. Le traçage via *syslog* peut être effectué par *splogger*. Voici un schéma montrant la structure de *qmail* lorsqu'il est chargé en mémoire :

```
UTILISATEUR PROGRAMME                 FONCTION
qmails      qmail-send                gestion sortie file d'attente
qmaill      \_ splogger qmail         enregistrement via syslog
qmailq      \_ qmail-clean            nettoyage de la file d'attente
qmailr      \_ qmail-rspawn           gestionnaire messages distants
root        \_ qmail-lspawn ./Mailbox gestionnaire messages locaux
```

La directive `./Mailbox` passée en paramètre à *qmail-lspawn* est la directive de livraison par défaut. Elle peut très être un tube (pipe) vers un MDA classique comme *procmail*. C'est par exemple le cas sur la *Debian GNU/Linux*.

Voici un petit schéma explicatif emprunté au fichier `INTERNALS` livré avec les sources qui contient une présentation de la structure de la file d'attente de *qmail* et sa gestion

```
 qmail-smtpd -- qmail-queue - qmail-send -- qmail-rspawn - qmail-remote
               /                    |      \
qmail-inject _/                qmail-clean  \_ qmail-lspawn - qmail-local
```

qmail-send gère les messages placés dans la file d'attente par *qmail-queue*. Il fait appel à *qmail-lspawn* lui même invoquant *qmail-local* et *qmail-rspawn* lui même invoquant *qmail-remote*. Voir leur page de manuel respectives pour les détails. Un **man qmail** donne une présentation globale de *qmail*.

Bien entendu, ces pages de manuel ne seront disponibles que lorsque *qmail* sera installé. Cependant, une version HTML de ces pages existe à l'URL http://qmail-docs.surfdirect.com.au/docs/qmail-1.03-man/.

Avant de passer à la suite, voici la structure standard du répertoire *qmail* `/var/qmail/`

```
REPERTOIRE          FONCTION
/var/qmail          répertoire principal
 \_ alias/          réceptacle des courriels «spéciaux» (root...)
 \_ bin/            binaires de la distribution qmail
 \_ control/        fichiers de configuration
 \_ man/            pages de manuel
 \_ queue/          la file d'attente
 \_ users/          plan d'adressage (super-fichier d'alias)
```

Sur *Debian GNU/linux*, la structure est un peu différente et plus conventionnelle. Les binaires sont répartis entre `/usr/sbin/` et `/usr/bin/`, les pages de manuel sont dans `/usr/man/`, le répertoire `/var/qmail/control/` devient `/etc/qmail/` et `/var/qmail/queue/` devient `/var/spool/qmail` via des liens symboliques.

Pour finir cette section, je vous signale l'existence de la «qmail big picture» à l'URL http://www.nrg4u.com/ qui donne une vue synthétique de l'achitecture de *qmail*.

Principales différences avec la messagerie sendmail

Le mieux serait de lire les fichiers `BLURB`, `BLURB1`, `BLURB2` et surtout `BLURB3` livrés avec les sources mais en voici un pâle résumé :

- plus de sécurité grâce à la structure «éclatée» de *qmail*, un minimum tourne en setuid ou en root;
- de très bonnes performances grâce entre autres choses à une gestion de file d'attente très rapide (numérotée par i-noeuds);
- configuration très simple via un ensemble de fichiers de contrôle et de variables d'environnement. Conformité aux RFC-822 et RFC-1123. Gestionnaire SMTP *qmail-smtpd* «8-bit clean», gestionnaire *qmail-qmtpd* implémentant le futur standard QMTP;
- disparition du fichier `/etc/aliases`, remplacé par la super base d'alias `/var/qmail/users/assign`. Disparition des `.forward` au profit des `.qmail`, à la fois plus puissants et plus simples;
- configuration et gestion du courrier très simple grâce aux fichiers `.qmail` interprétés par le MDA *qmail-local*; chaque utilisateur login controle toutes les adresses login-*. La livraison des courriels à destination de login-chaine est controlée par le fichier `~login/.qmail-chaine` ou à défaut par `~login/.qmail-default`;
- tout utilisateur peut créer et gérer ses propres listes de diffusions, soit avec *qlist*, soit avec le puissant *ezmlm*, écrit aussi par Dan Bernstein;

- nouveau format de boite au lettres nommé maildir, plus fiable que le format mbox et particulièrement utile lors de l'utilisation de NFS. Les utilitaires *maildir2mbox* et *maildirmake* sont fournis.

Avec *qmail*, nous dit Dan Bernstein, vous pouvez être sûr qu'un message entrant dans la file d'attente ne sera jamais perdu.

Chapitre 2. Compilation et installation

La licence qu'avait choisie Dan Bernstein pour *qmail* a longtemps interdit les distributions binaires sans son approbation préalable. *qmail* a donc été disponible uniquement sous forme d'archive source non modifiée. Une myriade de patchs d'origines diverses implémentant certaines fonctionnalités supplémentaires étaient alors disponibles.

À la fin de l'été 1998, Dan Bernstein a enfin décidé d'autoriser les distributions binaires sans approbation préalable mais sous certaines conditions bien précises, voir http://pobox.com/~djb/qmail/dist.html. Espérons que les distributions Linux proposeront à l'avenir d'installer *qmail* comme messagerie à l'installation, comme elles le font déjà pour *smail* et *sendmail*.

À l'heure actuelle, je ne connais pas de distribution *Linux* qui fournisse un paquetage binaire de *qmail*. Il va donc falloir compiler les sources.

Quelque soit la méthode que vous utilisez pour l'installation, il vous faudra ensuite faire très attention aux droits d'accès aux répertoires personnels des utilisateurs. En effet, le MDA *qmail-local* refusera de livrer le courrier à un utilisateur dont le répertoire personnel est autorisé en écriture pour le groupe ou pour tout le monde. Idem pour le bit «sticky». La même remarque s'applique aux fichiers `.qmail*`.

Par paquet source

Des paquets sources existent pour certaines distributions *Linux*, c'est le cas de la *Debian GNU/Linux* et la *RedHat Linux*. Cette méthode d'installation est recommendée dans la mesure ou les paquets en question sont censés être le mieux adaptés possible à votre distribution. D'un autre côté, vous trouverez rarement la dernière version en paquet source.

L'installation se déroule alors en trois étapes : installation du paquet source, compilation du paquet source et obtention du paquet binaire, installation du paquet binaire et suppression du paquet source.

Notons que pour l'instant, *qmail* n'écoute ni le port 25 (SMTP), ni le port 110 (POP3). La mise en place du support de ces protocoles fait justement l'objet du Chapitre 3.

Sans paquet source

Si aucun paquet source n'existe pour votre distribution, vous devez récupérer les sources de *qmail* sous forme d'archive. La suite de cette section ne concerne que la compilation à partir de l'archive.

Vous trouverez l'archive dans toutes les bonnes crèmeries FTP de la planète. Rien ne vous empèche non plus de la récupérer sur le site même de *qmail* http://www.qmail.org/ ou l'un de ses nombreux miroirs.

Récupérer et décompresser l'archive est un jeu d'enfant. Il convient ensuite de lire le fichier `INSTALL` et d'exécuter les étapes qui y sont décrites pas à pas.

La première chose à faire avant la compilation est sans doute d'éditer le fichier `conf-qmail` pour éventuellement modifier les réglages par défaut comme par exemple le répertoire *qmail*. `/var/qmail/`. La mise en place des utilisateurs et groupes de *qmail* est elle aussi capitale. Les réglages se font dans les fichiers `conf-users` et `conf-groups`. La distribution *Debian GNU/Linux* intègre déjà les utilisateurs/groupes *qmail*, l'installation n'en est que facilitée.

Une fois la compilation et l'installation faites, il vous faut mettre en place un script de lancement/arrêt automatique dans `/etc/init.d/` ou encore `/etc/rc.d/init.d/` selon votre distribution. Il en existe plusieurs, ils dépendent des spécificités de votre installation. En gros, ils font tous la même chose : lancer *qmail-start* avec de bons paramètres et récupérer son PID. Vous lirez avec profit la page de manuel de *qmail-start*.

Notons que pour l'instant, *qmail* n'écoute ni le port 25 (SMTP), ni le port 110 (POP3). La mise en place du support de ces protocoles fait justement l'objet de la section suivante.

Chapitre 3. Mise en place de SMTP, POP3, UCSPI et QMTP

Sont présentées ici les installations des programmes implémentant les protocoles SMTP, POP3 puis QMTP. Les installations sans *inetd* se feront grâce à UCSPI.

SMTP par qmail-smtpd via inetd

Il faut remplacer la ligne de SMTP dans `/etc/inetd.conf` par la ligne suivante et désactiver l'ancienne en la préfixant par une dièse :

```
smtp stream tcp nowait qmaild /var/qmail/bin/tcp-env \
tcp-env /var/qmail/bin/qmail-smtpd
```

Il est recommandé de désactiver (en la préfixant par une dièse si ce n'est pas déjà fait) la ligne COMSAT de ce même fichier. Réinitialiser enfin *inetd* en faisant un

```
root:bash$ kill -HUP $(pidof inetd)
```

Vous lirez avec profit la section *rcpthosts* du Chapitre 4 sur le fichier de contrôle `rcpthosts` ainsi que la section *Contrôle de relayage* du Chapitre 5 sur le contrôle de relayage.

POP3 par qmail-pop3d via inetd

Récupérer *checkpassword* sur http://pobox.com/~djb/checkpwd.html, le compiler puis l'installer dans `/usr/local/sbin/`. Je signale au passage que pour compiler avec les shadows sous Linux, vous serez sans doute amené à remplacer `SHADOWLIBS=-lshadow` par `SHADOWLIBS= -lcrypt` dans le `Makefile`.

Remplacer la ligne POP3 du fichier `/etc/inetd.conf` par celle-ci :

```
pop3 stream tcp nowait root /var/qmail/bin/qmail-popup qmail-popup \
votre.host.name /usr/local/sbin/checkpassword /var/qmail/bin/qmail-
pop3d Maildir
```

puis réinitialiser *inetd* par un

```
root:bash$ kill -HUP $(pidof inetd)
```

Faites attention, le nom du service POP3 peut être pop-3 et non pop3, regardez dans votre `/etc/services`.

UCSPI-TCP et TCPcontrol

UCSPI signifie Unix Client-Server Program Interface. Un outil UCSPI est un programme qui sait communiquer avec un certain medium de communication (par exple TCP). Il fournit de plus deux descripteurs de fichiers, l'un pour lire et l'autre pour écrire sur le medium. On peut alors lui associer un programme qui lira et écrira sur ces descripteurs sans se soucier du medium. Celà permet de créer à partir d'un programme UNIX quelconque une application client/serveur sur le medium en question. L'avantage est que le programme ainsi enveloppé ne se soucie plus du medium qu'il utilise, c'est désormais le problème de l'enveloppe UCSPI.

Les outils UCSPI sont de deux types : les clients et les serveurs

- Un client UCSPI se connecte à un serveur à un certain port et communique avec le programme qu'on lui spécifie via les descripteurs 6 et 7. Le client n'établit aucune connexion supplémentaire et meurt avec le programme.
- Un serveur UCSPI attend les connexions des clients sur un certain port et comunique avec le programme qu'on lui spécifie via les descripteurs 0 et 1. Il permet d'avoir plusieurs connexions simultanées.

Je vous conseille fortement la lecture du texte http://pobox.com/~djb/proto/ucspi.txt de Dan Bernstein dont je me suis largement inspiré.

Un exemple est *UCSPI-TCP* écrit par Dan Bernstein qui est constitué de *tcpclient* et *tcpserver*. En gros, on peut dire qu'ils permettent de faire pour un port et un programme particulier ce que fait *inetd* à la différence qu'ils sont très performants et paramétrables. Voici l'URL de *UCSPI-TCP* : http://pobox.com/~djb/ucspi-tcp.html.

UCSPI-TCP va se révéler très pratique dans le système *qmail*. Il perméttra de transformer les programmes *qmail-smtpd*, *qmail-qmtpd* et *qmail-pop3d* qui ne connaissent rien à TCP en de véritables serveurs sur certains ports TCP indépendament d'*inetd*. On gagnera ainsi en configurabilité et surtout en performances pures.

TCPcontrol quand à lui est un contrôleur d'accès écrit par Dan Bernstein destiné à être utilisé avec *UCSPI-TCP*. C'est ce que l'on appelle un «TCP wrapper». Voir http://pobox.com/~djb/tcpcontrol.html. Il est à *tcpserver* ce que *tcpd* est à *inetd*. Consultez sa documentation pour en savoir plus. Je n'ai pas utilisé *TCPcontrol* dans la suite. Vous pouvez lire avec profit la FAQ de *qmail* qui explique comment s'en servir pour le port SMTP.

SMTP par qmail-smtpd via UCSPI-TCP

Pour ce faire, se procurer *UCSPI-TCP* sur http://pobox.com/~djb/ucsp-tcp.html, compiler et installer les binaires dans `/usr/local/bin/`.

Ensuite, il faut préfixer la ligne SMTP de `/etc/inetd.conf` par une dièse et relancer le daemon *inetd* par un

```
root:bash$ kill -HUP $(pidof inetd)
```

Puis il faut ajouter les deux lignes suivantes à la section start du script `/etc/rc.d/init.d/qmail` :

```
/usr/local/bin/tcpserver -v -u 7770 -g 2108 0 smtp /var/qmail/bin/qmail-
smtpd \
   2>&1 | /var/qmail/bin/splogger smtpd 3 &;
echo $! > /var/lock/subsys/qmail-smtpd
```

(voir les pages de manuel pour les détails) et ces deux-ci dans la section stop :

```
kill `cat /var/lock/subsys/qmail-smtpd`
rm -f /var/lock/subsys/qmail-smtpd
```

Voilà, il ne reste plus qu'à stopper puis relancer *qmail*.

POP3 par qmail-pop3d via UCSPI-TCP

Il faut se procurer UCSPI-TPC sur http://pobox.com/~djb/ucsp-tcp.html, compiler puis installer les binaires dans `/usr/local/bin/`.

De même, récupérer *checkpassword* si besoin sur http://pobox.com/~djb/checkpwd.html, le compiler puis l'installer dans `/usr/local/sbin/`. Je signale au passage que pour compiler avec les shadows sous *Linux*, vous serez sans doute amené à remplacer `SHADOWLIBS=-lshadow` par `SHADOWLIBS= -lcrypt` dans le `Makefile`.

Ensuite, commenter la ligne POP3 dans `/etc/inetd.conf` en la préfixant par une dièse et réinitialiser *inetd* par un

```
root:bash$ kill -HUP $(pidof inetd)
```

Il ne reste plus qu'à ajouter les deux lignes suivantes à la fin de la partie start du fichier `/etc/rc.d/init.d/qmail` :

```
/usr/local/bin/tcpserver 0 pop3 /var/qmail/bin/qmail-
popup votre.host.name \
/usr/local/sbin/checkpassword /var/qmail/bin/qmail-pop3d Maildir &
echo $! > /var/lock/subsys/qmail-pop3d
```

et ces deux-ci au début de la section stop :

```
kill `cat /var/lock/subsys/qmail-pop3d`
rm -f /var/lock/subsys/qmail-pop3d
```

Faites attention, le nom du service POP3 peut être pop-3 et non pop3, regardez dans votre `/etc/services`. Il ne reste plus qu'à stopper puis relancer *qmail*.

QMTP, le petit dernier

C'est un nouveau protocole proposé par Dan Bernstein censé remplacer SMTP, il est associé au port TCP 209. Laissons à son auteur le soin de le présenter :

The Quick Mail Transfer Protocol (QMTP) is a replacement for the Simple Mail Transfer Protocol (SMTP). QMTP eliminates any need for end-of-line scanning between hosts with the same end-of-line convention. It features automatic pipelining and chunking, 8-bit transmission, prior declaration of the message size, and efficient batching. It is designed to be very easy to implement.

qmail bien sûr le supporte à travers le programme *qmail-qmtpd* qui se configure exactement comme *qmail-smtpd*.

Chapitre 4. Configuration : la base

Fichiers de contrôle

La configuration de *qmail* se fait presque entièrement via un ensemble de petits fichiers texte de «contrôle» tous situés dans le répertoire `/var/qmail/control/` ou son équivalent le cas échéant. Les différents programmes composant *qmail* lisent le contenu de ces fichiers et positionnent leurs variables en conséquence.

Des options par défaut sont prises pour chaque fichier inexistant. Par défaut, un seul de ces fichiers existe, il s'agit de `me` qui contient le nom qualifié de la machine.

Un **man qmail-control** donne une liste complète de tous ces fichiers, des programmes *qmail* qui les utilisent ainsi que des options prises par défaut. C'est très pratique.

Vous pouvez utiliser la commande **qmail-showctl** pour obtenir un rapport sur les différents fichiers de contrôle présents sur votre système *qmail* ainsi que les valeurs par défaut utilisées.

Il ne faut surtout pas oublier de demander à *qmail* de relire ses fichiers de contrôle après la modification de l'un d'entre eux. Pour ce faire, il suffit de faire un

```
root:bash$ kill -HUP $(pidof qmail-send)
```

aliasempty

Le premier fichier de contrôle que vous serez amené à modifier est sans doute `/var/qmail/control/aliasempty`. Il contient le nom de la BAL dans laquelle le MDA *qmail-local* livre le courriel lorsqu'il n'y a pas de `.qmail`. Sa valeur par défaut est `./Mailbox`.

Cette règle de livraison par défaut peut néanmoins être spécifiée en ligne de commande à *qmail-start* lors du lancement de *qmail*, voir la page de manuel. Elle sera transmise à *qmail-lspawn*.

Pour conserver la livraison par défaut dans `/var/spool/mail`, vous pouvez imiter le paquetage *qmail* de la *Debian GNU/Linux* : se servir du MDA *procmail* et lancer *qmail* par

```
qmail-start |/usr/local/sbin/qmail-procmail
```

Le script exécutable `/usr/local/sbin/qmail-procmail` contenant

```
#!/bin/sh

/var/qmail/bin/preline /usr/bin/procmail && exit 0

# Erreur EX_TEMPFAIL ?
[ $? = 75 ] && exit 111

# Sinon on renvoie une «Permanent Error»
exit 100
```

locals

Un autre fichier très important pour vous sera sans doute `/var/qmail/control/locals`, utilisé par *qmail-send*. Il contient les noms des hôtes considérés comme locaux, c'est à dire ceux pour lesquels les courriels que vous recevez et qui leurs sont destinés seront livrés sur votre machine. Par défaut, cette variable vaut la même valeur que le fichier de contrôle `me` (votre nom de machine). Un petit exemple valant mieux qu'un long discours, voici un exemple de `locals`

```
mondomaine.amoi
mamachine
mamachine.mondomaine.amoi
ftp
ftp.mondomaine.amoi
www
www.mondomaine.amoi
```

`/var/qmail/control/me` contenant

```
mamachine.mondomaine.amoi
```

Bien sûr, il faudra de plus que les champs MX dans le DNS de `ftp.mondomaine.amoi` et `www.mondomaine.amoi` pointent sur `mamachine.mondomaine.amoi`.

N'oubliez pas d'avertir *qmail-send* après toute modification du fichier `locals` en lui envoyant un signal HUP :

```
root:bash$ kill -HUP $(pidof qmail-send)
```

defaulthost et defaultdomain

Les deux fichiers de contrôle `/var/qmail/control/defaulthost` et `/var/qmail/control/defaultdomain` s'avèrent très utiles. Il permettent de spécifier la façon dont *qmail-inject* complète les adresses destinataires incomplètes. Ainsi, une adresse ne contenant pas de partie domaine sera complétée par `defaulthost` et une adresse dont la partie domaine ne contient pas de point sera complétée par `defaultdomain`. La valeur par défaut est la même que celle de `me`, ce qui ne correspond pas forcément à ce que vous souhaitez.

Les variables d'environnement `QMAILDEFAULTDOMAIN` et `QMAILDEFAULTHOST` ont priorité sur le contenu des fichiers de contrôles correspondants.

rcpthosts

Le fichier de contrôle `/var/qmail/control/rcpthosts` est utilisé par *qmail-smtpd*. Il permet de contrôler le relayage de courriel par la machine via SMTP.

Lorsque le fichier `rcpthosts` existe, *qmail-smtpd* refusera tout courriel dont la partie domaine de l'adresse destinataire ne figure pas dans `rcpthosts`. Par défaut, c'est à dire en l'abscence de `rcpthosts`, *qmail-smtpd* accepte tous les courriels.

Voici un exemple de `rcpthosts`:

```
.mondomaine.amoi
mondomaine.amoi
```

qmail-smtpd refusera ici tout courriel à destination d'une adresse extérieure au domaine `mondomaine.amoi`.

Il est bien entendu très important que `rcpthosts` autorise au moins les machines figurant dans le fichier de contrôle `locals` pour que vous puissiez recevoir les courriels qui leurs sont destinés par SMTP.

`rcpthosts` est ignoré si la variable d'environnement `RELAYCLIENT` est positionnée. Auquel cas, *qmail-smtpd* ajoute son contenu à l'adresse de destination des courriels qui transitent par lui.

Lisez la page de manuel de *qmail-smtpd* pour en savoir plus.

Le contrôle de relayage par `rcpthosts` est souvent associé à un contrôle au niveau TCP par un «wrapper» TCP. Voir la section *Contrôle de relayage* du Chapitre 5. Pour effectuer un rejet des courriels entrants par SMTP suivant l'expéditeur, reportez vous à la section *badmailfrom*.

badmailfrom

`badmailfrom` est très utile pour éviter le courriel entrant non sollicité. Il permet de spécifier les adresses d'expéditeurs interdits. *qmail-smtpd* refusera tout courriel provenant d'eux. Voici un exemple :

```
@polution.com
polueur@poubelle.com
```

qmail-smtpd refusera tout courriel provenant du domaine `polution.com` ou de l'adresse `polueur@poubelle.com`.

Pour faire du contrôle de relayage, se reportez à la section *Contrôle de relayage* du Chapitre 5 et la section *rcpthosts*.

smtproutes

Le fichier de contrôle `/var/qmail/control/smtproutes` permet de spécifier les relais SMTP que *qmail-remote* devra contacter pour router les courriels sortants. voir la page de manuel pour plus de détails. Voici un exemple :

```
premier.domaine:mail.premier.domaine
second.domaine:mail.second.domaine:24
:mail.autre.domaine
```

Les courriels à destination de `premier.domaine` seront transmis par SMTP au port TCP 25 de `mail.premier.domaine`, ceux à destination de `second.domaine` seront transmis par SMTP au port TCP 24 de `mail.second.domaine`. Quant aux autres courriels, ils seront transmis au port TCP 25 de `mail.autre.domaine`.

qmail-remote utilise aussi les fichiers de contrôle `helohost`, `timeoutconnect` et `timeoutremote`, voir la page de manuel de *qmail-remote*.

virtualdomains

Le fichier de contrôle `/var/qmail/control/virtualdomains` contient une liste de domaines virtuels, un par ligne. Un domaine virtuel est un domaine pour lequel la livraison se fera vers une seule adresse. Les domaines virtuels sont pris en compte par *qmail-send*. Par défaut, aucun domaine virtuel n'est défini. Voici un exemple de fichier `virtualdomains` :

```
mondomaine.amoi:
.mondomaine.amoi:
.uucp:alias-uucp
:alias-ppp
```

Dans cet exemple, les courriels à destination d'adresses dans le domaine `mondomaine.amoi` sont livrés normalement, pour celles se terminant par `.uucp`, la livraison a lieu à l'adresse `alias-uucp`, quant aux autres, ils sont livrés à l'adresse `alias-ppp`.

N'oubliez pas d'avertir *qmail-send* après toute modification du fichier `locals` en lui envoyant un signal HUP :

```
root$ kill -HUP pid_de_qmail-send
```

Allez voir la page de manuel de *qmail-send* pour plus de détails.

Et les autres ?

Je n'ai parlé que des principaux fichiers de contrôle, il y en a beaucoup d'autres. Je vous conseille de faire un «**man qmail-control**». Vous y trouverez une liste de tout les fichiers de configuration, les binaires qui les utilisent ainsi que les valeurs prises par défaut.

Livraison par qmail-local et utilisateur alias

Dans le système *qmail*, un utilisateur nommé `login` contrôle toutes les adresses courriel de la forme `login-*` où l'étoile représente une chaîne quelconque formée de caractères valides pour une adresse. Cette règle générale peut être altérée par la mise en place d'un fichier d'alias, voir la la section *Le fichier d'alias /var/qmail/users/assign* du Chapitre 5.

Pour chaque adresse `login-chaine`, l'utilisateur peut décider de créer le fichier `~login/.qmail-chaine` qui contiendra les instructions à suivre par *qmail-local*, le livreur de courriel (MDA) de *qmail*. Par convention, le fichier `~login/.qmail` correspond à l'adresse `login` tout court. De plus, le fichier spécial `~login/.qmail-default` lorsqu'il existe, joue le rôle de fichier par défaut.

En l'abscence des fichiers `~login/.qmail` et `~login/.qmail-default`, *qmail-local* livrera le courriel à destination de `~login` en suivant la règle de livraison par défaut spécifiée au lancement de *qmail-send*, voir la la section *aliasempty*. En général, cette livraison se fait dans la BAL au format mbox `~login/Mailbox` mais celà relève du choix de l'administrateur.

Il en va tout autrement pour les courriels à destination de `login-chaine`. *qmail-local* ne livrera pas un courriel à destination de `login-chaine` s'il ne trouve ni `~login/.qmail-chaine` ni `~login/.qmail-default` au moment de livrer. En fait, ce courriel sera intercepté par l'utilisateur spécial `alias` si son fichier `~alias/.qmail-default` existe et retournera à l'expéditeur sinon.

Les fichiers `~login/.qmail-*` ont tous la même syntaxe.

Remarques sur qmail-local

Je rappelle qu'il faut faire très attention aux droits d'accès aux répertoires personnels des utilisateurs. En effet, le MDA *qmail-local* refusera de livrer le courrier à un utilisateur dont le répertoire personnel est autorisé en écriture pour le groupe ou pour tout le monde. Idem pour le bit «sticky». La même remarque s'applique aux fichiers `.qmail*`. Voir la page de manuel *dot-qmail*.

Il faut aussi signaler que *qmail-local* convertit toutes les majuscules en minuscules dans les noms de fichiers `.qmail-*`. Il convertit aussi le caractère point en caractère deux-points et réciproquement. Ainsi, pour que le courriel à destination de `jojo.lapin` soit livré à l'utilisateur `jojo`, il suffit que jojo possède un `.qmail-jojo:lapin` dans son répertoire personnel `~jojo`.

Syntaxe des .qmail*

Plongeons-nous à présent dans le contenu de ces fameux fichiers `.qmail*`. Ces fichiers contiennent les instructions que doit suivre le MDA *qmail-local* pour livrer le courriel. Çelà peut rappeler vaguement le mécanisme des fichiers `.forward` du monde de *sendmail*. Voici un exemple :

```
# mon .qmail à moi
# Un pipe
|qbiff
# Un forward
&moi@ailleurs.laba
# Livraison dans une BAL maildir
./maildir/
```

```
# Livraison dans une BAL mbox
./mbox
```

On le voit, chaque ligne contient une et une seule instruction de livraison. Les lignes sont toutes analysées à chaque courriel livré par *qmail-local*. Elles peuvent contenir des commentaires préfixés par une dièse. Dans cet exemple, chaque courriel sera tubé dans *qbiff*, redirigé (forwarded) vers `moi@ailleurs.laba` et livré dans les boites aux lettres `~/maildir/` et `~/mailbox`.

- *qbiff* est un petit programme de notification d'arrivée de courriel livré avec *qmail*. Il annonce l'arrivée d'un courriel par un bip et un petit message sur le terminal;
- La directive de redirection (forwarding) se fait par une perluette. En fait on peut se passer de la perluette lorsque l'adresse courriel commence par une lettre ou un chiffre;
- Les nom de boites aux lettres commencent toujours par un point suivit immédiatement d'une barre de répertoire (`./`). Il se terminent toujours par une barre de répertoire pour les boites au format maildir et jamais pour les boites au format mbox. Les boites au format maildir doivent être créées avant d'être utilisées au moyen de l'utilitaire *maildirmake* qui fait partie de la distribution *qmail*;

qmail-local exécute les livraisons vers tubes de la forme

```
|commande
```

en écrivant le courriel dans l'entrée standard d'un

```
sh -c commande
```

exécuté dans le répertoire personnel du destinataire. L'entête du courriel ne contient alors pas les champs `Return-Path` et `Delivered-To` mais *qmail-local* passe beaucoup d'informations par variables d'environnement, par exemple `SENDER` et `RECIPIENT` (voir la page de manuel `qmail-command`). Pour les commandes qui ne ne prennent pas en compte ces variables, comme *procmail* ou *formail* par exemple, il suffit d'utiliser le binaire *preline* livré avec *qmail* qui insère les champs `Return-Path` et `Delivered-To` dans le message avant de l'écrire dans l'entrée standard de la commande. Par exemple :

```
|preline formail -A "Status: RO" | procmail
```

Je vous conseille fortement de faire un «**man dot-qmail**» pour en savoir plus sur la syntaxe et les subtilités du format des fichiers `.qmail`.

D'autres binaires peuvent vous être très utiles dans vos `.qmail*`. Jetez un coup d'oeil par exemple à la page de manuel de *condredirect* et à celle de *forward*.

Pour effectuer une modification sûre de vos fichiers `.qmail*`, Dan Bernstein recommande de positionner le bit «sticky» sur son répertoire personnel par un

```
chmod +t $HOME
```

pendant la durée de la modification. En effet, *qmail-local* refusera de livrer dans un répertoire dont le bit sticky est positionné. Une fois la modification effectuée, il vous suffit de désactiver le bit sticky de votre répertoire par un

```
chmod -t $HOME
```

Le plus simple est de se faire un alias pour son shell favori. Par exemple, pour *bash*, je rajoute la ligne :

```
alias edqm='chmod +t ~; && vi $1 && chmod -t ~;'
```

à mon `~/.bash_profile`.

Gestion de la file d'attente (à compléter)

La file d'attente de *qmail* est stockée dans `/var/qmail/queue/`. Elle est constituée de plusieurs répertoires. Chaque message est stocké dans un fichier dont le nom est un numéro d'i-noeud, ce qui assure son unicité. La structure de la file d'attente est expliquée dans le fichier `INTERNALS` livré avec les sources.

La numérotation par i-noeud rend malheureusement un peu problématique la restauration d'une file d'attente sauvegardée. Il existe cependant des utilitaires qui permettent de passer outre. Allez voir sur le site de *qmail*.

Les deux binaires *qmail-qstat* et *qmail-qread* permettent d'en connaitre son contenu. Ils font partie de la distribution *qmail* et ont donc leur page de manuel respectives.

queue-fix est un programme écrit par Eric Huss e-huss@netmeridian.com qui permet de vérifier l'intégrité de la file d'attente de *qmail* et de la réparer au besoin.

qmHandle est un script Perl écrit par Michele Beltrame mick@io.com. Il permet de lister en couleurs le contenu de la file d'attente de *qmail* ainsi que d'en supprimer des messages.

queue-fix et *qmHandle* ne font pas partie de la distribution *qmail*. Vous trouverez des pointeurs sur eux deux sur le site de *qmail*.

Listes de diffusion par qlist (à faire)

Chapitre 5. Configuration : pour aller plus loin

Le fichier d'alias /var/qmail/users/assign

Le système *qmail* peut posséder une liste d'alias compilée au format CDB stockée dans le fichier `/var/qmail/users/cdb` et dont la version ASCII est stockée dans le fichier `/var/qmail/users/assign`. La commande **qmail-newu** permet de transformer la version ASCII en version compilée CDB. Cette dernière est lue par *qmail-lspawn* avant l'appel à *qmail-local*.

La version ASCII ressemble à un fichier `passwd`. Voir la page de manuel *qmail-users* pour une description de son format. La commande **qmail-pw2u** permet d'ailleurs de fabriquer la base de données `/var/qmail/users/assign` à partir de votre fichier `/etc/passwd` :

```
root$ qmail-pw2u < /etc/passwd > /var/qmail/users/assign
root$ qmail-newu
```

Le fichier d'alias compilé est lu par *qmail-lspawn*. Ce dernier s'en sert pour déterminer l'utilisateur qui recevra la livraison d'un courriel par *qmail-local* en mode utilisateur. *qmail-local* se charge de la lecture et de l'interprétation des fichiers `.qmail*`. Il faut bien comprendre que la liste d'alias est lue par *qmail-lspawn* alors que les `.qmail*` sont lus par *qmail-local*. Les deux mécanismes ne sont pas mutuellement exclusifs mais successifs. L'un opère en UID 0 et l'autre en mode utilisateur.

Considérons par exemple un courriel à destination locale sortant de la file d'attente. Il est transmis à *qmail-lspawn* par son père le daemon *qmail-send*. *qmail-lspawn*, qui opère en UID 0, détermine en lisant le fichier d'alias compilé le login `login`, l'UID `uid`, le GID `gid` et le fichier `~login/.qmail*` en vertu desquels il faut faire la livraison par *qmail-send*. *qmail-send* est donc executé en UID `uid` et effectue la livraison conformément au fichier `~login/.qmail*` et aux règles expliquées dans la section *Livraison par qmail-local et utilisateur alias* du Chapitre 4.

```
 qmail-send
   \_ qmail-lspawn UID 0
        \_ qmail-local UID uid
```

Voici un exemple de fichier `/var/qmail/users/assign` :

```
=root:alias:70:65534:/var/qmail/alias:-:postmaster:
=abuse:alias:70:65534:/var/qmail/alias:-:postmaster:
=operator:alias:70:65534:/var/qmail/alias:-:postmaster:
=uucp:alias:70:65534:/var/qmail/alias:-:postmaster:
=anonymous:alias:70:65534:/var/qmail/alias:-:postmaster:
=hostmaster:alias:70:65534:/var/qmail/alias:-:postmaster:
=mailer-daemon:alias:70:65534:/var/qmail/alias:-:postmaster:
=manager:alias:70:65534:/var/qmail/alias:-:postmaster:
=warnings:alias:70:65534:/var/qmail/alias:-:postmaster:
=webmaster:alias:70:65534:/var/qmail/alias:-:postmaster:
=ftpmaster:alias:70:65534:/var/qmail/alias:-:postmaster:
=FaxMaster:alias:70:65534:/var/qmail/alias:-:postmaster:
```

```
+djalil:djay:1000:1000:/home/djay:::
.
```

Chaque ligne correspond à une directive de livraison. Voir la page de manuel `qmail-users` pour une description des champs. Les alias habituels sont tous redirigés vers `alias-postmaster`, à charge du `.qmail-postmaster` de router les messages vers un autre utilisateur. Le fichier doit toujours se terminer par une ligne ne contenant qu'un point. La dernière ligne demande à *qmail-lspawn* de livrer tout courriel à destination d'une adresse commençant par `djalil` vers `djay`.

La commande **qmail-getpw** permet d'obtenir la ligne du fichier d'alias correspondant à l'adresse courriel qu'on lui passe en argument.

Contrôle de relayage

Lisez pour commencer la section *rcpthosts* du Chapitre 4 sur le fichier de contrôle `rcpthosts`.

On associe souvent le contrôle de relayage par `rcpthosts` à un contrôle de connexion TCP par un «wrapper» TCP. Le cas typique est l'utilisation de *tcpd* via *inetd* pour lancer *qmail-smtpd*. Le contrôle TCP est alors effectué par les fichiers `/etc/hosts.allow` et `/etc/hosts.deny`. Voici par exemple ce que l'on peut mettre dans `/etc/inetd.conf` :

```
smtp    stream  tcp     nowait.200  qmaild  /usr/sbin/tcpd smtp
```

et dans `/etc/hosts.allow` :

```
smtp: .mondomaine.amoi 127.0.0.1: \
      setenv RELAYCLIENT:\
      twist { { /usr/bin/tcp-env /usr/sbin/qmail-smtpd 1>&3;} \
      2>&1|splogger qmail;} 3>&1
smtp: ALL: \
      twist { { /usr/bin/tcp-env /usr/sbin/qmail-smtpd 1>&3;} \
      2>&1|splogger qmail;} 3>&1
```

Ainsi, le contrôle de relayage par `rcpthosts` ne sera actif que pour les courriels provenants de connexions SMTP avec des machines extérieures au domaine `mondomaine.amoi`.

Bien sûr, il ne faudra pas oublier de faire un

```
root:bash$ kill -HUP $(pidof inetd)
```

pour que *inetd* relise son fichier de configuration.

Voir les pages de manuel `inetd.conf`(5), `tcpd`(8), `hosts_access`(5) et `hosts_options`(5).

PPP et serialmail

Lorsque la machine n'est connectée à l'Internet que de façon intermittante en PPP par modem par exemple, les courriels à destination de l'extérieur restent dans la file d'attente de *qmail* jusqu'à la prochaine connexion. Il est alors utile d'envoyer un signal ALRM à *qmail-send* à l'établissement de la lisaison IP (par exemple dans le script `/etc/ppp/ip-up.local` ou équivalent) pour qu'il traite les courriels présents dans la file d'attente immédiatement.

J'en profite au passage pour signaler que sur la *Debian GNU/Linux*, les fichiers `/etc/ppp/ip-up.local` et `/etc/ppp/ip-down.local` de PPP sont remplacés par des répertoires `/etc/ppp/ip-up.d/` et `/etc/ppp/ip-down.d/` destinés à accueillir des scripts lancés automatiquement par *run-parts* respectivement à l'établissement et à la chute de la lisaison IP via PPP).

Cette méthode a cependant un inconvénient. Supposons qu'aucune connexion n'est lieu pendant une période assez longue, les courriels en question seront alors tout bètement renvoyés à l'expéditeur, c'est un mécanisme présent dans toute gestion de file d'attente. La variable *qmail* `queuelifetime` represéente la durée de vie en seconde d'un courriel dans la file d'attente. Elle vaut par défaut 604800 secondes, soit une semaine.

Une solution élégante à ce problème consiste à faire en sorte que tous les courriels devant transiter par PPP soit livrés dans une boite aux lettres spécialement prévue à cet effet. Il suffira ensuite dès l'établissement de la liaison IP par PPP de vider cette BAL sur le port SMTP de la passerelle courriel de notre FAI. Ce n'est pas sans rappeler le mecanisme «store-and-forward» d'UUCP. On peut éditer cette BAL avec n'importe quel MUA, ce qui est tout de même mieux que de toucher à la file d'attente de *qmail*.

Nous créons donc une BAL nommée `~alias/ppp` par :

```
root$ maildirmake ~alias/pppdir
root$ chown -R alias ~alias/pppdir
```

Il nous faut ensuite créer le fichier `~alias/.qmail-ppp-default` qui doit contenir la ligne

```
./pppdir/
```

Enfin il faut ajouter une ligne à la fin du fichier `/var/qmail/control/virtualdomains` :

```
mondomaine.amoi:
.mondomaine.amoi:
:alias-ppp
```

ainsi, la livraison vers notre BAL se fera pour les courriels à destination des adresses extérieures au domaine `mondomaine.amoi`. Voir la la section *virtualdomains* du Chapitre 4.

Reste maintenant à trouver un moyen pour vider cette BAL sur le port 25 de notre FAI. C'est là qu'intervient *serialmail*, que l'on trouve sur http://pobox.com/~djb/serialmail.html. *serialmail* existe aussi sous forme de paquetage source *Debian*, je recommande fortement cette méthode à ceux qui possèdent une *Debian GNU/Linux*.

serialmail est un ensemble de petits programmes écrits par Dan Bernstein dont *maildirsmtp* qui permet justement de vider une BAL au format maildir sur un port SMTP.

maildirsmtp fait appel à *tcpclient* lui même faisant partie de UCSPI-TCP (voir http://pobox.com/~djb/ucspi-tcp.html). Un paquetage source *Debian GNU/Linux* est disponible.

Voici ce qu'il faut rajouter dans votre script `/etc/ppp/ip-up.local` ou équivalent :

```
  maildirsmtp ~alias/pppdir \
```

```
alias-ppp- mail.monfai.amoi \
mamachine.mondomaine.amoi 2» /var/log/maildir2smtp.log
```

où `mail.monfai.amoi` désigne la passerelle courriel de votre FAI.

La principale critique que l'on pourrait porter à ce système est que *serialmail* vide les messages sur le port SMTP séquentiellement. Le moindre bloquage empêche le traitement du reste de la BAL.

Avant de terminer cette section, je vous rappelle que le fichier de contrôle `/var/qmail/control/smtproutes` permet de spécifier à *qmail-remote* les relais de courriels à contacter (cf. la page de manuel). L'idéal pour une machine isolée, connectée de façon intermittante à l'Internet par PPP, est de demander à *qmail-remote* de transmettre tout les courriels sortants à la passerelle de courriel du FAI (un «smarthost» pour parler le langage de *sendmail*). Cela évite à *qmail-remote* de contacter lui même les machines destinataires et de faire des requettes DNS. La ligne PPP est donc moins sollicitée. On peut par exemple mettre ceci dans le fichier `smtproutes` :

```
:mail.monfai.amoi
```

Voir la la section *smtproutes* du Chapitre 4.

Fetchmail

Pour récupérer votre courriel sur le port POP3 de votre FAI, je vous conseille d'utiliser l'exellent *fetchmail*. Voici un exemple de `~/.fetchmailrc` possible :

```
poll mail.monfai.amoi protocol pop3:
 username monpseudo
 password monsesame
 fetchall
 batchlimit 1 # utile si le FAI utilise qmail
 is loginlocal here
 mda "env -u QMAILINJECT qmail-inject -f%F %T"
```

fetchmail fera donc appel à *qmail-inject* pour insérer les courriels dans la file d'attente de votre *qmail*. *qmail* se chargera ensuite de livrer le courriel dans votre BAL via le MDA *qmail-local*.

Vous pouvez aussi décider d'utiliser plutôt votre propre port SMTP pour l'insertion dans la file d'attente de votre *qmail*. Voici le `~/.fetchmailrc` correspondant :

```
poll mail.monfai.amoi protocol pop3:
 username monpseudo
 password monsesame
 fetchall
 batchlimit 1 # utile si le FAI utilise qmail
 forcecr # RFC 822 strict
 is loginlocal here
```

En fait, l'insertion dans la file d'attente de votre *qmail* ne sert pas à grand chose. Autant faire appel directement à un MDA qui va directement livrer le courriel dans votre BAL. Le plus simple est par exemple d'utiliser *procmail*. On courcircuite *qmail* car il ne sert à rien ici ! Voici le `~/.fetchmailrc` correspondant :

```
poll mail.monfai.amoi protocol pop3:
 username monpseudo
 password monsesame
 fetchall
 batchlimit 1 # utile si le FAI utilise qmail
 is loginlocal here
 mda "procmail -d loginlocal"
```

Réécriture d'adresse d'expéditeur

Si vous désirez modifier votre identité apparente dans les courriels que vous envoyez, il vous suffit d'utiliser les variables d'environnement `QMAILHOST`, `QMAILUSER` et `QMAILNAME`. Ces variables sont prises en considération par *qmail-inject* lors de l'injection de vos courriels dans la queue (cf. la page de manuel). Les champs `From:` et `Return-Path:` de vos courriels sont alors modifiés automatiquement. Celà s'avère très pratique lorsque votre machine n'est pas en permanence sur l'Internet et ne s'y connecte que de façon intermittante via une adresse IP dynamique.

On peut contrôler les modifications qu'apporte *qmail-inject* aux entêtes en utilisant la variable d'environnement `QMAILINJECT`. Voir la page de manuel de *qmail-inject* pour les détails.

Voici par exemple ce que j'ai rajouté à mon fichier `.bashrc` :

```
export QMAILHOST="mail.dotcom.fr"
export QMAILUSER="dchafai"
export QMAILNAME="Djalil Chafai"
export QMAILINJECT=f
```

Ainsi, les messages que j'envoie apparaissent comme provenant de `dchafai@mail.dotcom.fr (Djalil Chafai)`. Cette solution est interressante car elle marche avec tous les MUA faisant appel au wrapper *sendmail* de *qmail*, lui même appelant *qmail-inject*. De plus, cela permet de se débarrasser des problèmes de paranoïa des passerelles de courriels de certains FAI.

En fait, pour déterminer le nom d'utilisateur, *qmail-inject* utilise les variables d'environnement suivantes par ordre décroissant de préférence : `QMAILUSER`, `MAILUSER`, `USER` et `LOGNAME`. Pour le nom de machine, *qmail-inject* utilise la variable *qmail* `defaulthost`. Elle peut être cependant forcée par les variables d'environnement `QMAILHOST` et `MAILHOST`. Le nom personnel est quant à lui donné par les variables d'environnement `QMAILNAME`, `MAILNAME` et `NAME`.

UUCP

L'idée est un peu la même que celle utilisée pour le PPP. Elle consiste à mettre une ligne dans `virtualdomains` (voir la la section *virtualdomains* du Chapitre 4)

```
.uucp:alias-uucp
```

qui routera les courriels devant transiter par UUCP vers `alias-uucp`. Le fichier `~alias/.qmail-uucp-default` devra alors contenir un tube appelant le binaire qui injecte le courriel dans le spool UUCP (*uux* par exemple).

```
|preline -df /usr/bin/uux - -r -
a "$SENDER" relaisuucp!rmail "($EXT2@$HOST)"
```

Voir les pages de manuel de *preline*, *qmail-local* et *qmail-command* pour les détails. Ainsi tout courriel à destination de `nom@machine.uucp` sera envoyé par UUCP au relais `relaisuucp`.

Listes de diffusion par ezmlm (à faire)

ezmlm est un gestionnaire de listes de diffusions écrit pas Dan Berstein spécialement étudié pour *qmail*. *ezmlm* est beaucoup plus sophistiqué que le gestionnaire de listes minimal *qlist* que contient *qmail* nativement.

Le site http://www.ezmlm.org/ est dédié à *ezmlm*.

Statistiques par qmailanalog

qmailanalog est un ensemble de scripts *awk* écrits par Dan Bernstein qui fabriquent des statistiques sur le courriel en analysant les fichiers d'audit (logs) de *qmail*.

En général, *qmailanalog* est lancé par une crontab hébdomadaire ou mensuelle. Vous trouverez un pointeur sur *qmailanalog* dans sur site de *qmail*.

Dan Bernstein compte réécrire tout ou partie des scripts *awk* en langage C. Les prochaines versions de *qmailanalog* seront donc plus performantes.

Chapitre 6. Quelques configurations type

Sont présentées ici de façon très rudimentaire quelques configurations type qui pourront vous servir de point de départ.

Machine isolée

Est-ce bien utile d'utiliser une messagerie ? :-)

- `/var/qmail/control/me`

  ```
  mamachine.mondomaine.amoi
  ```

Machine isolée + liaison Internet intermittante

Reportez-vous à la section *PPP et serialmail* du Chapitre 5 pour tout ce qui concerne les connexions intermittantes à l'Internet.

- `/var/qmail/control/me`

  ```
  mamachine.mondomaine.amoi
  ```

- `/var/qmail/control/smtproutes`

  ```
  :mail.monfai.amoi
  ```

- `/var/qmail/control/defaultdomain`

  ```
  mondomaine.amoi
  ```

À l'établissement de la connexion à l'Internet, il suffit de demander à *qmail* de tenter de vider sa queue par un :

```
root:bash$ kill -ALRM $(pidof qmail-send)
```

Bien sûr, tout celà peut parfaitement être automatisé, par exemple dans un script `ip-up`.

Machine isolée + liaison Internet intermittante multi-FAI

Reportez-vous à la section *PPP et serialmail* du Chapitre 5 pour tout ce qui concerne les connexions intermittantes à l'Internet.

- `/var/qmail/control/me`

  `mamachine.mondomaine.amoi`

- `/var/qmail/control/defaultdomain`

  `mondomaine.amoi`

- `/var/qmail/control/smtproutes-fai-1`

  `:mail.monfai-1.amoi`

- `/var/qmail/control/smtproutes-fai-2`

  `:mail.monfai-2.amoi`

- `/var/qmail/control/smtproutes` lien symbolique vers l'un des deux précédents.

À l'établissement de la connexion à l'Internet, il suffit de positionner le lien symbolique vers le fichier correspondant au FAI utilisé puis de demander à *qmail* de tenter de vider sa queue par un :

```
root:bash$ kill -ALRM $(pidof qmail-send)
```

Bien sûr, tout celà peut parfaitement être automatisé, par exemple dans un script `ip-up`.

Client sur réseau local sans port SMTP actif

Je suppose que la machine a pour nom `mamachine.mondomaine.amoi`. Elle envoie tout le courriel qui ne lui est pas destiné vers la machine `mail.mondomaine.amoi`. N'ayant pas de port SMTP actif, elle devrait en toute logique avoir pour MX `mail.mondomaine.amoi` dans le DNS du réseau local.

- `/var/qmail/control/me`

  `mamachine.mondomaine.amoi`

- `/var/qmail/control/defaultdomain`

  `mondomaine.amoi`

- `/var/qmail/control/defaulthost`

  `mondomaine.amoi`

- `/var/qmail/control/smtproutes`

  ```
  :mail.mondomaine.amoi
  ```

Dans le cas d'une petite machine, on peut se passer de *qmail* et utiliser un programme très léger comme *sSMTP* qui n'est chargé qu'en cas de besoin et qui permet quand même de faire de la réécriture d'adresses.

Client sur réseau local avec port SMTP actif

Vous devez mettre en place le service SMTP. Reportez-vous au Chapitre 3.

Je suppose que la machine a pour nom `mamachine.mondomaine.amoi`. Elle envoie tout le courriel qui ne lui est pas destiné vers la machine `mail.mondomaine.amoi`. Ayant un port SMTP actif, elle devrait en toute logique être son propre MX dans le DNS du réseau local. Elle n'accepte en SMTP que le courriel qui lui est destiné.

- `/var/qmail/control/me`

  ```
  mamachine.mondomaine.amoi
  ```

- `/var/qmail/control/defaultdomain`

  ```
  mondomaine.amoi
  ```

- `/var/qmail/control/rcpthosts`

  ```
  mamachine
  mamachine.mondomaine.amoi
  ```

- `/var/qmail/control/defaulthost`

  ```
  mondomaine.amoi
  ```

- `/var/qmail/control/smtproutes`

  ```
  :mail.mondomaine.amoi
  ```

Serveur de courriel pour réseau local isolé

Vous devez mettre en place les services SMTP et POP3. Reportez-vous au Chapitre 3.

Je suppose que le serveur a pour nom `mail.mondomaine.amoi` et qu'il est MX pour `mondomaine.amoi`, `mail.mondomaine.amoi`, `client1.mondomaine.amoi` et `client2.mondomaine.amoi` dans le DNS du réseau local.

Le serveur n'accepte que le courriel à destination du domaine `mondomaine.amoi`.

Le courriel dont la destination n'est pas `mondomaine.amoi`, `mail.mondomaine.amoi`, `client1.mondomaine.amoi` ou `client2.mondomaine.amoi` ne sera pas considéré comme local, le serveur tentera donc de contacter les MX correspondants en SMTP, ce qui suppose l'existence de machines dont un MX n'est pas `mail.mondomaine.amoi` dans le réseau local.

- `/var/qmail/control/me`

  ```
  mail.mondomaine.amoi
  ```

- `/var/qmail/control/defaultdomain`

  ```
  mondomaine.amoi
  ```

- `/var/qmail/control/defaulthost`

  ```
  mondomaine.amoi
  ```

- `/var/qmail/control/rcpthosts`

  ```
  mondomaine.amoi
  .mondomaine.amoi
  ```

- `/var/qmail/control/locals`

  ```
  mondomaine.amoi
  mail
  mail.mondomaine.amoi
  client1
  client1.mondomaine.amoi
  client2
  client2.mondomaine.amoi
  ```

Serveur de courriel pour réseau local + liaison Internet intermittante

Vous devez mettre en place les services SMTP et POP3. Reportez-vous au Chapitre 3. Reportez-vous à la section *PPP et serialmail* du Chapitre 5 pour tout ce qui concerne les connexions intermittantes à l'Internet.

Je suppose que le serveur a pour nom `mail.mondomaine.amoi` et qu'il est MX pour `mondomaine.amoi`, `mail.mondomaine.amoi`, `client1.mondomaine.amoi` et `client2.mondomaine.amoi` dans le DNS du réseau local.

Le serveur n'accepte que le courriel en STMP à destination du domaine `mondomaine.amoi`.

Le courriel dont la destination n'est pas `mondomaine.amoi`, `mail.mondomaine.amoi`, `client1.mondomaine.amoi` ou `client2.mondomaine.amoi` ne sera pas considéré comme local, le serveur tentera donc de contacter la passèrelle SMTP `mail.monfai.amoi`.

- `/var/qmail/control/me`

  ```
  mail.mondomaine.amoi
  ```

- `/var/qmail/control/defaultdomain`

  ```
  mondomaine.amoi
  ```

- `/var/qmail/control/defaulthost`

  ```
  mondomaine.amoi
  ```

- `/var/qmail/control/rcpthosts`

  ```
  mondomaine.amoi
  .mondomaine.amoi
  ```

- `/var/qmail/control/locals`

  ```
  mondomaine.amoi
  mail
  mail.mondomaine.amoi
  client1
  client1.mondomaine.amoi
  client2
  client2.mondomaine.amoi
  ```

- `/var/qmail/control/smtproutes`

  ```
  :mail.monfai.amoi
  ```

À l'établissement de la connexion à l'Internet, il suffit de demander à *qmail* de tenter de vider sa queue par un :

```
root:bash$ kill -ALRM $(pidof qmail-send)
```

Bien sûr, tout celà peut parfaitement être automatisé, par exemple dans un script `ip-up`.

Serveur de courriel sur l'Internet

Vous devez mettre en place les services SMTP et POP3. Reportez-vous au Chapitre 3. Si vous souhaitez contrôler l'accès SMTP à votre serveur en fonction de l'adresse IP du client, reportez-vous à la section *Contrôle de relayage* du Chapitre 5.

Je suppose que le serveur a pour nom `mail.mondomaine.amoi` et qu'il est MX pour `mondomaine.amoi`, `mail.mondomaine.amoi`, `client1.mondomaine.amoi` et `client2.mondomaine.amoi` dans les DNS.

Le serveur n'accepte que le courriel à destination du domaine `mondomaine.amoi`.

Le courriel dont la destination n'est pas `mondomaine.amoi`, `mail.mondomaine.amoi`, `client1.mondomaine.amoi` ou `client2.mondomaine.amoi` ne sera pas considéré comme local, le serveur tentera donc de contacter les MX associés en SMTP. Le serveur rejettera tout courriel dont l'adresse de l'expéditeur appartient au domaine `pub.com`.

- `/var/qmail/control/me`

  ```
  mail.mondomaine.amoi
  ```

- `/var/qmail/control/defaultdomain`

  ```
  mondomaine.amoi
  ```

- `/var/qmail/control/defaulthost`

  ```
  mondomaine.amoi
  ```

- `/var/qmail/control/rcpthosts`

  ```
  mondomaine.amoi
  .mondomaine.amoi
  ```

- `/var/qmail/control/badmailfrom`

  ```
  @pub.com
  ```

- `/var/qmail/control/locals`

  ```
  mondomaine.amoi
  mail
  mail.mondomaine.amoi
  client1
  client1.mondomaine.amoi
  client2
  client2.mondomaine.amoi
  ```

Serveur de courriel multi-domaines sur l'Internet

Vous devez mettre en place les services SMTP et POP3. Reportez-vous au Chapitre 3. Si vous souhaitez contrôler l'accès SMTP à votre serveur en fonction de l'adresse IP du client, reportez-vous à la section *Contrôle de relayage* du Chapitre 5.

Je suppose que le serveur a pour nom `mail.mondomaine.amoi` et qu'il est MX pour `mondomaine.amoi`, `mondomaine.aussi`, `mail.mondomaine.amoi`, `client1.mondomaine.amoi` et `client2.mondomaine.aussi` dans les DNS.

Le serveur n'accepte que le courriel à destination des domaines `mondomaine.amoi` et `mondomaine.aussi`.

Le courriel dont la destination n'est pas `mondomaine.amoi`, `mondomaine.aussi`, `mail.mondomaine.amoi`, `client1.mondomaine.amoi` ou `client2.mondomaine.aussi` ne sera pas considéré comme local, le serveur tentera donc de contacter les MX associés en SMTP. Le serveur rejettera tout courriel dont l'adresse de l'expéditeur appartient au domaine `pub.com`.

- `/var/qmail/control/me`

  ```
  mail.mondomaine.amoi
  ```

- `/var/qmail/control/defaultdomain`

  ```
  mondomaine.amoi
  ```

- `/var/qmail/control/defaulthost`

  ```
  mondomaine.amoi
  ```

- `/var/qmail/control/rcpthosts`

  ```
  mondomaine.amoi
  .mondomaine.amoi
  mondomaine.aussi
  .mondomaine.aussi
  ```

- `/var/qmail/control/badmailfrom`

  ```
  @pub.com
  ```

- `/var/qmail/control/locals`

  ```
  mondomaine.amoi
  mail
  mail.mondomaine.amoi
  client1
  client1.mondomaine.amoi
  client2
  client2.mondomaine.amoi
  mondomaine.aussi
  client2.mondomaine.aussi
  ```

Chapitre 7. Petit léxique

Par ordre alphabétique :

- `BAL` = Boite aux lettres;
- `DNS` = Domain Name Service (base de données IP - nom);
- `FAI` = Fournisseur d'Accès à l'Internet;
- `MDA` = Mail Delivery Agent (par exemple *deliver*, *qmail-local*, *procmail*);
- `MTA` = Mail Transport Agent (par exemple *sendmail*);
- `MUA` = Mail User Agent (par exemple *elm*, *pine* ou *mutt*);
- `MX` = Mail Exchanger. Type d'enregistrement DNS;
- `NFS` = Network File System;
- `POP3` = Post Office Protocol version 3 (port TCP 110);
- `QMTP` = Quick Mail Transport Protocol (port TCP 209);
- `RCF 822` = Request For Comment n° 822 (standardisant le courriel);
- `SMTP` = Simple Mail Transport Protocol (port TCP 25);
- `TCP` = Transmission Control Protocol;
- `UCSPI` = Unix Client-Server Program Interface
- `UUCP` = Unix to Unix CoPy
- `maildir` = Format de boite aux lettres reposant sur des répertoires. Un fichier pour chaque courriel. Faire «**man maildir**»;
- `mbox` = Format de boite aux lettres reposant sur un seul fichier. Les courriels sont regroupés dans un seul et unique fichier. Faire «**man mbox**».

Chapitre 8. Quelques adresses

- Ying Zhang zhangg@sfu.ca a écrit un mini-HOWTO sur la mise en place du SMTP, POP, IMAP et la configuration des MUA (Unix et Windows) avec *qmail*. Voir http://www.sfu.ca/~yzhang/linux;
- Adam D. McKenna adam@flounder.net a commencé l'écriture d'un HOWTO. Voir http://www.flounder.net/qmail/qmail-howto.html;
- Le site officiel de *qmail* http://www.qmail.org/, son miroir français http://www.delsys.fr/qmail/ et la page http://pobox.com/~djb/qmail.html que Dan Bernstein http://pobox.com/~djb/ lui consacre;
- *sendmail* http://www.sendmail.com/ qui est passé en .com aprés avoir longtemps été en .org;
- *smail* http://www.smail.org/ qui est livré en standard avec la distribution *Debian GNU/Linux* http://www.debian.org/;
- La «qmail big picture» http://www.nrg4u.com/ permet d'avoir une vue synthétique de l'achitecture de *qmail*;
- Le *qmail* documentation project http://qmail-docs.surfdirect.com.au/;
- L'indispendable manuel en ligne de *qmail* http://www.delsys.fr/qmail/qmail-manual-html/index.html. En fait ce sont les pages de manuels et autres documents ASCII que l'on trouve avec les sources traduits en HTML avec des liens entres eux. Le tout est disponible sous forme d'archive http://www.delsys.fr/qmail/qmail-manual-html.tar.gz. Entre autres choses, vous y trouverez la FAQ *qmail* http://www.delsys.fr/qmail/qmail-manual-html/misc/FAQ.html, qui répond aux questions les plus courantes;
- Le MUA Mutt http://www.mutt.org/ connait très bien les boites aux lettres au format maildir. C'est sans doute l'un des meilleurs MUA mode-texte sous *Unix*;
- Le forum de discussion francophone fr.comp.mail.

Chapitre 9. À propos de ce document

Ce document n'est sans doute pas exempt d'erreurs. L'informatique n'est pas mon métier, je ne suis après tout qu'un amateur qui a décidé de faire profiter tout le monde de sa petite expérience.

Ce document a été rédigé en SGML conformément au DTD *Davenport Docbook v3.0* http://www.ora.com/davenport/docbook/ puis converti en des formats finaux grâce à *Jade* http://www.jclark.com/ et à une version légèrement adaptée des feuilles de style DSSSL pour *DocBook* de Norman Walsh http://www.berkshire.net/~norm/docbook/.

Licence d'utilisation

Ce document est distribué sous les termes de l'OpenContent Licence OPL v1.0. Voici le texte intégral de la licence :

Disponibilité

La dernière version de ce document est disponible en ligne sur le site du Club des Utilisateurs de Linux Toulousains et des environs (CULTe) http://savage.iut-blagnac.fr/ à l'URL http://savage.iut-blagnac.fr/projets/doc/qmail/.

Vous trouverez aussi la dernière version sur Linux-France http://www.linux-france.com/ à l'URL http://www.linux-france.com/article/mail/qmail/intro/.

Enfin, en dernier recours, vous pouvez aller voir sur ma page au Laboratoire de Statistique et Probabilités http://www-sv.cict.fr/lsp/ à l'URL http://www-sv.cict.fr/lsp/Chafai/Docs/intro-a-qmail/.

Vous trouverez l'archive de la dernière version de ce document aux URL http://www-sv.cict.fr/lsp/Chafai/Fichiers/intro-a-qmail_v0.4 .tar.gz et http://www-sv.cict.fr/lsp/Chafai/Fichiers/intro-a-qmail_v0.4 .tar.bz2 .

Remerciments

Le logo utilisé dans ce document est l'oeuvre de Dave Sill de5@sws5.ctd.ornl.gov.

Je tiens tout particulièrement à remercier Hédi Berriche et Éric Marsden pour les discussions fructueuses que j'ai eu avec chacun d'entre eux et pour leur lecture attentive des différentes versions du document. Ce document n'aurait sans doute jamais vu le jour sans Éric qui m'a beaucoup encouragé au tout début.

Contributions par ordre alphabétique

- Hédi Berriche berriche@mip.ups-tlse.fr, http://mip.ups-tlse.fr/~berriche/;
- Éric Marsden emarsden@mail.dotcom.fr, http://www.chez.com/emarsden/.