

Poster et recevoir du courrier avec Postfix

Éric Jacoboni

Version 0.7 – 29 octobre 2000

Table des matières

1	Introduction	2
1.1	À propos de ce document	2
1.2	Présentation	2
2	Installation de Postfix	3
2.1	Installation à partir de la distribution source	3
2.1.1	Compilation	3
2.1.2	Installation de la documentation	3
2.1.3	Installation des fichiers de configuration	3
2.1.4	Création des files d'attente	4
2.1.5	Installation des exécutable	4
2.1.6	Droits d'accès à la file d'attente	4
2.2	Utilisation de paquetages pré-compilés	5
3	Première configuration	6
3.1	Oublier sendmail!	6
3.2	Small is beautiful...	6
3.3	Démarrage de postfix	7
3.4	Configuration minimale	7
3.5	Test de la configuration locale	10
4	Pour aller un peu plus loin...	12
4.1	Les files d'attente	12
4.2	Les commandes de postfix	13
4.3	Envoyer du courrier non local	14
5	La réécriture de l'adresse de l'expéditeur	16
5.1	Énoncé du problème	16
5.2	La réécriture et le masquage du champ From :	17
6	Gestion des adresses locales	18
7	Conclusion et remerciements	19

1 Introduction

1.1 À propos de ce document

Ce document, comme d'habitude, ne se veut pas être un guide de référence ou une adaptation française de la documentation de `postfix`, très bien faite au demeurant. Il se borne à relater les manipulations que j'ai été amené à faire pour configurer le logiciel afin que je puisse recevoir et poster du courrier avec lui. Toutes critiques, positives ou négatives sont évidemment les bienvenues. Le site de référence de cette documentation est Linux-France qui contiendra toujours la version la plus récente. À partir de cette page, vous pouvez télécharger les sources DocBook/XML, la version au format PostScript et la version au format PDF de ce document.

1.2 Présentation

La configuration décrite ici a été testée avec les systèmes d'exploitation Debian GNU/Linux, et FreeBSD.

En fait, le système utilisé a peu d'importance du moment qu'il est reconnu par `Postfix`. Ce qui est décrit ci-après devrait donc s'appliquer dans tous les cas, le cas échéant à quelques modifications mineures près. Notons que, pour Linux, des paquetages précompilés existent : la Debian, notamment, permet d'installer directement le programme via `dpkg` ou `apt-get` et j'imagine qu'il existe également des paquetages `rpm`. Le problème, avec ces paquetages binaires, est qu'ils ne permettent pas de suivre les fréquents changements du logiciel ; or, celui-ci étant en phase de développement, les nouvelles versions ajoutent de nouvelles fonctionnalités souvent fort intéressantes...

Concernant les autres Unix, il n'y a aucune raison pour que le fonctionnement soit différent : la documentation cite d'ailleurs une liste de tous les systèmes pris en charge. Je n'ai personnellement vu aucune différence entre l'installation sous Linux et sous FreeBSD (dont les scripts d'initialisation sont bien distincts).

La distribution officielle de `Postfix` peut être récupérée sur son site officiel, ou sur l'un des sites miroirs français. La distribution actuelle est *postfix-19991231.08*.

Il ne sera pas traité ici de la récupération du courrier venant de l'extérieur (du serveur de courrier de son fournisseur d'accès si l'on est, comme moi, relié via une connexion téléphonique). Pour cela, d'autres documentations existent, consultez notamment le site Linux-France.

2 Installation de Postfix

2.1 Installation à partir de la distribution source

2.1.1 Compilation

Le logiciel se récupère sous la forme d'un fichier *.tar.gz* que l'on décompresse dans un répertoire d'installation :

```
# tar xvzf postfix-19991231-pl08.tar.gz -C /tmp
# cd /tmp/postfix-19991231-pl08/
```

Dans ce répertoire se trouve un fichier *INSTALL* fort clair (mais en anglais) qu'il suffit de suivre pas à pas pour installer et configurer *postfix*. La première étape consiste à générer les exécutables :

```
# make
```

Il faut ensuite déplacer les fichiers binaires, de configuration et les pages de manuel dans les répertoires adéquats de votre système. Sous le répertoire d'installation, vous devez normalement avoir les répertoires *conf*, *bin*, *libexec*, *html* et *man* (les autres répertoires sont ceux contenant les sources, des exemples et les fichiers objets générés par la compilation).

2.1.2 Installation de la documentation

Pour installer les pages de manuel, il suffit de déplacer le contenu des répertoires */tmp/postfix-19991231-pl08/man/man** dans les répertoires correspondants sur votre système (*/usr/man/man** sur une machine Linux, */usr/local/man/man** sur une machine FreeBSD).

Puis, vous pouvez installer le reste de la documentation. Pour rester compatible avec les documentations des autres logiciels de mon système Linux, vous pouvez créer un répertoire */usr/doc/postfix* dans lequel vous déplacerez tous les fichiers de documentation du répertoire d'installation (leurs noms sont en majuscules). Créez également le répertoire */usr/doc/postfix/html* et déplacez-y le contenu du répertoire *html* de l'installation (pour FreeBSD, on fait de même et on crée l'arborescence */usr/local/share/doc/postfix*).

2.1.3 Installation des fichiers de configuration

Sous Linux, tous ces fichiers vont dans un seul emplacement : */etc/postfix*, sous FreeBSD, ils iront dans */usr/local/etc/postfix*.

Je me bornerai ici à suivre ce qui est indiqué dans *INSTALL* (adaptez le chemin des fichiers de configuration à votre système) :

```
# mkdir /etc/postfix
# chmod 0755 /etc/postfix
# mv /tmp/postfix-19991231-pl08/conf/* /etc/postfix
# chmod 0644 /etc/postfix/*
```

```
# chmod 0755 /etc/postfix/postfix-script*
```

2.1.4 Création des files d'attente

Postfix utilise une arborescence beaucoup plus élaborée que celle de ses prédécesseurs pour placer les messages en attente de délivrance. Il suffit ici d'en indiquer la racine (*/var/spool/postfix/*, généralement) car tous les autres sous-répertoires y seront créés lors du premier démarrage de postfix :

```
# mkdir /var/spool/postfix
# chmod 0755 /var/spool/postfix
```

Vous pouvez choisir un autre emplacement : celui-ci devra être indiqué lors de la configuration que nous étudions plus loin.

2.1.5 Installation des exécutables

Là encore, vous êtes libres de choisir l'emplacement qui vous convient car il suffira ensuite de l'indiquer lors de la configuration.

La méthode généralement pratiquée consiste à séparer les *commandes* des *démons* : les premières sont initialement dans le répertoire *bin* et iront dans */usr/local/sbin* avec des liens de (*/usr/sbin* vers eux), les seconds sont initialement dans *libexec* et iront dans */usr/local/libexec/postfix* :

```
# cd /tmp/postfix-19991231-pl08/bin
# cp post* sendmail /usr/local/sbin
# mkdir /usr/local/libexec/postfix
# cp 'ls | egrep -v 'post|fsstone|smtp-|sendmail'' /usr/local/libexec/postfix
```

2.1.6 Droits d'accès à la file d'attente

Il s'agit ici de permettre l'accès des utilisateurs locaux au répertoire */var/spool/postfix/maildrop*. Par défaut, tous les sous-répertoires de */var/spool/postfix* appartiennent au groupe *root*. Or, lorsque les utilisateurs postent un courrier, celui-ci transite d'abord par le répertoire *maildrop*. Avec les droits actuels, ces courriers seraient donc refusés.

Plusieurs possibilités, décrites dans le fichier *INSTALL*, sont possibles : rendre le répertoire *maildrop* accessible à tout le monde, ou utiliser la commande *postdrop* en *sgid*. Nous avons retenu la première solution. Pour ce faire, il suffit de modifier les droits d'accès du répertoire *maildrop* :

```
# chmod 1733 /var/spool/postfix/maildrop
```

Cette solution suppose que vous ayez confiance en vos utilisateurs locaux... Si ce n'est pas le cas, préférez-lui la solution du `postdrop` en `sgid` (décrite dans le fichier *INSTALL*).

L'invocation de la commande `postdrop` lors de l'envoi d'un message est réalisée automatiquement via l'appel du script `postfix-script` qu'il s'agit donc de rendre accessible à tout le monde :

```
# cp /etc/postfix/postfix-script-nosgid /etc/postfix/postfix-script
```

2.2 Utilisation de paquetages pré-compilés

Si vous préférez utiliser des paquetages *rpm* ou *deb*, vérifiez les emplacements des différents fichiers et adaptez les chemins utilisés ici à votre configuration.

3 Première configuration

3.1 Oublier sendmail !

Pour que postfix devienne votre agent de transport de courrier, vous devrez probablement désactiver sendmail. Même si vous ne l'avez jamais activé ou configuré, il y a fort à parier que ce dernier ait été installé par votre distribution Linux !

Pour cela, il suffit, encore une fois, de suivre ce que propose le fichier *INSTALL* :

```
# cd /usr/sbin
# mv sendmail sendmail.OFF
# ./sendmail.OFF -q
# mv /usr/bin/newaliases /usr/bin/newaliases.OFF
# mv /usr/bin/mailq /usr/bin/mailq.OFF
# chmod 0 /usr/sbin/sendmail.OFF /usr/bin/newaliases.OFF \
    /usr/bin/mailq.OFF
# ln -s /usr/local/sbin/sendmail /usr/sbin/sendmail
# ln -s /usr/local/sbin/sendmail /usr/bin/mailq
# ln -s /usr/local/sbin/sendmail /usr/bin/newaliases
```

Et c'est fini...

Ces quelques lignes sauvegardent les exécutables originaux de sendmail, vident sa file d'attente, ôtent toutes les permissions pour les protéger, et créent des liens symboliques ayant les mêmes noms que les originaux vers la commande sendmail de postfix.

3.2 Small is beautiful...

La première chose qui frappe, quand on se plonge dans la documentation fournie avec postfix est son apparente simplicité... Finie la syntaxe ésotérique du *sendmail.cf*, finie la nécessité de passer par un ensemble de macros pour oser espérer comprendre un tant soit peu ce que l'on est en train de configurer : avec postfix tout se passe par l'adaptation d'un seul fichier, *main.cf*, normalement placé, comme tous ses autres fichiers de configuration, dans le répertoire */etc/postfix/* (ou */usr/local/etc/postfix/*). Enfin, presque... nous verrons plus tard que d'autres fichiers doivent être adaptés, mais tous sont humainement lisibles et richement commentés.

À la différence de sendmail, programme monolithique par excellence, postfix est composé de nombreux petits programmes réalisant chacun une tâche bien définie. Pour la plupart, ces programmes ne sont pas vus de l'utilisateur mais appelés directement par le programme serveur */usr/sbin/postfix*. Le lancement de celui-ci au démarrage du système dépend de l'Unix utilisé : avec un System V (Linux en fait partie, sauf la distribution Slackware), il est lancé via le script */etc/init.d/postfix* selon la méthode habituelle des *runlevels* ; sur FreeBSD, il est lancé via la ligne *sendmail_enable="YES"* dans le fichier */etc/rc.conf* (vous comprendrez plus loin ce que sendmail vient faire ici...). Sur les systèmes BSD, on peut également utiliser le fichier */etc/rc.local* :

```
% cat /etc/rc.local
```

```
postfix start
```

3.3 Démarrage de postfix

Ainsi que nous venons de le dire, quel que soit le système utilisé, c'est un script qui lance le serveur `/usr/sbin/postfix` en lui passant le paramètre `start`. Celui-ci lance à son tour le serveur principal, `/usr/libexec/postfix/master` qui prend alors les choses en main et lancera les autres démons lorsque cela sera nécessaire. Ces derniers se termineront après avoir accompli leurs tâches ou après une certaine période d'inactivité. Seul, le démon de gestion de la file d'attente, `/usr/libexec/postfix/qmgr` reste en permanence en activité.

Tout ceci peut se vérifier par une simple commande `ps` (ici sous FreeBSD, d'où la présence de ces serveurs sous `/usr/local/`) :

```
% ps axf

...
583 ? S    0:00 /usr/local/libexec/postfix/master
584 ? S    0:00 \_ pickup -t fifo -c
585 ? S    0:00 \_ qmgr -t fifo -u -c
...
```

qui met en évidence la présence du démon `master` et le fait qu'il a lui-même lancé les démons `pickup` et `qmgr` (la commande `ps` utilisée ici est celle de GNU, l'option `-f` n'a pas la même signification avec un `ps` BSD).

`pickup` est responsable de la récupération des courriers locaux : comme nous l'écrivions plus haut, pour des raisons de compatibilité, `postfix` utilise un programme nommé `/usr/sbin/sendmail` (qui *n'est pas* le programme `sendmail` bien connu, mais un homonyme). Ce programme est utilisé pour déposer les courriers locaux dans la file d'attente `maildrop` : tous les courriers qui sont postés par tous les utilisateurs sont déposés dans cette file.

`pickup` les récupère alors et les passe au démon `cleanup` qui remplira les en-têtes manquants, gèrera les enveloppes des messages et les déposera enfin dans une autre file d'attente, nommée `incoming`. Puis `cleanup` avertira le gestionnaire de file d'attente, `qmgr`, qu'un nouveau courrier est arrivé.

`qmgr` s'occupera alors de délivrer le courrier dans les boîtes aux lettres de leurs destinataires et de gérer les erreurs.

Nous verrons plus loin les autres démons entrant en jeu dans la délivrance du courrier. Passons maintenant à une configuration minimum pour tester localement `postfix`.

3.4 Configuration minimale

Notre but, ici, est d'arriver à poster et recevoir du courrier en local : par exemple, `root` doit pouvoir poster un message à l'utilisateur `babe`. Ce dernier doit pouvoir récupérer le message, le lire et répondre à `root`. Pour simplifier, nous utiliserons le programme canonique `mail`.

Nous supposerons que notre machine s'appelle `alex` et que notre domaine s'appelle `linux-france.org`. Vérifions tout de suite que c'est le cas :

```
% hostname
alex.linux-france.org
```

Ainsi que nous l'avons déjà dit, la majeure partie du travail de configuration consiste à adapter le fichier `/etc/postfix/main.cf` (ou `/usr/local/etc/postfix/main.cf`) à nos besoins. Bien entendu, tout cela doit se faire sous le compte `root`.

La première chose à faire est de sauvegarder le fichier original :

```
# cp /etc/postfix/main.cf /etc/postfix/main.cf.0
```

Puis, chargez `main.cf` dans votre éditeur de texte favori.

Normalement, un certain nombre d'options sont déjà en place. Certaines conviennent, d'autres non. Voici les lignes qui nous intéressent (les commentaires et les lignes non modifiées ont été supprimés) :

```
# INFORMATIONS SUR LES REPERTOIRES LOCAUX
queue_directory = /var/spool/postfix
command_directory = /usr/local/sbin
daemon_directory = /usr/local/libexec/postfix

# POSSESSION DES FILES D'ATTENTE ET DES PROCESSUS
mail_owner = postfix

# NOMS DE LA MACHINE ET DU DOMAINE
myhostname = alex.linux-france.org

# POUR L'ENVOI DU COURRIER
myorigin = $myhostname

# MODE DE TRANSPORT
default_transport = smtp

# GESTION DES ALIAS
alias_maps = hash:/etc/postfix/aliases
alias_database = hash:/etc/postfix/aliases

# DELIVRANCE DU COURRIER
mailbox_command = /usr/local/bin/procmail
```

Assurez-vous que toutes les autres possibilités pour ces lignes soient considérées comme des commentaires en les faisant précéder du caractère dièse (#) et ne modifiez pas les autres.

Avant de tester tout cela, détaillons rapidement les options choisies (pour des renseignements plus précis, reportez-vous aux pages de manuel et à la documentation fournie avec le programme).

La première section sert à spécifier les emplacements :

`/var/spool/postfix`

est le répertoire de base pour toutes les files d'attente de `postfix`, Lors de son premier lancement, `postfix` créera tous les sous-répertoires pour ses files sous ce répertoire ;

/usr/local/sbin

est le répertoire où se trouvent les commandes de `postfix` (les exécutable dont le nom commence par `post`, et sa version de `sendmail`);

/usr/local/libexec/postfix

est le répertoire contenant les démons de `postfix` : c'est là que se trouvent tous les programmes serveurs qu'il utilise.

La deuxième section précise qui est le propriétaire de la file d'attente et de la plupart des processus serveurs de `postfix`. Ici, nous avons conservé la proposition, après avoir créé l'utilisateur `postfix`. Voici son entrée dans notre fichier `/etc/passwd` :

```
postfix:x:101:101:./var/spool/postfix:/bin/false
```

Le 'x' dans la partie mot de passe vient du fait que nous utilisons les `shadow passwords`. Le groupe 101 correspond au groupe `postfix`, lui aussi créé pour l'occasion :

```
# grep 101 /etc/group
postfix:x:101:
```

La troisième section sert à indiquer le nom complet de notre machine.

La section suivante concerne l'envoi du courrier : elle permet de renseigner `postfix` sur la machine qui a posté. Pour le moment, nous considérerons que c'est ce que contient la variable `$myhostname`.

Nous précisons ensuite le protocole utilisé pour l'acheminement du courrier. Pour l'instant, `postfix` ne reconnaît que `smtp` et `uucp` (en réalité, on peut créer des transports dans `/etc/postfix/master.cf`, ce qui permet de changer des paramètres en fonction de multiples critères. On peut ainsi dupliquer le transport `smtp` et en changer les caractéristiques en fonction des courriers entrants ou sortants ce qui est très souple. Toutefois, ne l'ayant pas pratiqué, je n'en dirais pas plus...).

La gestion des alias peut faire appel au fichier `/etc/aliases` utilisé par ses prédécesseurs mais nous préférons en utiliser un autre : `/etc/postfix/aliases` (ou `/usr/local/etc/postfix/aliases`). La commande `man aliases` vous renseignera en détail sur le format de ce fichier. Disons simplement que, comme son nom l'indique, il permet de définir des alias entre des noms de destinataires. Ainsi, par exemple, un serveur de news poste quotidiennement un rapport sur ses activités à l'utilisateur `news` (ou `usenet`). Supposons que `babe` soit l'administrateur des news sur `alex` : pour qu'il puisse recevoir ces messages, et si `root` est d'accord, bien entendu, il suffit d'indiquer que les destinataires `news` et `usenet` ont pour alias `babe`. Ceci est réalisé par l'ajout de la ligne suivante dans `/etc/postfix/aliases` :

```
news: babe
usenet: babe
```

Pour des raisons d'optimisation, `postfix`, comme ses prédécesseurs, demande à ce que ce fichier soit traité comme une base de données au format `DBM` ou `DB`. Pour générer ces formats, on utilise l'utilitaire `/usr/sbin/postalias` (qui, rappelons-le, est un lien vers `/usr/local/sbin/postalias`). Ma machine ne reconnaissant pas le format `DBM`, j'ai donc opté pour le second et produit la base à l'aide de la commande :

```
# postalias hash:/etc/postfix/aliases
```

qui a engendré le fichier */etc/postfix/aliases.db*.

La section concernant la délivrance du courrier local indique ici que nous souhaitons utiliser *procmail* pour cette tâche. Tout autre programme ayant la même fonction peut convenir (*deliver*, par exemple), mais *procmail* est le plus connu dans le monde Linux et FreeBSD. Cette section ne concerne que l'acheminement *local* du courrier, *ie.* son écriture dans les boîtes aux lettres des destinataires.

3.5 Test de la configuration locale

Après toute modification de l'un des fichiers que nous venons d'étudier, il faut demander à *postfix* de relire sa configuration :

```
# postfix reload
```

À l'aide de la commande *ps ax*, vérifiez la présence des démons *master*, *pickup* et *qmgr*. Si vous ne les voyez pas, c'est qu'il y a eu un problème : consultez les fichiers */var/log/mail.** pour tenter d'en rechercher la cause.

Si tout s'est bien passé, *root* va pouvoir envoyer un courrier à *babe* :

```
# mail babe -s test
premier test local
.
Cc:
#
```

Immédiatement, *babe* a dû recevoir ce courrier :

```
% mail
Mail version 8.1 6/6/93. Type ? for help.
"/var/spool/mail/babe": 1 message 1 new
>N 1 root@alex.linux-france.org. Wed Jan 20 01:46 12/424 "test"
& 1
Message 1:
From root@alex.linux-france.org Wed Jan 20 01:46:10 1999
Delivered-To: babe@alex.linux-france.org
To: babe@alex.linux-france.org
Subject: test
Date: Wed, 20 Jan 1999 01:46:10 +0100 (CET)
From: root@alex.linux-france.org

premier test local
& d
& q
```

On notera la présence d'un champ `Delivered-To` : : il est ajouté par postfix afin d'éviter les boucles dans la délivrance du courrier et ne sera pas affiché par défaut avec la plupart des logiciels de lecture du courrier (en tous cas, avec Gnu et Netscape...).

Essayons encore : maintenant postez sous le compte *babe* un message à *root* et vite, très vite, faites `ps axf`. Sous Linux, vous devriez voir les lignes suivantes :

```
1271 ? S    0:00 /usr/local/libexec/postfix/master
1272 ? S    0:00 \_ pickup -t fifo -c
1273 ? S    0:00 \_ qmgr -t fifo -u -c
1286 ? S    0:00 \_ cleanup -t unix -u -c
1287 ? S    0:00 \_ trivial-rewrite -n rewrite -t unix -u -c
1288 ? S    0:00 \_ local -t unix
```

Assez rapidement, vous remarquerez, si vous faites la même commande, que `cleanup` et `local` se sont terminés, tandis que `trivial-rewrite` survit plus longtemps, puis se termine.

Tout ceci correspond ce que nous disions plus haut : `pickup` a récupéré le message et l'a passé à `cleanup`. `trivial-rewrite` s'est chargé de réécrire l'adresse en rajoutant le nom de la machine locale derrière le nom de l'utilisateur. `local` est le démon responsable de la délivrance locale du message dans la boîte aux lettres du destinataire. C'est à ce moment que le fichier des alias est pris en compte et, si le destinataire utilise un fichier `.forward`, `local` le fait suivre à l'adresse indiquée. C'est `local` qui ajoute le champ `Delivered-To` : pour éviter un bouclage intempestif et c'est lui qui remplit le champ `From` de l'enveloppe (à ne pas confondre avec le champ `From` :...).

Pour finir cette partie, il ne vous reste plus qu'à essayer de faire la même chose avec vos logiciels de lecture de courrier favoris : cela ne devrait pas poser de problème puisque `local` délivre les messages à l'endroit où la plupart des logiciels s'attendent à les trouver.

4 Pour aller un peu plus loin...

4.1 Les files d'attente

Lorsqu'on a l'habitude d'utiliser `sendmail`, la multiplicité des files d'attente de `postfix` a tendance à dérouter. En effet, en lieu et place du classique et unique `/var/spool/mqueue/` on se retrouve avec l'arborescence suivante :

```
# tree /var/spool/postfix/
/var/spool/postfix/
|-- active
...

|-- deferred
|   |-- 048ED779D1
|   |-- 09085779E0
|   |-- 17101779D6
|   |-- 31DE0779DA
|   |-- 3D15D779D4
|   |-- 4D2BD779D7
|   |-- 7BFA6779D3
|   |-- 7C65D779CF
|   |-- B10C3779D0
|   |-- C3272779D2
...

|-- incoming
...

|-- maildrop
...
```

Nous avons abrégé la sortie de cette commande pour ne retenir que les noms de répertoires qui nous intéressent ici.

`postfix` utilise quatre files d'attentes :

maildrop

contient les messages locaux ;

incoming

contient les messages qui ont été prélevés dans *maildrop* par le démon `pickup`, puis qui ont été traités par le démon `cleanup`. Cette file contient aussi les messages venant de l'extérieur. En bref, elle contient les messages qui n'ont pas encore été traités par le gestionnaire de file d'attente `qmgr` ;

active

est une file contenant les messages en cours de délivrance par `qmgr` ;

deferred

contient les messages qui n'ont pas pu être délivrés (il y en a 10 dans notre exemple).

De plus, le répertoire `/var/spool/postfix/defer` contient les mails en attente plus longue et des répertoires qui sont hachés afin de ne pas avoir des répertoires contenant trop de fichiers : ainsi, le fichier `7B345AC0B1`, par exemple, sera dans `defer/7/B/7B345AC0B1`.

La documentation HTML de la distribution postfix dispose d'un schéma présentant clairement les interactions entre les différentes files d'attente et les différents démons. C'est d'ailleurs de lui que je me suis inspiré...

Le contenu de la file d'attente peut être consulté avec la commande `mailq` (les habitués de `sendmail` ne seront pas dépayés...) : normalement celle-ci ne doit produire que les messages dont la délivrance n'a pas encore eu lieu (dans notre cas, les 10 messages contenus dans la file *deferred*).

4.2 Les commandes de postfix

Nous avons déjà vu que postfix s'invoquait en lui passant un paramètre qui précisait l'action à entreprendre. Ainsi, les paramètres `stop` et `start` arrêtent et démarrent le serveur, respectivement. D'autres commandes existent :

reload

force postfix à relire ses fichiers de configuration (en fait, il s'agit de deux appels consécutifs à postfix : l'un avec le paramètre `stop`, l'autre avec le paramètre `start`) : cette commande s'avère donc nécessaire après la modification du fichier *main.cf*, par exemple ;

check

permet de vérifier la configuration du système de courrier. Ce paramètre permet aussi de surveiller les différentes permissions des répertoires utilisés par postfix, et de créer ces répertoires s'ils n'existent pas. Si la configuration est correcte, et si l'option `-v` n'a pas été utilisée, cette commande ne produit aucune sortie ;

flush

force postfix à tenter de vider la file *deferred*, donc à envoyer les messages en attente de délivrance.

Enfin, *last but not least*, la distribution postfix fournit le programme `postconf` pour afficher les paramètres modifiés par notre configuration (`postconf -n`), les valeurs par défaut du logiciel (`postconf -d`) et l'ensemble des valeurs courantes des paramètres (`postconf`). Voici, à titre d'exemple, ce que donne la première commande sur ma configuration personnelle actuelle :

```
# postconf -n
alias_database = hash:/etc/postfix/aliases
alias_maps = hash:/etc/postfix/aliases
command_directory = /usr/local/sbin
daemon_directory = /usr/local/libexec/postfix
debug_peer_level = 2
default_destination_concurrency_limit = 10
default_transport = smtp
defer_transports = smtp
local_destination_concurrency_limit = 2
mail_owner = postfix
mailbox_command = /usr/local/bin/procmail
myhostname = alex.linux-france.org
myorigin = $myhostname
program_directory = /usr/local/libexec/postfix
queue_directory = /var/spool/postfix
relayhost = [smtp.mon.fai]
sender_canonical_maps = hash:/etc/postfix/canonical
```

Nous n'avons pas encore vu certains de ces paramètres ? C'est normal. Nous allons maintenant nous en occuper...

4.3 Envoyer du courrier non local

Bien, nous savons maintenant que `postfix` fonctionne correctement avec les adresses locales à notre machine (si ce n'est pas le cas, n'allez pas plus loin et revenez ici lorsque le problème sera réglé...). Allons maintenant à la conquête du monde...

La meilleure chose à faire, tant que tous nos tests n'ont pas donné satisfaction, est d'utiliser un *réchoz* : *ie.* une adresse où nous pourrions envoyer un courrier qui nous sera retourné tel qu'il a été reçu. Ceci nous permettra de vérifier au moins deux choses :

- que notre message a bien été acheminé à cette adresse ;
- que les entêtes de notre message sont corrects (c'est important car certains sites refusent les messages provenant de sites non connus : voir la section sur le *rémasquagez* des adresses).

`echo@cnam.fr` est une adresse de ce type, et nous l'utiliserons ici.

Tout le courrier sortant de notre machine est d'abord envoyé au serveur de courrier de notre fournisseur d'accès qui se chargera ensuite de l'envoyer sur l'Internet. Dans la terminologie classique, cela s'appelle un *hôte relais* (ou *relayhost*, en anglais). Il faut donc indiquer à `postfix` le nom de cet hôte relais. Si l'on suppose que nous sommes abonnés au fournisseur d'accès bien connu `mon.fai` et que la machine s'occupant du courrier chez ce fournisseur s'appelle `smtp.mon.fai`, il faut modifier le fichier `main.cf`, rechercher les lignes contenant `relayhost`, en décommenter une et mettre :

```
relayhost = [smtp.mon.fai]
```

Le format indiqué ici suppose que nous utilisons SMTP (spécifié par la variable `default_transport`) pour transporter notre courrier. En réalité, le format de `relayhost` permet aussi d'indiquer une machine UUCP bien que le transport par défaut reste SMTP pour un intranet local, par exemple (voir les commentaires associés à cette variable dans `main.cf` pour connaître toutes les possibilités).

Puis, comme à chaque modification de ce fichier, il faut faire `postfix reload`.

Essayons maintenant la commande suivante :

```
% mail echo@cnam.fr -s test
essai d'envoi de courrier via postfix
.
Cc:
%
```

Puis, examinons la file d'attente :

```
% mailq
-Queue ID- --Size-- ----Arrival Time---- -Sender/Recipient-----
9C26D779D0      351 Wed Jan 20 21:23:55  babe@alex.linux-france.org
(Name service error for domain cnam.fr: Host not found, try again)
```

N'étant relié à l'Internet qu'épisodiquement, via une connexion PPP à un fournisseur d'accès (F.A.I.), le domaine `cnam.fr` ne pourra être connu que lorsque nous serons connectés, par consultation du serveur de noms de notre F.A.I. Ce n'était pas le cas ici, et ce ne le sera pas souvent : pour économiser sur la facture téléphonique, on a tout intérêt à composer son courrier en étant déconnecté et à tout expédier d'un seul coup lors de la connexion suivante.

`postfix` nous prévient de cet état de fait grâce au message *Name service error for domain cnam.fr* de la commande `mailq`. Lors de notre prochaine connexion, il suffira simplement d'appeler la commande `postfix flush` (ou `sendmail -q` pour les nostalgiques...) pour que le courrier en attente soit à nouveau délivré.

sur ma machine, par exemple, seul `root` a le droit de faire `postfix flush`, un utilisateur normal devra utiliser `/usr/sbin/sendmail -q` pour envoyer les messages. La documentation de `postfix` vous expliquera pourquoi.

Lorsque l'on est, comme moi, connecté épisodiquement, le mieux est donc d'indiquer à `postfix` de déferer la délivrance des courriers. Ceux-ci seront alors placés dans la file d'attente et il ne tentera plus de les envoyer sauf si on le lui demande explicitement via un `sendmail -q`. Ce type de fonctionnement est tout à fait adapté aux connexions épisodiques (on peut mettre le `sendmail -q` dans un script de post-connexion) et est piloté par la variable `defer_transports` du fichier `main.cf` :

```
defer_transports = smtp
```

Ceux qui se sont longtemps battu pour contourner ce problème avec `sendmail`, le vrai, apprécieront cette simplicité...

Nous pouvons aussi aller directement inspecter le contenu des files d'attente de `postfix` : vous noterez alors que le numéro `9C26D779D0` apparaît en deux endroits :

- dans le répertoire `/var/spool/postfix/defer/9/C/` : si vous regardez le contenu de ce fichier, vous noterez qu'il contient simplement le message indiquant le problème de résolution de l'adresse de destination.
- dans le répertoire `/var/spool/postfix/deferred` : l'édition du contenu du fichier vous permettra de constater que toutes les informations y sont, mais formatées d'une façon peu lisible...

Il existe une commande, assez primitive pour l'instant, appelée `postcat` qui permet d'afficher sous une forme plus lisible les messages dans les files de `postfix`.

5 La réécriture de l'adresse de l'expéditeur

5.1 Énoncé du problème

Vous avez probablement passé beaucoup de temps à trouver un nom pour votre chère machine, et vous êtes sûr d'avoir fait preuve d'imagination en l'ayant appelé `warlordz.kill.windows` : je doute que ce soit original, mais cela change de `localhost.localdomain`...

Quoi qu'il en soit, un problème va maintenant se poser : avec cette configuration, et un tel nom de machine, le champ `From` : de vos courriers sera, par exemple `bugs@warlordz.kill.windows`, ou `lamer@warlordz.kill.windows`. Cela ne posera pas de problème quand l'utilisateur `bugs` postera un courrier à `lamer` puisqu'il s'agira d'une délivrance locale, mais cela risque d'en poser lorsque `bugs` enverra un courrier à `postmaster@site.serieux`... En effet, pour éviter les courriers indésirables (aka SPAM ou UCE) postés par de tristes sires, le système de courrier de `site.serieux` a sûrement mis en place une protection toute simple : tout message dont l'adresse d'expéditeur n'appartient pas à un domaine connu sera directement mis au panier (il y a même des sites qui vont plus loin en bannissant les messages venant de domaines connus... pour être assez laxistes quant à l'émission de ces fameux SPAMs).

Qu'est-ce qu'un rdomaine connu ? Un domaine dûment enregistré auprès des instances de l'Internet et je suppose que vous n'avez pas payé le NIC pour déposer le domaine `warlordz.kill.windows`, non ? En conséquence, `site.serieux`, lorsqu'il recevra un message venant de votre domaine, recherchera dans ses tablettes s'il connaît ce nom et, comme ce ne sera pas le cas, votre précieuse missive ira se perdre dans les oubliettes de l'histoire, sans même que vous en soyez averti..

Un autre problème est que, même si le message n'était pas rejeté et que `postmaster@site.serieux` lui répondait, le message de réponse serait donc envoyé à `bugs@warlordz.kill.windows`. Là encore, le site n'étant pas connu, pas même de votre fournisseur d'accès, la réponse se perdra dans la nature (ou plutôt non, elle reviendra à `postmaster@site.serieux`, ce qui ne manquera pas de l'agacer...).

Pour corriger ces deux problèmes, il faut donc faire en sorte que l'adresse `bugs@warlordz.kill.windows` soit remplacée, lors de l'expédition du message par une adresse valide au sens de l'Internet, donc dûment déclarée.

Pour ce faire, nous supposerons que vous êtes abonné au fournisseur d'accès `fai.fr` (qui, lui, est enregistré et donc connu de tous les sites Internet). Lorsque vous vous êtes abonné, le fournisseur vous a octroyé un nom pour votre boîte aux lettres (selon les cas, c'est vous qui proposez le nom, ou bien lui qui vous l'impose). Admettons que cette adresse soit `jdupont@fai.fr` (ce qui, je vous l'accorde, est bien moins fun que `bugs@warlordz.kill.windows`... que les Dupont me pardonnent, mais ce n'est qu'un exemple).

Nous avons trois méthodes pour régler le problème :

- la première consiste à se plier aux exigences et, la mort dans l'âme, à rebaptiser sa machine `fai.fr` et à renommer l'utilisateur `bugs` en `jdupont` : ainsi, l'adresse d'expédition sera correcte. Pas terrible... et adieu la rébellion *against ze oueurlde* ! (il manque des fautes d'orthographe pour faire vrai...);
- la deuxième consiste tout simplement à faire enregistrer son domaine auprès d'un organisme dûment accrédité, et donc à dépenser les économies que l'on réservait à l'achat de la prochaine cartouche pour sa Gomme Baille Color (il existe également des possibilités pour enregistrer gratuitement un nom de domaine, si l'on dispose des ressources matérielles et des compétences nécessaires : voir le site www.eu.org). Cette solution implique également d'autres contraintes purement techniques et nous ne la détaillerons donc pas ici.
- la troisième, celle qui sera étudiée ici, consiste à configurer `postfix` pour qu'il réécrive l'adresse de l'expéditeur, qu'il n'importe l'adresse `bugs@warlordz.kill.windows` en `jdupont@fai.fr`.

5.2 La réécriture et le masquage du champ From :

`postfix`, comme son prédécesseur, permet de modifier le champ `From` , qui contient l'adresse de l'expéditeur. Par défaut, cette option n'est pas active et il va donc falloir modifier notre configuration pour l'utiliser.

Le principe est très simple :

- on indique à `postfix` qu'il doit utiliser une table de réécriture des adresses des expéditeurs. Cette table s'appelle *canonical* et sera placée dans `/etc/postfix/` (ou `/usr/local/etc/postfix/`), son format est décrit par la commande `man 5 canonical`. Dans notre cas, le fichier *canonical* devrait donc contenir :

```
bugs jdupont@fai.fr
```

- à partir de ce fichier, on génère un fichier au format DB avec la commande `postmap /etc/postfix/canonical`, comme on l'a fait pour le fichier des alias ;
- dans la section `ADDRESS REWRITING` (*Réécriture des adresses*), prévue à cet effet dans `main.cf` (et initialement vide), on ajoute la ligne suivante :

```
sender_canonical_maps = hash:/etc/postfix/canonical
```

- on force `postfix` à relire ce fichier en faisant `postfix reload`.

Et c'est tout... Il reste à notre ami Jean Dupont, euh... pardon, à *bugs* à envoyer un message de test (local ou à une adresse écho) afin de vérifier si la réécriture s'est bien effectuée.

6 Gestion des adresses locales

La gestion des adresses locales s'effectue très simplement par `postfix`. Dans la plupart des cas, vous n'aurez rien à faire. Ce qui suit ne doit donc être fait que dans certains cas bien précis.

Supposons que votre machine reçoive des courriers à destination de `gus@machin.uucp.fr` (ce qui peut être le cas si vous utilisez également UUCP pour recevoir du courrier). Le problème consiste maintenant à faire comprendre à `postfix` que les messages à destination des adresses `machin.uucp.fr` doivent être considérées comme locales à votre machine, sinon il essaiera de les renvoyer...

Pour ce faire, on utilisera un autre fichier de configuration : *transport* un peu comme on l'a fait pour *aliases*. La syntaxe de ce fichier est très simple et décrite par sa page de manuel (`man transport`). Dans notre cas, voici quel serait son contenu :

```
alex.linux-france.org  local:
localhost              local:
machin.uucp.fr         local:
```

Toutes les adresses à destination de `alex.linux-france.org`, de `machin.uucp.fr`, ainsi, bien sûr, que les adresses locales seront alors considérées comme du courrier local à la machine.

Comme *aliases*, le fichier *transport* doit être traité pour produire un fichier au format *db* :

```
# postmap /usr/local/etc/postfix/transport
```

Il reste maintenant à indiquer à `postfix` qu'il doit utiliser cette table pour acheminer le courrier. Pour cela, on rajoute dans *main.cf* la ligne suivante :

```
transport_maps = hash:/usr/local/etc/postfix/transport
```

qui aura priorité sur le contenu de la variable `$mydestination` (c'est pour ça que le contenu de celle-ci doit apparaître dans le fichier décrivant les adresses locales).

La lecture de la page de manuel vous donnera toutes les autres informations nécessaires pour, par exemple, préciser pour une entrée donnée un type de transport particulier : si, par exemple, vous désirez utiliser UUCP pour expédier vos courriers à destination de `truc.uucp.fr`, il suffira d'ajouter l'entrée

```
truc.uucp.fr      uucp:
```

7 Conclusion et remerciements

Ceux que `sendmail` rebutait peuvent essayer `postfix` pour le transport de leur courrier : sa syntaxe est claire et l'ensemble est richement commenté. De plus, ses concepteurs ont pris soin de garder une compatibilité avec son géant de prédécesseur : sur ma machine, tous les programmes qui utilisaient `sendmail` pour lire et envoyer le courrier n'ont eu besoin d'aucune modification pour continuer à fonctionner, car cette compatibilité est un critère important pour les concepteurs de `postfix` (et essentielle pour espérer le remplacer).

Je ne suis pas suffisamment expert pour comparer les fonctionnalités avancées des deux programmes mais, dans le cadre d'une utilisation personnelle, en machine isolée, aucune fonctionnalité de `sendmail` ne m'a manqué. Il faudrait voir ce que cela donne sur une configuration lourde (gestion du courrier sur un réseau découpé en sous-réseaux, gestion des listes de diffusion, etc.). La lecture du forum de discussion `fr.comp.mail` vous en apprendra plus sur ce sujet.

Parmi ce qui manque – bien que peu de particuliers en aient besoin – ce sont les réécritures via des expressions rationnelles, par exemple. Ce sera probablement implanté dans une future version.

Cet article est évidemment incomplet, imprécis et contient probablement des erreurs grossières... Je remercie donc d'avance tous ceux et celles qui prendront la peine de lire ce document, de le corriger et de l'annoter. J'essaierai de faire en sorte qu'il suive les évolutions futures de `postfix` (et celle de ma connaissance du transport de courrier) et toute aide est la bienvenue dans le contexte de cet article qui se veut simplement un guide d'initiation.

La version actuelle de ce document a été soumise à la sagacité d'Olivier Robert, de Nat Maka-révitch et d'Olivier Tharan. Je les remercie particulièrement de leur aide en n'oubliant pas les contributeurs zélés de `fr.comp.mail`.