Mise en oeuvre d'une messagerie avec POSTFIX

Cas d'un serveur de messagerie d'un Laboratoire

Centre d'Océanologie de Marseille
UMS 2196 CNRS

Maurice.Libes@com.univ-mrs.fr

GERET Strasbourg - 28 Mars 2003

Introduction

- Architecture du système : les binaires, les queues
- Configuration, administration de Postfix
  - Exemple de config : un serveur, un « null » client
  - Sécurité :
    - Limiter le Relayage,
    - Contrôles anti spam : filtrage d'adresses
    - mise en cage
    - limiter les dénis de service :contrôles de charge
    - couplage avec un antivirus
- Conclusions : intérêts de Postfix

Introduction : Pourquoi Postfix?

- écrit et maintenu par wietse Venema

  “Postfix attempts to be fast, easy to administer, and hopefully secure, while at the same time being sendmail compatible enough to not upset your users.”

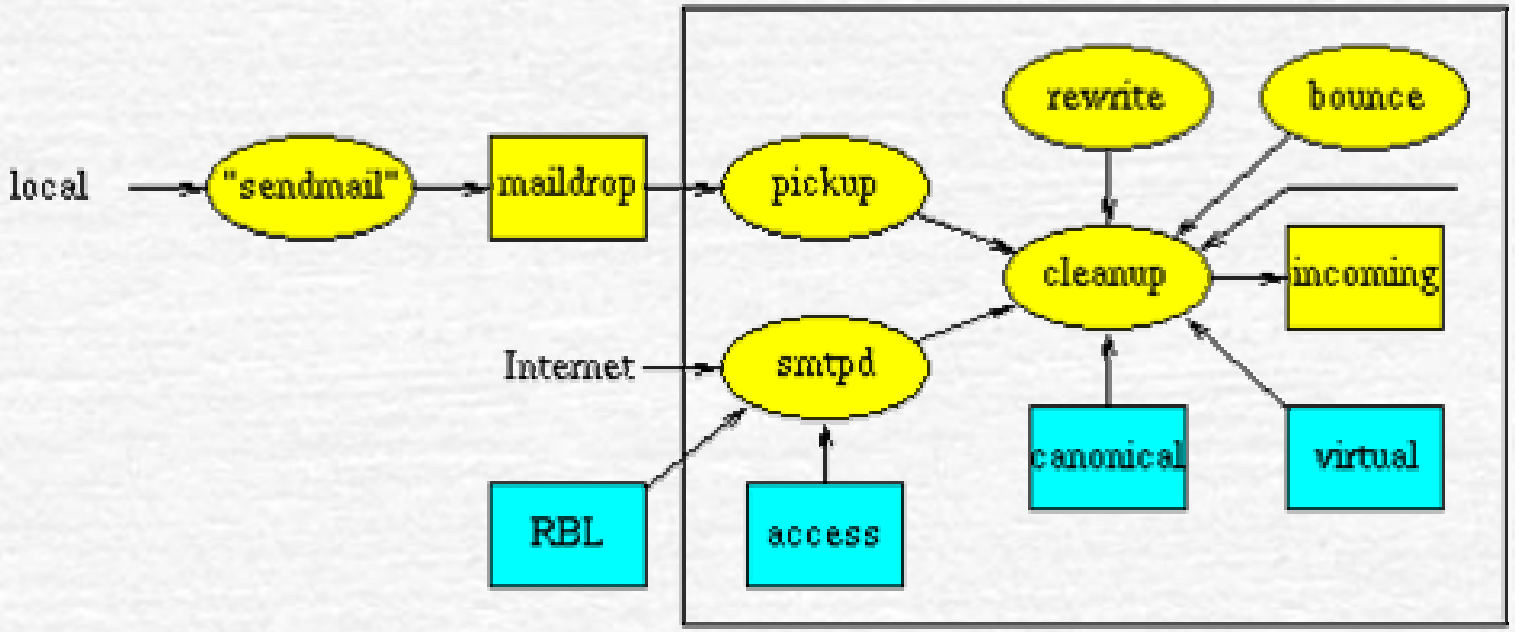
- Se présente comme une alternative à sendmail *(canal historique)* apportant nativement des fonctionnalités intéressantes
- Conçu pour
  - apporter une bonne compatibilité avec sendmail au niveau de la ligne de commande, des fichiers de conf (alias, .forward) et les queues
  - Apporter une simplicité d’administration par une configuration facile à comprendre : *un fichier de conf, des variables « parlantes »*
  - Être rapide et sûr, par une architecture modulaire : *plusieurs binaires (semi résidents en mémoire), et différentes queues de gestion des mails*

Postfix au Centre d'Océanologie de Marseille

- Un laboratoire CNRS - environ 250 personnes sur 2 sites
- Un mailhost central (MX) qui redistribue sur un site « feuille »
  - Gestion des boites par POPs + IMAPs
  - Postfix couplé avec Amavis + antivirus Sophos
  - Serveur SYMPA
- Environ 10000 mails /jour en moyenne (entrées + sorties)
- Pentium III 600Mhz : 512 Mo RAM, 2 disques 9Go
  - Charge CPU assez importante en pointe (40% idle)

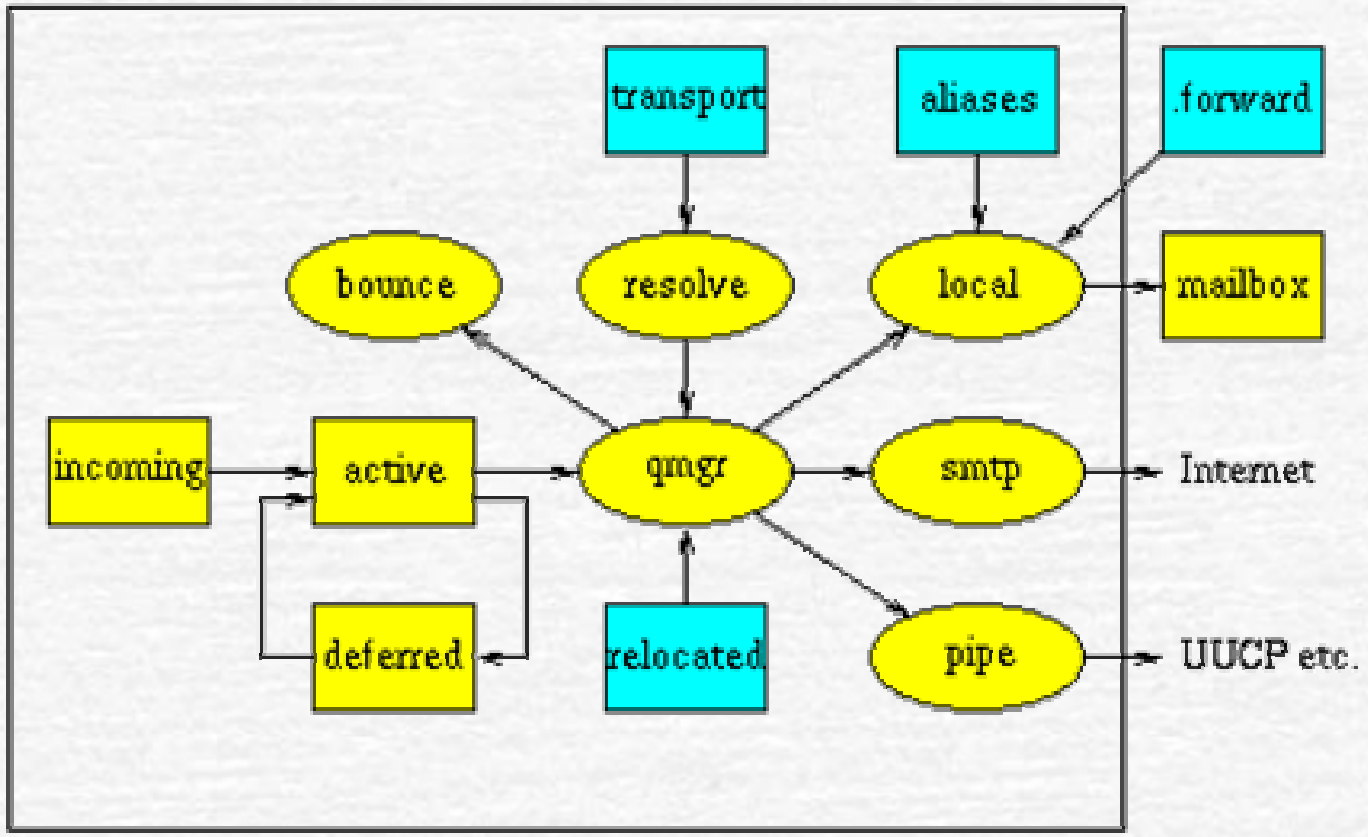
Postfix :architecture du système

- Pas d'architecture monolithique... Postfix est composé d'une dizaine de process assurant chacun une tâche et communiquant entre eux par socket ou queues
- Réception de mail



Postfix :architecture du système

- Délivrance et envoi de mail



Postfix : phase 1 Réception/Insertion

- *Le daemon « master » contrôle le lancement des tous les autres daemons*

- ***Sendmail / postdrop*** : créé pour l'homonymie et compatibilité de l'interface avec *sendmail « CH » :* recoit les messages des utilisateurs locaux
  - Dépose le message dans la queue « maildrop »
  - Via l'intermédiaire du daemon « *postdrop* »
  - Quelques compatibilités avec sendmail:
    - Sendmail –bp ⇔/usr/bin/mailq -> ../sbin/sendmail
    - Sendmail –i ⇔/usr/bin/newaliases -> ../sbin/sendmail

- ***Postdrop*** : dépose un mail dans la queue « *maildrop* » pour ne pas que */var/spool/postfix/maildrop* soit « *world-writeable* »

Postfix : phase 1 Réception/Insertion

- **Smtpd** : réception de mail d'hôtes distants par smtp. Lancé par « master » « smtpd » recoit les connexions réseau (port 25) et engage la transaction smtp. Chaque message reçu est envoyé au daemon « ***cleanup*** » et le message entrant est déposé dans la queue « ***incoming*** »

- Réalise les premiers contrôles (directives de « main.cf)
  - *content_filter* : possibilité de passer le message à un process qui va analyser le contenu (*amavis*) et rejeter le message ou le réinjecter dans la chaine postfix.
  - *message_size_limit* : taille max des messages
  - *smtpd_recipient_limit* : nombre max de destinataires dans un message
  - Contrôle sur les phases *HELO*, *Mail From:, RCPT to*: en fonction de fichiers indexés

Postfix : phase 1 Réception/Insertion

- ***pickup***: simple dépilage de la queue « *maildrop* », réceptionne les messages et les envoie au daemon « *cleanup* »
- Tourne sous root, mais sans aucune interaction avec l'extérieur. Peut être « *chrooté* » dans *master.cf*
- Prend en charge également la directive *content_filter* de « *main.cf* » qui permet de passer le mail à un programme extérieur pour analyse (*amavis, spamassin*) avant de le réinjecter dans le circuit, ou le jeter!

Postfix : Phase 2 formatage des entêtes

- ***Cleanup + trivial_rewrite*** : vérification de toutes les entêtes smtp avant de le déposer localement. Examen de Conformité au RFC822. Canonicalise les messages reçus avant de les déposer dans la queue « *incoming* », puis avertit le gestionnaire de queue *qmgr* si tout est correct.
  - Insère headers manquants : (Resent-) From: Message-Id: Date:
  - Extrait les adresses de destination de l'enveloppe To: Cc: Bcc:
  - Accès aux tables des aliases et reverse aliases
  - Élimine les doublons dans les adresses de destination
  - Traite les directives de « main.cf » d'examen des headers et du body et de masquerade d'adresses
    - *Body_checks, Header_checks, Masquerade_domains*
- ***Trivial-rewrite*** *:* appelé optionnellement si les adresses ne sont pas FQDN. destiné à réécrire les adresses au format FQDN nom@hote.domaine

Postfix : Phase 3 stockage – livraison - envoi

- ***Qmgr*** : dépile les messages de « *incoming* » et les remet à un agent de livraison en fonction de l'adresse de destination
  - « *local* » | « *procmail* » délivrance locale */var/spool/mail*
    - Gestion du *.forward*
  - *smtp* : résolution DNS de la destination et envoi vers l'extérieur par smtp
  - *Pipe* : envoi des messages vers un autre programme externe (amavis)
  - *Bounce* : gestion des messages non délivrables : empilement dans la queue bounce, et envoie de message à l'émetteur.

Postfix : Les queues de mail

- Répartition intelligente des mails selon leur état de progression dans */var/spool/postfix/*
- →Reprise après crash sans aucun problème
  - *Maildrop* : dépot des messages émis localement
  - *Incoming* : dépot des messages entrants émis localement + extérieur smtp
  - *Active :* mail prêts...en cours de délivrance locale par *qmgr*, taux de dépot contrôlé et limité
  - *Deferred :* mail ne pouvant pas être délivrés tempo-rairement (*mailq* affiche le status de la queue “defer”)
  - Bounce : messages d’erreur, livraison impossible

- Gestion intelligente des files pour préserver les ressources de la machine: *leaky bucket, fairness, exponential backoff, slow start ...*

Postfix : les fichiers de conf

- 2 fichiers de configuration principaux:
  - ***/etc/postfix/main.cf*** et ***/etc/postfix/master.cf***
- Des fichiers indexés (lookup table) pour rechercher des correspondances (pas de langage de réécriture d'adresses)
  - *Aliases ml: libes@com.univ-mrs.fr*
  - *sender_canonical_maps* (=reverse aliases)
  - *Access* (=liste noire)
  - *Protected* : liste des listes internes à protéger (all@labo.univ-xxx.fr)
  - *Insiders* : les copains extérieurs autorisés à utiliser nos listes internes
  - *Header.regexp, body.regexp :* recherche de motifs dans les header ou le body
  - *Relocated* : liste d'utilisateurs ayant changé d'adresse

Postfix : Les utilitaires d'administration

- *Postfix* [start | stop | reload |check | flush]
- *Postmap* : création des fichiers indexés (.db .dbm)
- *Postconf* : utilitaire d'affichage ou configuration de main.cf
- *Postalias* : maintenance des tables d'alias (compatible sendmail)
- *Postcat* : affichage lisible des mails dans les files
- *Postsuper* : gestion et maintenance des queues de mail (purge après crash)
- *Postlog* : envoyer un message à syslog

Postfix : configuration de base dans « main.cf »

- Indiquer d'où viennent les mails (masquerade From:
  - myhostname = mailhost.labo.univ-mrs.fr
  - mydomain = labo.univ-mrs.fr
  - myorigin = $myhostname *(par défaut)*
  - myorigin = $mydomain *(souvent plus adapté)*
  - masquerade_domains = $mydomain
- Indiquer les messages conservés localement
  - mydestination = $myhostname localhost.$mydomain www.$mydomain ftp.$mydomain (cas où le serveur a plusieurs CNAME ou A record)
- Le relayage : qui autoriser?
  - mynetworks = subnet | host | 139.124.128.0/22, 127.0.0.0/8
  - mynetwork_style = subnet (par défaut)

Postfix : configuration de base

- Un « NULL » client :

Postfix d'une machine qui ne conserve rien… se contente de tout renvoyer vers le mailhost officiel du site.

- myhostname = pcml.com.univ-mrs.fr
- mydomain = com.univ-mrs.fr
- myorigin = $mydomain
- **relayhost** = mailhost.com.univ-mrs.fr
- mynetworks = 127.0.0.0/8 139.124.2.0/24
- mydestination = (rien)

Postfix : Sécurité

- Aucun binaire tournant avec le bit suid, Pas d'exécution sous l'uid root
- Prémices du filtrage AntiSPAM : filtrage d’adresses au niveau de chaque phase de la transaction SMTP
  - Contrôle lors de la connexion du client sur le port 25
  - Contrôle sur HELO
  - Contrôle sur le « Mail From »
  - Contrôles sur les différents Headers par regexp (Subject)
- Possibilité de mise en cage (chroot) de Postfix
- limitation des dénis de service par contôle de charge

Postfix : premiers contrôles antispam

1. Contrôle sur le client SMTP *(celui qui initie la connexion)*

smtpd_**client**_restrictions = permit_mynetworks,

hash:/etc/postfix/access, [rejet si liste noire locale]

reject_maps_rbl [si présence dans $maps_rbl_domains]

reject_unknown_client *[client sans enregistrement PTR dans le DNS]*

maps_rbl_domains = blackholes.mail-abuse.org, dialups.mail-abuse.org, relays.mail-abuse.org

2. Sur le HELO

smtpd_helo_required = **yes** *(no par defaut)*

smtpd_helo_restriction = reject_invalid_hostname, (adresse malformée)

reject_unknown_hostname, (adresse HELO sans A ou MX record)

reject_non_fqdn_hostname (adresse HELO non FQDN)

Postfix : premiers contrôles antispam

- 3. Contrôles sur le « sender » mail from:

  smtpd_**sender**_restrictions = hash:/etc/postfix/access,
  reject_unknown_sender_domain, reject_non_fqdn_sender,
  check_sender_access hash:/etc/postfix/access, reject_maps_rbl

- 4. Contrôles sur le destinataire rcpt to: (essentiellement pour le relayage)

  smtpd_**recipient**_restrictions = hash:/etc/postfix/protected,

  check_relay_domains [verifie que le message est à destination de $mydestination ou $relay_domains]

  smtpd_restriction_classes = insiders_only [protection des listes internes]

  insiders_only = check_sender_access, hash:/etc/postfix/insiders, reject

- 5. Filtrage sur des motifs (regexp) des header
  - header_checks = regexp:/etc/postfix/header.regexp
    - /^Content-Type: multipart.*"----[A-F0-9]+_Outlook_Express_boundary"/i REJECT
    - /^Subject:.*\$[0-9]+.*$/ REJECT

Postfix : Directives de contrôle de charge

Un luxe de directives pour réguler la charge du système et lutter contre les dénis de services:

- limitation du nombre de process
  - *default_process_limit* (défaut: 50) : contrôle le taux d'entrées sorties de mails via le nombre de process concurrents de postfix (smtp, smtpd)
- régulation du nombre de connexions simultanées
  - *local_destination_concurrency_limit* : max de délivrance simultanées locale identique
  - default_destination_concurrency_limit : nombre max de mail simultanés vers un site distant (protection des sites extérieurs)
- limitation du nombre de destinataires par mail
  - *smtpd_recipient_limit* : nombre maximal de destinataires dans un mail
- gestion des tentatives de réémission vers sites inaccessibles
  - *maximal_queue_lifetime* (5 jours) : temps de rétention avant que le message soit déclaré non délivrable
  - *queue_run_delay* (1000s) : fréquence de retetatives
- pénalisation des clients envoyant du SPAM

Postfix :contrôle de ressources mémoire

Postfix possède des directives de limite d'utilisation de la mémoire afin de limiter les dénis de services.

« The idea is to keep running under conditions of stress, without making the problem worse. »

- message_size_limit = 5000000
- Header_size_limit = 2048
- line_length_limit = 102400
- bounce_size_limit = 50000

Postfix : Directives de réécriture d’adresses

Pas de langage complexe de réécriture. Tout se fait à travers des tables de recherche indexées (lookup table)

- réécriture en prénom.nom
  - sender_canonical_maps = /etc/postfix/revaliases
- masquage d ’adresse
  - masquerade_domains = com.univ-mrs.fr
- redirection d ’adresses virtuelles
  - virtual_alias_maps = hash:/etc/postfix/virtual

Postfix Protection des listes de mail

- Protection des listes de mail
  - smtpd_recipient_restrictions = hash:/etc/postfix/protected,
  - smtpd_restriction_classes = insiders_only
  - insiders_only = check_sender_access, hash:/etc/postfix/insiders, reject

- **Dans /etc/postfix/insiders_only**

m.libes@wanadoo.fr OK
Judas.bricot@free.fr OK
Gerard.manvussa@lseet.univ-tln.fr OK

- **Dans /etc/postfix/protected**

info-soc@com.univ-mrs.fr	insiders_only
liste-ita@com.univ-mrs.fr	insiders_only
Labo@com.univ-mrs.fr	insiders_only

Postfix+Amavis : Couplage avec Antivirus

Utilisation de la possibilité d'envoyer le message à un programme extérieur pour analyse (daemon « pipe » de postfix)

1. Installer un antivirus tournant sur unix
2. Installer amavis *http://amavis.sourceforge.net/*
3. Dans */etc/postfix/main.cf*
   - *content_filter = vscan:* *[vscan est un le nom du service que l'on va utiliser]*

4. Dans */etc/postfix/master.cf*
   - *##service type private unpriv chroot wakeup maxproc command + args*
   - *smtp inet n - n - - smtpd*
   - **vscan** unix - n n - 10 **pipe** user=vscan argv=/usr/sbin/amavis ${sender} ${recipient}
   - localhost:10025 inet n - n - - smtpd -o content_filter=

```
$ lsof -i:10025
master  24539 root   66u  IPv4 14213601      TCP localhost.localdomain:10025 (LISTEN)
```

- *http://www.com.univ-mrs.fr/ssc/info/cours/install-amavis-postfix.html*

Postfix : environnement chrooté

- mkdir /var/spool/postfix/etc
- mkdir /var/spool/postfix/lib
- mkdir -p /var/spool/postfix/usr/lib/zoneinfo
- cp /etc/localtime /etc/services /etc/resolv.conf /etc/nsswitch.conf ~etc
- ln –s /etc/localtime ~usr/lib/zoneinfo
- cp /lib/libnss_* ~lib

```
# ==========================================================
# service type  private unpriv  chroot  wakeup  maxproc(50) command + args
# ==========================================================
smtp      inet  n       -       y       -               -               smtpd
pickup    fifo  n       -       y       60      1               pickup
cleanup   unix  n       -       y       -       0               cleanup
#qmgr     fifo  n       -       n       300     1               qmgr
qmgr      fifo  n       -       y       300     1               nqmgr
rewrite   unix  -       -       y       -       -               trivial-rewrite
bounce    unix  -       -       y       -       0               bounce
defer     unix  -       -       y       -       0               bounce
flush     unix  n       -       y       1000?   0               flush
smtp      unix  -       -       y       -       -               smtp
```

Conclusions : Pourquoi Postfix?

- Simplicité d'administration *(majorité d'options par défaut!)*
- configuration facile à comprendre : *un fichier de conf, des variables « parlantes »*
- fonctionnalités intéressantes (filtrages, contrôles de charge, collaboration avec autres programmes externes)
- *Excellente tenue en charge en cas de déni de service*
- Architecture modulaire : *plusieurs binaires , et queues de gestion des mails*
- Sécurité prise en compte nativement : *relayage, mise en cage, contrôle antispam basique, pas de binaires setuid, ...*

Conclusions : Pourquoi Postfix?

- bonne compatibilité avec les fichiers de conf de sendmail, (alias, .forward)
- Excellente portabilité sur plusieurs plateformes
- Pas de mauvaises surprises en exploitation, ça fait ce que ça dit...
- Exploitation sans problème depuis 4ans : ça marche tout seul
- Large communauté d'utilisateurs et de développeurs : Beaucoup de logiciels complémentaires

→*http://www.postfix.org*