Sendmail X

La nouvelle génération de sendmail

José-Marcio Martins da Cruz

Ecole des Mines de Paris

JRES 2005 – Marseille

Jose-Marcio.Martins@ensmp.fr - http://j-chkmail.ensmp.fr

Plan

- L'histoire de sendmail
- Sendmail 8
- Sendmail X – Vue générale
- Mise en oeuvre
- Quelques astuces
- Conclusions

sendmail : son histoire

- 1979 – delivermail – routage de messages à l'Université de Berkeley
- 1981 – première version de sendmail
- Des temps sombres – développements parallèles (IDA et KJS)
- 1991 – Kit de Configuration de Jussieu – Pierre David et Jacky Thibault
- 1992 – Eric Allman reprend le développement – début sendmail 8
- 1999 – Création Sendmail Inc. -> version open source/version commerciale
  - STARTTLS, AUTH, libmilter (FFR), ...
  - Version libre : Eric Allman, Gregory Shapiro, Claus Assman et Murray Kucherawy
- 2001 – sendmail 8.12.0
  - libmilter (version officielle), libsm, sendmail ne tourne plus en tant que *root*, ...
- 2002 – Début développement sendmail X – Claus Assmann

sendmail 8

- sendmail 8.1 : 07/Jun/1993 (sendmail 8.6 : Oct/1993)
- Fichier de «configuration» contient :
  - Configuration
  - Programmes – les paramètres sont calculés par le binaire et les décisions prises par le programme dans le fichier de configuration.
- Un processus en écoute plus processus serveurs fils (un par connexion) -> fork !
- Traitement périodique de la file de messages :
  - Enregistrement des messages sur deux fichiers : enveloppe et contenu
  - stratégie d'ordonnancement par défaut peu efficace
- API libmilter pour traitements externes : callbacks pour chaque «étape» du protocole SMTP (versions >= 8.12.0)
- Qualité du code : http://www.sendmail.org/CodingStandard.html

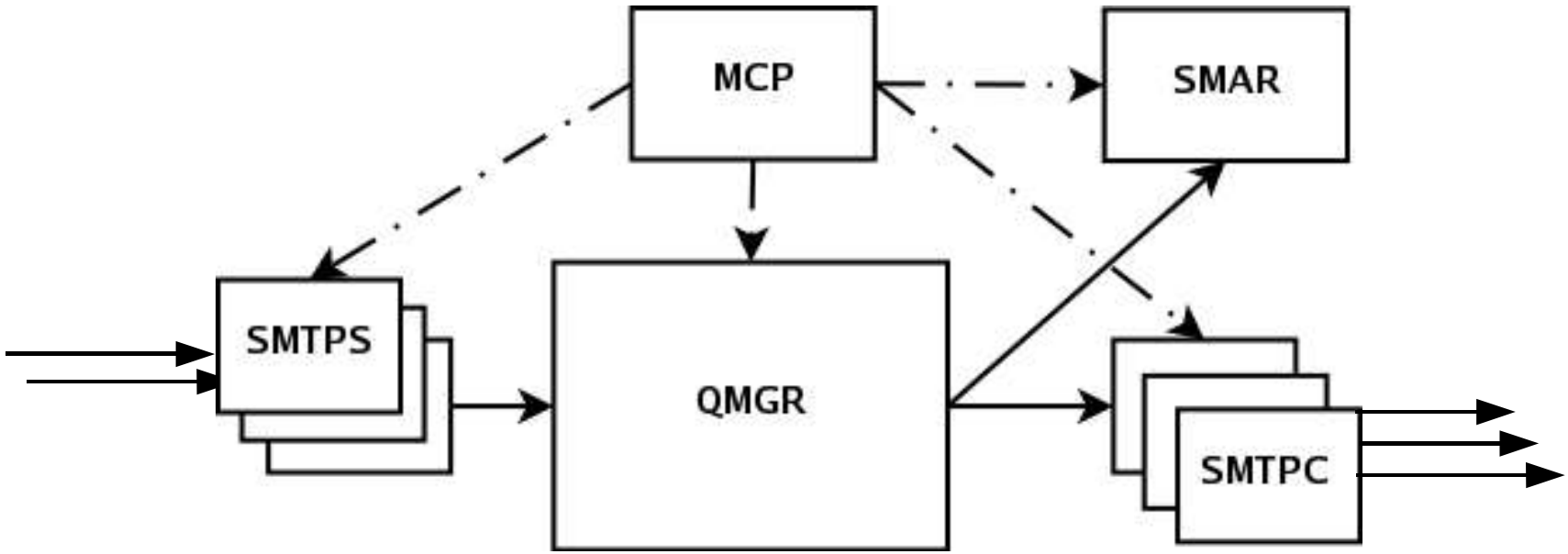
sendmail X – Pourquoi ?

- sendmail 8 a été conçu il y a 13 ans déjà
  - Monolithique – trop de fonctions intégrées dans le MTA lui-même
  - Configuration difficile
  - Technologie dépassée
  - Mais toujours capable de satisfaire les besoins les plus courants
- Une nouvelle durée de vie d'au moins 10 ans
- Capable de s'adapter aux nouvelles technologies pendant son cycle de vie
- Compatibilité avec sendmail 8 ??? – ce n'est pas une contrainte
  - sm 8 a été créé avec les besoins du début des années 90...
  - Dans la mesure du possible donner les moyens de faire autrement
  - Ex : redirection (fichier **.forward)**

sendmail X – Caractéristiques de projet

- Contraintes
  - Sécurité – impossibilité d'introduction et de compromission du serveur
  - Fiabilité – pas de perte de messages pour des raisons «frivoles»
- Objectifs
  - Robustesse – les conséquences des défaillances restent locales
  - Flexibilité – on peut ajouter ou remplacer des modules
  - Scalabilité – doit pouvoir profiter des améliorations matérielles
  - Extensibilité – ajouter des nouvelles fonctionnalités
  - Maintenabilité – détection et correction d'erreurs facile, comportement prévisible
  - Portabilité – doit fonctionner sur tous les OS compatibles POSIX + Windows (???)
  - Tests – procédure de teste automatique lors de la compilation

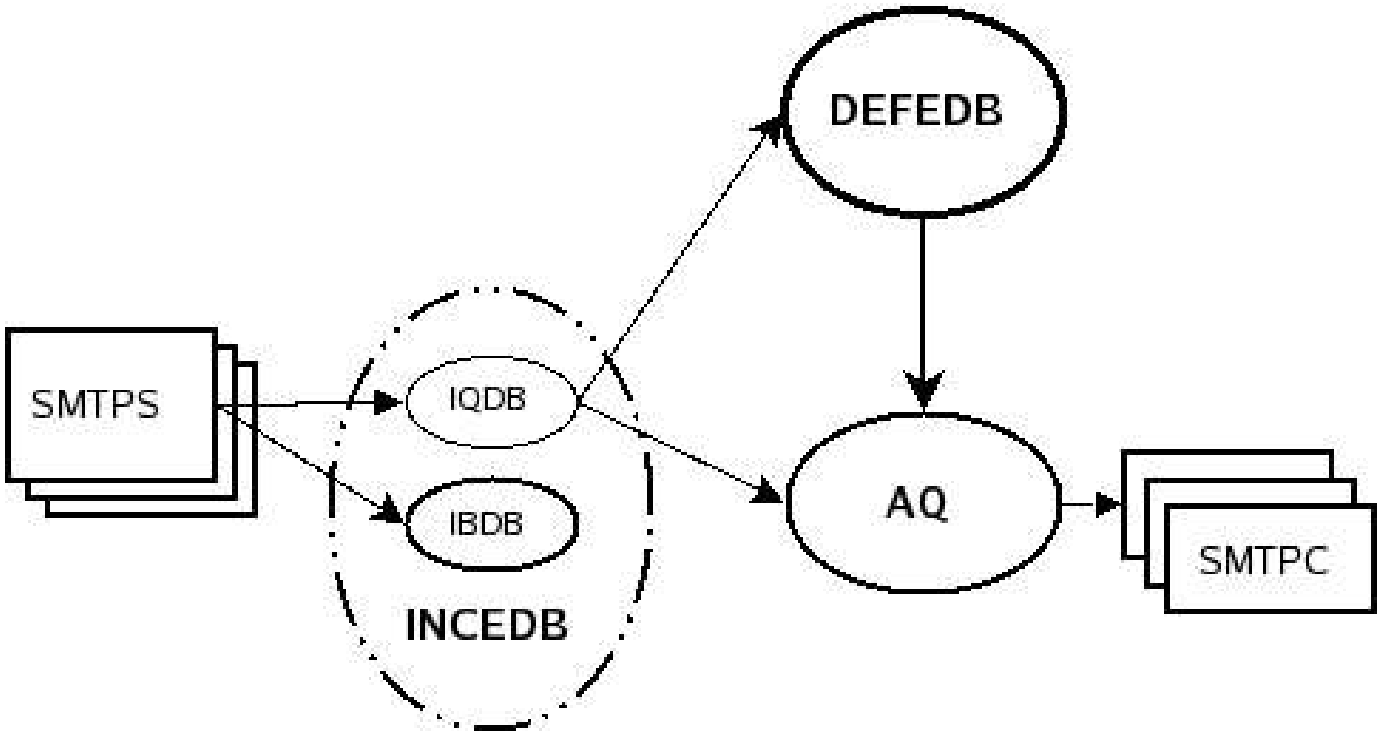
sendmail X - architecture



sendmail X – internals...

- Threads :
  - Statethreads pour smtps et smtpc, pthreads pour les autres
  - Event threads -> implémente «pool of workers»
- Abstractions
  - Maps (tables) : hash, socket, passwd, ...
  - RCBs (communication asynchrone entre modules)
  - ...
- libmta (équivalent de libsm sous sendmail 8)
- Qualité
  - http://www.sendmail.org/CodingStandard.html

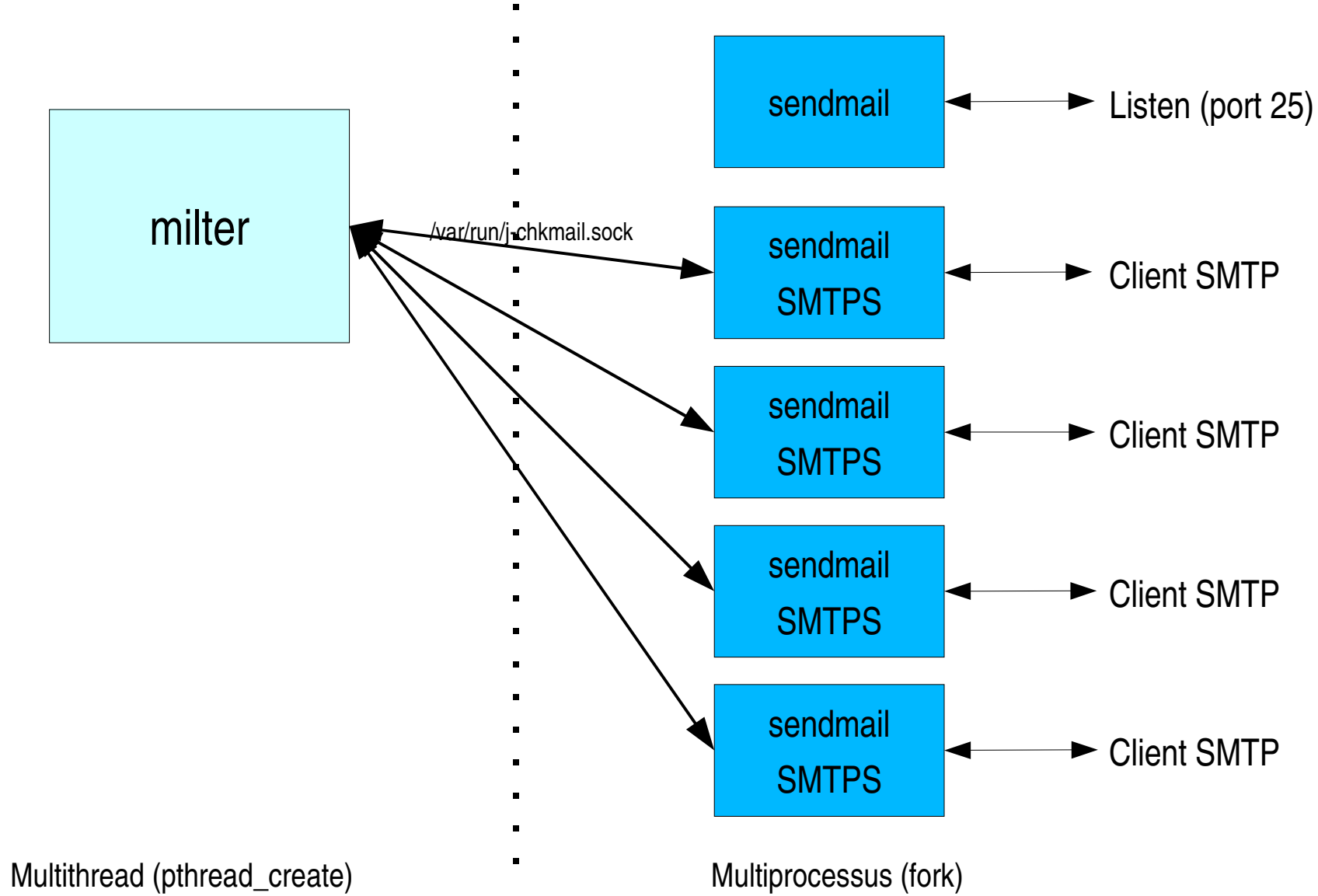
sendmail X – Gestion de la file de messages



sendmail X – Gestion de la file de messages

- Gérée par des événements (échéancier) et non pas par traitement périodique
- Traitement d'un message arrivant
  - Tentative immédiate de remise puis mise dans file différée, si échec
- Deux niveaux d'ordonnancement :
  - Macro – gère l'échéancier (passage DEFEDB -> AQ)
  - Micro – ordre du traitement des messages dans la file active
- Stockage de la file (sur disque) :
  - EDB (enveloppe database) : table de hachage (BerkeleyDB)
  - CDB (content database) : un fichier texte par message

sendmail 8 - libmilter



sendmail X - libpmilter

- Différences principales
  - Un seul canal de communication entre *smtps* et *pmilter* – les transferts sont multiplexés sur un seul descripteur de fichier
  - Sendmail X utilise un modèle événementiel («pool of workers»), tandis que sendmail 8 utilise un thread par connexion SMTP active.
- Limitations actuelles de la libpmilter (seront levées sur la prochaine version)
  - Il n'est pas encore possible de modifier le corps d'un message (y compris ajouter ou supprimer des en-têtes)
  - L'API n'est pas encore compatible avec celle de sendmail 8. Une couche de compatibilité est prévue.
  - Conséquence : les solutions de filtrage pour sendmail 8 ne sont pas portables sur sendmail X sans modification.

sendmail X - benchmark

- Capacité de relayage de 108 messages/seconde
  - Sun E450 (4 x 400 Mhz)
  - 100 K messages – 4000 caractères
  - 100 sessions simultanées (10K sessions - 10 messages par session)
- Pas de comparaison possible avec les benchmarks existants
  - Nombre de messages et connexions simultanées bien plus petit (risque d'écroulement)
  - http://www-dt.e-technik.uni-dortmund.de/~ma/postfix/vsqmail.html
- Besoin d'un benchmark avec mesure en charge soutenue au lieu de rafale de messages

Configuration de sendmail X

- smx.conf – fichier de configuration principal
  - Une section par instance de module :

    lmtp { ... }

    smtps MTA-EXT { ... }

    smtps MTA-INT { ... }

    smtpc { ... }

    qmgr { ... }

    smar { ... }
- aliases.db - inclut aliases et virtusertable dans sm8
- mt.db - mailertable dans sm8
- access.db - équivalent de access.db de sm8
- qmgr_conf.db - nouveau
- Autres tables - à venir ou définies par l'utilisateur

Configuration – smx.conf

```
smtps MTA-EXT {
        log_level = 11;
        log {
                facility = mail; ident = "MTA-EXT";
        }
        CDB_gid = 262;
         listen_socket {
                address = 194.214.158.1; type = inet;  port = 25;
        }

        start_action = pass;
        pass_fd_socket = smtps/mtaextfd;
        user = smxs;
        path = "/opt/smx/libexec/smtps";
        arguments = "smtps -I 2 -N MTA-EXT -f /etc/smx/smx.conf";

        flags = { access };

        pmilter {
             socket { type = inet; port = 2001; address = 127.0.0.1; }
        }
}
```

Configuration – smx.conf

```
qmgr {
        smtpc { initial_connections = 10; max_connections = 20; }

        queue_return_timeout = 4d;
        retry_min_delay  = 10m;
        retry_max_delay = 4h;

        control_socket = qmgrctrl.sock;

        log_level = 12;
        log { facility = local6; ident = "qmgr"; }

        wait_for_server = 4;
        wait_for_client = 3;
        start_action = wait;
        user = smxq;
        restart_dependencies = { smtps, smtpc, smar };
        path = "/opt/smx/libexec/qmgr";
        arguments = "qmgr -f /etc/smx/smx.conf";
        conf="qmgr_conf.db";
}
```

Configuration – smx.conf

```
smar {
        log_level = 12;
        log { facility = local6;  ident = "smar"; }

        nameserver = { 127.0.0.1, 194.214.158.200 };

        map lusers { type = hash; file = "/etc/smx/localuser.db"; }
        map password { type = passwd; }
        map valid_users { type = sequence; maps = { password, lusers };}

        local_user_map { name = valid_users; flags = { implicitly_match_detail };}

        map smx_map { type = socket; mapname = aliases; path = /var/run/sockmap.sock; }
        aliases {
                name = smx_map;
                flags = { local_domains, implicitly_match_detail, replace_macros }
        };

        start_action = wait;
        user = smxm;
        restart_dependencies = { smtps, qmgr };
        path = "/opt/smx/libexec/smar";
        arguments = "smar -f /etc/smx/smx.conf";
}
```

Mise en oeuvre
Serveur d'arrivée

- Dans /etc/smx/smx.conf

```
smar {
    map utilisateurs { type = passwd; }
    local_user_map { name = utilisateurs; flags ={ implicitly_match_detail }; }
}
```

- Dans /etc/smx/mt

```
math.universite.fr          lmtp:
```

- OBS :
  - On peut aussi avoir des destinataires définis dans la table *aliases* et non pas dans la table «*utilisateurs*». Dans ce cas, le LDA doit être capable de le traiter. Ex :

```
un_alias    :    local:
```

Mise en oeuvre
Serveur d'arrivée avec Cyrus IMAP

- Dans /etc/smx/smx.conf

```
smar {
        map lusers { type = hash; file = "/etc/smx/localusr.db";}
        local_user_map { name = lusers; flags ={ implicitly_match_detail }; }
}
smtpc { LMTP_socket="lmtpsock"; }
```

- Dans /etc/smx/mt

```
        # si on a défini une socket UNIX dans la section smtpc de smx.conf
        math.universite.fr                  lmtp:
        # si l'on veut une socket INET
        math.universite.fr                  2001^lmtp:[127.0.0.1]
```

- La table /etc/smx/localusr.db est créée, par exemple, avec l'aide du script :

  http://www.sendmail-fr.org/smx/scripts/imap2users

Mise en oeuvre - Passerelle

- Dans /etc/smx/access

to:@math.universite.fr	relay
to:@physique.universite.fr	relay
cltaddr:10	quick:relay

- Dans /etc/smx/mt

math.universite.fr	esmtp:[10.1.0.1]
physique.universite.fr	esmtp:[10.2.0.1]

- Obs : il faut :
  - Soit autoriser le relayage vers les serveurs internes (cas présent)
  - Soit déclarer une liste d'utilisateurs valables dans la base *access*
- Problème : pas facile de valider les adresses dans cette config.

Mise en oeuvre
Passerelle avec validation des adresses

- Dans /etc/smx/smx.conf

```
smar {
    map math_users { type = socket; mapname = users; address=10.1.0.1; port=2101; }
    map phys_users { type = socket; mapname = users; address=10.2.0.1; port=2101; }
    map valid_users { type = sequence; maps = { math_users, phys_users }; }
    local_user_map { name = valid_users; flags = { implicitly_match_detail }; }
}
```

- Dans /etc/smx/mt

```
math.universite.fr          esmtp:[10.1.0.1]
physique.universite.fr      esmtp:[10.2.0.1]
```

- Obs :
  - Voir exemple de serveur de table du type socket dans *contrib/socketMapServer.pl*
  - Pour éviter des requêtes multiples, on peut utiliser une table locale (de type socket) pour rédiriger les requêtes selon le sous-domaine, ou même convertir la requête en une requête LDAP, mySQL, ...

Mise en oeuvre - Serveur d'arrivée et de stockage sur des machines différentes

- Dans /etc/smx/smx.conf

```
smar {
        map lusers { type = passwd;}
        local_user_map { name = lusers; flags ={ implicitly_match_detail }; }
        map smx_map { type = socket; mapname = aliases; path = /var/run/sockmap.sock; }
        aliases { name = smx_map; flags = { implicitly_match_detail, replace_macros } };
}
smtpc { LMTP_socket="lmtpsock"; }
```

- Dans /etc/smx/mt

```
ccr.universite.fr          2001^lmtp:[10.1.0.1]
math.universite.fr         2001^lmtp:[10.2.0.1]
physique.universite.fr     2001^lmtp:[10.3.0.1]
```

- La «socket table» *smx_map* peut s'inspirer du script
  - http://www.sendmail-fr.org/smx/scripts/imap2users + *contrib/socketMapServer.pl*

Comment faire ? Blacklistes

- Blacklistes (RBLs)
  - Dans smx.conf :
    - smar { dnsbl { domain = mail-abuse.org; tag = mail-abuse; } }
  - Dans access database :

mail-abuse:127.0.0.1	error:550 5.7.1 listed at mail-abuse.org as open relay
mail-abuse:127.0.0.2	error:550 5.7.1 listed at mail-abuse.org as spam source
mail-abuse:127.0.0.9	error:451 4.7.1 listed at mail-abuse.org as suspicious
mail-abuse:temp	error:451 4.7.1 temporary lookup failure at mail-abuse.org

Comment faire ? Greylisting

- Seul l'adresse IP est utilisé
- Dans /etc/smx/smx.conf

```
smtps { flags = greylisting };
smar {
    greylisting {
        grey_wait = 10m; grey_expire = 1d;
        white_expire = 15d; netmask = 0xFFFFFF00;
    }
}
```

- Dans /etc/smx/access

```
cltaddr:10          OK
```

Comment faire ? Adresses protégées

- Dans /etc/smx/aliases :

  mes_amis : <jean@domain> <claude@domain> <pierre@domain>

  mes_eleves : <marc@domain> <jean@domain>

  tous_labo : <jean@domain> <claude@domain> <pierre@domain> <fabrice@domain>

- Dans /etc/smx/smx.conf

  smtps { protect_recipients { allow_by {sender, client_IP}}}

- Dans /etc/smx/access

  # tous les amis peuvent envoyer

  protectrcpt:mes_amis list:<mes_amis@domain>

  # que le maître

  protectrcpt:mes_eleves from:<moi@domain>

  # en interne uniquement

  protectrcpt:tous_labo cltaddr:10

Alors, on ne peux pas tout faire ?

- Fonctionnalités supprimées
  - Je veux mon fichier .forward !!!
    - Utilisez le script http://www.sendmail-fr.org/smx/scripts/forward2aliases.pl pour récolter les fichiers .forward dans les répertoires HOME ou répertoire commun et les ajouter dans la table des aliases
  - ...
- Fonctionnalités qui seront ajoutées plus tard
  - La commande Mail ne marche plus ! Je ne peux plus envoyer des messages en ligne de commande ???
    - Dans la version actuelle de sendmail X, il n'y a pas encore de module de soumission de messages. Utiliser sendmail 8 ou mini_sendmail ou nbSMTP.
  - ...

Conclusions

- Sendmail X est un MTA moderne et performant.
- Sa conception a privilégie la simplicité, les performances et la sécurité.
- Sa flexibilité permet de trouver des solutions élégantes et simples à des problèmes que l'on trouve actuellement dans les structures typiques des universités.
- Des nouvelles fonctionnalités (MSP, ...) doivent encore s'ajouter à des prochaines versions.
- Retours de expérience souhaités (ca *plus* sm9 *at* sendmailx *dot* org)