

Installer et administrer des listes de diffusion

Serge Aumont
serge.aumont@cru.fr

Novembre 1997



Résumé

Ce cours a pour objectif de vous aider à choisir un gestionnaire de listes de diffusion, à l'installer, à l'administrer pour faire vivre vos listes de diffusion ou pour ouvrir un service d'hébergement de listes, proposer des solutions pour une interface www au service de liste.



Table des matières

1	Pourquoi des listes?	3
2	Les intervenants	4
3	Principe technique sous Unix	5
4	Les fonctionnalités des gestionnaires de listes	6
4.1	Du point de vue utilisateur	6
4.2	Les fonctions d'administrateur	7
4.2.1	Les fonctions normales	7
4.2.2	Les traitements d'erreur	7
4.2.3	Prévention des boucles	8
5	Critères de choix d'un gestionnaire de liste	10
5.1	Choisir un produit simple ou un produit complet	10
5.2	Avec les sources ou avec un contrat de maintenance?	10
5.3	Chaque jour, des centaines ou des dizaines de milliers messages?	11
5.4	Quelle répartition gestionnaire de liste / MTA	11
5.5	Quelques produits pour ceux qui ont le courage de se passer d'Unix	13
5.6	Quelques produits pour les autres...	13
6	SmartList à la loupe	15
6.1	la face visible des utilisateurs	15
6.2	L'installation	16
7	Performances	18
7.1	Quelques réglages des performances	19
7.2	répartition de charge	20
8	Vers une interface WWW d'accès aux services de listes de diffusion	22
8.1	Abonnement désabonnement par formulaire	24
8.2	Serveur d'archives	25
8.3	Serveur d'archives : l'indexation	27
8.4	Outils www d'administration	27
8.4.1	Analyse des bounces assistée par web	29
8.4.2	La modération par web	29
9	Netiquette, obligations administratives	29
9.1	CNIL	31

1 Pourquoi des listes ?

Il existe de nombreux services de diffusion d'information sur l'Internet, dans quel cas recourir aux listes de diffusion ?

Petit rappel sur des alternatives possibles

1. Tout UA (interface de messagerie) qui se respecte dispose d'un calepin d'adresses avec la possibilité de désigner une liste de destinataires par un mnémonique.
2. le WEB est fait pour une diffusion unilatérale. Il existe bien des formulaires qui permettent de déposer des messages sur un serveur. Les fonctionnalités disponibles sont très limitées et l'utilisation très contraignante. Ces formulaires sont plutôt adaptés à la fonctionnalité de livre d'Or mais ne conviennent pas vraiment à des forums qui supposent une forme d'adhésion à un groupe et à ses règles de fonctionnement. Citons les produits connus :
 - HyperNews logiciel sous Unix (<http://www.hypernews.org/>)
 - Frontpage forum
 - Altavista Forum
 - O-reilly webboard
3. Les news : ce service est fonctionnellement proche de celui des listes de diffusion, il présente de nombreux avantages sur celui-ci :
 - pas d'info non sollicitée dans ma boîte aux lettres;
 - pas d'abonnement, donc pas de PB d'adresses périmées
 - expiration automatique des messages anciens
 - catalogue des news group disponibles

mais...

1. nombreux sont les abonnés qui n'aiment pas le service news.
2. on ne sait pas très bien à qui on s'adresse, on contrôle la liste des machines qui accèdent à un forum donné, mais pas la liste des utilisateurs.
3. l'expiration n'est pas faite à la demande de l'utilisateur, mais en fonction de la place disponible sur son serveur de news...
4. les forums à distribution planétaire sont bien trop copieux.
5. quand un forum est créé, suite à une procédure de vote, on ne peut plus en distinguer celui qui en a eu l'initiative.

A la différence des listes de diffusion, les news sont basées sur une inondation universelle des messages à échanger vers toutes les machines du réseau de news. Ceci a deux conséquences :

1. contrairement à une idée répandue, il n'est pas certain que le processus de diffusion des news soit plus économique en bande passante. Dans les news, un message n'est transféré qu'une fois vers chaque serveur où il peut être lu par plusieurs personnes d'où une économie. Cependant contrairement aux listes de diffusion, chaque machine réceptrice des news reçoit un grand nombre de message qui ne sera jamais ni lu, ni relayé dans le cas des serveurs en bout de la chaîne de diffusion.
2. les spammeurs utilisent sans vergogne les logs des news pour construire leurs bases d'adresses. S'exprimer dans les news, c'est s'exposer au SPAM. Ce problème est quasi insoluble dans les news.

A consulter : deux articles en français sur le SPAM

- (a) Un article de Bortzmeyer : <http://www.pasteur.fr/other/computer/JRES97/AntiSpam/>
- (b) Article de Demeester : <http://www.imagnet.fr/ime/spam.htm>

Très souvent, le recours aux listes de diffusion n'est pas le résultat d'un choix rationnel, en particulier quand on utilise les listes comme des forums ouverts de grande audience.

2 Les intervenants

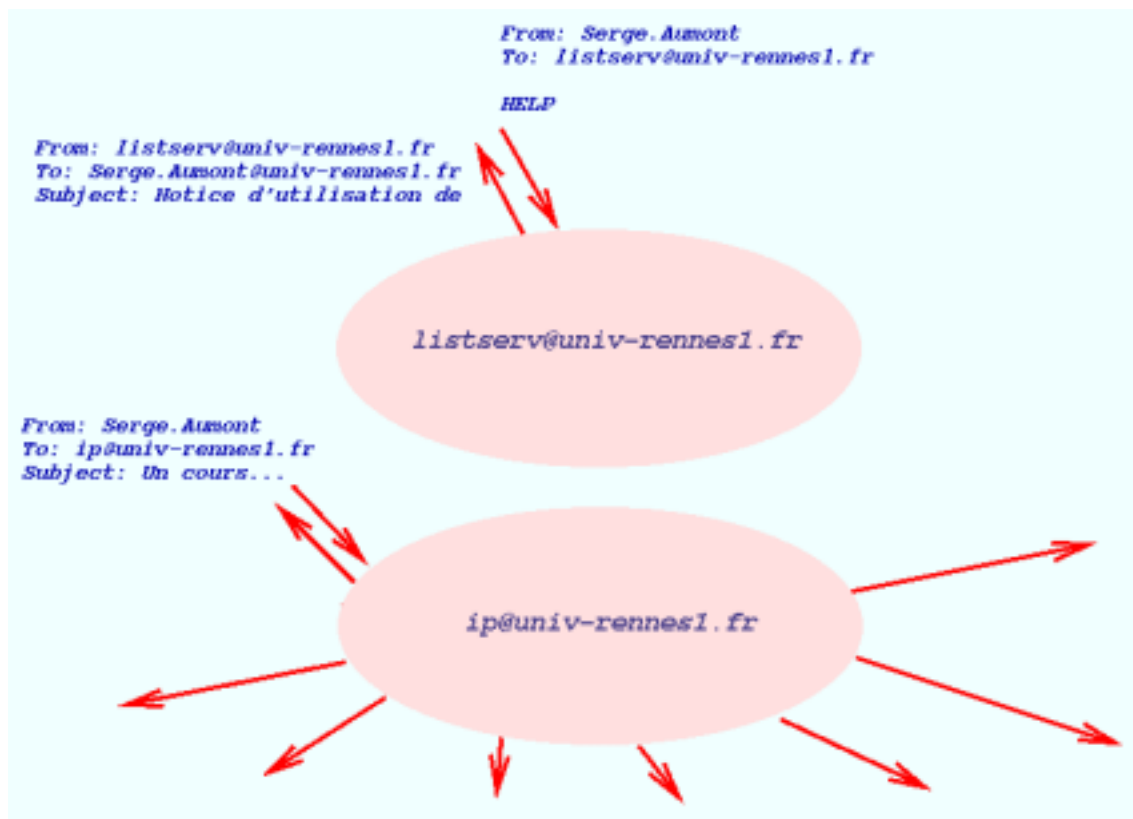
Pour bien comprendre le fonctionnement et l'organisation des services de listes de diffusion, il convient de distinguer les différents intervenants d'un service de listes de diffusion.

1. L'hébergeur. C'est celui qui exploite le service de listes, il a un rôle de prestataire de service qui fait tourner les machines mais qui s'engage sur la pérennité et la disponibilité de son service et qui applique des règles de déontologie professionnelle.
2. Le propriétaire de la liste, c'est celui qui en a eu l'initiative ou qui exerce une délégation sur le fonctionnement de la liste. Il assure l'interface avec l'hébergeur et les abonnés, il a surtout des devoirs et quelques privilèges (ce rôle est souvent partagé par plusieurs personnes) :
 - Il choisit en concertation avec l'hébergeur les paramètres de fonctionnement de la liste. (contrôle des abonnements ou non, liste visible de tous ou adresse de liste confidentielle, etc)
 - Il a accès à la liste des abonnés (commande REV DEL ADD)
 - Il doit signaler les incidents de fonctionnement de sa liste à l'hébergeur.
 - Il assiste les abonnés
 - le plus souvent il gère les messages d'erreurs.

- le modérateur Si la liste est modérée, il est le seul habilité à s'exprimer dans la liste, il valide ou invalide la diffusion de tout message. Sa responsabilité éditoriale est évidente.

Ces fonctions sont parfois cumulées par une seule personne. Aujourd'hui, en France, on identifie trop rarement la fonction d'hébergeur et les relations contractuelles entre hébergeur et propriétaires de listes.

3 Principe technique sous Unix



Sendmail propose avec les alias un système intégré de liste diffusion ; exemple d'alias :

```

petiteliste: pierre@truc.fr,paul@bidule.fr,jacques@chose.fr
grosseliste: :include :/etc/listes/grosseliste.abonnes
    
```

L'adresse `petiteliste` est expensée vers les trois destinataires Pierre, Paul et Jacques. L'adresse `grosseliste` est expensée vers la liste des destinataires contenu dans le fichier `grosseliste.abonnes`. Dans les deux cas, le service de liste est limité à la diffusion des messages ; aucun service additionnel par exemple pour traiter abonnement et desabonnement.

L'alternative est basée sur des alias qui "pipent" le message dans un programme.

```
ip: "|/usr/local/bin/deliver.pl ip"
```

En première approximation, écrire un programme genre "deliver.pl" est trivial. En fait, la simplicité des concepts utilisés dans la messagerie ne permet pas d'intégrer facilement dans le modèle utilisé une multitude de cas particuliers.

4 Les fonctionnalités des gestionnaires de listes

Il existe de nombreux logiciels de gestion de liste (Listserv, majordomo, SmartList, Sympa) et depuis peu, il existe même une offre pour des plateformes windows (95 ou nt) et MacOS.

4.1 Du point de vue utilisateur

1. Mail de 1 vers N (de N=3 à N =< quelques milliers)
2. Abonnement / désabonnement (libre ou contrôlé)
3. Liste des abonnés
4. Modération
5. Archivage
6. Mode résumé (digest)
7. Authentification
8. Help

La forme visible par les utilisateurs dépend du gestionnaire de liste choisi. Exemple, l'adresse du gestionnaire de liste :

- Quand le gestionnaire de liste est un humain : nom-liste-request@...
- Listserv : listserv@...
- Majordomo : majordomo@...
- Smartlist : nom-liste-server ou nom-liste-request
- etc

Une norme pour cette adresse aiderait les utilisateurs. A cause de ce désordre (et aussi parfois faute de faire le moindre effort pour lire la doc), les utilisateurs adressent souvent des requêtes de gestion dans les listes plutôt qu'au gestionnaire de listes.

- Sympa : filtre les requêtes adressées aux listes via un module perl optionnel, adresse un rejet motivé à l'auteur de l'erreur.
- Majordomo adresse l'erreur au propriétaire de la liste

- ListProc, SmartList et LISTSERV détectent les commandes même avec certaines fautes de frappe.
- SmartList, en outre, exécute les commandes de gestion qu’il a reconnues.

Le jeu de commande varie :

- Smartlist : une tentative de reconnaissance du langage naturel ?
- LISTSERV : un standard de fait pour les services de base auxquels de nombreux utilisateurs sont habitués. De nombreux robots (dont sympa) ont choisi de coller à ce standard, au moins pour les services de bases.
- ListProc : un libre choix atypique

4.2 Les fonctions d’administrateur

4.2.1 Les fonctions normales

- Certains propriétaires de liste n’ont pas d’accès `login` sur la machine de liste. Ils font plus ou moins recours à l’administrateur selon le niveau des fonctions accessibles par l’automate en mode messagerie. Quelques interfaces de listes de diffusion en mode WWW apparaissent, certaines donnant accès à des fonctions d’administrations. Elles font l’objet d’un paragraphe de cet article.
- Paramétrage liste par liste ou global ? (ex: action en cas d’erreur utilisateur, épuración des entêtes inutiles, etc)
- Utilisation d’une signature pour contrôler certains accès
 - From:
 - Confirmation par retour de courrier
 - Mot de passe
 - Signature PGP

4.2.2 Les traitements d’erreur

- Traitement des messages d’erreur : dispositif auto-videur. Récupérer les messages d’erreurs dans le but de supprimer les adresses erronées des listes est complexe. Il faut faire le tri entre des erreurs temporaires et des erreurs sans espoir d’amélioration. Les formats des messages d’erreurs sont multiples même s’il existe un RFC à ce sujet. Un rapport d’anomalie peut concerner plusieurs destinataires... Même pour un humain à qui ces compte-rendus d’anomalie sont censés être adressés ...
 - Listserv et ListProc font des traitements de base.
 - Sympa rien **pour le moment...**

- SmartList offre une série de filtres successifs pour identifier l'adresse responsable de l'erreur, il compte les erreurs pour chaque adresse et (après notification) désabonne l'adresse si le nombre d'erreurs atteint la limite définie par l'administrateur.

Consulter les conseils : Lire les rapports de non-remise (<http://www.cru.fr/listes/atelier/bounce.html>)

4.2.3 Prévention des boucles

Une des pires choses qui puisse arriver à votre liste c'est une boucle non-détectée par votre gestionnaire de liste. Exemple : un mauvais plaisantin abonne une liste à elle-même ...

La prévention des boucles doit être basée sur la distinction SMTP entre enveloppe et entêtes : From et From:

L'usage standard de cette distinction est le suivant :

- cas d'un message diffusé dans une liste

```
Return-path : listes-owner@cru.fr
Received : from home.cru.fr (aumont@localhost.cru.fr [127.0.0.1])
          by home.cru.fr (8.8.5/jtpda-5.2) with ESMTp id NAA11624
          for <listes> ; Mon, 3 Nov 1997 13 :45 :47 +0100
Message-id : <199711031245.NAA11624@home.cru.fr>
To : admin-list@cru.fr
Subject : atelier
Date : Mon, 03 Nov 1997 13 :45 :47 +0100
From : Aumont - Comite Reseaux des Universites <Serge.Aumont@cru.fr>

...

```

Le message a été posté par Aumont dans la liste admin-list@cru.fr. Le gestionnaire de liste marque son empreinte dans le champ Return-path. Le processus d'acheminement de ce message, s'il génère un rapport d'anomalie, doit l'adresser à listes-request. Aucune réponse automatique ne peut engendrer une boucle.

Un usager utilisant son bouton reply fait une réponse à l'auteur du message dans le cas de cette liste.

- cas d'un message généré par l'automate de gestion

```
Return-path : sympa-request@cru.fr
Message-id : <199711031303.OAA30993@listes.cru.fr>

```



```
Date : Mon, 3 Nov 1997 14 :03 :10 +0100
From : sympa-request@cru.fr (Systeme de Multi-Postage Automatique)
To : serge.aumont@cru.fr
Subject : Output of your commands
...
```

Beaucoup de passerelles de messagerie ne respectent pas cette distinction (PROFS et Micro-Soft Mail !). MSmil ne conserve que l'adresse d'enveloppe et les utilisateurs adressent souvent des requêtes à listserv-request...

Une solution pour faciliter l'usage des listes aux utilisateurs de ces produits non conformes à SMTP est de positionner From et From: à la même adresse, mais attention aux boucles. Adopter cette démarche impose alors de détecter les boucles ce qui n'est pas simple et pas sans inconvénient. C'est aussi une manière d'encourager des produits qui polluent l'Internet.

Il existe trop de cas où les RFCs sont bafoués pour se contenter de cette méthode.

- vacation
- passerelles vers les news
- passerelles vers des systèmes non SMTP
- rapports de remise exotiques
- mailers zozotants
- abonnements croisés de listes

Pour prévenir les boucles, il est fortement déconseillé à tout service en mode messagerie de répondre ou distribuer un message qui contient dans l'adresse d'émetteur :

- autoanswer
- echo
- listserv
- mailerdaemon
- mirror
- netserv
- server
- vacation
- postmaster
- ...

Enfin, il faut parfois tenter de détecter des boucles pour les casser ou en limiter l'impact. Ce n'est pas facile ; on peut

- Utiliser la convention X-Loop : , elle consiste à ajouter une entête X-Loop : adresse de liste dans les messages. On peut reconnaître à cette marque les messages déjà diffusés dans une liste et bloquer leur rediffusion.
- gérer une table des Message-Id : , ne pas rediffuser un message dont le Message-Id : est déjà dans les tables.
- Utiliser un checksum sur le corps du message (après avoir filtré certains caractères comme les espaces) pour détecter qu'un message a déjà été distribué.

- Rejeter des messages avec certains Subject : comme Delivery delayed, chercher la présence de plusieurs Message-ID : etc
- Limiter le nombre de messages diffusés par personne, par listes pour une période de temps donnée permet de limiter la gravité des boucles.

Listproc est semble-t-il le mieux armé dans ce domaine.

5 Critères de choix d'un gestionnaire de liste

Les critères de choix sont multiples ; la panoplie des logiciels disponibles très complète.

Il faut d'abord choisir un produit sous Unix ou pas.

Les raisons de ce choix débordent largement la problématique des listes de diffusion.

5.1 Choisir un produit simple ou un produit complet

Listserv est une marque déposée. Ce nom désigne donc le produit d'Eric Thomas né dans Earn. Il dispose de la plus large palette de fonctionnalités. Rien ne manque ou presque mais ... le mieux est l'ennemi du bien. Administrateurs, propriétaires de listes et utilisateurs risquent de se perdre dans un luxe d'options et de nuances.

A l'opposé, Majordomo et Sympa sont peut-être un peu frustes ?

Listserv est le gestionnaire le plus intégré : c'est celui qui fait le moins appel à des applications externes, en particulier à un MTA (agent de transfert de messages) externe (en général sendmail). A l'opposé, Majordomo utilise massivement sendmail par la distribution des messages (alias include), *ftpmail* pour la mise à disposition des archives ; Smartlist utilise Procmail comme interpréteur du format SMTP, etc. Bien entendu, ceci permet une grande simplicité du code et parfois de la configuration du gestionnaire de service.

5.2 Avec les sources ou avec un contrat de maintenance ?

Vous pouvez avoir envie de modifier le code de votre serveur. Difficile de le savoir à l'avance ...

- Almanac, BMW, Majordomo, Mreply, Smartlist et Sympa sont donnés avec les sources.
- Listproc 6.0.c avec restrictions
- CREN Listproc, LISTSERV, Mailbase, Mercury, Maiser, Mx, Petidomo, Smof sont distribués en binaire, mais peut-on avoir un contrat de maintenance pour autant ?

LISTSERV et CREN ListProc sont des produits commerciaux dont le support peut être acheté (ou en tout cas payé).

5.3 Chaque jour, des centaines ou des dizaines de milliers messages ?

Certains gros serveurs de listes peuvent traiter, en pic, jusqu'à un million de messages par jour !

Deux points fondamentaux de l'architecture des serveurs de listes conditionnent la résistance aux problèmes de charge :

1. existence ou non d'un spool des messages en attente de diffusion ;
2. part du processus de distribution des messages traitée par le gestionnaire de liste et part laissée au MTA local.

– Certains gestionnaires de listes sont activés par le MTA (sendmail) et n'existent donc pas sur le système quand il n'y a aucun message (Majordomo et SmartList). Inconvénients : pour chaque message, le gestionnaire de fichier doit être chargé en mémoire puis, il doit initialiser complètement son contexte général et spécifique à une liste donnée

Si plusieurs messages arrivent au même moment (par exemple après un arrêt prolongé de la machine), plusieurs processus identiques doivent se partager la mémoire de la machine. De plus, ces tâches utilisant toutes les mêmes ressources ont tendance à écrouler la machine.

(Majordomo est écrit en perl, SmartList utilise l'agent de remise de message ProcMail, tout deux sont interprétés.

– D'autres sont lancés au démarrage du système et utilisent une queue des messages en attente. ListProc, Sympa et LISTSERV tournent en partie en daemon et traitent séquentiellement les messages. Ils résistent donc mieux aux pics de trafic qui caractérisent le mail (mais le daemon occupe des ressources même quand il n'y a pas de message à traiter). Sympa garde en mémoire la liste des abonnés, le daemon sympas est donc assez gros, mais chaque demande d'abonnement ne passe pas par une phase de chargement d'une image de la liste.

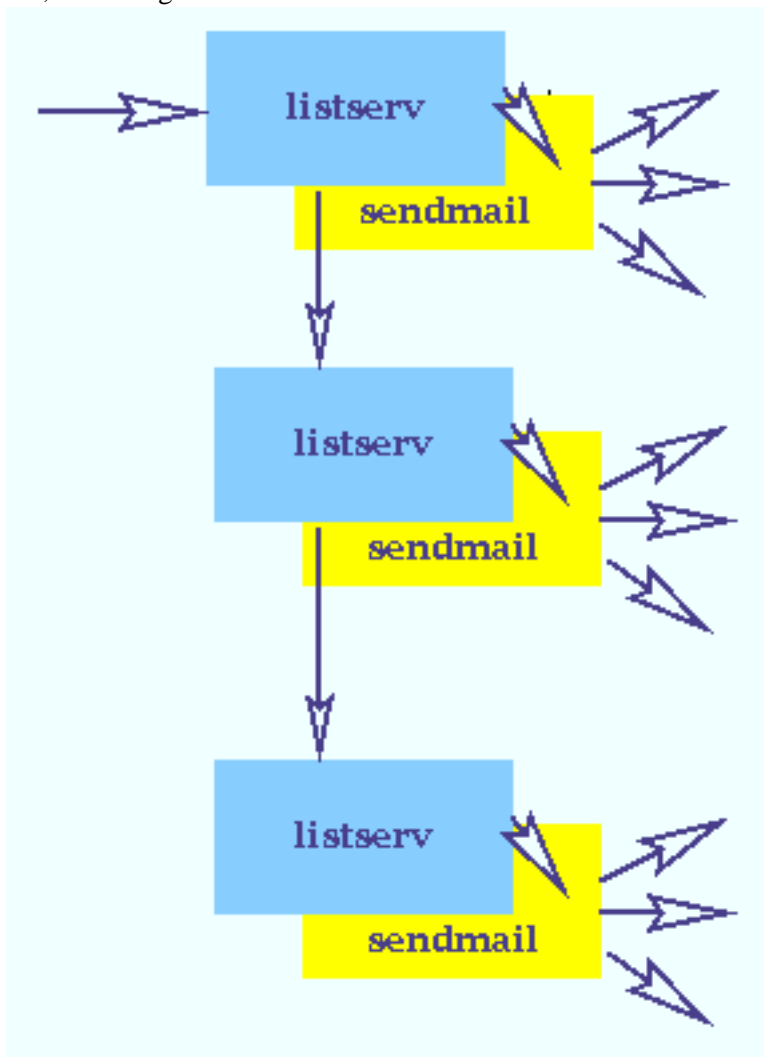
A titre d'illustration, nous avons mesuré avec sympas sur un P200 Pro sous linux 25 secondes pour la première opération sur une liste de plus de 6000 abonnés, puis moins d'une seconde pour les opérations suivantes sur cette même liste chargée en mémoire.

5.4 Quelle répartition gestionnaire de liste / MTA

Pour les grosses listes, la question cruciale est relative à la performance du processus de distribution des messages.

– Solution Majordomo, Petidomo,... : tout donner à sendmail. Majordomo passe les messages à sendmail avec la liste des destinataires. Toute la charge repose sur sendmail. Si les destinataires ont des adresses éparpillées (au sens MX record), sendmail achemine autant de messages que de destinataires. Avec deux cents destinataires outre Atlantique, le processus de remise peut être de plusieurs heures selon les aléas de disponibilité de l'Internet (avec le risque qu'une discussion soit entamée par les premiers servis avant que tous ne reçoivent le premier message).

- Solution listserv : traiter lui même un maximum de remise. Ci dessous, un schéma du backbone listserv, un héritage de bitnet.



1. remise aux destinataires locaux via une session SMTP avec le MTA local
2. remise aux listserv voisins : 1 seul exemplaire. Le gain est là !
3. remise au reste via une session SMTP avec le mailer local.

Inconvénient : complexité de gestion de listserv qui doit disposer d'une table de routage spécifique. Cette solution était parfaitement adaptée à bitnet dont la topologie était stable et sur lequel le "store and forward" était la règle pour tout cheminement de message. De plus cette méthode a fait l'objet d'un RFC, mais seul LISTSERV l'implémente ce qui lui enlève une grosse part de son intérêt.

- Solution Sympa, Smartlist, ... : passage du message au MTA par appels de commande . Le nombre de destinataires pour chaque appel à sendmail est limité à N. Les adresses sont triées par domaine pour favoriser le groupage des destinations par sendmail. (il faut alors tenir compte du coup de la fonction de tri pour les abonnements et désabonnements). Chaque processus sendmail est raisonnablement gros. Plusieurs processus peuvent tourner en parallèle mais il y a des risques de pics de charge liés au nombre de sendmail. La configuration de sympa permet de contrôler le nombre maximum de processus sendmail lancé. SmartList comme Sympa permettent de configurer liste par liste le nombre maximum de sendmail lancés en parallèle. En outre Sympa permet de choisir finement le facteur de groupage des destinataires.
- Listproc utilise une méthode très proche mais il intègre un client SMTP et lance sendmail via une session vers le port 25 de la machine mailhost. Avantage, on peut plus facilement séparer la machine de listes de la machine de diffusion.

5.5 Quelques produits pour ceux qui ont le courage de se passer d'Unix

- **LetterRip** MacOS <http://www.fogcity.com/letterrip.html>
- **ListSTAR** MacOS <http://www.starnine.com/liststar/liststar.html>
- **Macjordomo** MacOS <http://www.greatcircle.com/majordomo/>
- **Lyriss** Windows/NT ou 95 <http://www.lyris.com>
- **MDaemon** windows-nt et 95 <http://www.mdaemon.com>
- **Mustang Web Essentials ListCaster** windows-nt et 95 <http://www.mustang.com/we/>
- **Pegasus Mercury** windows-nt et 95 (liste perso) <http://www.pegasus.usa.com/>
- **fastraq post server**, windows nt4ou 95 . <http://www.fastraq.demon.co.uk/post/introduction.html>
- **SLMail** Windows NT, 95 <http://www.seattlelab.com/>
- **SVList** windows95 <http://www.cadvision.com/softventures/svlist.htm>

5.6 Quelques produits pour les autres...

La liste des produits plus ou moins aboutis est longue. Voici une sélection de références de produits que je ne connais pas forcément très bien.

1. **Cerberus Mailer**, produit commercial distribué en source perl <http://www.abiogenesis.com/cerberus/>
2. **Petidomo** <http://www.petidomo.com/> produit commercial sous solaris, irix, linux, netbsd ... Produit commercial mais gratuit pour usage scientifique ou éducation. Ce produit s'installe très simplement avec en standard une interface d'administration de listes par serveur WWW. Le robot de listes est donc très facile à mettre en oeuvre mais son architecture (pas

de daemon) n'est pas du tout fait pour supporter une charge importante. L'interface www d'administration pose des problèmes de sécurité puisqu'elle permet d'agir directement dans les alias !

Ci-dessous le fichier de configuration d'une liste qui peut être édité par un cgi :

```
# Config file for mailing list "petidemon"
# generated automatically by petidomoconf
#
ListType                open
AllowPublicSubscription yes
AllowAlienSubscription  yes
AllowMembersCommand    no
ShowOnIndex            yes
#Hostname                listes.cru.fr
AdminPassword          minidomo
PostingPassword        minidomo
#ReplyTo                none
#PostingFilter           "/bin/cat"
#Archive                "log"
```

A noter deux bonnes idées :

- le paramètre `PostingFilter` dont la valeur par défaut `"/bin/cat"` peut être remplacée par n'importe quel script pour personnaliser les messages envoyés dans la liste.
- un fichier de contrôle d'accès `acl` permet de maîtriser l'accès à la liste.

Petidomo semble être une très bonne solution pour des sites de petites tailles principalement pour des listes privées.

3. **IDG** (Internet Discussion Group) envoyer un message à `lt.ew@dot.ca.gov` avec `help` dans le corps du message.

Plusieurs originalités de IDG réservent probablement son usage à de petites listes internes:

- Une seule adresse pour les commandes et la liste. Ce qui n'est pas une commande est considéré comme un message...
- Possibilité pour chaque utilisateur de définir un filtre par mots clefs pour sélectionner les messages que l'on désire recevoir.
- Système simple de vote

4. **ListProc**, deux versions

- V6 gratuite, non supportée. Parfois appelée Unix Listserv.
`ftp://cs-ftp.bu.edu/pub/listserv/current/_version.Z`
- CREN ListProc V7 `gopher://info.cren.net` (unix, windows-nt?)

5. **LISTSERV**, vendue par L-soft (E.Thomas). <http://www.lsoft.com/listserv-facts.html> Windows-nt, Unix. (La doc d'administration fait 250 pages, celle à usage des propriétaires de listes 150 ...).
6. **Mailbase**: <http://www.mailbase.ac.uk>, impose de disposer de ingres et perl utilisé pour le service académique de Janet: 1800 listes sur un serveur, probablement une bonne solution pour les gros sites.
7. **Majordomo**: très populaire sur l'Internet parce que libre de droit, écrit en PERL et longtemps sans concurrents. Attention aux problèmes de performances l'architecture de Majordomo n'est pas adaptée aux grosses listes. <ftp://ftp.greatcircle.com/pub/majordomo>. De plus, il semble que MAJORDOMO pose quelques problèmes d'installation et de sécurité. Alors pourquoi ce succès de majordomo?
8. **MReply**: Mail Forwarder Information, une boîte à outil avec laquelle on peut faire des listes de diffusion. Configuration assistée par serveur WWW <http://clever.net/self/faq/mreply.html>
9. **SmartList**: largement appuyé sur "procmail" écrit par le même programmeur. <ftp://ftp.informatik.rwth-aachen.de>. Ce gestionnaire de liste est assez original et offre de bons outils d'administration.
10. **SYMPA**: Système de Multi-Postage Automatique, une réécriture de TULP par le même auteur (C.Wolfhugel). <ftp://ftp.cru.fr/CRU/Listes/sympa>. Les objectifs de la première version: maintenabilité du code, performances, authentification et purge sont atteints. L'Urec et le CRU y ont adaptée une première interface d'administration par WWW: weblist (fureter dans <ftp://ftp.cru.fr/CRU/Listes>).

6 SmartList à la loupe

6.1 la face visible des utilisateurs

Dans SmartList, toutes les commandes de gestion doivent être envoyées à l'adresse: `liste-request@site`. SmartList a donc la prétention d'être un automate capable de traiter des remarques qui sont normalement adressées à un humain !

- SmartList interprète le champ Subject :
- Pour s'abonner: `subscribe`, se désabonner: `unsubscribe`.
 - Une seule commande d'abonnement par message.
 - Pas de champ Free form name dans les abonnements (prévu dans une prochaine version).
 - Seul l'administrateur des listes peut consulter la liste des inscrits.

- Le serveur d'archive essaye de simuler le système de gestion de fichier Unix en mode messagerie. Il connaît les commandes:
 - get filename ...
 - ls directory ...
 - egrep case_insensitive_regular_expression filename ...
 - maxfiles nnn (par sécurité, le maximum de fichier attendu en retour).
 - version
- le tri des requêtes de gestion et des messages à distribuer semble assez performant, assez pour choisir de traiter les requêtes mal aiguillées.
- L'administrateur de liste peut adresser des commandes de gestion sophistiquées avec une entête de type :


```
X-Command: <adresse du propriétaire> <password> <command>
```

 Les commandes disponibles permettent d'abonner, désabonner, consulter la liste des abonnés, consulter et remettre à zéro les logs de la liste.

6.2 L'installation

L'installation est remarquablement simple et bien documentée. Il faut :

- Installer sendmail
- Installer procmail
- enregistrer l'utilisateur et le groupe `list` (pas de privilège, partition avec un peu de place).
- donner à cet utilisateur le droit d'utiliser l'option -f de sendmail (trusted users, variable Tlist de sendmail.cf).
- créer le home dir de l'utilisateur `list`.
- exécuter `sh install.sh ~list`
- créer une liste `maliste` en exécutant `createlist maliste <adresse du propriétaire>`

À l'issue de ce travail, une structure de fichier assez complexe est créée dans le home dir de `listserv`. Cette structure a pour objectif de permettre de jouer sur tous les paramètres liste par liste tout en autorisant un paramétrage par famille de listes ou pour l'ensemble du système.

Voici la structure des données après installation :

```
.bin/                .examples/          maliste/
.etc/                .procmailrc
```

- `.bin/`
Ce répertoire contient les binaires par défaut,

– .etc/

Ce répertoire contient la configuration par défaut des listes :

rc.archive	rc.init.orig	rc.request	unsubscribe.txt
Manual	rc.custom	rc.submit	archive.txt
rc.init	rc.main	reject	help.txt
rc.init.dist	rc.post	subscribe.txt	

Si vous voulez personnaliser les fichiers de doc de toutes les listes, intervenez sur les .txt de ce répertoire.

– .procmailrc le fichier de config de procmail (pas de modif)

– maliste

Contient la configuration de cette liste :

accept	help.txt	rc.custom	reject
subscribe.txt	archive.txt	rc.init	
tmp.from	log	rc.request	tmp.request
dist	msgid.cache	rc.submit	unsubscribe.txt
archive/	bounce/		

– dist: la liste des abonnés

– log: le log de cette liste

– rc.init permet entre autre de changer les règles de recherche liste par liste grâce à la variable `PATH`. On peut ainsi modifier un binaire pour une liste seulement !

– rc.custom contient les paramètres généraux:

– adresse du propriétaire de la liste (`maintainer`)

– taille max des messages

– localisation et fonction du log

– paramètre du processus d'élimination des adresses en erreur

– `foreign_submit` (autorisation de soumission) = ouvert ou limité aux adresses du fichier `accept` (qui peut être un lien sur le fichier des abonnés)

– `restrict_archive` (autorisation d'accès aux archives)

– `cc_unsubrequests` (envoyer ou pas au `maintainer` les demandes de désabonnement)

– `reply_to`

– `digest_flag` (mise en route de la fonctionnalité de digest)

– `X_COMMAND_PASSWORD` mot de passe pour le propriétaire de liste

– etc

- archive/ contient les archives. A noter, en posant un lien vers le fichier des abonnés dans le répertoire des archives, vous pouvez simuler la fonctionnalité REView de TULP.
- bounce/ historique des rapports de non remise. Exemple d'un message reçu par le propriétaire d'une liste après plusieurs rapports de non remise pour une même adresse :
- reject liste des adresses indésirables : permet d'empêcher l'abonnement et la soumission de messages pour certaines adresses.

```
Return-path: <www-fr-request@univ-rennes1.fr>
X-diagnostic: Mail to oarditi@sparta.sjsu.edu bounced 4 times
X-diagnostic: Bounces exceed threshold of 4
X-diagnostic: Removed: oarditi@sparta.sjsu.edu 32760
X-diagnostic: Mail coming from a daemon, ignored
X-diagnostic: Possible loopback problem
X-envelope-to: www-fr-request
Date: Thu, 2 May 1996 08:58:06 +0200
From: Mailer-Daemon@univ-rennes1.fr (Mail Delivery Subsystem)
Subject: Returned mail: User unknown
To: www-fr-request@univ-rennes1.fr
```

...

SmartList ne peut traiter tout les rapports de non remise. Dans certains cas (exemple plusieurs diagnostics d'erreur dans 1 seul message), il ne supprime pas l'abonné de la liste mais envoie un rapport avec :

```
X-diagnostic: Mail to xxx@muguet.saclay.cea.fr bounced 5 times
X-diagnostic: Bounces exceed threshold of 4
X-diagnostic: Not confident enough to autoremove the offending address
```

...

En effet, on ne peut supprimer un abonné qu'à coup sur.

- msgid.cache, la mémoire des identificateurs des messages déjà traités, utilisé dans la prévention des boucles.

7 Performances

Nous hébergeons de très nombreuses listes et nous devons faire face à un trafic extrêmement élevé. Le nombre de messages émis depuis nos systèmes de listes de diffusion successifs est passé de 180 000 messages par mois il y a 3 ans à 1 500 000 messages par mois en octobre 97.

Cette situation particulière nous a poussé à bricoler, au jour le jour, un outil de gestion de notre messagerie avec :

- surveillance des différents spools et daemon, édition d’alarme quand certains seuils sont atteints.
- enchaînement d’actions fréquentes (création de liste, création d’archive, redémarrage d’un daemon, modification des alias, ...)

7.1 Quelques réglages des performances

La tâche critique au regard de la charge est la remise des messages, et plus spécifiquement, la canonisation des adresses dans le processus de remise. Beaucoup d’anomalies de fonctionnement de l’Internet affectent les temps de réponse du DNS. Les sessions SMTP peuvent alors s’avérer très longues. Le nombre maximum de sessions SMTP parallèles sur une machine étant limité, quelques messages adressés à une liste de quelques milliers d’abonnés peuvent saturer le processus de diffusion. Les solutions possibles sont les suivantes :

- Utiliser la notion de priorité de sympa pour éviter l’engorgement de petites listes locales par de grosses listes publiques. Sympa permet d’affecter une priorité à chaque liste de diffusion. Après l’envoi de chaque message, Sympa réexamine son spool dans l’ordre des priorités.
- Une session SMTP peut durer plusieurs heures. Chaque évènement de la session est contrôlé par un time-out. Dans sendmail 8.8 ces timers peuvent être réglés finement évènement par évènement. L’option `timeout.command` permet de renseigner la valeur par défaut des timers ; une valeur de deux heures peut s’avérer indispensable pour joindre quelques sites mal foutus mais cette valeur conduit le processus de diffusion des messages à saturation. L’option `Timeout.connect` contrôle de délai maximum d’attente lors de l’établissement de la connexion TCP, 5 à 10 minutes sont indispensables. `Timeout.iconnect` désigne le même timer mais uniquement dans le cas de la première tentative de remise du message. Avec une valeur de 20 secondes pour ce timer, les processus sendmail lancés par sympa abandonnent les tentatives de remise lors que la connexion TCP est longue à établir. Le message est alors mis en spool. Le processus sendmail qui réexplore le spool périodiquement utilisera lui le timer `Timeout.connect`. Ce fonctionnement permet de ne pas encombrer sympa avec des sessions SMTP lentes (lors que la connexion TCP est longue à établir, le plus souvent la totalité de la session est très longue).
- augmenter la mémoire de la machine afin de pouvoir supporter un grand nombre de sessions SMTP en parallèles.
- régler le nombre de destinataires par appel de sendmail. Sympa (et d’autres robots) permet de configurer le nombre de destinataires mis dans chaque appel de sendmail. En augmentant ce nombre, on diminue le nombre d’appel à sendmail nécessaire pour servir une liste donnée d’abonnés. Ce nombre ne peut être augmenté inconsidérément sans rendre inefficace chaque processus sendmail devenu trop complexe. Ajoutons que Sympa permet de spécifier le facteur de groupage maximum (nombre de destinataires d’un même domaine dans

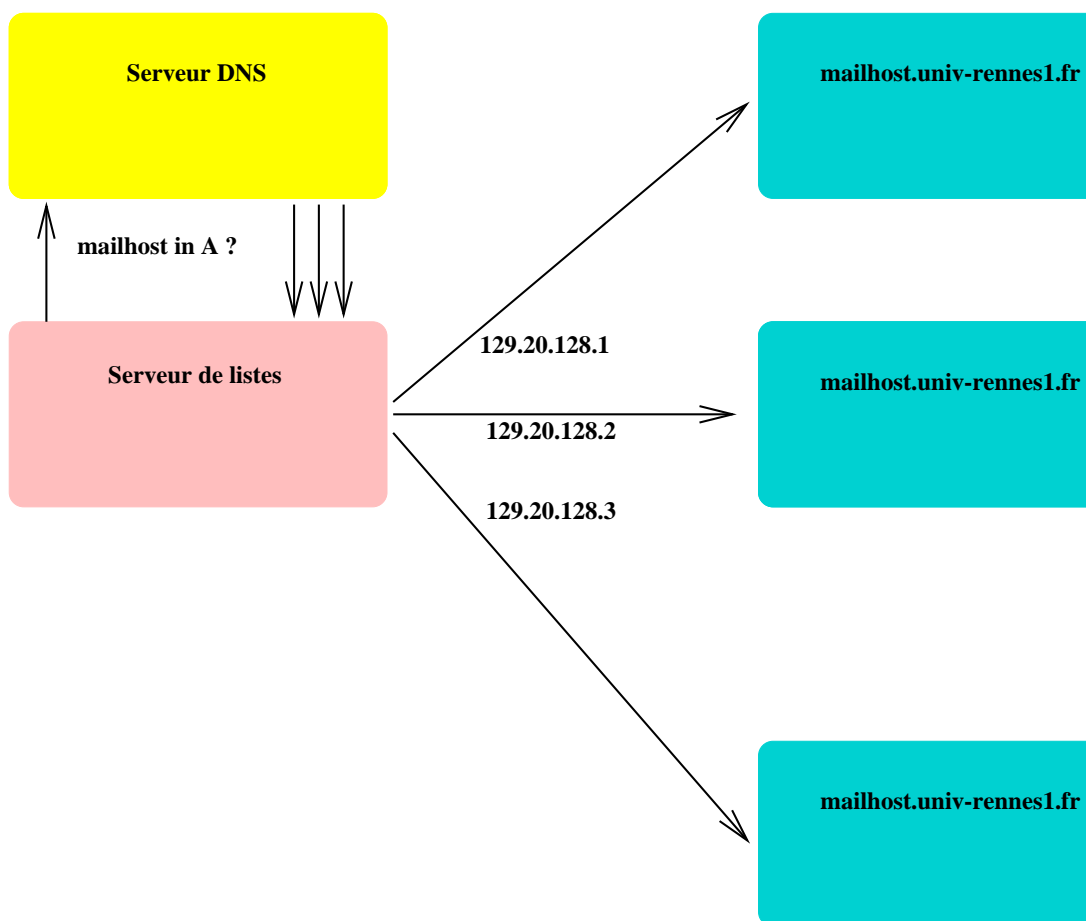
un même appel à sendmail). Ce facteur permet bien sur une grande économie de ressources puisque le message peut être posté en une session SMTP pour plusieurs dizaines d'adresses, comme c'est le cas par exemple pour les abonnés du domaine compuserv.com de certaines grosses listes. Au total, le nombre maximum de sessions SMTP, le nombre de destinataires par appel de sendmail et le facteur de groupage s'avèrent difficiles à régler autrement que par essais successifs. Ces paramètres interfèrent fortement avec des paramètres du système ou de sendmail. S'inspirer des valeurs par défaut de la distribution de sympa.

- La répartition de charge. cette méthode consiste à séparer la fonction robot de liste de la fonction moteur SMTP et de répartir cette dernière sur plusieurs machines. Cette méthode a été utilisée par le passé avec TULP sur un sparc. L'adoption de sympa et de linux rend inutile pour le moment ce système qui est décrit dans le paragraphe suivant.

7.2 répartition de charge

Quelque soit la finesse du dispositif de diffusion et quelque soit la puissance de la machine de gestion de liste, on peut arriver à saturation. En effet, la machine de diffusion doit canoniser **chaque** adresse destinataire en utilisant le DNS (recherche de MX). Cette recherche de MX peut durer plusieurs secondes pendant lesquelles le processus sendmail reste en mémoire. Une solution consiste à exploiter le DNS pour répartir ce travail sur plusieurs stations.

Bind, compilé avec l'option ROUND-ROBIN permet de faire une rotation de l'ordre dans lequel il retourne les adresses. On exploite cette possibilité pour répartir uniformément la charge entre plusieurs machines ainsi :



Pour le faire il suffit de :

1. Installer sur la machine d'hébergement de listes un sendmail feuille, c'est à dire un sendmail qui passe tout les messages à une machine de relais sans canoniser les adresses. Appelons cette machine `relay.mon.domaine`
2. choisir quelques machines aux quelles on confie le rôle de diffusion des messages. Installer sur ces machines une configuration sendmail complète (qui utilise le DNS et accepte de relayer en provenance du serveur de listes).
3. définir dans le DNS les adresses IP de `relay.mon.domaine` avec autant de `a` record que de machines de diffusion.

Ainsi par exemple, TULP sur `mailimailo.univ-rennes1.fr` fait appel à un sendmail qui passe tout à la machine virtuelle `exploreur.univ-rennes1.fr`. Celle-ci définie avec les deux adresses IP des deux machines de diffusion; on trouve dans les tables DNS du domaine `univ-rennes1.fr` :

```
mailimailo      IN      A      129.20.128.39
ariane          IN      A      129.20.128.45
exploseur       IN      A      129.20.128.39
exploseur       IN      A      129.20.128.45
```

Une consultation DNS pour `exploseur.univ-rennes1.fr` retourne :

```
%nslookup
> exploseur.univ-rennes1.fr

Name :      exploseur.univ-rennes1.fr
Addresses : 129.20.128.45, 129.20.128.39

> exploseur.univ-rennes1.fr

Name :      exploseur.univ-rennes1.fr
Addresses : 129.20.128.39, 129.20.128.45
```

8 Vers une interface WWW d'accès aux services de listes de diffusion

L'évolution marquante de ces derniers mois dans le domaine des listes de diffusion concerne les produits connexes aux robots de listes via le WEB. En effet, l'interface d'accès en mode messagerie des fonctions d'un robot de listes (abonnement, désabonnement, archives, administration) est peu satisfaisante.

- Le fonctionnement par message est un fonctionnement batch
- L'authentification des messages est impossible ou lourde faute de signature dans le protocole de messagerie lui même.
- Le service de liste s'adresse maintenant à des gens qui découvrent l'Internet à travers le web alors que les robots en mode messagerie ont été conçus quand le mail était presque le seul service des réseaux. Aussi simple que soit le mode d'emploi d'un robot en mode messagerie, s'en servir impose de lire quelques lignes de doc. Privé de menu déroulant et d'interface presse-bouttons, nombreux sont les utilisateurs déconcertés :-)

Un service de liste de diffusion ne saurait maintenant être conçu sans son versant www. Le minimum consiste en une description des listes disponibles, mais on a vite besoin :

- de formulaire d'abonnement et désabonnement
- d'archives en HTML
- de moteur de recherche dans les archives
- d'une interface d'administration

Citons quelques solutions trouvées dans la nature.

- the Mailing List WWW Gateway lwwgate <http://www.netspace.org/users/dwb/lwwgate.html> qui permet d'interfacer LISTSERV(TM), List Processor 6, Majordomo et SmartList (adaptation probablement facile avec tout robot en mode messagerie). lwwgate propose des archives et l'abonnement et le désabonnement.



- Mailserv,

<http://advicom.net/fitz/www/mailex/fitz/mailex/> une interface uniquement pour vous affranchir de la syntaxe de commande de divers robots. Pas d'interface d'administration sauf si le robot concerné contient des fonctions d'administration en mode messagerie. Les résultats sont toujours envoyés par messagerie...



- Regan http://www.peak.org/peak_info/mlists/Majordomo.html, une interface permettant seulement d'envoyer des commandes à tout robot majordomo.
- MajorCool <http://ncrinfo.ncr.com/pub/contrib/unix/MajorCool/>



- Majordomo-WWW Gateway <http://www.inf.utfsm.cl/marcos/majordomo/www.html> propose depuis peu un module spécifique pour les propriétaires de listes.

Notre contribution dans ce domaine est encore partielle. Elle se compose de briques :

1. Interface utilisateur : un mécanisme d'authentification et des cgi d'abonnements et désabonnements ainsi qu'un moteur d'archivage : MhonArc.
2. Interface administration en cours d'élaboration.

8.1 Abonnement désabonnement par formulaire

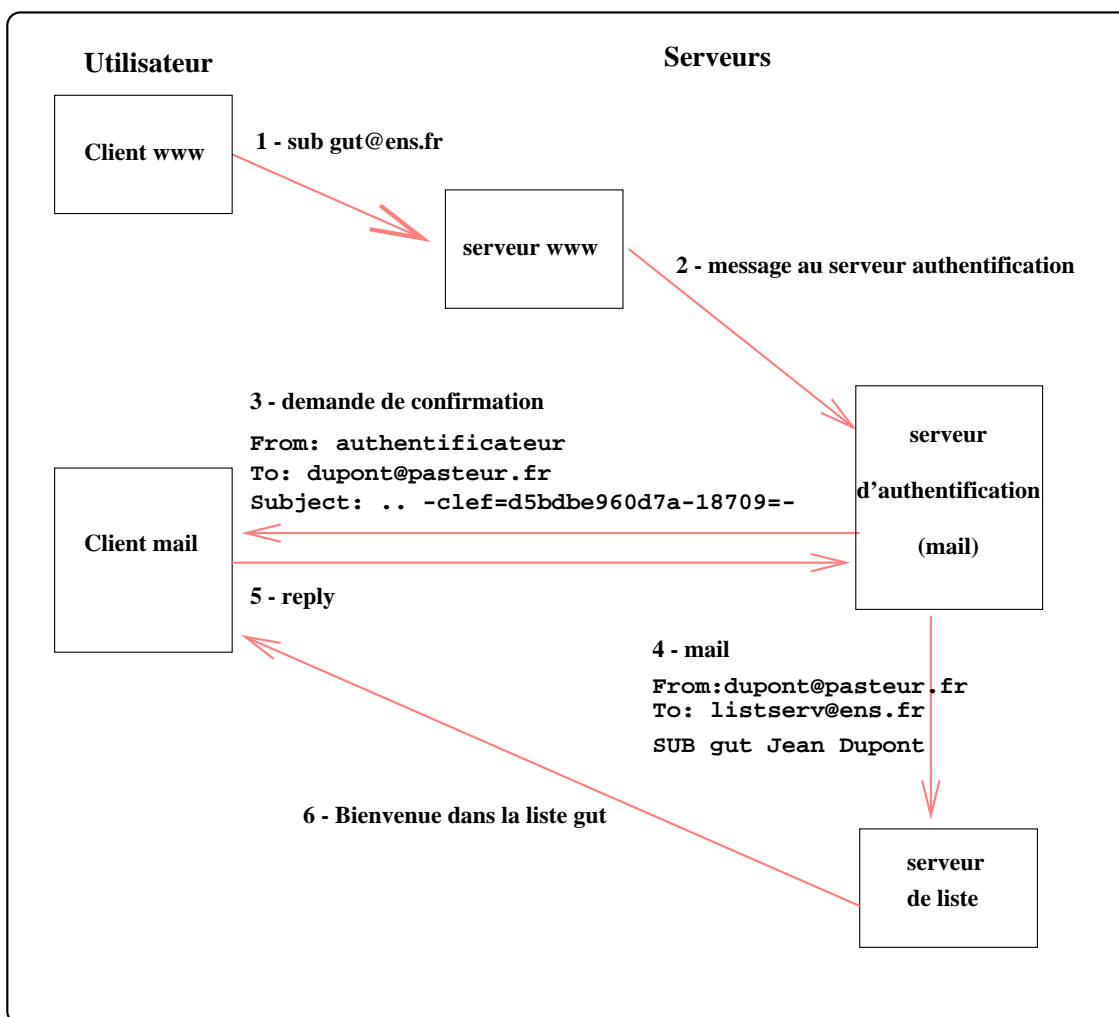
Plusieurs solutions sont possibles pour offrir un formulaire d'abonnement à une liste par web. Nous avons choisi de saisir l'adresse du demandeur parce qu'il n'existe heureusement pas de moyen de récupérer l'adresse e-mail d'un client WWW. Le but est de forger un message à destination du robot passé en paramètre en positionnant le champs `FROM` : à l'adresse du demandeur. Il nous semblerait irresponsable de proposer un tel formulaire sans aucune vérification de la provenance des requêtes ainsi formulées. Même si tout un chacun peut facilement usurper l'adresse d'un tiers, rien ne doit être fait qui facilite cela. Nous passons donc par une phase de demande de confirmation par mail.

Le principe de l'authentification utilisée consiste à demander par mail à l'intéressé une confirmation du service demandé. Si la demande de confirmation est reçue, cela prouve que l'adresse utilisée est valide. (Sympa intègre maintenant cette fonction d'authentification, il est possible alors de court-circuiter l'authentification dans les cgi d'abonnement)

Les paramètres de demandes de services nécessitant une identification du demandeur sont stockés et une demande de confirmation par mail est envoyée par le serveur à l'adresse concernée. On identifie la confirmation (et la demande de service en spool) à l'aide d'une clef numérique (calculée avec md5) qui figure dans le sujet du message de demande de confirmation.

Un simple reply (dans un délai de 48 heures) permet de valider la demande.

Ce n'est pas une signature PGP, (moins souple, moins sûr), mais on ne fait aucune hypothèse sur les outils des utilisateurs.



Le serveur d'authentification par demande de confirmation est conçu pour être utilisé pour valider toute sorte de demande de service déposé par interface WWW. Il est utilisé par exemple pour des inscriptions à des cours, ou les demandes de mise à jour des zones inverses du DNS.

Le serveur d'authentification et les formulaires adaptés à une vingtaine de robots différents (les robots sont décrits dans un fichier de config) sont disponibles sur

```
ftp://ftp.cru.fr/pub/CRU/Listes/authentificateur/
ftp://ftp.cru.fr/pub/CRU/Listes/weblast/
```

8.2 Serveur d'archives

Le traitement des archives demande de convertir un flot de message SMTP en un ensemble de documents HTML.

Hypermail et MHonArc sont deux produits voisins, ils offrent tous deux :

- la présentation des messages en HTML

- l'édition de listes de messages chaînés entre eux et triés par date, par sujet, par conversation ("thread").

Dans les deux cas, les accents 8 bits sont présentés, les URLs sont traduites en ancres, mais :

1. Hypermil permet de paramétrer la présentation des listes de messages, mais seulement à la compilation.
2. MHonArc présente bien les accents dans l'orthodoxie de MIME, c.à.d. même quand ils sont codés en "Quoted Printable".
3. MHonArc offre la possibilité de définir des filtres pour toute partie de corps MIME



MhonArc est un superbe produit écrit et mis à disposition par Elvis Hood <http://www.oac.uci.edu/indiv/ehood/MHonArc/doc/mhonarc.html>. Ce produit est utilisé sans modification pour fabriquer les archives de Francopholistes. MhonArc convertit des messages SMTP en HTML. Il est hautement configurable avec des fichiers de ressources permettant de modifier en profondeur l'aspect des archives obtenues sans modifier le code perl de MhonArc. Il est ainsi possible de faire des archives paginées comme dans Francopholistes ou bien d'utiliser des Frames. Bien entendu, les fichiers de config de MhonArc sont assez complexes, mais MhoArc prévoit une valeur par défaut pour toutes les ressources et on peut l'utiliser simplement sans fichier de ressources.

MHonArc est remarquablement simple à mettre oeuvre et il serait dommage de ne pas penser à son installation en même temps que celle d'une liste de distribution. Le principe d'application est d'abonner un archiveur à la liste. L'adresse de l'archiveur correspondant a un alias du type : `archiveur-java-u-strasbg.fr: "| /usr/local/bin/mhonarc - ... "`

En fait on a besoin d'enrober l'appel de mhonarc par quelques lignes de shell (certains préco-nisent d'utiliser procmail).

Nous vous proposons à titre d'information notre solution. <ftp://ftp.cru.fr/pub/CRU/Listes/archiveur>

8.3 Serveur d'archives : l'indexation

Compte tenu de la taille des archives, une recherche indexée est vite indispensable.

Marc-search (Eric D. Friedman) et MailSearch (Jerry Stratton) sont deux solutions proposées en contribution à MhonArc. Marc-Search offre des fonctionnalités intéressantes, mais n'est pas adapté à un usage intensif sur de grosses archives pour des raisons de performances. Nous avons dû renoncer à l'utiliser pour Francopholistes bien que n'ayant pas encore de solution de remplacement.

Glimpse est un des produits utilisés dans le projet Harvest. L'objectif de glimpse est d'indexer tout ou partie d'un système de fichier avec des possibilités de recherches très sophistiquées.

Les recherches ressemblent à `grep` mais outre les expressions régulières, on peut utiliser :

- et/ou logique (appliqué au message entier et non à une ligne)
- recherche approchée
- filtrage avant indexation (`gunzip` pour `*.gz` etc)

En outre Glimpse permet une indexation incrémentale très utile pour les archives de listes ainsi qu'un mode client serveur (`glimpsed`) permettant de maintenir en mémoire une base d'index.

Glimpse était utilisé dans une précédente version de Francopholistes avec une relative satisfaction.

8.4 Outils www d'administration

Rien de ce que nous connaissons n'est vraiment satisfaisant. Pour autant que nous avons pu le vérifier, la plupart des solutions citées plus haut sont simplement des habillages en mode www permettant de masquer l'interface mail du robot ; le retour des commandes se fait la plupart du temps par messagerie ; ces solutions ne permettent donc pas de s'affranchir des problèmes d'authentification et de l'inconfort du mode batch lié mode messagerie. De véritables fonctions d'administration en mode www imposent de pouvoir interagir directement avec les données du robot à partir d'un serveur http pour certaines fonctions, l'inconvénient majeur de cette solution réside dans une dépendance de l'interface d'administration vis à vis du robot de listes en mode messagerie.

La premier problème à résoudre est le problème d'identification de l'administrateur. Notre prototype d'interface d'administration utilise le mécanisme du serveur Apache (`.htaccess` `.htgroup` `.htpasswd`) pour faire ce contrôle. Quelques outils complémentaires pour gérer les mots de passe s'avèrent indispensables si vous vous adressez à un grand nombre de propriétaires de listes :

- changer de mot de passe
(`change_passwd.cgi` <http://www.genome.wi.mit.edu/ftp/pub/software/WWW/passwd/>)
- allouer et réallouer un mot de passe par mail

Moyennant ce contrôle, on peut proposer aux propriétaires de listes une page d'administration avec quelques données statistiques, l'accès à la liste des abonnés, un formulaire permettant de

modifier certains paramètres de la liste, une interface assistée pour la gestion des bounces, une interface pour assurer la modération.

La fenêtre ci-dessous présente la première maquette de cette page d'administration.

Administration de la liste windows-nt-fr
 semaine du 1/11 au 7/11

Le flux des messages

Nb d'articles distribués	76
Nb d'abonnés	558
Nb de messages générés	42408
Nb de rapports de non-remise générés	909
Estimation du % d'adresses erronées parmi les abonnés	2.14%

Le flux des abonnés

Nb d'abonnés	558		
Nb d'abonnements	par messagerie	par web	total
	15	0	15
Nb de désabonnements	par messagerie	par web	total
	16	0	16
Nb de ADD	0		
Nb de DEL	0		

La provenance des abonnés

8.4.1 Analyse des bounces assistée par web

Analyser les rapports de non remise est particulièrement difficile à automatiser, mais il est possible de simplifier le travail du propriétaire de liste en séparant sa boîte aux lettres de celle dans laquelle il reçoit les rapports de non remise. Notre idée est de construire sur le serveur WWW une archive des rapports de non remise. La présentation HTML de ces messages permet de mettre en évidence les adresses incriminées, de proposer une fonction de rapprochement des adresses présentes dans les rapports de non remise de celles de la liste des abonnés ainsi qu'un bouton pour activer le désabonnement d'une adresse sélectionnée.

En complément, nous encourageons les propriétaires de listes à utiliser régulièrement la fonction purge de SYMPA. Ce système permet de demander aux utilisateurs de confirmer leur abonnement et de désabonner ceux qui ne le font pas (voir documentation SYMPA).

8.4.2 La modération par web

Un service en projet depuis longtemps... Si on laisse le robot de liste de diffusion envoyer à chaque modérateur de liste un exemplaire du message à modérer, il est alors très difficile de proposer des outils compatibles avec la modération par plusieurs personnes. En effet, le message étant dupliqué dans les boîtes aux lettres de chaque modérateur, il n'est plus possible de faire coopérer ceux-ci pour assurer que chaque message soit traité une fois et une seule.

Pourquoi ne pas envoyer les articles à modérer dans une boîte à lettre spécifique et partagée par tous les modérateurs? Une archive HTML des messages en attente de modération permettrait de centraliser les messages et de donner un accès partagé et contrôlé par mot de passe. Il suffit alors d'ajouter dans chaque message un bouton pour le rejet ou au contraire validation et la diffusion aux abonnés. On dispose alors d'un bon outil pour la modération si l'on n'a pas choisi d'éditer systématiquement les messages avant diffusion.. Autre avantage : la méthode pour valider la modération devient indépendante du type d'UA utilisé par le modérateur. En effet, la modération utilisant des entêtes SMTP n'est pas fiable : chaque UA utilise différemment les entêtes SMTP Resent ou X-Sender dans les fonctions redirect ou redistribute par ailleurs ces entêtes ne sont pas plus sûrs que le champs From.

9 Netiquette, obligations administratives

L'internet étant sorti des labos de recherche, il est légitime de s'inquiéter des conséquences que peuvent avoir des dérapages dans l'utilisation des listes de diffusion.

Sans fantasmer sur les risques juridiques liés aux listes (ou aux serveurs web), il convient de prendre quelques précautions. Rappelons les principaux incidents auxquels sont exposés les listes de diffusions :

- diffusion de pub vers la liste des abonnés
- exploitation des listes d'abonnés pour diffusion de pub
- propos constituant des infractions ou délits tenus dans la liste (diffamation, haine raciale, sexisme, etc)

- polémiques stériles et mortelles pour l'intérêt de la liste
- propos hors sujet, perte de l'objet de la liste
- ...

Pour prévenir ces dérives ou en atténuer l'impact, il convient de prendre quelques précautions. La première d'entre elle concerne l'investissement du propriétaire de la liste dans ses contenus. A cet égard, nous vous conseillons de mettre en place deux chartes d'utilisation des listes de diffusion : une charte type à usage des abonnés, une charte à usage des propriétaires de listes.

Au delà des aspects strictement légaux, la netiquette (le savoir vivre appliqué au comportement sur le réseau) est indispensable pour maintenir l'intérêt des listes de diffusions. Il revient aux propriétaires de listes de rappeler si besoin quelques règles et conseils :

1. Respecter l'objet de la liste (défini par un thème ou une population visée).
2. Attention au prosélytisme de tout poil engendré par la passion. Sur le réseau, plus encore que dans la vie civile, les sollicitations abusives sont mal acceptées (en particulier le harcèlement commercial).
3. Gare au effet du langage parlé-écrit qui caractérise la messagerie. Cette façon de communiquer favorise la spontanéité et le parlé direct, mais on doit garder à l'esprit que si les messages sont parfois écrits en langage parlé par leur auteur, ils sont bel et bien reçus comme une communication écrite. Cette dissymétrie explique souvent les traits d'humour mal compris, la polémiques qui fait rage, les contre-sens fréquents, voire les messages complètement incompréhensibles.
4. Plus le nombre d'abonnés est important, plus vous devez prendre soins à la rédaction de vos messages. Prenez pour référence des modes de communication par l'écrit (courrier papier, article de presse, article scientifique).
5. Si vous commentez un autre message, conservez en uniquement les passages significatifs et abstenez vous de redistribuer un long message affublé de je suis d'accord.
6. Accentuez vos messages, dirigez les discussions sur le codage des accents dans le mail vers la liste smtp-fr et recommandez la lecture de <http://www.cru.fr/listes/apropos/accents.html> à ceux que les aspects techniques de l'accentuation intéressent.
7. Ne pas diffuser de document attaché gigantesque s'ils peuvent être installés dans un serveur d'information. Ne pas diffuser en mode ouvert des objets au format propriétaire (tout le monde n'a pas Microsoft et Word6 !)
8. Ne propager qu'avec parcimonie les messages reçus par d'autres listes.
9. Attention à la diffusion des reply. Positionnez éventuellement un Reply-To .

Il appartient aux propriétaires de liste de donner le ton. Le meilleur atout pour obtenir une liste pertinente c'est de la modérer. Vous pouvez le faire périodiquement : à la création de la liste pour filtrer le bruit et mettre en valeur les informations, et chaque fois qu'un relâchement est constaté ou dès qu'une polémique hors sujet s'allume. Il est indispensable que le propriétaire de la liste s'investisse sur le sujet et mette un point d'honneur à ce que les questions pertinentes reçoivent toutes une réponse de qualité.

9.1 CNIL

Un fichier d'adresses est un fichier sur les personnes. Au terme de la loi du 6 janvier 78, le traitement de ce fichier doit faire l'objet d'une déclaration auprès de la CNIL (d'une demande d'avis dans le cas d'une administration). Cette loi et les obligations administratives qui en résultent ne doivent pas être considérées seulement comme des contraintes. Ce sont aussi des protections dont nous avons besoin.

L'application de l'esprit de cette loi aux listes de diffusion conduit à réaffirmer les principes suivants :

- tout doit être fait pour faciliter le désabonnement ;
- chaque abonné doit être informé précisément de l'usage qui est fait de son adresse ;
- chacun a le droit d'accéder aux informations le concernant ;
- la liste des abonnés d'une liste sur un sujet donné constitue une information nominative sur l'utilisation des médias, c'est donc une information qu'il appartient aux gérants de site et aux propriétaires de protéger.

Pour respecter ces principes, nous vous conseillons de

- Documentez avec soin votre système de listes dans les messages de bienvenue et si vos logiciels de liste le permet ajoutez dans chaque message une entête `X-url-désabonnement :`
- Poster à chaque abonné un bilan individuel de ses abonnements avec un mode d'emploi pour se désabonner. Vous trouverez notre script pour faire ce travail dans `ftp://ftp.cru.fr/pub/CRU/Listes/bilan`
- Faites une purge périodique des abonnés en demandant confirmation de l'abonnement (certains logiciels de listes sont équipés pour, sympa en particulier. Il n'est pas très difficile de simuler leur fonctionnement quand le logiciel n'en dispose pas. Cette étape est très redoutée des propriétaires de listes qui surveillent leur audimat car l'effet sur le nombre d'abonnés est drastique. C'est cependant la seule méthode radicale pour expurger les listes d'abonnés d'adresses périmées et surtout, c'est le moyen de recueillir périodiquement l'assentiment des abonnés sur les termes de la charte d'utilisation de chaque liste.
- Ne laissez pas les listes d'abonnés publiques

- Ne laissez pas aux personnes non abonnées le droit de s'exprimer dans la liste.
- Installer un dispositif anti-spam sur votre moteur SMTP.
- Déclarez votre service de listes de diffusion à la CNIL. C'est un bon moyen de faire le point sur l'ensemble de ces aspects et de sensibiliser **propriétaires** et abonnés sur leurs responsabilités.

Le CRU a rencontré la CNIL pour lui présenter le principe des listes et élaborer une demande d'avis type. Vous pouvez vous aider avec le dossier <http://www.cru.fr/securite/CNIL/listes/index.html>, vous y trouverez en particulier des exemples de chartes.

