This document is a description of Sympa project direction. The goal is to fix priorities and to publish it in order to collect user comments. The initial version (November 2005) describes development of project from Sympa version 5.1. We will try to keep it up to date but no guaranties.

Of course, Sympa forge server includes a bug report and feature request system that are used too.

We kindly suggest you to comment this document using the wiki feature.

Performances

Performances becomes a limitation because many sites now create thousands of lists. Many universities create lists automatically (through families) using data coming from the information system. This performances issue concerns mainly the **web interface**.

web interface with many lists

We already listed some problems:

- startup delay
- response time
- memory usage

In version 5.2 startup is already optimized using binary version of list config file (so initial loading of configs is much faster) and the number of *which* calls is limited when operation is related to a list.

We still need to improve the response time for all other request. We will explore the following directions :

- ~~sharing memory storage of each scenario (currently duplicated for each list)~~
- ~~optimize and limits number of SQL request~~
- ~~do_log subroutine calls might be expensive~~
- ~~List::new calls might be expensive (example: testing the date of config files)~~

List families

List family instantiation is currently only available via the command line, as a bulk process. You provide a set of data and a family definition and sympa.pl instantiates the family to create all the lists.

We plan to extend the family features and allow list creation from the web interface and also from the mail interface (to create one-time lists).

task manager

- spool management needs to be optimized

perl memory leaks

- Not sure, comments welcome!

restart email distribution

Currently (version before 6.0) the distribution process of a particular message for one list can't be stopped. There are a few issues with that.

- Sending TERM signal to Sympa process may take a while to stop the process if sympa.pl daemon is distributing a message for a huge list.
- Sending signal 9 may result in duplicated messages. Restarting Sympa the distribution process will start again at the beginning and some subscribers will receive the message twice.
- The priority process is evaluated each time Sympa reads the spool content but a message with high priority will have to wait until current distribution is terminated.

It should not be too difficult to change Sympa in following way.

- sympa.pl first process should prepare the distribution making small subsets of recipients that a second process could handle.
- The second process could then evaluate the priority after distributing each subset. Stop would be very fast and restart would just finish the distribution.

This architecture is implemented in version 6. See server_architecture for big services.

Scalability

First, it could be very useful to separate different daemons on separate servers. This can be done already but there are some limitations because everything is not in Sympa database (configs, archives and spools).

Second, a daemon could need load balancing.

lock files

In order to ease repartition of Sympa service on multiple server, lock files should be done using a MySQL table.

Using memcached

memcached is a free software tool that allows storing data in RAM and share access to those data using a inet or unix socket. May be this tool can be used in order to solve sharing of data between different daemons. This solution would provide high performance and load balancing but it seems not

compatible with our need of hi disponibility.

moving config files and spools to the database

This would make the application more complex and difficult to manage using your favorite editor file, but as long as the administration web interface is complete and sophisticated this should not be a problem.

This work as started in Sympa 6.0. A new sophisticated bulk mailer is now running on our development server. It uses the database to store sets of email recipients and a new spool for "ready to send" messages. A new daemon `bulk.pl` submits those messages to MTAs. The new bulk mailer properties are the following.

1. Scalability. Multiple servers can run multiple `bulk.pl` without any other configuration than a copy of sympa.conf and each robot.conf so those daemons can run on different servers.
2. Better priority management. Now a message for a high priority list can be distributed before a message for a low priority list message even if this message distribution has started before the new high priority message is received by Sympa server.
3. Easy stop and restart, check point. Until 6.0 is Sympa is stopped with a `kill -9` during the distribution of a message to a large list. The distribution will restart at the beginning when starting again sympa.pl and many subscriber can receive this message twice. This problem is now solved and `kill -TERM` stops Sympa with a very short delay even if it is distributing a message to a huge list.

Authentication architecture

Some improvements of security and usability are expected.

password storage (done in version 5.4)

~~We plan to replace password encryption with hashing (MD5 or SHA1) in version 5.5. So reminder feature will be replaced by~~ *reset password* ~~in a similar way as other web application.~~

1. ~~WWSympa creates a~~ *one time ticket*~~, store it in a dedicated table (one_time_ticket_table) and sends it to the user.~~
2. ~~If the user goes to the web interface, this ticket is~~ *burned*~~, the user is authenticated and prompted to change/choose password. This way a user never has to use a password he didn't choose.~~

The method using a "one time ticket" could be used elsewhere in Sympa. In many cases Sympa sends some URL to users, it is useful that those URL are considered as a way for authentication.

auth.conf

- ~~auth.conf is currently a global file. We will make it possible to define different auth.conf for each virtual robot.~~(done)
- Some want to use HTTPS only for encryption of login form. This needs to be introduced in auth.conf

export password in LDAP

If Sympa could export passwords in a LDAP directory, it would be possible to use this LDAP directory for some other application (single sign on such as CAS).

SSO without email patch

Done: ~~We received a big contribution from John-Paul Robinson from University of Alabama at Birmingham. The goal of this patch is to change the Shibboleth-based authentication so that Sympa does not use email addresses as user identifier. We will introduce this contribution in main Sympa version quickly.~~

n-tier architecture

~~This section concerns the SOAP server. It is needed for some remote application acting as soap client of Sympa server to make some authentication assertion about the end user. For example a CMS with its own authentication method can request services to Sympa as proxy of the end user. That can be done if Sympa SOAP server trusts the remote application and accepts a set of assertions from this SOAP client. Sympa would use incoming variables in authorization scenario.~~ (done)

A Sympa server could also request services from another Sympa server. Imagine that "which" action returns could tell the user about his subscriptions on another mailing list server.

DKIM

W DKIM is an emerging standard for message authentication based on signature. Sympa will check DKIM signature to the benefit of authorization scenario. It is also needed to verify that message modification made by Sympa (headers and body) doesn't alter original DKIM signature. In some case Sympa could sign messages itself when authentication of the original sender is good enough.

Alternate email addresses

Sympa allows you to declare an alternative_email_attribute LDAP attribute in your auth.conf file.

When authenticating a user via LDAP this information is retrieved. We currently have VERY limited use for this data.

- Stored in a HTTP cookie
- Used to propose email unification (subscriptions with address A are changed to address B).

What should be done in the near future to extend this future

Customizing mail content per subscriber

A lot of work has been done last year in order to allow parsing of message footer and header using all variables of Sympa templates including the subscriber email. This would make it possible to add a message footer with personalized unsubscription instructions or to use VERP for better bounce management. This needs to bypass the Sympa internal bulk mailer that saves some bandwidth. Now everything is ready to start this new development. We plan to allow both current mail grouping and mail parsing depending on the need (example use parse for unique return path only for subscribers which get bounces or every 10 messages send to the list etc.).

User preference

In addition to session attributes, user preferences and attributes could be introduced.

- Subscriber attributes defined by list owner. The management of subscriber attributes could be performed by the list owner or by subscribers themselves. Attributes could be hidden to other subscribers or not. Attributes could be used in scenario?
- List owners need to tag some boor users in order that any mail from them is submitted to list editor before distribution.

Shared document repository

Some more features in the shared repository document.

- Get a complete folder as a zipped/compressed file.
- Display a header and a footer of a folder.
- Encoding file name. There is big bug related to file name containing some 8bits chars.
- Move, remove, rename a folder or a file. Needed if you have to reorganize a folder.
- Display folder content recursively. (Something that look like `ls -lR` unix command.)
- Provide a basic search engine (using filenames, not contents).
- Some want to restrict access to some part of the shared, some want to open more access on sub-folder. The solution we have chosen is something like Unix access heritage from directory to subdirectory but it is difficult to understand (and to explain). We plan to introduce a "inherited access" attribute to a folder. With this attribute any subfolder of that folder will have exactly the same access control rules.
- Replace HTMLArea WYSIWYG-HTML editor because it is not supported any more.

Internals. Work has been started to rewrite an object oriented version of this code. The work needs to be finalized. …

Bounces management

Extending bounces notifications. See Pour une meilleurs remontée des bounces, in French.

Logs and OPT/IN traceability

subscription traceability

Many spammers claim their "service" is based on opt-in but that's false. Sometimes users claim they never subscribe to a legitimate mailing list, but in fact they did ! Subscription traceability will provide information about the users subscription, on the web interface (initial subscription message, confirmation). This is ready (nice work from Floriane Fiquet) and it will be integrated in a very near future.

Internals

- ~~Add a particular session attribute that allows debug mode.~~
- ~~Add a particular attribute that displays variables available for TT2 parsing so no more documentation is needed for that.~~
- ~~Replace CVS by subversion to manage sympa sources.~~
- Check point during distribution process so it could be possible to stop sympa.pl during a huge message distribution without duplicate any message
- Currently listmaster is listowner of any list. In some cases listmaster must have more restriction then listowner.
- Archive should include new message as soon as possible preferably just when the distribution process starts. A message should be copied to the outgoing spool before the distribution process start.
- ~~Test PAPI authentication method.~~
- ~~utf8 encoding. Charset encoding is depending on server file system, .po catalogs and client. We need to fix some bugs but we also need to consider global organization for encoding usage.~~
- ~~Full virtual robot. Different virtual robots should be able to manage different lists using the same name. The work is almost finished in main the CVS branch.~~

Documentation

The reference manual needs to be completed regarding some topics. Other documentation

documents would be of interest (quick start, developers doc, list owner documentation, online documentation).

List members management

~~This chapter should explain the difference between subscription and inclusion. It should also explain how include2 works.~~

Internationalization

~~This chapter should explain how Sympa manages translations and how new translations can be added. It would also cope with common problems with encoding (mail, web, archives,..).~~

List management

A new chapter should deal about list management (rename, close, import, export).

Owner / Moderator howto

WWSympa should provide online help for owners/moderators, describing their role, tools.

Miscellaneous

List management

- List import and export features for listmasters.
- Import subscribers via upload of a csv file.
- Import archives via upload of archive in another MLM software format.

SOAP related

- Allow scenario testing. Submitting a message via the SOAP interface could be a good API for those who need a milter in order to reject message during SMTP session.
- soap_extensions

Other

- For a message distributed to multiple list, block multiple distribution to same subscriber.
- Prefered skins per user.
- ~~CSS. Fix some display bugs.~~
- ~~Display last login date and IP when login.~~
- Display multiple message in a single web page in archive.
- Check compatibility with W IPv6, for example allow access condition based on IPv6 address in scenario.
- Attributs utilisateurs par liste (French only)
- External plugins in archived.pl
- Handling multiple similar notifications.

From:
https://www.sympa.org/ - **Sympa mailing list server**

Permanent link: ❌
https://www.sympa.org/dev/project_direction?rev=1282804334

Last update: **2010/08/26 08:32**