The Exim Mail Transfer Agent in a Large Scale Deployment.

Dr Yann Golanski*
The Planet Online Ltd.
1 April 200

Abstract
This paper is a recommendation on how to use Exim in a large scale deployment environment. It focuses on the pros and cons of using Exim and reports on how Planet Online uses Exim in a successful way for its Connect and Go system.

1 Introduction
1.1 History
Exim\(^1\) was written in 1997 by Philip Hazel\(^2\) of the University of Cambridge Computing Service as an experimental mailer, loosely based on Smail 3. Since then, Exim has been used by many universities, private companies and internet service providers – Planet Online fitting the latter category.

1.2 Philosophy
Exim uses a simple, yet powerful philosophy stating that on a well connect to the internet host, 99% of messages can be delivered right away. Thus exim does not have, nor needs a complicated queuing strategy, and special actions need only be taken either in time of local stress or if the first attempt at delivery fails.

The design of exim centers around the delivery of messages on reception either to a localhost, local user or to a remote host. There is only one pool of messages, that can be scan by exim queue runners. There term queue runner is misleading, as exim does not have a queue. All the waiting messages are in a so-called spool directory where they are all available. Exim queue runner is just a process that will open each header file and decide to take action or not. Those queue runners run at fixed

\(^*\)yann@theplanet.net
\(^1\)Available from www.exim.org
\(^2\)email: ph@cus.cam.ac.uk
times, and will die only after having looked at all messages in the spool.
The different exim processes do talk to each other via a hints database
(which is not required, they are only hints) and log files. This design
assures that there will never be a single long lived process apart from the
exim daemon\(^3\) who listens to port 25.

1.3 Process Interactions and configuration

A full review of how exim processes interacts is beyond the scope of this
paper, but a small overview follows.

When exim gets a mail messages on its SMTP port, it forks a new
process that write that messages in the spool. The same thing happens
when a local user's mail user agent called /usr/lib/sendmail\(^4\) to send a
message out. Once this message is in the spool, a queue runner will pick
it up and try to deliver it.

If the message is local, a director is used to deliver the message. If it is
not then the message is passed to a router that tries to route the message
to its final destination. Routers can, if they find that the message is local
after all, send it back to a director. This is due mainly to the Small 3
inheritance. Philip Hazel has stated that if he were to re-write exim then
only one of those would be necessary.

Whichever way the message takes (either router or director) it is passed
to a special transport that transports the message to its destination –
either a remote site, a local user mailbox, or whatever else the system was
configured to use.

Exim can use alias files, rewrite certain headers and call on global
filter files to curb things such as spam, or to perform any other task the
administrator wants. Exim can use a wide range of tools, such as ksearch,
data bases (cbl, ldap, mysql) lookups and perl-like regular expressions. A
full description of exim's configuration can be found at the exim web site
located at \url{www.exim.org}.

2 The Connect and Go Architecture

2.1 Background

The Connect and Go (CnG) was implement at Planet Online to meet the
need for a dial-up ISP solution for the launch of freeserve\(^5\). A set a Linux
servers was chosen by Nigel Metheringham now of Vdata and Steve Clarke
to host all the systems which includes sign-up, databases, domain name,
web, news and mail servers.

\(^3\)This is not strictly true. The main exim daemon forks a new exim process that is dealing
with the every new connection. That forked process should be set not to run as a privileged
user.

\(^4\)Which, of course, is a symbolic link to exim.

\(^5\)Freeserve is a sub branch of Dixon PLC. It was the first ISP to provide free internet
connection in the UK, with free web space and email. Local calls are still paid by the end
user, but there are no other hidden costs. At the time of writing, freeserve is the biggest UK
ISP with 2.5 million users. More information can be found at \url{www.freeserve.net}
Figure 1: Network MRTG graph of outgoing (top) and incoming (bottom) servers.

So it is against this Linux background that the following paper will focus. Of course, exim runs on many different implementations of Unix and your local system may perform better if you tune Exim to your local OS.

2.2 Overall Architecture of the System

Mail is handled by a cluster of Linux machines, all running the same kernel and exim configuration. The hardware is constant over all machines. Currently, those are Compaq 1800 pentium 2 with 256 Mbits of RAM and 5 times 9 Gigs SCSI hard disk, one used as a boot disk\(^6\) and the other 4 disk as exim main spool. The latter are in a RAID set, allowing for one of those disks to fail\(^7\) without lose of data.

The email service is split into three different servers: incoming, outgoing and fallbacks. Incoming servers are design to take all email traffic from the world to our CnG domains\(^8\). This is done by having the MX records in DNS of all node.vendor addresses pointing to the incoming cluster. The outgoing servers are designed to relay all mail for our CnG users. All mail

\(^6\)This disk contains all the Linux OS, and its utilities. Note that it is a cut down version, and as such does not have things like a compiler and/or linker to avoid problems in case of a crack of the machine.

\(^7\)This is a question of when rather than if

\(^8\)which are of the form user@node.vendor where the node name is the user ID, and the vendor is the CnG vendor (for example freerve.co.uk). The user part can be set up by the customer to have multiples emails with the same node same.
they send, will go via the outgoing servers. Those are in the majority in our system. The latter servers as well are used as POP servers for the CnG users. Utilisation can be seen in Figure 2. If mail in the outgoing server cannot be delivered for some reasons, it is then forwarded to the fall-back servers that will take care of it in its own time. This is mainly to reduce the size of the spool on the outgoing servers. This can be seen in Figure 4, and will be discussed later on.

Figure 1 shows the MRTG network traffic graphs for an outgoing and one incoming server. The strange wave-like pattern exhibited by both server is mainly due to UK phone tariffs which are cheaper after 18:00. Most users will come home from work then log in when the rates are cheaper. Weekends tend to be quieter periods.

CnG users mail is stored on a Network Appliances Filers. Those are mounted via NFS so that all mailers can access them. At present, the CnG uses clusters of Network Appliance filers with several terabites of capacity. See www.netapp.com for more information on their large scale storage systems.

2.3 Exim’s Architecture

2.3.1 General Installation and Peripherals

Three points need to be address before delving into a a more detailed discussion.

Currently, version 3.13 of exim is used. It is important to keep up to date with new versions, as small bug fixes and performance improvement are constantly added. Of course, one runs the risk of bugs creeping into new code, but so far no major bugs were found. It is easy to upgrade only one mailer and test the code, then roll out the new version onto all other mailers once one is sure of its stability.

Secondly, Exim will make a lot of DNS queries, to retrieve all MX records for all the mails it has to deliver. It is obvious that a local dedicated DNS cache server with plenty of memory will improve the delay between lookups.

Lately, there is no need to stress that network traffic will be great on any large scale systems, making the network a possible bottle neck to mail transfer. Figure 1 shows what typical networking you may have on one server.

2.3.2 Server and Exim Configuration

Exim uses two main directories namely the spool and hints ones, for all its operations. Both those directories need to be on fast I/O disks – section 2.3.3 will address this issue in more details.

The hints databases are just that, hints. If one exim process fails to write to them, it does not matter. No email will be lost. The advantage is very similar to caching. Every domain and user will have a list of hints associated with it. Say a domain is down and the hints say it is down. All subsequent received messages for that domain will not be delivered unless the retry time is reached. This will therefore save CPU time, memory
and bandwidth as less connections need to be made. Hints will help your system a lot, but in some cases it is beneficial to delete all hints and start afresh. Such a case might be when a major internet connection fails for a few hours.

The spool directory will be very heavy on IO. Exim writes two files for each email it receive. One contains the message body and the other contains the message headers. They are labeled with a -D and -H flag respectively which is added to the message ID. By default Exim, stores all those files in one directory. If the traffic of mail is large enough, then this will be inefficient. If the option split_spool_dir is set, the spool will be split into 62 sub-directories and messages will be split between them. Those directories are based on the hex representation to the time part of the exim message ID. In version 3.14, Philip Hazel has promised an improvement in the way the queue runners deal with the split spool. Currently, when a queue runner is created (either by the daemon or by cron) it stat the top directory, and then goes down it. The new version will state sub directories, and deal with messages only contain therein. In either cases, the subdirectories are scanned randomly.

Another directory tree that exim need access to is the local mail is stored. The classic mailbox format is not suitable as each delivery needs to lock the whole mailbox before actually saving the message. The Mailbox format, in which each email is kept as a separate file, is preferred as multiple deliveries can happen at the same time. In case the user has a problem with one message, it is easier to deal with rather than scanning his whole mailbox. Another not as yet implemented feature would be to add the size of each message to the name of the file which contains this mail. The benefit is not from a exim point of view, but is a POP one. As POP needs the exact size of messages, it needs to stat every file within the directory. If the correct size is in the name of the file, stating the directory will provide all the necessary information.

Figure 2 shows the number of POP processes for a POP server – which doubles up as an outgoing server.

The structure of those directories need to be set up in a fast and continent way. One way to do that, is to use exim nhash function. Nhash

---

9Exim ids its mail with the Unix time in hex a dash a hash on the message sender another dash and a hex number in case of collision. A typical message id would look like 12cb2-0008DY-00.
will take a user name and hash it in a mod-div that will return two numbers. Say your system directories are divided into ten volumes on a filer. Each one of those volumes has 600 sub directories. Within each of those, lays the users maildir. For example, user foobar.freeserve.co.uk will have her maildir in /mail/spool/9/986/foobar.freeserve.co.uk. The hash is fairly good and at the time of creation was able to split two million users into roughly equal groups. Another hashing function is available in exim ($hash) but its split is not as good as the one given by the above nhash.

Queues or rather the spool tree needs to be kept to a minimum as well. This is a direct result of exim’s implementation philosophy, as the shorter the queue the faster new messages will be delivered. Note that it is not efficient to create queue runners at short time intervals. If too many queue runners are created, then some will try to examine a message while another queue runner is dealing with it. This will cause the first exim process to freeze waiting for the lock to be lifted.

Figure 3 shows the number of exim processes during a typical week. Of those, some are delivering messages, some are queue runners and one is a daemon.

Figure 4 shows the number of messages on the queue for both kind of servers. Incoming servers tend to have very light queues, with some exceptions at time of high load. The one shown in the top graph of figure 4 was due to a user who was getting many mails form an excessive number of mailing lists. Graphs such as this one can alert the administrator to those kind of problems. On outgoing servers, the mail spool mainly consists of frozen messages, leaving an average queue of 10 messages. Those messages are frozen when they match a set of strict patterns that suggest that this message could potentially contain spam. More advanced system checks are then needed weather to pass them onto the Planet Abuse Team or to
un-freeze that message for delivery. However, any queue runner will still have to open all those messages and check if they are frozen or not. There is a compile time option to move those messages in a frozen spool, but so far only a Planet (and thus Linux) specific tool\(^\text{10}\) exist for dealing with them.

Another good way to keep the queue short is to use the `$message_age` with the fallback servers. If a mail cannot be delivered in a short time (say a few hours) then it is passed onto the fallback server which can afford to have a longer queue. If a message cannot be delivered after a few hours, it is unlikely that the reason will go away soon, and thus the message becomes none-urgent.

Unsolicited commercial email, also known as Spam, is a major problem for any ISP. The amount of junk mail that arrives can slow down the service and will certainly annoy most users. Exim has some spam fighting capacities, but a thine line need to be walked between overload and under-checking. Exim can use the Real-time Blackhole List (RBL) of maps.vix.com and any other you can to subscribe to. Not setting those is asking for your servers to be abused. Exim global filters can perform some tests on emails, but here is really where the thine line starts. It is easy to write a small perl script that will scan a mail and return a value of 'spam likeness'. However, this option is not useful at all as it will take too long to run and will kill the system. However, it is possible to curb spam that way and freeze messages for humans to look at.

All of those are aimed at creating a service that can easily be expanded sideways, by the additions of new servers. However, the main bottle neck that exim has is the disk IO, which is discussed below.

---

\(^{10}\)The tool, called eXim, is available under the GNU Public Licence (as is exim) and can be downloaded from the ftp (ftp.exim.org) site.
2.3.3 Exim’s Disk IO

Figure 5 shows the average disk IO, as given by the sard utility for all the disks in the system. This includes reads and writes, but makes no difference between them. The blue lines are all in the RAID set, the black one is the boot disk – which contains the exim binaries.

2.3.4 The Competition

Why use exim and not some other MTA such as Sendmail or qMail?

Exim provides a secure and simple MTA, that can be configured to the administrators’ content to do a wide variety of jobs. Its whole design makes it ideally suited for large or small systems. Sendmail is notorious difficult to configure and has a reputation to be very insecure. Qmail is limited by CPU, and does not have as many configuration possibility as exim.

As so many systems vary in little details, a comparison between all MTAs is not really relevant. One could conceive of a set up where the incoming and outgoing servers would run exim and the fallback qmail due to its queue management utilities. Figures 6 and 7 and 8 are given as an indication of the CPU, load and memory utilisation on typical outgoing and incoming servers. Sadly corresponding graphs are not avaible for Sendmail and Qmail as Planet uses neither of those MTAs.

Exim, while mainly written by Philip Hazel, has many contributors. Many individuals and organizations have asked for features that are resulting is still small and feature rich MTA.
Figure 6: Percentage of central processing unit used in different user modes.

Figure 7: Load of under which the server is running. This is an average over 5 minutes.
3 Conclusion

This paper shows what can be done to improve exim’s performance on a heavily loaded and large system. It is by no means a definitive guide, but it should give mail administrators and developers a better understanding of how exim can be used.

The author would like to thank Philip Hazel for his constant efforts that makes Exim what it is.