IMAP, POP, and Mail Servers

Or - now that I have distributed computing, why can't I have distributed email?

The Problem?

- We live in a world of distributed computing and multiple systems. Many of us have accounts on very large numbers of systems. It's annoying to only be able to read email on the one machine you've singled out for your mail delivery host.
- Delivery to personal workstations can be a bad thing - they're often not backed up (at least not religiously) and may be down for extended periods. They're not carefully maintained machines by and large. There are a number of security issues - do you trust everyone on your cluster who knows root passwd?
- Maintenance of email receiving functionality on thousands of machines is a burden.
- Why must **every** machine run sendmail?

The Problem?

- Users often want to access email while off-site, in some way that doesn't involve logging in to a system at FERMILAB.
- Our current means of dealing with email delivery for mh is prone to error - users often botch exmh auto-incorporation setup, for example - and why should they **have to** understand these issues so intimately just to get their email delivered?
- The UNIX email spooling concept still stems from single-machine environments. The mail spool is inefficient (one big file) and easy to corrupt.
- Users in a cluster such as FNALU want to be able to read email from any node in the cluster, particularly if one is down for maintenance.

Solutions?

- One solution used at AFS sites is to deliver email into mailboxes in AFS space by having email sent to a mail server that does this. AMS mail uses this solution, and it has been used successfully by many sites.
  - Not very scalable
  - Tends to cause big AFS loads due to very large numbers of callbacks issued.
  - This can be done without AFS, but has even worse NFS problems.
- At FNAL we tend to have people use exmh and have them autoincorporate their email into their AFS home directory.
  - Easy for users to break this.
  - All hell breaks loose if AFS servers go down, disks go bad, networks fail, etc.
  - Pretty specific to MH.

Solutions?

- Many sites use NFS mounted mail boxes so that a single server can hold user boxes mounted across many clients.
  - » This is hard to get right - different file-locking algorithms in different clients, NFS interoperability (lockd is always iffy) , interoperability between different clients (why can't vendors all use the same mail UID, for example?).
  - » Sites that do this often have mystery problems - garbled email, dropped messages, etc.
  - » What happens when you open 2 email clients on different machines at the same time? It's unpredictable.

POP - the traditional solution

- One common solution for this problem is to use POP protocol. POP is a common wire protocol for email retrieval. In this model, email is sent to a commonly-shared server, where incoming email is stored. Users retrieve incoming email from the POP server using either a POP-compatible email client, or using a POP program that retrieves email from the server into their normal local inbox. There are many POP servers and clients available on most platforms. A list of some of these is in the handouts. This is a big win in many respects.
- Think of POP as a read-only source of email for users. You start the client, connect to the POP server, and download all your new email.

POP deficiencies

- POP is an “offline” protocol - you download the email, and then work offline. Users often want to work “online” where they work on the email directly.
- POP requires that you get all of an email message at once, whether you want to read all of it or not. If they’re big MIME messages, you get all the enclosures as well. Many clients automatically retrieve your whole mail spool.
- Therefore, POP can chew up a lot of network bandwidth for someone who just wants to see what their new email is.
- If you’re on a laptop at a conference, for example, you now have all your mail on your laptop, not where you really want it to be, like at work, foldered away.

POP advantages

- Main one is that there are **many** POP clients available, and many POP servers, and they exist on just about any client you might wish. It's ubiquitous.

IMAP - the solution of the week

- IMAP is a more capable wire protocol, but is far from new. It was conceived at Stanford in 1986, and implemented on DEC-20. The IMAP2 standard dates to 1988, and has had several improvements since.
- Current standard is IMAP4, which is a Proposed Internet Standard. There are now many extensions to the basic protocol, and many of these are being put in the standard as well, including protocols to decide which extensions a server supports. It is quickly becoming a standard, with new commercial and free products coming out all the time. There is also an IMAP industry consortium.

IMAP Advantages

- IMAP can support email serving in three modes:
  - Offline, like POP
  - Online, where users may connect to the server, look at what email is available, and access it online. This looks to the user **very** much like having local spool files, but they're on the mail server.
  - Disconnected operation, where a mail client connects to the server, can make a "cache" copy of selected messages, and disconnects from the server. The user can then work on the messages offline, and connect to the server later and resynchronize the server status with the cache.
- The big difference is that with IMAP the primary storage is on the server, not on the local machine.
- Email being put away for storage can be foldered on local disk, or can be foldered on the IMAP server. Although IMAP can do Offline access, it's not a normal mode.

IMAP Advantages

- The protocol allows full user of remote folders, including a remote folder hierarchy and multiple inboxes.
- It keeps track of explicit status of messages, and allows for user-defined status - maybe marking messages *really-super-important.* This allows read messages to really be marked read, for example.
- Supports new mail notification explicitly.
- Extensible for non-email data, like netnews, document storage, etc.
- Selective fetching of individual MIME body parts.
- Server-based search to minimize data transfer.
- Servers may have extensions that can be negotiated (more later).

Proposed or current IMAP extensions

- ANNOTATE extension
- Internationalized Mailboxes
- STATUS extension
- QUOTA extension
- ACL extension
- SORT extension (proposed)
- SCAN extension (proposed)
- Low-Bandwidth extension (proposed, but implemented in ROAM)
- Server-based forwarding

IMAP problems

- Still not widely available, but this is changing fast. More commercial implementations are available all the time. There are MAC, UNIX, PC, NT, etc. clients, and this is where most of the industry thrust is headed.
- At the moment there are sometimes interoperability problems due to the changing protocols - for example, pine 3.91 has some problems with the Cyrus IMAP server - but 3.92 and later works fine.
- Not all clients support all the possible features - and none of them is a clear-cut winner in a lot of cases. Many clients are still based on IMAP 2 or IMAP2bis. IMAP2bis is mostly a matter of folklore rather than a standard. It is mostly used by pine, but pine is by far the most common IMAP client.

Project Cyrus - an IMAP Implementation

- Our current test server is based on the CMU Cyrus project IMAP server. This is part of the Andrew2 project at CMU, and is intended to serve as a very flexible mail system, with many extensions. It was done primarily as a replacement for the AMS mail system they had used, storing email in AFS space. AMS proved not to be scalable for large numbers of users.
- The Cyrus server handles several extensions:
  - Kerberos authentication of users - extensible to other authentication schemes.
  - ACL's based on AFS ACL structures - this means that we can use AFS ACL's from the AFS pts server on mailboxes.
  - public-access mailboxes and bulletin boards.
  - Zephyr notification of new email.

Cyrus Implementation

  - Export of news to imap boxes - a news spool area can be read directly through the IMAP protocol.
  - Quota
- Some integration to IMSP - what the heck is that, I hear you cry? IMSP is a companion protocol that allows mapping of a username/mailbox pair to a correct mail server - allowing for multiple servers and a directory structure. It also allows location-independent access to support files, like personal addressbooks, mailing lists, etc. IMSP is being replaced with ACAP,
- Designed to be very scalable, but requires a separate userless machine designed to be an IMAP server. All the email is kept MH-style - one message per file, and all files belong to CYRUS. Obviously, it makes a LOT of files.
- Does POP serving as well as IMAP. KPOP as well.

IMAP1 server implementation

- The gory details are in a separate paper.
- Small Sun workstation with RAID disk for the mail storage spool - about 2 gig currently available. This is being replaced by a SPARC 20, which is also the new client news server.
- All system, etc. disks are mirrored. Any disk can be pulled out and the system will just slow down.
- IMAP service has been in test mode since April. Things have generally gone quite well. Users are making use of it, and seem happy with the service provided.
- Handling about 2200 email messages per week for 25 users. About 2 megabytes of mail per week, and about 80 megs of mail is stored.

Extensions Implemented

- Access to FNAL news groups as mailboxes.
- EXMH as a POP client.
- Exploration of MIME/http integration into email messages. Some of the clients handle this pretty well.
- Testing of zephyr for email notification - this works fine, but is of little utility until zephyr becomes productized.
- Kerberized email clients would be nice - current system requires that users log in using their AFS password - this sends their password across the wire, and is kind of silly since FNALU users at least are already authenticated.
- Tests of Sun's new UNIX and windows clients - so far seems to work.
- Tests of other UNIX and non-UNIX client software - ML, tkrat, Ishmail, for example.

Futures

- IMSP/ACAP integration and testing - this is a ways off.
- Work out support/ownership issues.
- Tests of scalability - U. of Washington has 60,000 user accounts and 250,000 messages per day - 20,000 per hour at peaks, so we're probably okay.
- Possible investigation of shared mailboxes as a replacement for VAXNEWS that uses an industry-standard exchange protocol.

Implementation Proposal

- We consider the IMAP server a supported service for POP and IMAP serving.
- We would like to encourage FNALU users to have their email spooled on the IMAP server rather than on the individual FNALU interactive machines. This will allow us better reliability (if one machine is down for service they can still read their mail.), better backups, stronger security (as the email is going to one machine that is better controlled and that has all the latest email patches installed), and allows us to put user mailboxes under quota control.
- We primarily see this as a way for UNIX users on site to keep their email in a place that can be accessed from any node and that simplifies mail delivery both for the system admins and for the users.

Decisions to be made:

- Quota policy - what is the default email quota and what needs to be done to request an increase.
- Will we support BBS-style shared mailboxes?
- Integration of news into mail to be supported?
- Should we at some point **require** all UNIX email delivery to mail servers? CERN for example now will only support email through IMAP.

Interesting stuff

- Sun announced IMAP client and server on March 26 - currently testing the beta UNIX client against the Cyrus server. Seems okay. Clients also for Windows 3.1, NT, Win95.
- Nokia has announced cellular phones with IMAP support built in - you can read your mail on your cell phone. This apparently only works in Europe and Asia at the moment. This is a cell phone with 8 meg of RAM!
- Many more clients/progress being announced.
- Industry consortium met in January and meets again in 2 weeks.

For Future Reading

- Handouts
- Project Cyrus - http://andrew2.andrew.cmu.edu/cyrus/cyrus.html
- Email Resource list - http://andrew2.andrew.cmu.edu/cyrus/email/email.html

Using Email Clients with Imap1

- POP - simply point your POP client to use imap1.fnal.gov as a POP server. It will automatically use your INBOX folder for retrieving email. You won't have access to anything else.
- Pine - you need to point your email resources at the imap server. Some examples from your setup session or .pinerc file:
  - inbox-path=:{imap1.fnal.gov:143}INBOX
  - folder-collections=server {imap1.fnal.gov:143}INBOX.[],local mail/[]
  - default-fcc={imap1.fnal.gov:143}INBOX.sent-mail
  - read-message-folder=read
- Other IMAP clients may have different syntax - but the protocol on the Cyrus server believes your tree starts at INBOX (or user.username) and has dot-separated descending folders - thus user.hanson.folderworld.junkmail.