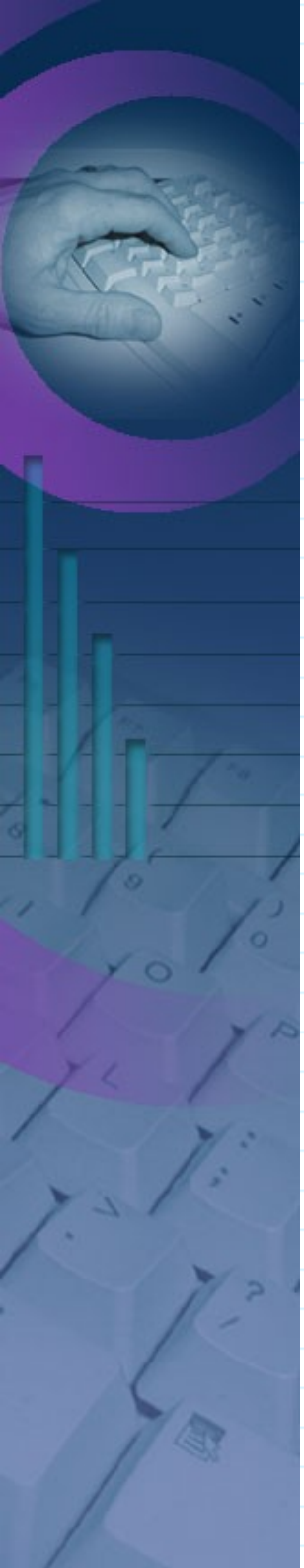
Introduction To Cyrus IMAP



Copyright

If you find this document useful or further it's distribution we would appreciate you letting us know.

Cyrus?

Cyrus IMAP is named after Cyrus the Great (559 - 530 BC), founder of the Persian kingdom, and “King of the World”.

Cyrus the great is remembered for his tolerance of the various minorities and generally benevolant leadership.

The tomb of Cyrus the Great.

Cyrus the Great also created the first modern postal system, whose motto was: “stopped by neither snow, rain, heat, or gloom of night”.

The Pony Express service was based on the Persian model, and the subsequent United States Postal Service adopted the motto from the Persian postal system.

Support

Supported Protocols

IMAP
IMAPS
POP3
POP3S
KPOP
LMTP

Server Features

- Multiple simultaneous operations on a single mailbox (concurrency)
- Per mailbox access control lists
- Storage quotas
- Private mailbox database
  - No /var/spool/mail
  - No mail stored in user home directories
- Shared mail folders
  - Any mail folder may be shared
- Bulletin board folders
- Server side e-mail filtering
- SSL and TLS
- Multiple load balanced mail partitions
- IMAP server aggregation
- Exporting UseNet groups as IMAP folders

Cyrus Users

Roanoke College	~2,700 users
North Carolina State University	~36,000 users
George Fox University	~3,000 users
NPG Cable	~15,000 mailboxes

(>150,000 messages a day on a Dual PIII Dell Poweredge 6300 with ReHat 7.1, kernel 2.4.17, and 512Mb of RAM)

College of William and Mary	~18,0000 users

(100,000 - 150,000 messages per day)

Amicus, Inc	~20,000 users

(75,000 - 100,000 messages per day, 30Gb of online mail)

Middle Tennessee State University	>20,000 users

Large Sites

Does Cyrus IMAP Scale? :)

http://www.hot.ee ("*The Hotmail of Estonia*") >300,000 users.

The commercial FastMail.FM service, which provides IMAP accounts via the Internet to customers, including the full feature set of Cyrus IMAPd. http://www.fastmail.fm./

Carnegie Mellon University

~15,000 users with ~25,000 folders
~160,000 public or shared folders
> 6000 active IMAP sessions every day.

PrimPosta/Post4CE with more than 160,000 users processing between 80,000 and 100,000 messages per day.

Cyrus is used as the IMAP provider in both Bynari's Insight server (http://www.bynari.com) and the kgroupware project (http://kroupware.kde.org).

Installation

RPMs of Cyrus SASL and Cyrus IMAP are available from:

http://home.teleport.ch/simix/

The packages should work, or the source package should be rebuildable, on any RedHat 7.x distribution. RedHat 8.x ships with both Cyrus SASL v1 and v2, RedHat 7.x ships with only Cyrus SASL v1 so you will need both the SASL and the IMAP packages.

The paths used by the RPM packages are not the standard Cyrus IMAP paths, they have been modified to conform to the FHS.

I strongly encourage the use of packaged versions of the Cyrus software. Building from source is non-trivial, and there are many configure options whose meaning is not at first apparent. This is a very complicated suite of software.

Do not use the 2.1.6, 2.1.7, or 2.1.8 versions. 2.1.9 is stable. 2.1.6, 2.1.7, 2.1.8, and 2.1.9 were all released withing 48 hours of each other. :)

Debian packages are available at debian.org

Paths

Ideally these two directory structures should exist on separate physical volumes.

/var/spool/imap

This is the default mail partition, where the content of mailboxes are stored here unles you instruct the server otherwise.

/var/lib/imap

The configuration and operational data for the server exists here; including quota information, duplicate delivery database, mail box list, and TLS session attributes.

/usr/libexec/cyrus

Most of the Cyrus IMAP related binaries and scripts live here; including cyradm, deliver, reconstruct, etc...

The above paths are the defualt for the RPM install, compliant with FHS. They are different than the “standard” paths of a Cyrus IMAPd installation.

The cyrus user

The Cyrus IMAP server and its related processors run as the user “cyrus”, a member of group “mail”.

In order to gain administrative access to the server via the “cyrus” account a SASL2 password entry must be created on the server; regardless of the authentication method used for mailbox access.

```
$ saslpasswd2 -c cyrus
Password:  ******
Again (for verification): ******
```

Services & /etc/cyrus.conf

The services offered by Cyrus IMAP are controlled via the configuration file cyrus.conf. You shouldn't need to modify this file unless you want to disable some services or adjust prefork options.

- The prefork directive determines the minimum number of threads of that service will be running at all times.

Installation will add the following services if they don't already exist:

pop3	110/tcp
imap	143/tcp
imsp	406/tcp
acap	674/tcp
imaps	993/tcp
pop3s	995/tcp
kpop	1109/tcp
sieve	2000/tcp
lmtp	2003/tcp
fud	4201/udp

```
SERVICES {
  # add or remove based on preferences
  imap       cmd="imapd" listen="imap" prefork=1
  imaps      cmd="imapd -s" listen="imaps" prefork=3
  pop3       cmd="pop3d" listen="pop3" prefork=0
  pop3s      cmd="pop3d -s" listen="pop3s" prefork=0
  sieve      cmd="timsieved" listen="sieve" prefork=0
  lmtpunix cmd="lmtpd" listen="/var/lib/imap/socket/lmtp" prefork=2
}
```

/etc/imapd.conf

The primary configuration file for controlling the specific behaviors of Cyrus IMAP is /etc/imapd.conf.

- There are ~80 directives available.
- "man imapd.conf"

```
configdirectory: /var/lib/imap
partition-default: /var/spool/imap
admins: cyrus
allowanonymouslogin: no
sieveuserhomedir: no
sievedir: /var/lib/imap/sieve
sendmail: /usr/sbin/sendmail
hashimapspool: true
sasl_pwcheck_method: saslauthd
sasl_mech_list: PLAIN
tls_cert_file: /usr/share/ssl/certs/cyrus-imapd.pem
tls_key_file: /usr/share/ssl/certs/cyrus-imapd.pem
```

Where to store master indexes

Default mail store

Administrative accounts

No anonymous users

How to authenticate users

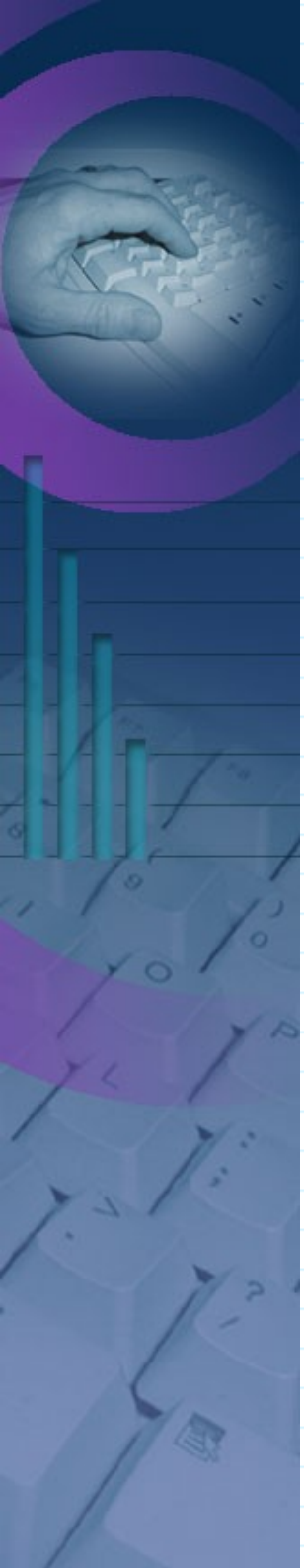
Administrative Shell

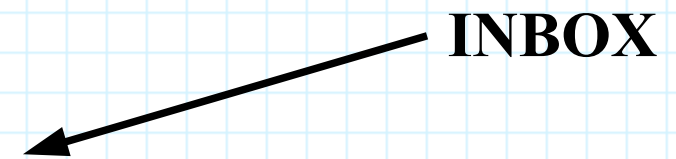
cyradm[--user user]
[--[no]rc]
[--systemrc file]
[--userrc file]
[--port n]
[--auth mechanism]
[--server] server

The cyradm shell is used to create and delete mailboxes, change access control lists, and perform most administrative tasks.

cm	Create mailbox
dam	Delete ACL on mailbox
dm	Delete mailbox
exit	Quit cyradm
lam	List ACLs on mailbox
lm	List mail boxes
lq	List quota information
lqm	List quota on mailbox
lqr	List quota on root
renm	Rename mailbox
sam	Set ACL on mailbox
sq	Set quota

The mailbox hierarchy





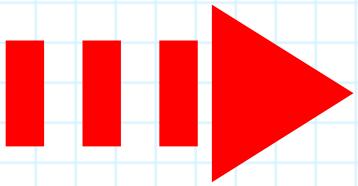
user.adam
user.adam.klug
user.adam.subsys.mail.mta.sendmail
user.adam.subsys.mail.mua.pine
user.adam.subsys.mail.mua.evolution
user.adam.subsys.mail.mua.outllook
user.adam.subsys.mail.mua.eudora
user.adam.subsys.gnome
user.adam.subsys.ldap.dsa.openldap
user.adam.subsys.ldap.clients.nss
user.adam.subsys.ldap.clients.pam
user.adam.subsys.ldap.clients.evolution
user.adam.subsys.ldap.clients.php
user.adam.subsys.rdbms.informix
user.adam.subsys.rdbms.postgresql

- Mailboxes exist in a hierarchy.
  - Levels of the hierarchy are seperated by the “.” character.
  - User names and folder names should not contain a dot.
  - User folders appear beneath the INBOX of the user.
  - Access is determined by ACLs, not the location in the hierarchy.
    - A user can share a folder from their hierarchy, and other users can subscribe.

altnamespace: yes

The hierarchical nature of the mailbox system can be confusing to both users and some client e-mail packages. The altnamespace feature causes Cyrus IMAPd to present the immediate children of the INBOX as present in the same hierarchival level as INBOX. This only effects how the clients percieve the structure, the mailbox hierarchy remains intact for all other purposes and operations (administration, ACL inheritance, quotas, etc...)

INBOX
INBOX.klug
INBOX.subsys.mail.mta.sendmail
INBOX.subsys.mail.mua.pine
INBOX.subsys.mail.mua.evolution
INBOX.subsys.mail.mua.outllook
INBOX.subsys.mail.mua.eudora
INBOX.subsys.gnome



INBOX
klug
subsys.mail.mta.sendmail
subsys.mail.mua.pine
subsys.mail.mua.evolution
subsys.mail.mua.outllook
subsys.mail.mua.eudora
subsys.gnome

ACL Attributes

- Each ACL has two elements:
  - Who (identifier)
  - What privileges
- An ACL can be either positive, the default, which grants rights, or negative which revokes rights.

ACLs are bound to identifiers

user
group:*group*
anonymous
anyone

"Anonymous" is a non-authenticated connection.
"Anyone" is any authenticated connection.

l	Visible
r	Read (View messages, perform searches)
s	Seen and recent flags preserved
w	Write (Modify flags other than Seen and Deleted)
i	Insert (May copy messages into the mail-box)
p	Post (May submit messages to the mailbox)
c	Create (May create new subfolders as well as delete and rename the mailbox)
d	Delete (May mark messages as deleted and perform expunges)
a	Administrator (May change the ACL)

ACL Shortcuts

none	-
read	lrs
post	lrsp
append	lrsip
write	lrswipcd
all	lrswipcda

Default and Inherited Access.

- A new user.* folder has all privileges granted to the user.
  - Users always have visible (“l”) and administrator (“a”) privileges to all folders beneath their INBOX.
- Users specified by the admins directive in imapd.conf have visible (“l”) and administrator (“a”) privileges on all folders.
- A folder created beneath another folder in a hierarchy inherits the ACL of its parent.
  - The ACL can then be modified.
- A folder which has no parent (top of a hierarchy) initially has the privileges specified by the defaultacl directive in the imapd.conf configuration file.

Calculating Access

Since a mail folder can have any number of ACLs, it is important to keep track of how ACLs interact with each other. Cyrus IMAP calculates a user's permissions in the following fashion:

- Start with the rights granted to "anyone".
- Add all the rights specifically granted to the user.
- Add all the rights specifically granted to groups to which the user belongs.
- Remove the sum of the rights specifically revoked to the user and any groups to which the user belongs.

NOTE: The IMAP administrative account "cyrus" does NOT have the automatic access to delete mail folders. By default only the owner of a folder has the permission required to delete the folder. To remove folders the cyrus administrator must first grant itself the appropriate access: "c".

Quotas

- A quota is a limit on the size (the accumulation of messages) of a folder hierarchy.
  - Indexes and other overhead are not included in the quota.
  - No folder may be under the domain of more than one quota.
    - A quota descends the folder hierarchy until another quota is encountered.
      - A folder is under the domain of the nearest quota.

user.adam
user.adam.klug
user.adam.personal
user.adam.vendorfiles
user.adam.subsys
user.adam.subsys.mail.mta.sendmail
user.adam.subsys.mail.mua.pine
user.adam.subsys.mail.mua.evolution
user.adam.subsys.mail.mua.eudora
user.adam.subsys.gnome

If a quota is created on user.adam and user.adam.subsys then user.adam.subsys.* are not included in the quota on user.adam, but user.adam.klug, user.adam.personal, and user.adam.vendorfiles would be.

Quota Details

- So long as a folder has not exceeded quota it may receive a message of any size.
  - If a folder has a quota of 100Mb, and contents 96Mb of messages, a 7Mb message could be successfully delivered. The mailbox would then be over-quota and refuse further messages.
- If a folder has exceeded quota and the MTA attempts to deliver a message, a temporary error is generated. The MTA should periodically attempt redelivery for a few days.
- Cyrus will automatically issue quota warnings to users who access folders near or exceeding their quotas.
  - Not all mail clients support quota warnings.
  - Cyrus only dispatches quota warnings to users with *delete* ("d") privileges on the relevant folder(s).
    - Users with delete privileges are the only ones who can do anything to reduce the storage in use (delete and expunge messages).

Bulletin Boards

Bulletin boards are shared folders that can receive messages via an e-mail address, and typically do not belong to a user.

Creating a Bulletin Board:

1. You may optionally set a postuser in the /etc/imapd.conf configuration file.

```
postuser shared
```

If you set the postuser as listed above address of the board will be shared+foldername@..., if no postuser is set the address of the board will be +foldername@....

2. Create the folder that will operate as the bulletin board.

```
cm service
```

3. Grant post access to anonymous on the folder, and additional access to those who should be able to view the board:

```
sam service anonymous p
sam service group:service lrwids
```

Sendmail tip: If you have multiple SMTP servers you may need to set LUSER_RELAY to your Cyrus host(s) in order to utilize bulletin board posting.

What is Sieve

Sieve is a standards-track mail filtering language defined in RFC3028. Sieve is designed to be platform independent and specifically serve the purpose of processing RFC822 compliant messages in a safe and secure fashion.

Why sieve?

- Client side filtering requires that the client connects to the server, and the script is tied to the client platform.
  - How is a users messages filtered and sorted while he or she is on vacation?
- Other server side scripting methods (primarily procmail) have severe disadvantages.
  - Require the forking of processes.
  - No standard way for users to manage scripts without direct (shell) access to the server.
  - Security is not a primary focus.

Sieve example

There are several web front ends that permit users to install, edit, and remove their own sieve scripts.

```
require "fileinto";
require "reject";
if header :contains "Subject" "[CIRCLE]" {
        fileinto "INBOX.Circle List";
}
if header :contains "Subject" "A great Shockwave flash movie" {
        reject "Possible virus?  Check your system!";
}
if header :contains "Received" ".da.uu.net" {
        fileinto "INBOX.Junk";
}
if header :contains "To:" "@bigfoot.com" {
        reject "Yeah, right.  Bugoff, hosier!";
}
if anyof ( not address :all :contains ["To", "Cc", "Bcc"] "yourname@youraddress.com" )
{
        fileinto "INBOX.Junk";
}
```

Authentication

- Cyrus IMAP uses Cyrus SASL version 2 to process all authentication.
- Cyrus SASL2 supports the following authentication mechanisms:
  - Plain text
  - PAM
  - Kerberos IV
  - Files (/etc/passwd, /etc/shadow)
  - Kerberos V (GSSAPI)
  - Shared Secrets (sasldb2)
    - CRAM-MD5
    - DIGEST-MD5
- If allowanonymouslogin is present in imapd.conf the server will permit the user anonymous to log in with any password.

PAM & saslauthd

If you wish to authenticate users using a non-SASL method, such as PAM, the saslauthd daemon provides a bridge between Cyrus and your external authentication method (called a mechanism).

Specify the appropriate lines in /etc/imapd.conf:

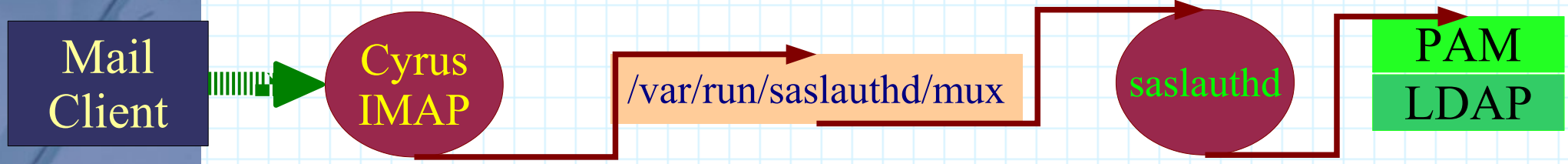
```
sasl_pwcheck_method: saslauthd
sasl_mech_list: PLAIN
```

and start the saslauthd daemon:

```
$ chkconfig saslauthd on
$ service saslauthd start
```

*Users will be authenticated using the PAM "imap" stack.

The file /etc/sysconfig/saslauthd controls the exact behavior of the saslauthd process
MECH sets how passwords are verified, the typical value is "pam".
FLAGS is additional parameters, typically specifing the number of threads with the -n option, i.e. "-n 5"



The PAM Stacks

The PAM stack for authenticating users only requires auth and account designates. Keeping this stack as simple as possible simplifies debugging authentication problems.

```
[adam@lnx01688 pam.d] $ cat imap
auth          required    /lib/security/pam_ldap.so
account       required    /lib/security/pam_ldap.so
```

Unless mail boxes are created for "system" accounts such as root, bin, ldap, etc... within Cyrus there is no reason to include modules cush as pam_unix in the IMAP PAM stack.*

If you provide pop and sieve services you will need PAM stacks appropriate to those services as well.

*Unless of course you still use these to authenticate system users.

Testing Authentication

```
lnx01688:~ $ imtest -m login -v -u adam lnx01688
S: * OK lnx01688.morrison.iserv.net Cyrus IMAP4 v2.1.5 server ready
C: C01 CAPABILITY
S: * CAPABILITY IMAP4 IMAP4rev1 ACL QUOTA LITERAL+
MAILBOX-REFERRALS NAMESPACE UIDPLUS ID
NO_ATOMIC_RENAME UNSELECT CHILDREN MULTIAPPEND SORT
THREAD=ORDEREDSUBJECT THREAD=REFERENCES IDLE
STARTTLS LISTEXT LIST-SUBSCRIBED ANNOTATEMORE X-
NETSCAPE
S: C01 OK Completed
Please enter your password: ********
C: L01 LOGIN adam {9}
S: + go ahead
C: <omitted>
S: L01 OK User logged in
Authenticated.
Security strength factor: 0
C: Q01 LOGOUT
Connection closed.
```

LMTP
(Local Mail Transfer Protocol)

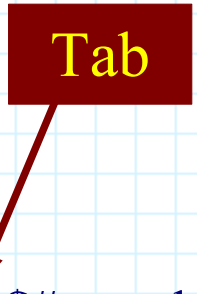
- LMTP is a derivative protocol from SMTP and is used to transfer messages from the MDA and the mail store.
- The use of LMTP provides several advantages:
  - If a message is sent to multiple users, only a single copy of the message is stored per mail partition. This is transparent to end users.
    - This functionality is controlled by the singleinstancestore directive in imapd.conf, valid values are yes or no.
  - Supression of duplicate messages
    - If a message with the same message-id and envelope receipient has been recieved within the last three days, the message will be silently discarded.
      - This functionality is controlled via the duplicatesuppression directive in imapd.conf, valid values are yes or no.
  - LMTP supports authenticated final delivery.

sendmail using deliver

Older versions of the sendmail-cf package (Sendmail M4 Configuration) are only able to generate a configuration file that uses the Cyrus deliver program via the cyrus mailer. If your sendmail packages support the cyrusv2 mailer (which uses LMTP), use that method as described on the next slide. If the file /usr/share/sendmail-cf/mailer/cyrusv2.m4 exists then your packages support delivery via LMTP.

```
define(`confBIND_OPTS',`-DNSRCH -DEFNAMES')dnl
define(`confLOCAL_MAILER', `cyrus')dnl
MAILER(`local')dnl
MAILER(`smtp')dnl
MAILER(`cyrus')dnl

LOCAL_RULE_0
Rbb + $+ < @ $=w . >    $#cyrusbb $: $1
```



You must define a LOCAL_RULE_0 if you wish to use the bulletin board support.

sendmail using LMTP

Due to the fact that the RPM packages are FHS compliant and do not use the standard paths, you must edit /usr/share/sendmail-cf/mailer/cyrusv2.m4 changing the line

```
ifdef(`CYRUSV2_MAILER_ARGS',, `define(`CYRUSV2_MAILER_ARGS',
`FILE /var/imap/socket/lmtp')')
```

to read

```
ifdef(`CYRUSV2_MAILER_ARGS',, `define(`CYRUSV2_MAILER_ARGS',
`FILE /var/lib/imap/socket/lmtp ')')
```

Then include

```
MAILER(cyrusv2)dnl
define(`confLOCAL_MAILER', `cyrusv2')dnl
```

in your m4 file. These lines must occur **after** the MAILER statements for local and procmail.

Preserve local plus detail

If you include FEATURE(`preserve_local_plus_detail')dnl in your sendmail m4 file, the generated configuration will pass e-mail addresses containing the "+" symbol to the LMTP socket without alteration. This permits *subordinate posting*.

Example:

awilliam@whitemice.org is delivered to user.awilliam
awilliam+presentations@whitemice.org is delvered to
    user.awilliam.presentations
awilliam+presentations.cyrus@whitemice.org is delivered to
    user.awilliam.presentations.cyrus

In order to recieve a message via this operation the submitter ("anonymous" if via unathenticated SMTP) must have post ("p") privilages to the mailbox.

If the sender does not have post privilages or the mail box does not exist the message is delivered to the INBOX as normal.

LUSER_RELAY

In sendmail parlance a luser is an address that appears to be local, but for which no account exists. lusers are typically users that have only e-mail access to a system, with no corresponding system privilages.

By default sendmail will bounce messages as "User Unknown" if a system account does not exists for a user. The m4 LUSER_RELAY directive permits this behaviour to be modified.

Example:

```
define(`LUSER_RELAY',`cyrusv2:localhost')dnl
```

The above will relay all luser messages to the cyrusv2 mailer, where they will either be delivered or generate and error.

```
define(`LUSER_RELAY',`cyrus-1.morrison-ind.com')dnl
```

The above will relay the message to the host cyrus-1.morrison-ind.com, where delivery will be attempted based upon the rules at that server.

postfix

To configure the postfix MTA to use LMTP add the following to /etc/postfix/main.cf -

```
mailbox_transport = lmtp:unix:/var/lib/imap/socket/lmtp
```

And ensure the following line exists in /etc/postfix/master.cf

```
lmtp    unix  -       -       n       -       -       lmtp
```

This assumes you are not running services in a change-root configuration.

The contents of this slide provided by Simon Matter

/etc/cyrus.conf: START

The START stanza of /etc/cyrus.conf is processed when the Cyrus IMAP service is first started.

ctl_cyrusdb is the Cyrus utility for validating and checkpoint the database.
-r = DatabaseRrecovery.

```
START {
  # do not delete this entry!
  recover       cmd="ctl_cyrusdb -r"

  # this is only necessary if using idled for IMAP IDLE
  #  idled                cmd="idled"
}
```

"Database Recovery" checks that the contents of the databse is valid, in case the last service stoppage was abnormal.

/etc/cyrus.conf: SERVICES

```
SERVICES {
  # add or remove based on preferences
  imap          cmd="imapd" listen="imap" prefork=5
  imaps         cmd="imapd -s" listen="imaps" prefork=1
  pop3          cmd="pop3d" listen="pop3" prefork=3
  pop3s         cmd="pop3d -s" listen="pop3s" prefork=1
  sieve         cmd="timsieved" listen="sieve" prefork=0

  # at least one LMTP is required for delivery
#  lmtp         cmd="lmtpd" listen="lmtp" prefork=0
  lmtpunix      cmd="lmtpd" listen="/var/lib/imap/socket/lmtp" prefork=1
}
```

/etc/cyrus.conf: EVENTS

The EVENTS stanza schedules various maintenance events.

```
EVENTS {
  # this is required
  checkpoint    cmd="ctl_cyrusdb -c" period=30

  # this is only necessary if using duplicate delivery
suppression
  delprune      cmd="ctl_deliver -E 3" period=1440

  # this is only necessary if caching TLS sessions
  tlsprune      cmd="tls_prune
}
```

Check point the database: make sure temporary files and old log files are removed, update hot backup, etc...

Run every ***n*** minutes.

Maintain the duplicate delivery database. -E *{days}* controls how far back in the delivery database to look for duplicates.

Makes sure the list of TLS sessions is clean, purges expired entiries.

ctl_mboxlist

The ctl_mboxlist allows you to dump the list of all the system mailboxes, and associated access control lists, to standard output.

ctl_mboxlist can also be used to restore the mail box list and access control rules.

```
$ cd /usr/libexec/cyrus
$ ./ctl_mboxlist -d
user.awilliam   default awilliam       lrswipcda
user.mwilliam   default mwilliam       lrswipcda
```

Only the user “cyrus” may use the ctl_mboxlist utility.

Performance

Cyrus IMAPd is a very I/O intensive application, beyond ensuring ample network bandwidth, etc... the following optimizations should be applied.

1. Place /var/spool/imap and /var/lib/imap on seperate physical volumes, each in its own filesystem.
2. Set "noatime" on both /var/spool/imap and /var/lib/imap, Cyrus does not utilize the last accessed meta-data.
3. Set the number of saslauthd threads to 1.5 times the number of connections received per second during a busy period.
4. Make sure sufficient file descriptors are available by adjusting fs.file-max via sysctl.

Any fairly recent CPU should have sufficient capacity to support a minimum of ~200 users. But if SSL/TLS is used extensively this will significantly effect CPU utilization and require additional resources.

emory Consumption	
saslauthd	8Mb @ 10thrds.
imapd	2.2Mb+.5Mb/ea
cyrus-master	5Mb
sendmail	3Mb+2.5Mb/ea

Corruption

On rare occasion a user's mailbox may become corrupted, causing any of a variety of errors or problems when they attempt to read their mail. Often they will be able to open their mailbox but not see any messages.

A good indication that a mailbox is corrupt is the presence of a core file in the mailbox directory.

```
[root@sardine /root]# ls -l /var/spool/imap/m/user/mdoozan/core
-rw-------    ...  12873728 Aug 13 09:25 /var/spool/imap/m/user/mdoozan/core
```

You may also see messages such as

```
master[910]: process 31003 exited, signaled to death by 11
```

in the system logs.

Chronic corruption most likely indicates a hardware or Operating System problem.

Reconstruct

```
[root@sardine /root]# ls -l /var/spool/imap/m/user/mdoozan/core
-rw-------  12873728 Aug 13 09:25 /var/spool/imap/m/user/mdoozan/core
[root@sardine mdoozan]# su - cyrus
bash-2.04$ /usr/libexec/cyrus/reconstruct  -r -f user.mdoozan
user.mdoozan
user.mdoozan.A. J. MORRISON
user.mdoozan.BOHINSKI
user.mdoozan.BRIGHTON_040TIRES
user.mdoozan.CARL_040BARNOSKY
user.mdoozan.CRUZ
...
user.mdoozan.mail
user.mdoozan.sent-mail
user.mdoozan.tmp
user.mdoozan.trash
bash-2.04$ ^D
```

The reconstruct command can be used to repair almost any kind of mailbox corruption.

reconstruct *{options}* *{mailbox}*

reconstruct options

- -C *{configuration file}*
  - Use alternate configuration file.
- -r
  - Reconstruct all subordinate mailboxes as well.
- -f
  - Examine filesystem to add any subordinate folders that are not listed in the mailbox index.

ACAP

ACAP is the Application Configuration Access Protocol, defined in RFC2244. The purpose of ACAP is to provide a remote store of application configuration information, preferences, address books, and other information useful to mobile users.

An ACAP daemin is available from the Cyrus software download page: http://asg.web.cmu.edu/cyrus/download/

ACAP is supported by:

Simeon 3.0.2
UW Pine
Mulberry
Eudora

Ximian intends to integrate ACAP into gconfd, making it available to all applications (no time line however).

Whether ACAP will become a widely supported protocol remains to be seen.

Migrating From UW

Probably the most formidable step of installing Cyrus IMAP is migration from your previous IMAP package. On Linux, this is typically the University of Washington IMAP server, or UW-IMAP.

This process is well documented and utilities exsist to make it fairly painless.

The Cyrus IMAP RPMs provide the following tools:

imapcreate - A bulk folder creation utility.
reconstruct - A folder re-indexing utility (required by batchreconstruct).

You must acquire the following migration scripts:

inboxfer (and cpmsg required by inboxfer and folderxfer)
bsd2cyrus
folderxfer
batchreconstruct

Available at - http://www.kalamazoolinux.org/projects/awilliam/

You will also need /usr/bin/formail provided by the procmail package.

Migrating From UW

1. Create a text file containing the list of users to be migrated - one per line.
2. Create an inbox for each user using the imapcreate utility.

```
# cat users.text | /usr/libexec/cyrus/imapcreate  -u cyrus -p cyc8ys sardine
Creating user.ksikkema on default
```

3. Transfer contents of existing UW inbox (/var/spool/mail) into the Cyrus inbox using the inboxfer utility.

```
# /usr/libexec/cyrus/inboxfer users.text
```

4. Generate a list of mail folders that need to be created and migrated for each user (using the bsd2cyrus utility)

```
# /usr/libexec/cyrus/bsd2cyrus users.text > mailboxes.txt
```

The contents of this file looks like -

```
. . .
ksikkema:user.ksikkema.sent-mail:/home/ksikkema/mail/sent-mail
ksikkema:user.ksikkema.friends:/home/ksikkema/mail/friends
ksikkema:user.ksikkema.internal:/home/ksikkema/mail/internal
ksikkema:user.ksikkema.steelcase:/home/ksikkema/mail/steelcase
ksikkema:user.ksikkema.drafts:/home/ksikkema/mail/drafts
. . .
```

Migrating From UW

5. Create Cyrus folders to receive e-mail from each migrated folder, again using the imapcreate utility.

```
# cat mailboxes.txt | cut -d":" -f2 | sed "s/^user\.//g" |
/usr/libexec/cyrus/imapcreate  -u cyrus -p cyc8ys sardine
Creating user.ksikkema.sent-mail on default
Creating user.ksikkema.Hanson_040Cold_040Storage on default
Creating user.ksikkema.Total_040Logistic_040Control on default
Creating user.ksikkema.friends on default
Creating user.ksikkema.internal on default
. . . .
```

6. Migrate mailfrom user UW folders into Cyrus using the folderxfer utility.

```
# /usr/libexec/cyrus/folderxfer mailboxes.text
/bin/cat "/home/ksikkema/mail/sent-mail" . . . .
```

This step may take some time.

7. Rebuild indexes, etc.., for each new Cyrus user using the batchreconstruct utility.

```
# su - cyrus -c "/usr/libexec/cyrus/batchreconstruct /tmp/users.text"
```

Your done, welcome to Cyrus IMAPd.

More Information

Managing IMAP by Dianna Mullet & Kevin Mullet
O'Reilly & Associates, Inc.
ISBN 0-596-00012-X