

Dovecot IMAP Server

<http://www.dovecot.org/>

Date: September, 2009

experience fanatical support[®]
WWW.RACKSPACE.COM

rackspace[®]
HOSTING 

Rackspace

- Rackspace Email uses Dovecot to serve IMAP for over a million paid mailboxes
 - MS Exchange also available
- Rackspace has sponsored Dovecot development for years
 - And employed me full time for year 2009

Overview

- Dovecot history & how Apple uses it
- Dovecot features
- IMAP & Dovecot performance
- Troubleshooting
- Future features

What is Dovecot?

- Dovecot is
 - IMAP server
 - POP3 server
 - Local mail delivery agent with Sieve filtering
 - Managesieve server
 - LMTP server (v2.0+)
- Dovecot is NOT
 - SMTP server
 - So it neither receives nor sends mails directly

Dovecot



Pictures from Wikipedia, by *Cyril Thomas* and *Carcharoth*

History

- Dovecot design was started around June 2002
- First release was July 2002
- Late 2003 a redesign started
- v1.0.0 released April 13th 2007
- v1.1.0 released June 21st 2008
- v1.2.0 released July 1st 2009
- v2.0 betas hopefully this year

Why Did Apple Switch to Dovecot from Cyrus?

-

Apple's Dovecot

- My test version: WWDC developers preview
- dovecot --version says: 1.1.14apple0.5
- Apple patches:
 - Open Directory authentication
 - Multiple connections per mail process
- Configuration in `/etc/dovecot/dovecot.conf`
 - Server admin changes some settings
 - Defaults more towards performance than security

Server Admin: sirkku.local: Mail

SERVERS

- Available Servers (0)
- sirkku.local
 - Mail
 - Open Directory

Overview Logs Connections Maintenance Settings

User	From	Connection Length	Type	Connection Count
tss	127.0.0.1	00:01:16	IMAP	2

2 IMAP connections and 0 POP connections (2 total)

+ . ⚙️ ↻ Stop Mail

Server Admin: sirkku.local: Mail

SERVERS

- Available Servers (0)
- sirkku.local
 - Mail
 - Open Directory

Overview Logs Connections Maintenance Settings

General Relay Filters Quotas Mailing Lists Logging Advanced

Domain name:
 Enter the local Internet domain name.

Host name:
 Enter the Internet host name of this mail system.

Push Notification Server: None

Enable SMTP

- Allow incoming mail
- Hold outgoing mail
- Relay outgoing mail through host:
 - Authenticate to relay with user name:
 - Password:
- Copy undeliverable mail to:
- Copy all mail to:

Enable IMAP with maximum of connections

Enable POP

Deliver to "/var/mail" when IMAP & POP are disabled

+ . ⚙️ ↻ Stop Mail

Features

- Often has better performance than competition.
 - Optimized for minimizing disk I/O (index/cache files)
 - Hosting my own mails on 10 years old Sparc helps
- Highly configurable for different environments
 - Standard mbox and Maildir with **transparent** indexing (external mailbox modifications are ok)
 - mbox: Dovecot's high-performance mailbox format
 - Many different ways of clustering
 - Extremely flexible authentication
 - Postfix and Exim support Dovecot for SMTP AUTH

Features

- Admin-friendly / self-healing
 - All errors are logged
 - Understandable error messages
 - Improved constantly (to reduce my email load)
 - Detected (index) corruption gets fixed automatically
- `file_dotlock_create(/home/timo/Maildir/dovecot-uidlist)` failed: Permission denied (euid=1000(timo) egid=1000(timo) missing +x perm: /home/timo)
- `chown(/home/timo/Maildir/.box, -1, 0(root))` failed: Operation not permitted (egid=1000(timo), group based on /home/timo/Maildir)

Authentication

- Password and user database separation
 - Passdb for verifying user's password
 - Userdb for looking up how to access mailbox
- Support for almost everything: SQL, LDAP, PAM, checkpassword scripts, etc.
 - Everything is configurable (e.g. full SQL queries)
 - Supports multiple dbs (e.g. system + virtual users)
- Auth mechanisms: PLAIN, CRAM-MD5, DIGEST-MD5, Kerberos, OTP, etc.
- Password schemes: Plaintext, CRYPT, MD5, SHA1, SHA256, SSHA, SSHA256, etc.

Authentication Cache

- Passdb and userdb lookups can be cached
- Password changes are automatically detected:
If auth is unsuccessful, and previous auth was
 - a) successful: do uncached passdb lookup
 - b) unsuccessful: fail login
- Negative caching can be disabled
 - User doesn't exist caching
 - Password failures (v1.2+)
- Avoids a need for imapproxy with webmails?

Maildir

- Apple: `/var/spool/mail/dovecot/<user-id>/`
- Maildir basics:
 - One file = one mail
 - Filename globally unique
 - Message flags stored in filename
- `1250461029.M8247P5745.host,W=1279,S=1243:2,S`
 - W = Virtual message size (CRLF linefeeds)
 - S = Physical message size (exactly the same as in disk) – for speeding up quota recalculation
 - :2, just means “version 2” and flags follow the comma. S = Seen
- Messages must never change!

Maildir++ Directory Layout

- **Maildir/** - INBOX
 - cur/, new/, tmp/
- **Maildir/.foo/** – folder called "foo"
 - cur/, new/, tmp/
- **Maildir/.foo.bar/** – foo's child folder "bar"
 - cur/, new/, tmp/
- '.' begins all folder directory names and separates hierarchies

Maildir Directories

- Saving messages:
 - first mail is written to **tmp/**
 - Once in a while old files (from crashes, etc.) are deleted
 - mail is moved to **new/** to finish saving
- Dovecot looks for mails in **new/** and moves to **cur/**
 - Scanning **new/** is faster than scanning **cur/**
 - So **cur/** will eventually contain all messages

Dovecot Files

- **dovecot-uidlist** maps filenames to IMAP UIDs
- **dovecot-keywords** maps a..z flags in filenames to IMAP keywords (aka. custom flags, labels)
- **subscriptions** tracks IMAP subscriptions

No state is lost if deleted:

- **dovecot-uidvalidity*** - for generating unique IMAP UIDVALIDITY values
- **dovecot.index*** - Index files
- **maildirsize** – Tracks quota usage

IMAP Protocol

- Base protocol is complex – difficult to implement it correctly (both client & server)
- Flexible – many different ways to implement a client (online & offline clients)
- Extensible – there are a lot of extensions
 - Clients rarely support more than some basic extensions, such as IDLE.
 - Thunderbird v3 adds support for several new extensions, such as CONDSTORE.

ImapTest IMAP Server Tester

- Written originally for Dovecot stress testing
 - Found a lot of crashes, hangs and mailbox corruption on other IMAP servers as well
- Tests IMAP server compliance with scripted tests and dynamic random stress testing.
- Dovecot is currently the only IMAP server that fully passes all of ImapTest tests.
 - Panda IMAP is practically there too
- Most other servers fail in many different ways.
- <http://imapwiki.org/ImapTest>

Offline IMAP Clients

- Typically download newly seen messages' bodies once and cache them locally
- Often can be configured to download immediately vs. download when reading
- Some use server side searches (Thunderbird) and some don't (Outlook – if some messages haven't been downloaded, those aren't searched)
- Usually also fetch messages' metadata once (headers, received date)
- Server-side caching may help, but not that much
 - It's extra disk I/O -> more likely just hurts

Online IMAP Clients

- Webmails often keep asking for the same information over and over and over again
- Pine and some webmails cache what they've already seen, but not permanently
- Mutt (without local cache) and some others fetch all messages' metadata every time when opening a mailbox
- Caching is very useful, but different clients want different metadata

IMAP Server Performance

- Difficult to benchmark
- Depends a lot on clients: Whether clients use a local cache makes a huge difference.
 - Online vs. offline clients
- What data to index/cache?
- SPECmail2009 adds support for IMAP
 - Emulates different IMAP clients. Client amounts are configurable.
 - The only benchmark giving realistic results.
 - Published results all run on different hardware -> results unusable for comparing software

Dovecot Cache File

- **dovecot.index.cache** files
- The main reason for Dovecot's good performance
- Dynamic: caches only what clients want.
 - Specific message headers (From:, Subject:, etc),
 - MIME structure information,
 - Sent / received date, etc.
- Caching decisions for each field: "no", "temporary", "permanent"
- Unused fields dropped after a month.
- Cached data never changes (IMAP guarantees)
- Cache file gets "compressed" once in a while
- Often about 10-20% of mailbox size

Dovecot Index Files

- **dovecot.index** contains messages' metadata
 - IMAP Unique ID number (**UID**) identifies messages
 - Flags (\Seen, \Answered, keywords, etc.)
 - Extension data: mbox file offsets, cache file offsets, modseq number (v1.2 CONDSTORE), etc.
- Lazily created/updated since v1.1
 - **dovecot.index.log** has all the latest changes.
dovecot.index is updated after 8 kB of new data has been written to the .log

Dovecot Index Files

- **dovecot.index.log** is a mailbox transaction log
 - Somewhat similar to databases' transaction logs or filesystem journals.
 - Contains all changes to be done to **dovecot.index**.
- **dovecot.index** is read to memory once and then updated from **dovecot.index.log**
 - Very efficient with NFS / clustered filesystems!
 - Very efficient to find out what changes another session had done!

Plugins

- Dovecot plugins can hook into almost anything and modify Dovecot's behavior. Some existing features implemented as plugins:
 - Access Control Lists
 - Quota
 - Full text search indexes
 - Reading compressed mbox/maildir files
- Can add new IMAP commands
- Implement new mail storage backends (virtual, SQL, IMAP proxying)

Dovecot Clustering

- Two different ways to do it:
- Globally shared filesystem
 - Many IMAP servers, each able to handle any user
 - NFS, cluster filesystems
- Sharding
 - Each user's data in different servers
 - maybe mirrored to 2-3 servers
 - IMAP proxy forwards users to correct server(s)

Apple Clustering

- I've only googled this information..
- Xsan, cluster filesystem
- Multiple mail servers connected to Xsan
 - Active-active setup
 - Load balancing with hardware, DNS, ..?
 - Performance probably best if user usually redirected to the same server
 - Or if not user, at least the same IP

Troubleshooting

- Logs! Dovecot logs all errors!
- top
- rawlog
- dtruss

Dovecot Processes

- Something's slow? Isolate it to a specific process first, then use e.g. dtruss:
- **dovecot** – master process, creates all other processes, all logging goes through it
- **dovecot-auth** – OD lookups
- **imap-login, pop3-login** – accepts new connections, handles commands until successful login, SSL proxying even after login
- **imap, pop3** – post-login handling

Client Troubleshooting

- Look at the IMAP/POP3 protocol traffic between Dovecot and client
 - Dovecot's **rawlog** tool
 - works also with SSL connections
 - Some other network sniffer such as Wireshark
- `imap/pop3_client_workarounds` settings not enabled in Apple's default config (?)

v1.2 New Features

- Virtual mailboxes (search views)
 - “All unread emails in all mailboxes”
 - All messages in all mailboxes (except Trash)
 - Virtual POP3 INBOX
 - For searching messages from all mailboxes
 - gmail-like conversation views
- Users can share mailboxes to each others
 - IMAP ACL commands
- New IMAP extensions, performance improvements

Dovecot v2.0

- Some new features already implemented:
 - Redesigned master process
 - Easy to add external services, e.g. ManageSieve
 - Redesigned configuration
 - Local/remote IP/mask -specific configuration
 - SSL certs
 - Allow changing config data source (e.g. SQL?)
 - LMTP server and proxy
 - dsync: Reliably and efficiently sync two mailboxes (e.g. via SSH)
 - mbox – high performance mailbox format

Dovecot v2.x

- Features not yet implemented, but hopefully will be by the end of this year:
 - Index file improvements
 - No locking (with atomic appends)
 - Small checksums all around for detecting corruption
 - In general make the code simpler and more robust
 - Multi-master replication
 - dbox cloud storage (for some existing cloud API(s)?)
 - Index sharing/replication between servers

Questions?