Survey of software architecture V0.0

Name of system: Sendmail
Reviewer: Oleksii Kononenko
Date: 09/29/2011
Author of software: Eric Allman
Author of book chapter: Eric Allman
Five star rating of book chapter: ***** Very valuable. Easy to follow
Purpose of system:

Sendmail has a very long history, and at a first time it was just a part-time unpaid project with no big goals for future, the only thing it did have to do is forwarding emails in easy and clear way between completely different networks. But as time goes and new standards become available, Sendmail evolves in more complex software – Mail Transfer Agent.

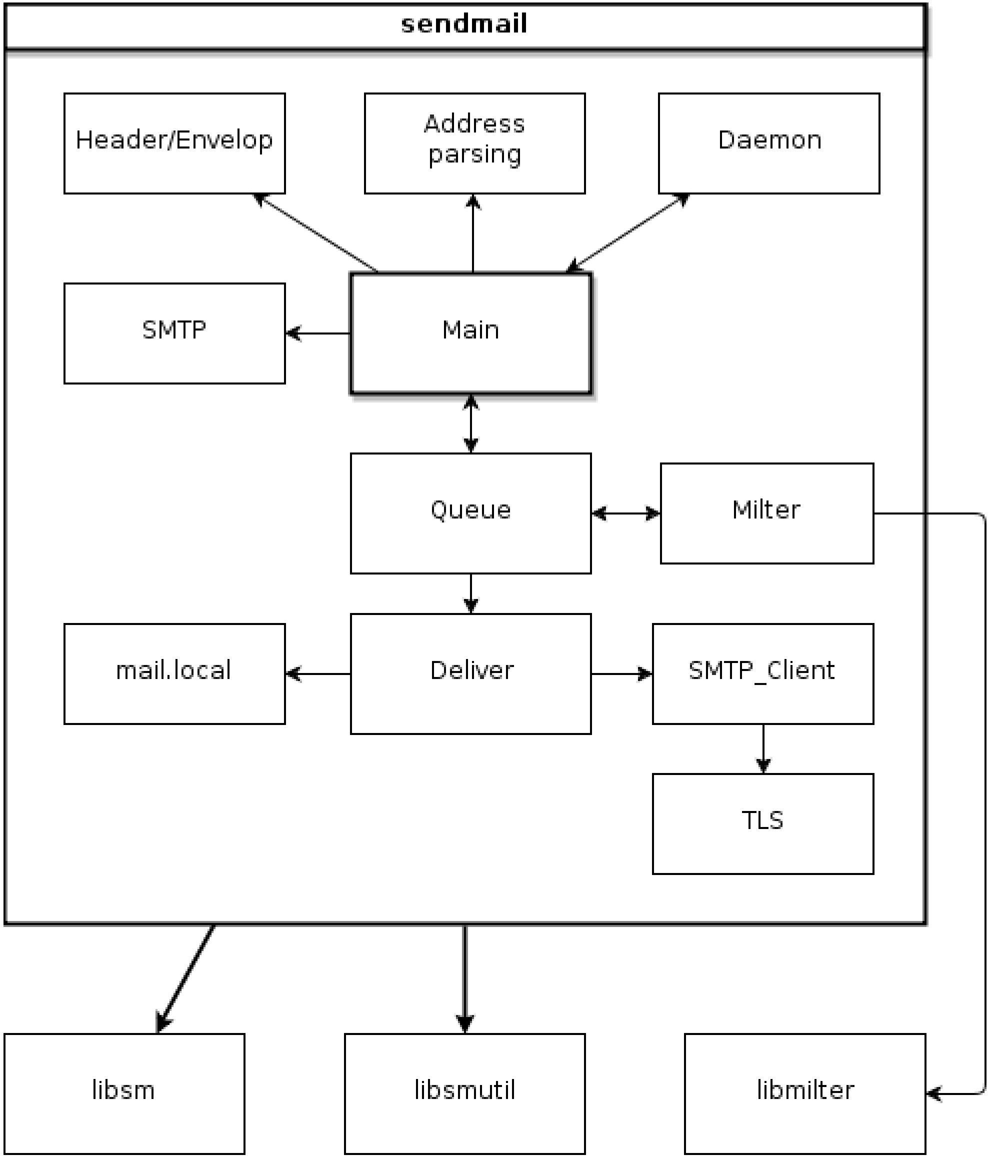
Today Sendmail is the most widespread cross-platform software that transfers electronic messages in the Internet. It does the most important, though invisible, part in the communication between people in the Internet: checking and fixing messages, queueing them, transferring via SMTP, providing sort of security to the electronic world (anti-spam and anti-virus defence, confidentiality).

Basic metrics

KLOC: 118 KLOC
Project start-up: 1980
Number of major releases: Sendmail has six major releases: delivermail 1 and 2, sendmail 3, sendmail 4, sendmail 5 and sendmail 8
Number of developers: <200
Size of user community or number of installations: 818,374,269 effective January 2011
Major stakeholders: Eric Allman
Use of concurrency: One part of sendmail is milter. It's kind of interface that allows user to add some plugins to the software. Each milter runs in a different thread.
Implementation language: C
Supporting software: Sendmail does not rely on any particular software. It is cross-platform software that can use features of different systems; moreover it can even work with different databases.

High level architecture

Diagram of software architecture

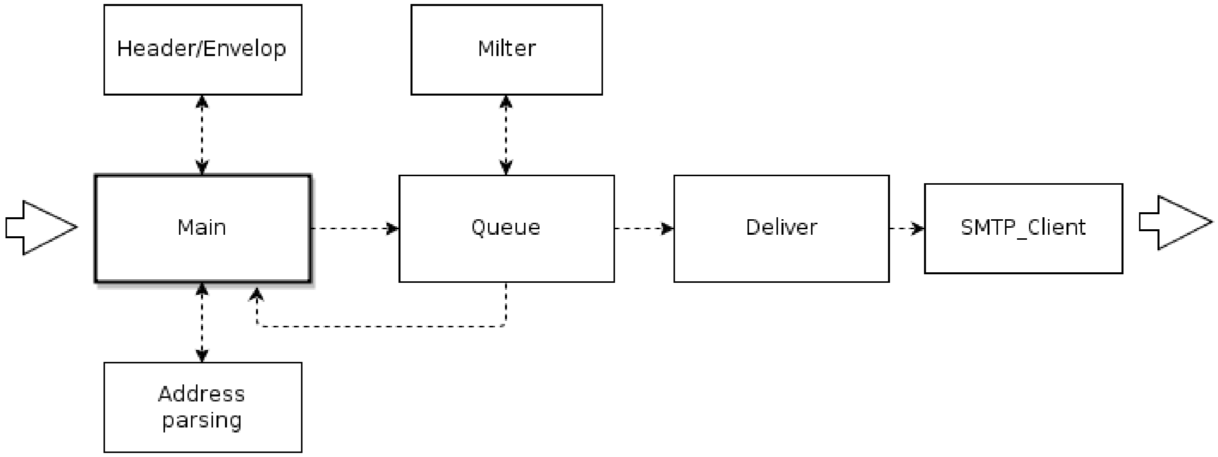


Each box represents a “logical” module; in some cases it’s a single c-file, and in the others – group of c-files. The size of a box somehow relates to the size of a module.

Arrows show a relation “depends on”. Two bold arrows, which go from the sendmail group, mean that the majority of elements in sendmail group use a particular module.

High level scenarios

Let’s look at the process of sending an e-mail (assumptions: sendmail is not in daemon mode, we will not use encrypted connection).



- Process starts when user forces the Mail User Agent to send an e-mail, and MUA calls the sendmail.
- Sendmail begins with initialization procedure (not shown on the image) – reading and processing of the configuration file
- Main module calls Address parsing module, which splits the address into 3 parts: network, host, user
- Main module creates/modifies header and envelop, and sends the message to the Queue
- Queue module calls all of the attached milters (mail filters). Queue module send message to the Deliver module if all milters accepted it; otherwise, Queue module rejects the message
- Deliver module sends the message using the information that was previously extracted from the address

Data structures or algorithms: Anything special

Control flow and/or data key to the architecture if any: Anything special

Architectural style: Though sendmail looks like a standalone software, it has Pipes/Filter architecture. It’s pipeline because a result of each module is an “input” for another module. Also milters in sendmail are kind of filters in Pipes/Filter architecture – each milter is independent from both the pipeline and other milters, in addition all milters run concurrently.

Major evolutionary changes: There were not any big changes in architecture during the history of sendmail. However, it is worth mentioning that all modules have faced a lot of changes to adjust to surrounding tech-world, and that some new modules (like SMTP and TLS) were added to the pipeline.

Performance bottlenecks: The only thing that I’ve found is that there were problems with early implementation of queue. As Eric Allman states in the chapter, queue crushed each time it had around 10000 messages

Real time: It seems to me that only Milter module may create some problems. Queue module cannot proceed and send the message to the Deliver module until it got responses from all milters (accept message/reject message/milter crush or an error message). So, if some milter takes a lot of time to proceed a message it may create some delays for whole process.
But on the other hand, because sendmail is a Mail Transfer Agent, it is not expected to complete its work immediately.

Notation for architecture: No information
Methodology: No information

Appendix:
Kruchten’s eight context attributes applied to Brown/Wilson systems

1. **Size:** M
2. **Criticality:** Hi
3. **Age of system:** XL
4. **Rate of change:** Lo
5. **Business model:** open source
6. **Stable architecture:** Lo
7. **Team distribution:** UnK
8. **Governance:** UnK