Architecture of a scalable mail system

Joel Jaeggli
for
PACNOG2
June 2006

The problem...

- E-mail is one of the most critical services beyond actual IP service that an ISP fields for it's customers.
- Yet, frequently email systems are simply assembled from various bits and pieces as new needs or problems arise without regard to the overall design of the system.

Growth

- Even in an ISP that is not rapidly expanding it's customer base, the growth in demand for email services can put substantial strain on the existing email system.
- The University of Oregon only has about 20% more email customers then it had 5 years ago.
- Yet the overall demands on the service, in terms of storage and throughput increased at least 8x over the same time-frame, due to higher mail volume, increased spam-filtering, and more frequent access by the end users.

The trap!

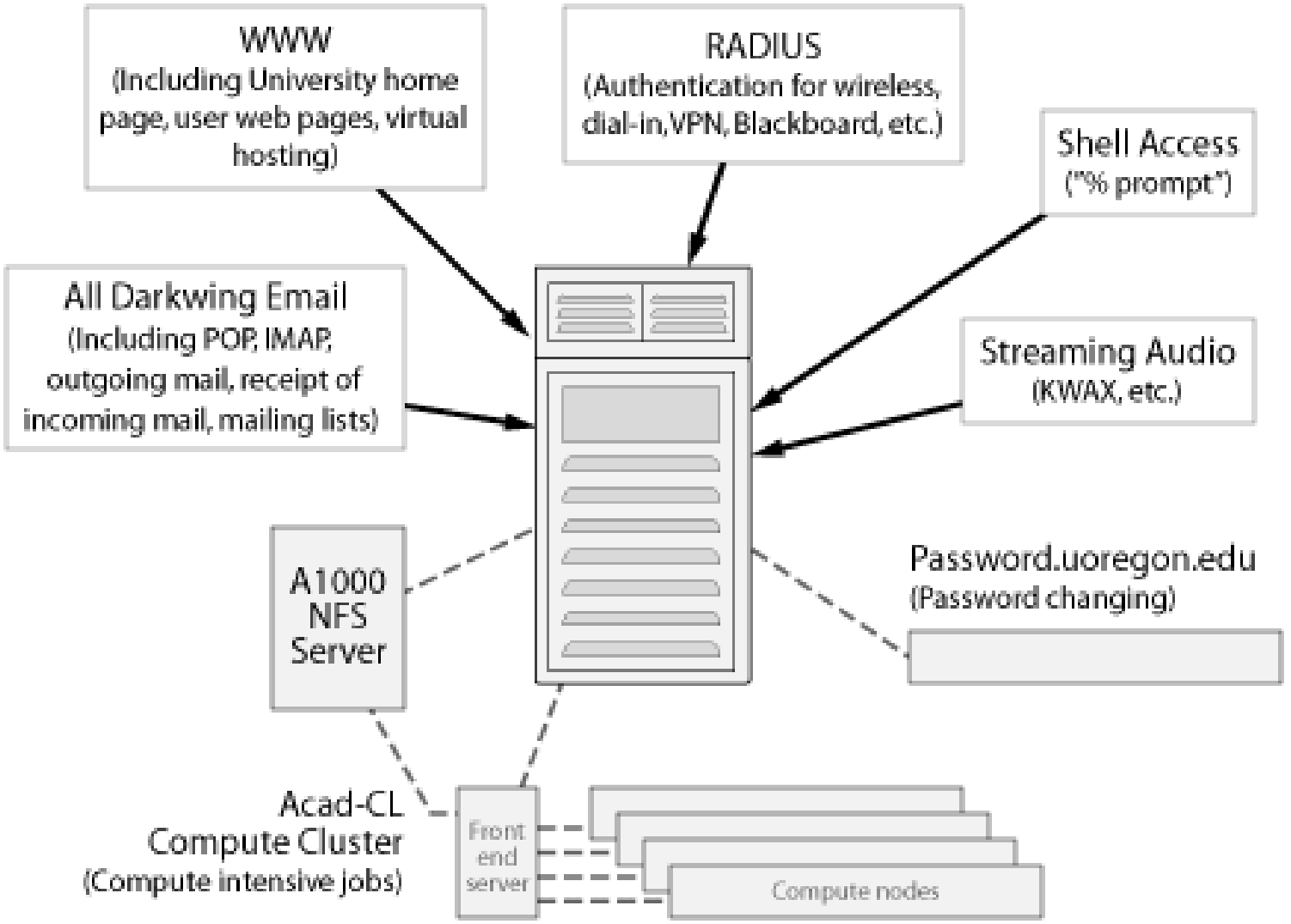
- It's pretty easy to configure a linux/unix machine to provide all services necessary to send and receive mail and present it to customers.
- Without some advance planning however, it can be hard to tease apart the monolithic server model without much pain and suffering.
- When the monolithic approach runs out of headroom, it's like plunging off the cliff, one more user, one more message, is to much and you take out the whole service.

Case-study – The University of Oregon

- A little self-criticism is healthy.
- Since the early 1990's we had served to customer's (faculty and staff, and then students) centralized computing needs including email from a pair of progressively larger UNIX machines (SUN 630MP, SPARC 1000, SPARC CENTER 2000, SUN Enterprise-5500).
- While email was always one of the principle applications for the systems it grew to consume virtually all of the resources to the exclusion of the other functionality the systems were supposed to be providing.

Classic Darkwing (Simplified) *

Everything ran on one box: A tangled, interconnected …



* slightly sanitized for political reasons

Case-study - Continued

- By 2002 it had become obvious that the 4x or so increase in performance that we could achieve by upgrading to a new generation of still larger Sun machines wasn't going to meet our current needs let-alone the the needs at the far-end of the hardware replacement cycle.
- E-mail services were becoming constrained as delivery volumes continued to increase.
  - We literally could not afford the cpu cycles on the machines for more effective spam filtering.
  - Process limitations were being put in place to limit the number of pop and imap users. In order to stave off a meltdown on the disk i/o side. We were actually having to deny people service.

Continued 2

- Lot's of ISP and Enterprise sysadmins have been down this road.
- Not all of them made it out the other-side.
- There is a reason why free email services are as popular as they are despite the fact that many of them aren't very good.

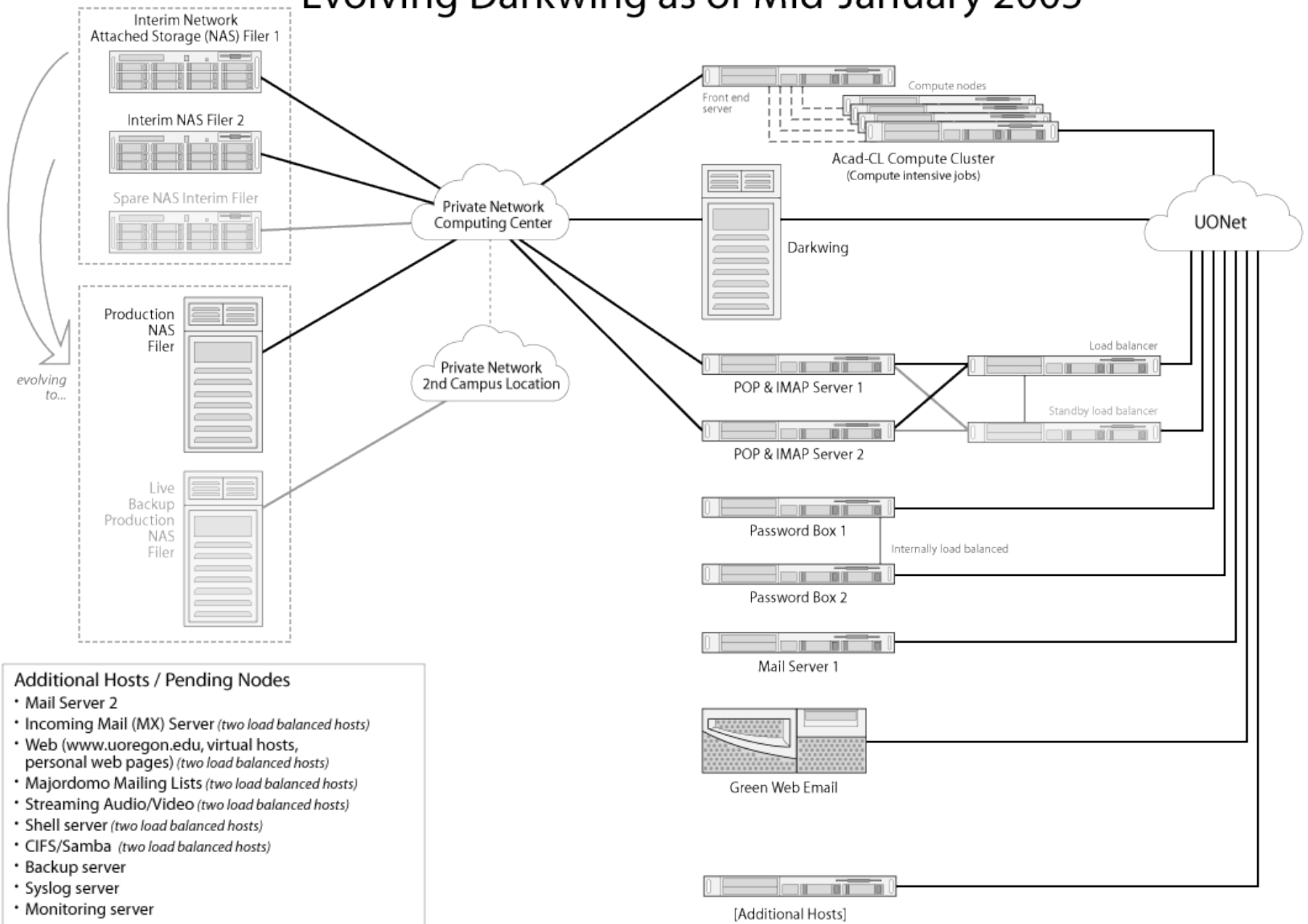
Case-Study - Denouement

- Faced with the options of:
  - Pulling a rabbit out a hat.
  - Looking for new jobs.
- We choose the former...
  - Only it was a bunch of rabbits.

Case-study - approach

- A mail system is not inherently monolithic. It's a series of interrelated tools that conspire to achieve a common goal...
- We had three immediate problems to solve:
  - Separate mail delivery from the other services so that we could actually implement the better filtering we had been holding off on.
  - Get the storage off Darkwing since the SUN's disk i/o bottle-neck was as effective a killer as the lack of cpu.
  - Get back to the point where we weren't dropping pop and imap connections (including from out webmail interface) on the floor due to lack of resources.

Evolving Darkwing as of Mid-January 2005



Additional Hosts / Pending Nodes

- Mail Server 2
- Incoming Mail (MX) Server *(two load balanced hosts)*
- Web (www.uoregon.edu, virtual hosts, personal web pages) *(two load balanced hosts)*
- Majordomo Mailing Lists *(two load balanced hosts)*
- Streaming Audio/Video *(two load balanced hosts)*
- Shell server *(two load balanced hosts)*
- CIFS/Samba *(two load balanced hosts)*
- Backup server
- Syslog server
- Monitoring server

Note: Multiple services may be delivered from the same physical pair of load balanced hosts, depending on system requirements and other factors

Case-Study – What did we do?

- As an interim solution, moved the storage off of Darkwing onto a pair of cheap linux based NFS servers.
  - We tried using only one but it was only twice as fast as darkwing's disk subsystem and we melted it.
- Deployed two pop/imap servers fronted by a layer-3 load balancer
  - This caused an immediate and massive drop in the load on darkwing, when this service went away.
- Pulled incoming and outgoing smtp off darkwing onto another box.

What did we do - 2

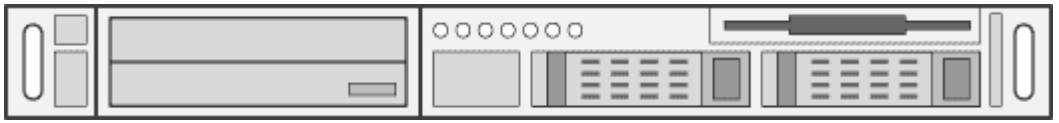
- We attempted to move services in such a way that there would be as few user visible interruptions and changes as possible.
- But there were still a few bad days...

Case study – The server building-block

- Obviously one of the goals here was to stop spending huge amounts of money on expensive proprietary unix machines.
- Yet we needed a lot more machines.
- If they were identical or at least similar then they could serve as drop-in replacements for each other.
  - Sparing would be much simpler,
  - no more expensive service contracts for huge machines that you can't afford to take an outage of any duration on.

The server building-block - 2

1u rackmount pc



The original aim was to have them cost about $2500ea. Actual costs were a bit further north of that since some of them were thought to need to be fairly high-end and we wanted at least the first few batches to be similar...

Machines were all dual-processor xeon or opteron 1u rackmounts with 2GB or more of ram and a pair of mirrored disks.

They were all sourced from regular commercial server vendors.

Use of local storage for user data is minimized as much as possible.

It seems entirely feasible that this could have been done with more smaller machines and indeed that may be the approach in the future.

The big changes, then till now - Storage

- Storage moved from linux NFS servers to a pair of netapp r200 filers, one serves as a backup.
- We went from having:
  - 1999 - 2(Suns) x 12 x 18GB drives
  - 2002 – 2(Suns) x 12 x 73GB drives
  - 2004 – 2(Linux) x 12 x 250GB drives
  - 2005 – 1(Netapp) x 56 x 300GB drives
- The Linux NFS servers were faster than the local disk on the Sun, the single Netapp is faster still. however a time will come when the second filer will be pressed into production or we'll need to purchase more filers.

The big changes – Hosts

- In january 2005 we had:
  - darkwing doing some smtp and mailing lists
  - 1 smtp server
  - 2 pop/imap servers
  - 1 webmail server
- As of now we have:
  - 4 smtp servers
  - 1 mailing list server
  - 5 pop/imap servers
  - 3 webmail servers
- And that's just the mail related machines.

The big changes - Software

- Deploying multiple SMTP servers gave us the luxury of deploying better mail filtering then we had before. This was done with an eye towards preserving some level of user control over areas where false positives are possible.
- Switching from uw-imap to dovecot bought us better performance due to server-side caching of mailbox indexes.
- migration to maildir is still an outstanding issue.

The big changes - Practices

- It is now possible with some care to roll out upgrades to various parts of the service without taking a visible outage simply by taking machines out of rotation on the load balancer.
- The loss of an individual machine in the redundant services results in at most a temporary loss of state and a modest degradation in overall performance.

How can we apply this experience?

- Even when building a significantly more modest mail-system than the one at the U of O we can apply a similar set of principles.
- With some intelligent design choices the components of mailv server collapsed onto a single machine can easily be separated at a later date if need be.
- Some design choices critical to scaling mail service at this point (maildir delivery, imap caching) can be costly (in terms of time) to migrate to when you already have a large user-base.

Plan of attack for this workshop

- Get Sendmail up and running.
- Relay for our customers only.
- Local delivery using Procmail.
- Delivery in maildir format into home directories.
- Get Clamav installed.
- Get Spamassassin installed.
- Get Dovecot installed .

Extra curricular activities

- Cyrus-SASL and STMP Submission
- Install a webmail front-end.

Why Sendmail?

- In other workshops we teach Exim or Postfix...
- But many people already have existing sendmail installs and some basic experience with it.
- Many of the pieces We're talking about here are easily usable with any MTA.

Why Maildir?

- Many modern MUA's (mail clients) assume that they live in a lock-free environment which is not possible with traditional mbox style mail folders.
- As mailboxes get larger the costs associated with moving messages from one mailbox to another (and rewriting the original mailbox) grow with mbox while they remain constant with maildir.
- But you've potentially traded one performance problem for another.

Why Clamav?

- Commercial anti-virus vendors are in the business of selling fear.
- Clamav actually works pretty well.
- The near-zero false postive rate means that you can safely run it across all mail coming in to your system.
- Integration into the early stages of mail delivery mean you can simply reject the message before the connection is closed rather than have to decide later if you want to bounce the message (in general bouncing virus messages is considered bad). This makes it better in this respect the most commercial products.

Why Spamassassin

- IP, DNS or URL based blackhole lists are by themselves a fairly gross tool for determining if mail is spam.
- Using blackholes in conjunction with forms on content analysis has a pretty high detection and low errror rate.
- The ability to customize spamassassin on a per-user basis can make it vastly more accurate.

Why Dovecot?

- Dovecot supports both maildir and mbox format mailboxes which can make migration a little easier.
- Dovecot implements a server-side indexing scheme that can vastly speed up the handling of large mailboxes.
- It's generally safe handling multiple concurrent connections to the same mailbox even on different machines.
- Out of the box it supports pop3/s imap/s.