Deploying the Apache HTTP Server within the Apache Software Foundation

Justin R. Erenkrantz
University of California, Irvine
http://www.erenkrantz.com/oscon/
justin@erenkrantz.com

Why should I pay attention?

- *One of the folks behind root@apache.org*
- *Responsible for maintaining servers*
- *Committer to Apache HTTP Server and APR*
  - *Familiar with the httpd 2.x codebase*
- *Committer to Subversion (we run that too)*
- *'Spare' time: PhD student at UC Irvine*

What is the ASF?

- *The Apache Software Foundation (ASF)*
- *Organizational, legal, and financial support*
- *Not-for-profit foundation*
- *Currently has 24 top-level projects (TLP)*
  - *Each TLP may have many codebases*
- *All of these have websites, code, mailing lists*

Apache HTTP Server History

- *Apache HTTP Server has been market leader for over eight years...and counting*
- *2.0 series went GA in April 2002*
  - *New architecture to fix real-world issues*
  - *Windows (and others) are first-class now*
  - *Includes mod_dav, mod_ssl out of box*

What does the ASF provide?

- *Three essential services:*
  - *Websites: http://jakarta.apache.org/*
  - *Version control: httpd-2.0 repository*
  - *Mailing Lists: general@xml.apache.org*
- *The failure of any impairs the project!*

What does the ASF use?

- *Desire to eat own dogfood, if possible*
- *Websites: Apache HTTP Server 2.0*
- *Version control: CVS and Subversion*
- *Mailing lists: ezmlm, qpsmtpd, qmail*
- *Platform: FreeBSD (4.x and 5.x)*

What level of service?

- *Users: ~900 committers & ~130 members*
- *Websites: ~40 million views/month*
- *Version control: ~8GB of source*
- *Mailing lists: ~400k inbound emails a day*
- *Expect these numbers to continue to rise*

Organizational Structure

- *Email is key to our coordination planning*
  - *IRC used for real-time fire-fighting*
  - *infrastructure@: General list*
  - *root@: People with root access*
  - *apmail@: People who can create lists*
  - *Physical access: Big red button hitters*

Where do the servers reside?

- *Like our contributors: all over the place*
- *Main servers now at UnitedLayer in SF*
- *Used to reside at CollabNet (for free!)*
  - *Strong desire to be self-reliant*
  - *Pay our own co-lo costs now*
- *Can now provide access to those in area*

Supporting Hardware

- *Still try to reduce need for physical access*
- *Terminal Server*
  - *Allows remote serial console*
  - *Not all machines support BIOS output!*
- *Power distributor*
  - *Allows power-cycling if all else fails*

The ASF machines

- *minotaur: Shell, CVS, WWW*
- *hermes: email server*
- *nagoya: bug-tracking, web mail archives*
- *brutus: Gump continuous builds*
- *moof: Apple test box*
- *ajax: European backup*

The ‘Not-quite-so-big’ Picture

minotaur (CVS, WWW)

Cyclades ACS

nagoya: E4500 (Webapps)

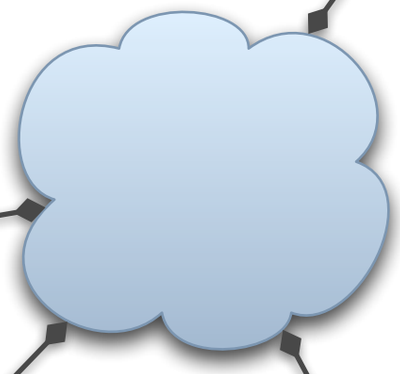
hermes: x345 (SMTP)

eris: x345 (Webapps?)

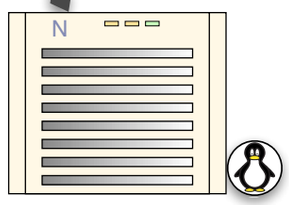
brutus: x345 (GUMP)

loki: x345 (VMWare?)

Cyclades PM20



moof: PowerMac (Testing)



ajax: Itanium (Backup)

Dual Virtual Host Strategy

- *www.apache.org = cvs.apache.org*
- *Traffic primarily on www instance*
  - *'cvs' requires Subversion, SSL, WebDAV*
    - *High(er) memory footprint!*
  - *Leverage 'IP alias': .194 and .195*
  - *Two instances with one optimized for space*

Two Parallel Instances

- *Other advantage of two instances*
  - *Allows 'unstable' testing, too*
    - *www.apache.org*
      - *httpd-2.0 APACHE_2_0_BRANCH*
    - *cvs.apache.org*
      - *httpd-2.0 HEAD (aka 2.1)*

Choice of MPM

- *Multi-Processing Module (MPM) in 2.0*
- *Prefork: Traditional Apache 1.3 model*
- *Worker MPM: smaller footprint w/ threads*
- *Configured for 800 clients.*
  - *Yet, we use prefork...Why?*
- *Threads busted in FreeBSD (5.3: first fix!)*

Off-loading Large Transfers

- *Main webpages are not mirrored*
- *Downloads are mirrored*
  - *Selected Mirroring Strategy*
    - *Importance of being fresh: 4x daily.*
    - *~5GB of releases served by mirrors*
    - *~225 mirrors around the world*

The Need for Mirroring

- *ASF had long history of informal mirrors*
  - *Only HTTP Server used them*
    - *Jakarta was majority of traffic*
  - *Users defaulted to our servers **not** mirrors*
  - *Too many downloads to serve ourselves*
    - *Had to respect our bandwidth cap!*

Mirroring Strategy

- *Made mirrors mandatory*
  - *Looked at how SourceForge did it*
    - *Found extra steps too cumbersome...*
- *Promote digital signatures*
  - *Mirrors could be corrupt or malicious*
  - *Signature links still point to our servers*

Building a Web of Trust

- *MD5 and PGP signatures available*
- *Provide that it hasn't been corrupted*
- *Yet also requires user participation*
  - ***You** need to verify the signature*
  - ***You** should also have a path to signer*
- *Keysigning parties are extremely beneficial*

Mirroring Overview

- *Developed Python script to read mirrors*
  - *Greg Stein's EZT template engine*
    - *Project-specific look and feel*
  - *Use GeoIP to find geographic region*
    - *Present locale-appropriate mirrors*
- *95% percentile usage now ~18Mbps*

Evolution of ASF Setup

- *Then: daedalus (mail/www); icarus (CVS)*
- *Now: minotaur (www/CVS); hermes (mail)*
  - *Icarus retired when minotaur came*
    - *Begin of switchover to UL co-lo*
  - *Hermes forced into production*
    - *Had been testing it, but daedalus died*

The role of minotaur

- *daedalus and icarus were initial boxes*
  - *Dual PIII/800s: ~2000-2003/2004*
    - *Obtained around early 2003*
      - *Xeon/2.4GHz with HyperThreading*
      - *RAID5 array with 400GB*
    - ***Two** months doing 'make world'*

Why was that history important?

- *We do not have a pure dedicated web server*
- *Most compete with other critical services*
  - *In minotaur's case, CVS is hosted on it*
- *Also acts as shell server for accounts*
  - *Our CVS setup mandates shell accounts*
- *Must keep load 'low': ~1 load average (CVS)*

No dynamic pages

- *Emphasize static content to reduce load*
- *Adopt tools like Anakia, Forrest, etc.*
  - *Transform XML into (X)HTML*
  - *Provides benefits of SSIs without cost!*
- *Python CGI scripts handle mirroring*
- *MoinMoin Python Wiki (wiki.apache.org)*

Deciding When to Deploy

- *Two reasons for deploying a new build*
- *New release pending*
  - *Prefer to see a release run for 48 hours*
  - *Minotaur has ‘honorary’ release vote*
- *Resolve issues seen on our servers*
  - *Fixes for unknown reproduction cases*

Deploying New Builds

- *Greg Ames and Jeff Trawick handle www*
  - *Will deploy a new build and send email*
- *Only one custom patch at the moment*
  - *Stores input in brigade*
    - *Facilitates crashdump reproduction*
  - *Rest of patches have been committed!*

Responding to Failures

- *httpd architecture is fairly resilient*
  - *Each client handled in separate process*
    - *Crashes cause new child to be spawned*
    - *Segfault triggers crash dump and log*
- *Most common install error with ‘suexec’*
  - *We often forget the suid root bit! (Oops)*

Adopting Subversion

- *Moving away from CVS to Subversion*
- *Developer advantages:*
  - *Renames, atomic commits, etc, etc.*
- *Administration advantages:*
  - *Easier incremental backups*
  - *Better access control*

Subversion Backups

- *CVS has fundamental flaw*
  - *Every commit changes an RCS file*
    - *Not possible to keep just ‘delta’*
- *SVN: Incremental backups post-commit*
  - *Atomic commit can be ‘replayed’ later*
- *Synchronized to off-site mirrors*

Access Control

- *As ASF has grown, blurred group lines*
  - *Avalon wants to give commit access to Cocoon and JAMES developers*
- *Complicated 'avail' system on top of CVS*
- *mod_authz_svn provides group control*
  - *First written by Sander Striker (root@ too)*

Migrating from CVS

- *More projects are starting to **ask** to migrate*
  - *No forced migration...yet.*
    - *Users beginning to feel comfortable*
- *Tried to make all tools available from CVS*
  - *ViewCVS still works, commit emails, etc.*
- *Still haven't obtained a valid SSL cert! Ugh.*

Spam, spam, spam

- *Spam has been an increasing problem*
- *Most email is for mailing lists*
  - *Human moderator gets requests*
  - *Hundred moderate emails/day for some*
  - *Reach tipping point for us*
- *Re-deployed a new mail architecture*

qmail

- *I wish I knew why we use qmail...Yet, we do.*
- *Have lots of hacks to handle our load*
  - *Remote Concurrency & Big ToDo*
- *Very hard for new admins to understand*
- *Can't migrate away from qmail easily*
  - *ezmlm is too central to our mail delivery*

qpsmtpd

- *Replacement for qmail SMTP component*
- *Written in Perl by Ask Bjørn Hansen*
  - *perl.org and mysql.org uses it too*
- *Allows easy introduction of plugins*
  - ***Major** difficulty with qmail by itself*
- *If you use qmail, highly recommended!*

clamav

- *Free GPL Virus Scanner*
- *Automatic updates through freshclam*
- *Daemon via persistent clamd*
- *Qmail-scanner was awful with clamav*
  - *Actually spurred us to qpsmtpd*
- *Rejects about 10,000 messages a day*

SpamAssassin

- *Part of our strategy to eat our dog food*
- *Using SA 3.0.0rc1 with spamd*
- *Reject if email over 10.0 score*
- *All network tests enabled now*
  - *No Bayes rejection...yet (How??)*
- *Rejects about 12,000 messages a day*

Real-time Blackhole Lists

- *Reject a message outright if you are on:*
  - *Spamhaus XBL/SBL*
  - *SORBS DUL RBL (Dynamic IP)*
  - *DSBL*
- *Rejects about 130,000 messages a day!*
- *Wish didn't have to reject dynamic IPs, but…*

Custom Plugins

- *check_virtualdomains: Mail to @apache.org*
- *check_badrcpto: Reject john@, clark@*
- *exe_filter: Blocks EXE & ZIP (~90k/day!)*
- *spamwatch: Custom rule sets*
  - *SpamAssassin guys showed not effective*
  - *Catches ~500 emails/day w/false positives*

Forced migration!

- *We received a donation of four IBM 345s*
- *Did initial testing; No urgent schedule*
- *Daedalus hard drive died...*
- *Placed hermes into production before 'ready'*
- *Later found problem with RAID controller*
  - *Deployed a kernel patch on faith (worked)*

Lessons Learned

- *We can do 40 million page views/month on one box. You can too. Be smart!*
- *Always re-evaluate what you are doing.*
- *Try to involve as many as feasible.*
- *Can you off-load the work?*
- *Stick to the basics.*

Useful links

- *WWW: http://www.apache.org/server-status/*
- *CVS: http://cvs.apache.org/server-status/*
- *Henk Penning: http://www.apache.org/~henkp/*
- *Vadim Gritsenko: http://www.apache.org/~vgritsenko/*