FreeBSD Release Engineering

[or, is it done yet?]

So much work, so little time

- 46 releases since 1993, 1.0 to 5.4
  - A few were 'oops' or 'web-only'
- Countless BETA and RC builds for each
- Developers Previews, monthly snapshots
- Lots of man hours!
  - Maintaining the release infrastructure
  - Maintaining sysinstall and the install media
  - Reviewing and approving commits
  - Writing release documentation

Why?

- Originally done to support CD sales for Walnut Creek CDROM
- Release Engineering previously a full-time, paid job (shared with other development and writing duties)
- BSDi/Wind River collapse, rise of broadband at home reduces CD sales
- Work now done by a team of volunteers

Who?

- Cast of characters
  - Me
  - Ken Smith
  - Doug White
  - Robert Watson
  - Hiroki Sato
  - Murray Stokely
  - Recently retired: John Baldwin, Steve Price, Bruce Mah
  - Previously retired: Jordan Hubbard, PHK, Rod Grimes, ...

How?

- Most release cycles are planned to be 4-5 weeks
  - Tree 'frozen' during the release cycle.
  - Minimum of 2 BETA/RC builds released to encourage testing
  - Create the release branch once release time is near.
- Coordinate with ports and docs teams
- Coordinate with mirror sites, bittorrent to ensure smooth downloading

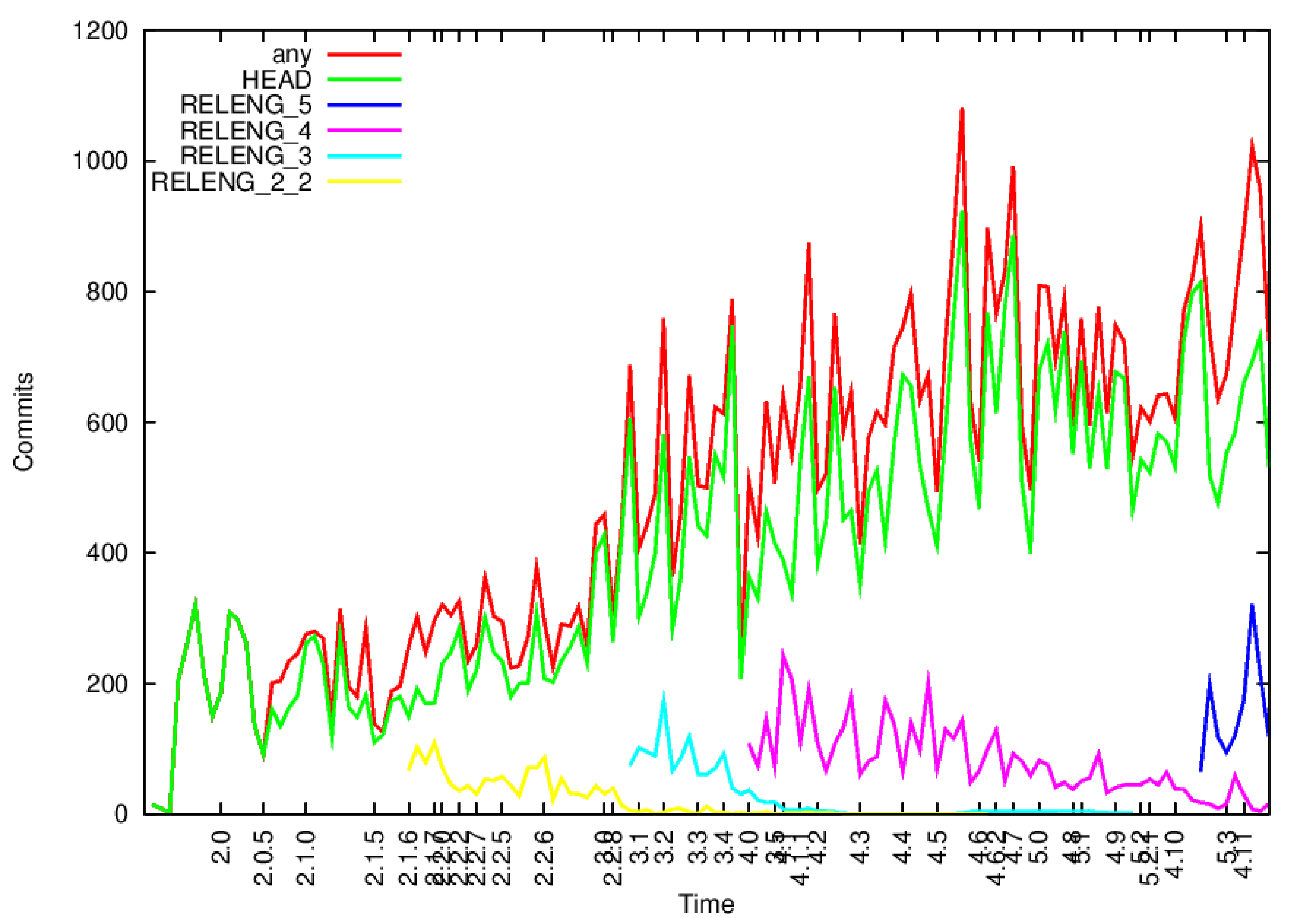
Branching

- Branching allows us to organize the commits and maintain a stable and consistent basis for releases.
- HEAD is the 'current' branch where experimental code goes
- RELENG_5 is the stable branch where tested code migrates to
- RELENG_5_4 is the release branch where errata and security fixes go.

“Are we still in code freeze?”

- Code freeze is an effective and important tool
  - Reduces the last-minute rush
  - Switches focus to fixing bugs, polishing the release.
- Delicate balance, don't want to frustrate developers and send them elsewhere
- Have tried to have philosophy of branching early to allow developers to continue to develope, test experimental fixes.

Commit graphs



Documentation

- Each release contains readme, release notes, install guide, errata notes, and hardware list for each supported platform.
- Migration guide for major releases
- Handbook also compiled and provided for install.
- Web page gets updated with similar information for each release.

Precompiled packages

- Build, or attempt to build, the entire 12,000+ ports tree for each release on every architecture
  - Valuable also for load testing the release candidate
- Package set placed online for download
- Selected popular packages also put onto the disc1 and disc2 ISOs

To ISO or not to ISO

- Releases were originally in FTP format only, CD images developed and kept in-house
- CD images made available for free download starting in 1998, pulled back for a while
- ISO format now the primary release medium
- Format changing from disc1 (dist + packages) + disc2 (livefs) to....

New Layout

- Combined dists + livefs for disc1, almost all packages on disc2
- Livefs allows for more interesting installers in the future
- Dedicated package discs allow for flexibility, customization by vendors
- Reduces work for RE, squeezing ever-bloating packages onto disc1 consumed too much time.

The Once and Future Sysinstall

- State of the art for 1995
- *“This is NOT the glorious semi-graphical sysinstall [...] this is simply the 'son of sysinstall' I've been promising to write in the interim”* -JKH 27 April 1995
- 10 years later....
- Libh? Bsdinstaller? Anaconda? YAST? Sysinstall?

5.4 and beyond

- FreeBSD 5.4 released 9 May 2005 (was really hoping for 5-4-2005, oh well)
- 5.5 in September
- 5.6 (maybe) in early 2006
- 6.0? ...

FreeBSD 6.0

- 5-CURRENT started in July 2000
- 5.0 Released Jan 2003, but not considered production quality, so 5-CURRENT continued
- 5.3 Released Nov 2004, 5-STABLE, 6-CURRENT finally started
- 4+ years for a CURRENT stream is a long time.

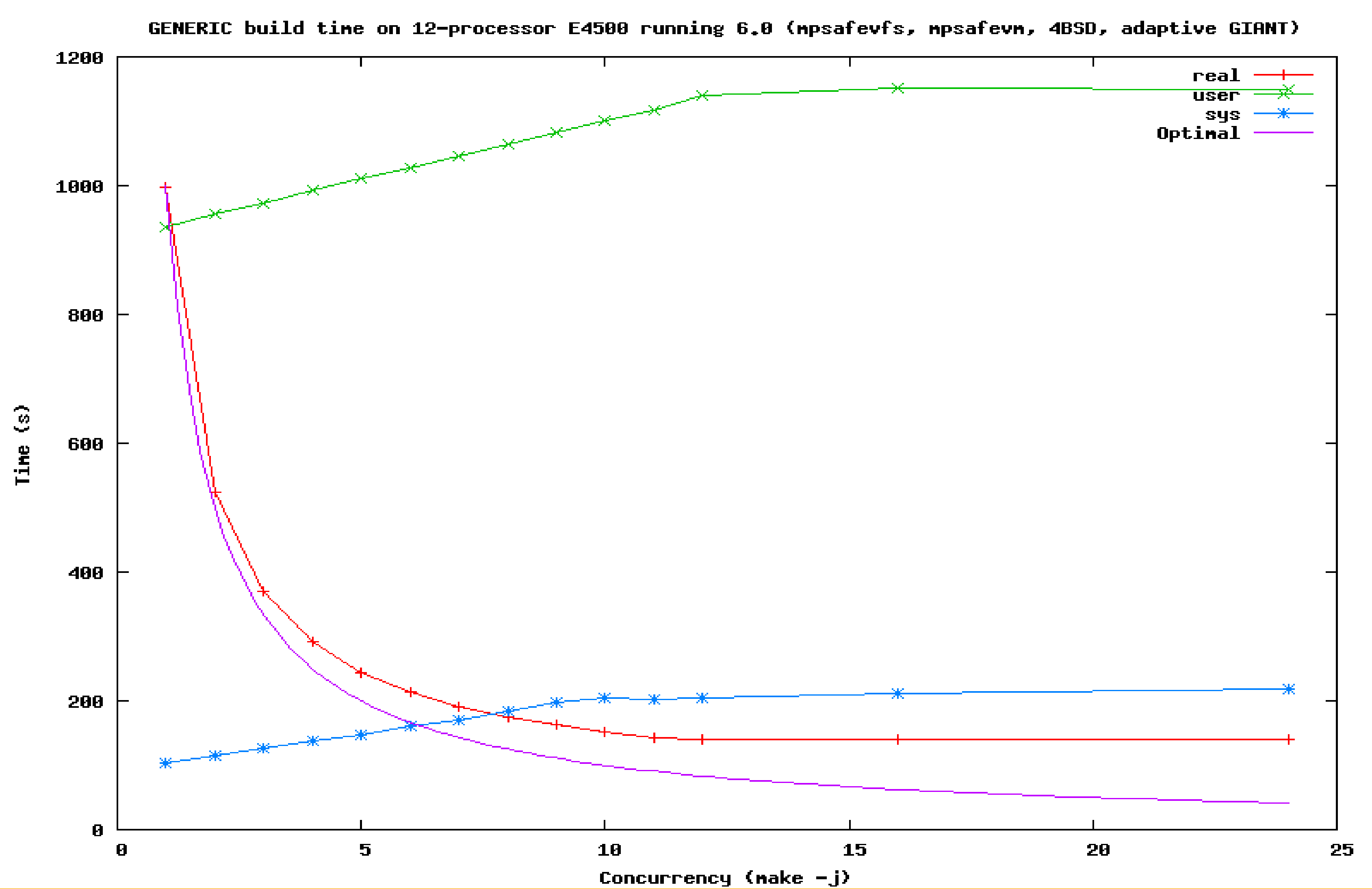
FreeBSD 6.0 cont.

- RELENG_4 stagnated because infrastructure required for new hardware (ACPI, drivers, APIC) only existed in 5-CURRENT.
- Long development cycle resulted in more bugs, not less.
- Solution is to do shorter, more focused dev cycles that are dictated by fixed dates, not desired features.

But what about performance?

- Major goal of 5.x was SMPng locking
  - Network, sockets
  - VM
  - Proc, file descriptors, pipes
- VFS is the last major subsystem, and it's almost done and undergoing testing in 6-CURRENT.
- Results? It scales!

12 CPU E4500


GENERIC build time on 12-processor E4500 running 6.0 (mpsafevfs, mpsafevm, 4BSD, adaptive GIANT)
Time (s)
1200
1000
800
600
400
200
0
0
5
10
15
20
25
Concurrency (make -j)
real
user
sys
Optimal

FreeBSD 6.0 cont.

- June 1, 2005 is the date for stopping 6-CURRENT development work and starting the 6.0 bug fixing.
- Why so soon since 5-STABLE/5.3? What does 6.0 have?
  - SMPVFS
  - WiFi / WPA
  - PHK-BUF

FreeBSD 6.0 cont.

- Things that might squeeze into 6-CURRENT
  - New WiFi infrastructure
  - OpenBSM
  - Remove ip6fw
  - Symbol versioning prototype, GCC 3.4.4
- June is almost here!

FreeBSD 6.1, 6.2

- Continue trying to do a release every 4 months.
- The parallel release cycle of 4.x and 5.x were hard.
- Need to make 6.x a fully viable replacement for 4.x and 5.x as soon as possible.

7-CURRENT

- What can we do in 12-18 months?
- What are the important pieces of infrastructure that need attention or repair?
- What are the important features that users need?
- My list:
  - Journaling UFS
  - CAM locking, iSCSI and SAS/SATA support
  - Installer

7-CURRENT cont.

- Other infrastructure:
  - Config(8), Kernel modularity
  - GCC 4
  - Multi-pass probe
  - TTY refcounting and locking
  - Locking, performance improvements
- Other features?
  - ARM/Xscale, PPC ports?
  - Xen?