Mozilla Release Engineering Brownbag for New Hires

(16dec2010)

How to make your first code change

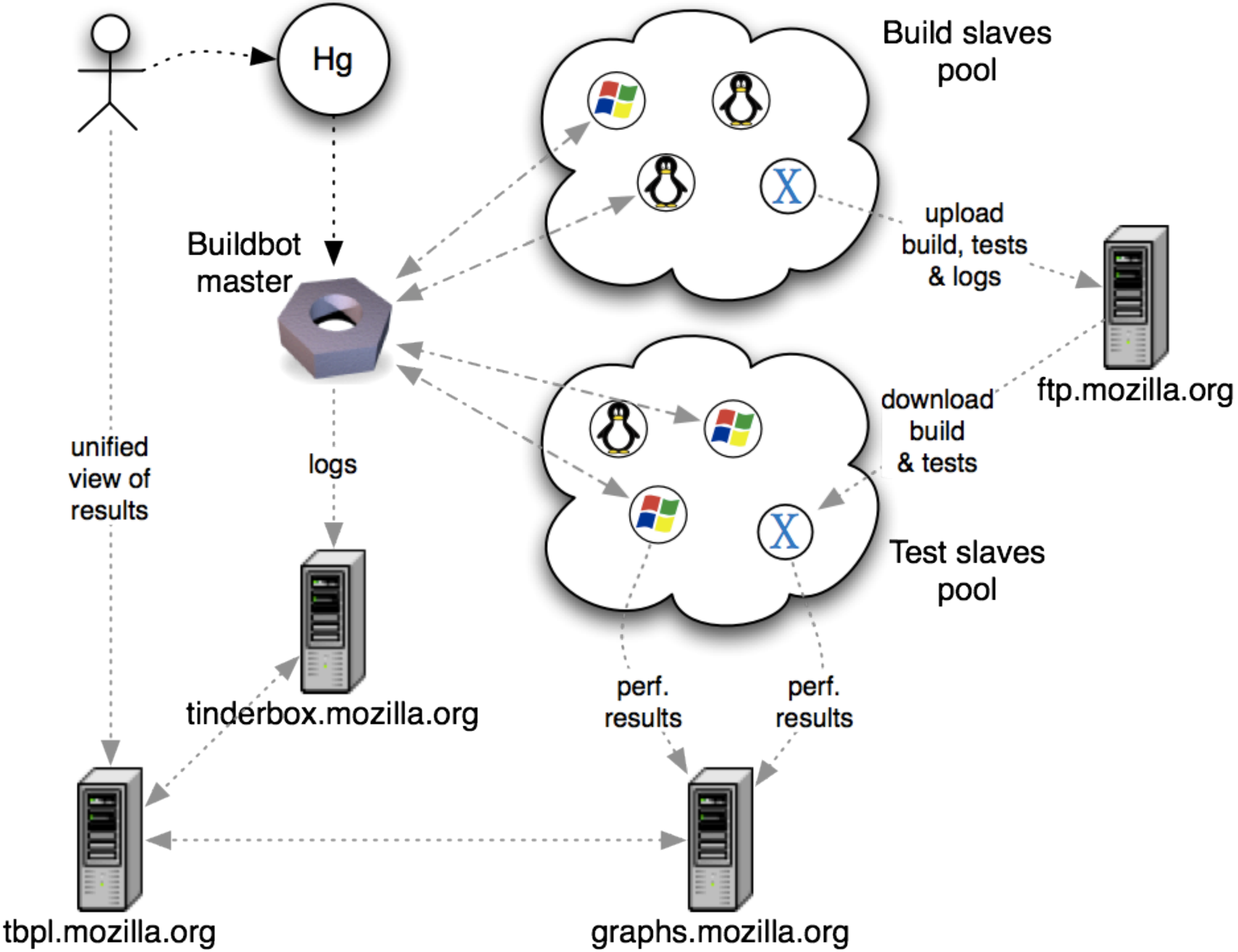
- **hg clone http://hg.mozilla.org/mozilla-central**
- Edit source files
- **hg diff –U 8 > foo.out**
- Attach foo.out to bug in bugzilla
  - set "r?", "sr?" to ask for review
- If you have commit access, you can checkin a reviewed patch using "hg push"
  - "checkin-needed" to ask someone else to commit

What happens when you checkin?

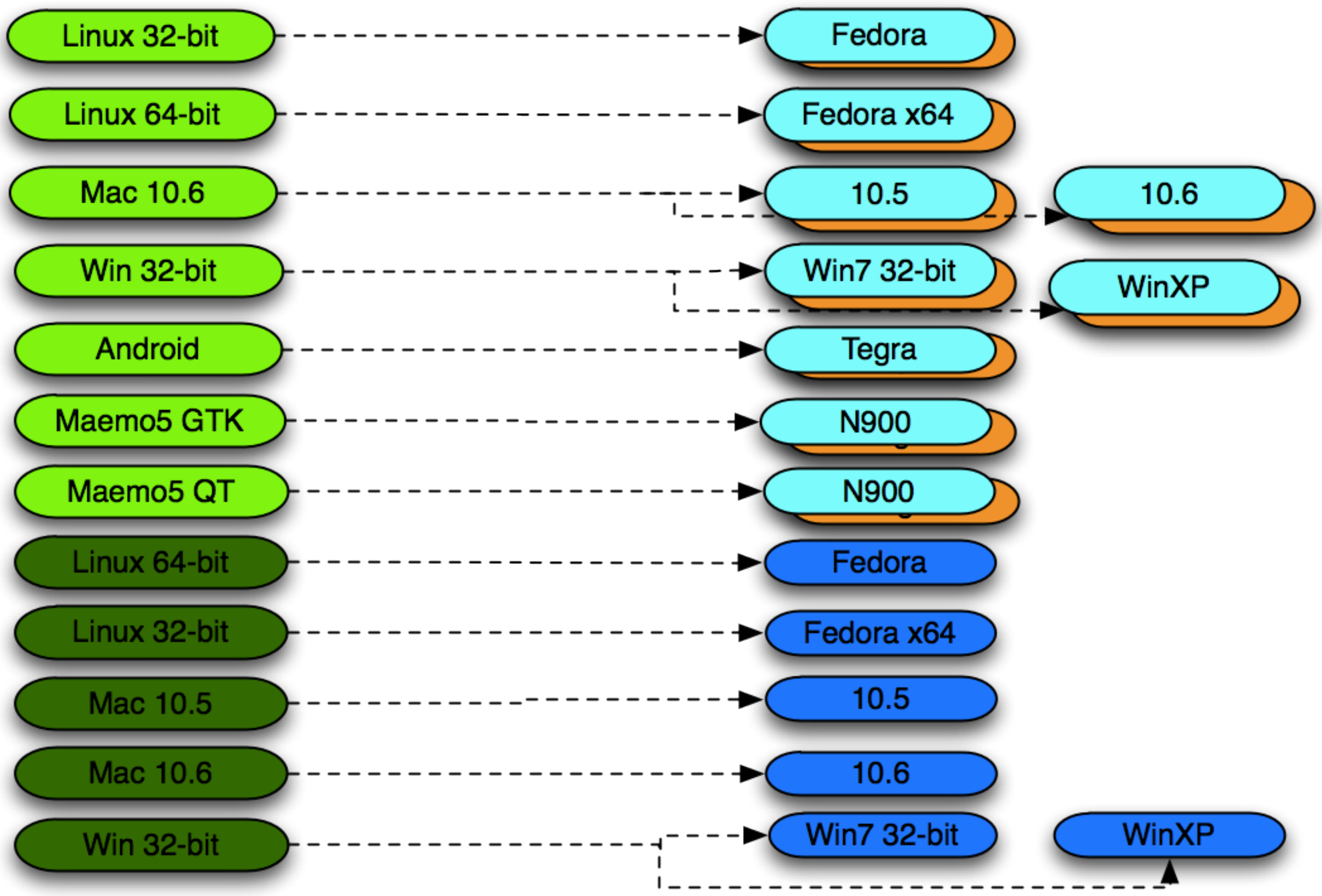
- Checkin to hg.m.o is detected
  - jobs to do builds are queued to run on next available machines
- Completed builds are put here
  - http://ftp.mozilla.org/pub/firefox/tinderbox-builds/[branch]-[OS]/[time]
- As builds become available:
  - Unit test and performance jobs are queued
- Results on tbpl.mozilla.org
- All with your change, and only your change.

What happens when you checkin?

- One check-in:
  - 170+ testing jobs (just desktop)
  - 70+ hours of desktop testing
  - 25+ hours of mobile testing
  - More than 4 days of CPU usage per check-in


Hg
Build slaves
pool
Buildbot
master
upload
build, tests
& logs
ftp.mozilla.org
download
build
& tests
unified
view of
results
logs
Test slaves
pool
tinderbox.mozilla.org
perf.
results
perf.
results
tbpl.mozilla.org
graphs.mozilla.org

Build, unit tests and talos coverage



🟩 optimized build

🟦 unit tests for opt. build

🟧 performance tests for opt. builds

🟩 debug build

🟦 unit tests for debug build

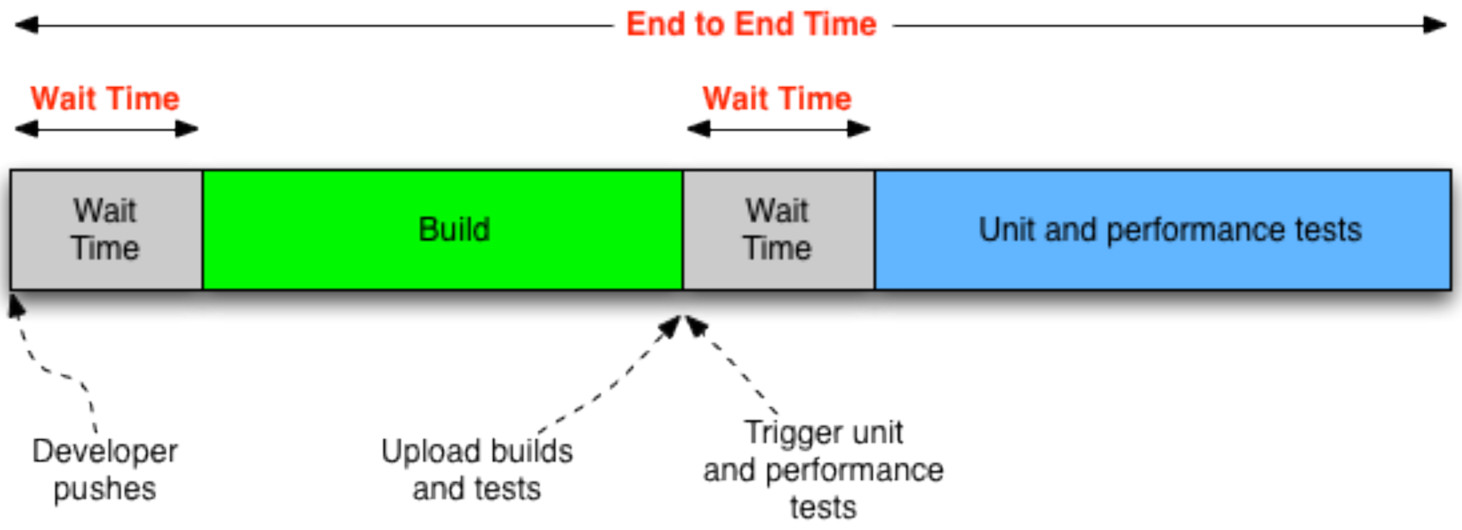
Unit test suites

- jetpack
- jsreftest
- mochitest
  - (1, 2, 3, 4, 5, other)
- mozmill-all
- reftest
- xpcshell
- crashtest
- a11y

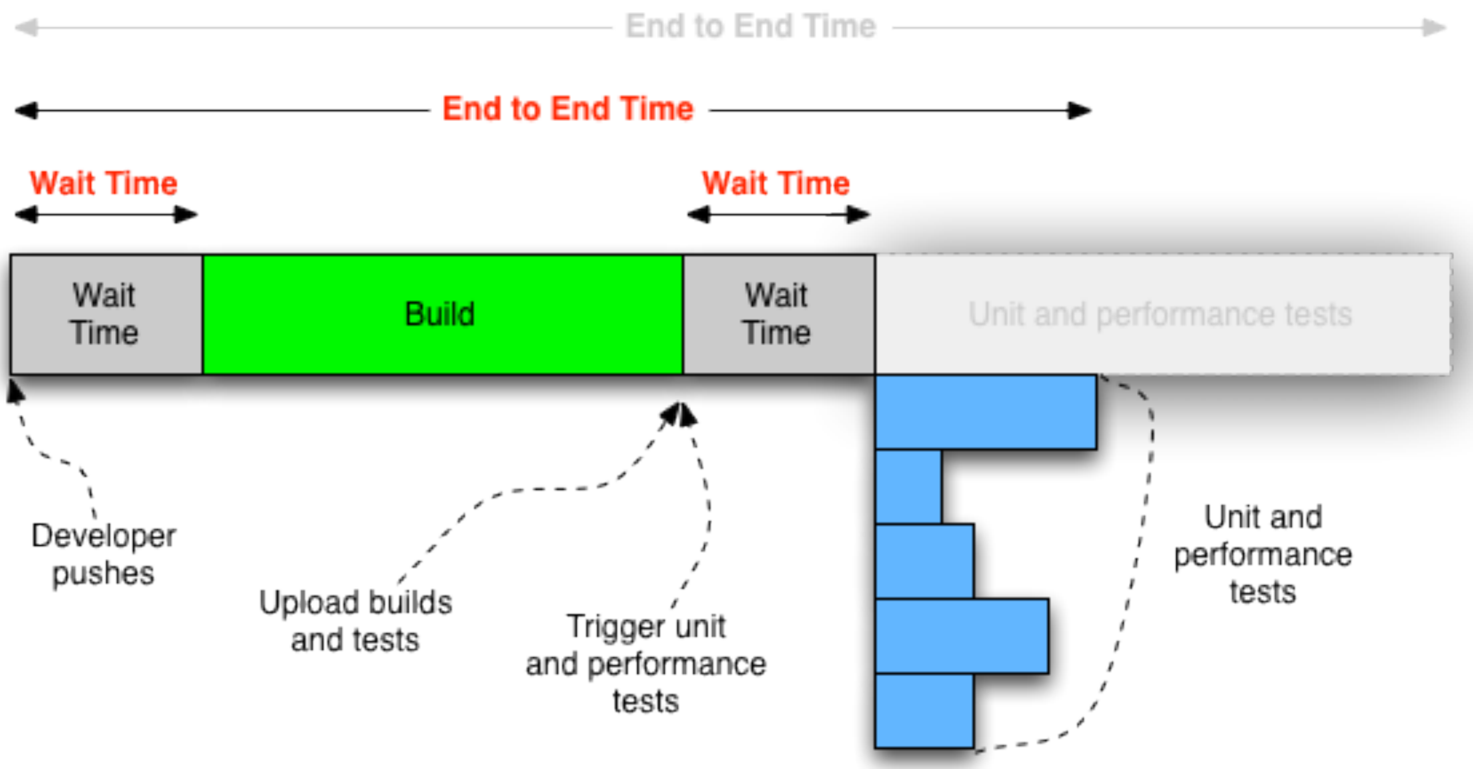
Talos suites

- chrome
- cold
- dirty
- dromaeo
- nochrome
- scroll
- svg
- tp4

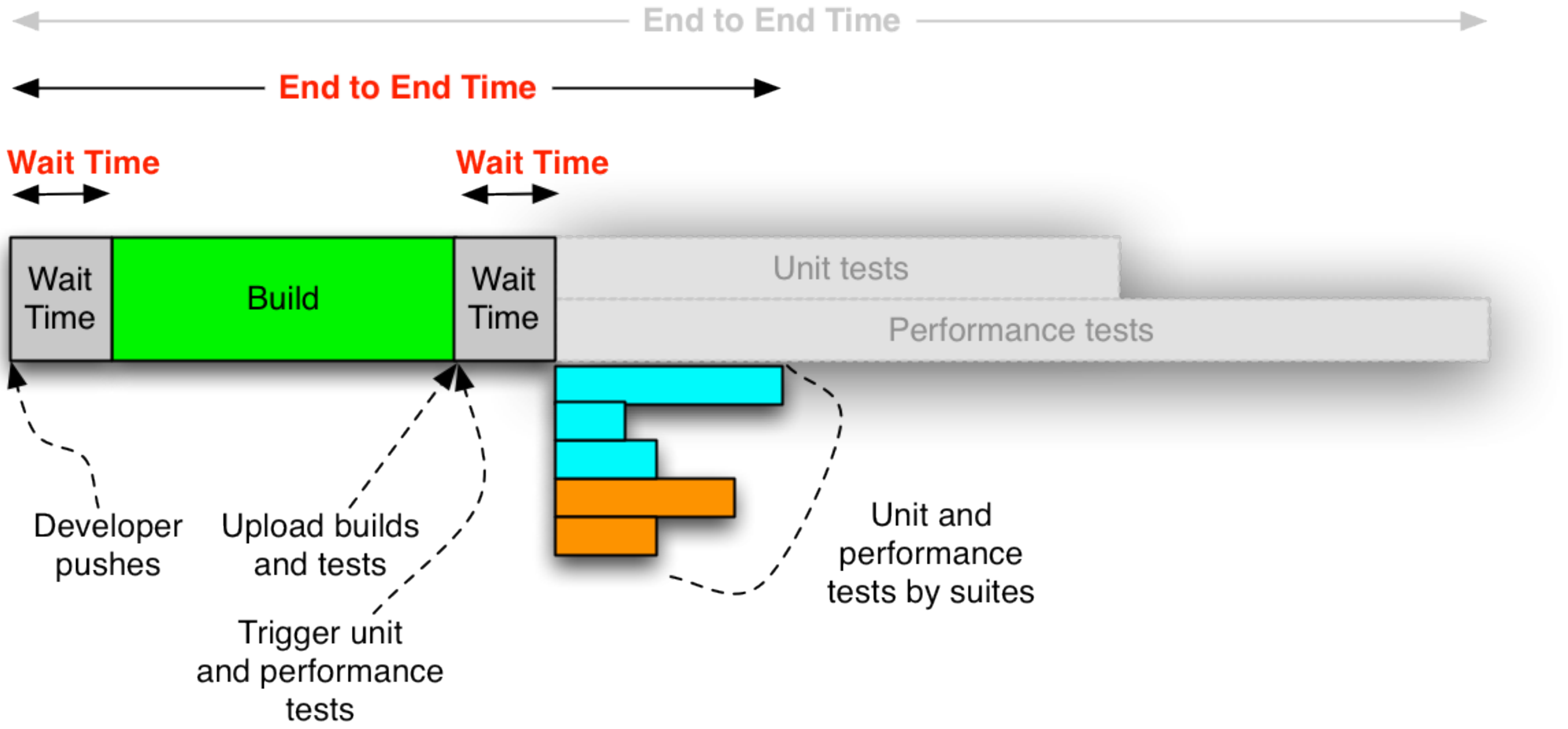
End to end – one block



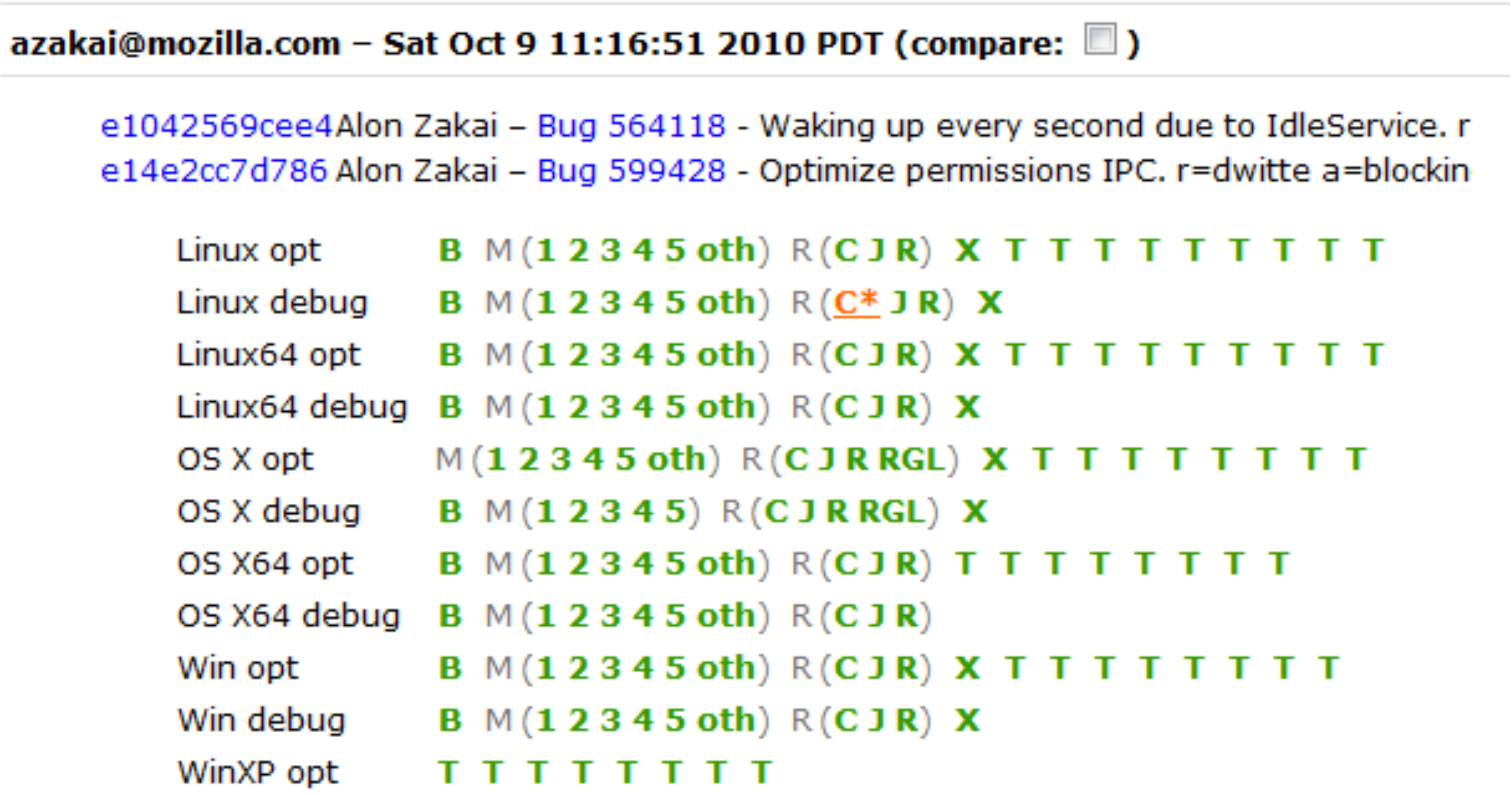
End to end – broken into pieces



End to end – our reality



tbpl.mozilla.org



azakai@mozilla.com – Sat Oct 9 11:16:51 2010 PDT (compare: □)

e1042569cee4Alon Zakai – Bug 564118 - Waking up every second due to IdleService. r
e14e2cc7d786 Alon Zakai – Bug 599428 - Optimize permissions IPC. r=dwitte a=blockin

Linux opt	B M(1 2 3 4 5 oth) R(C J R) X T T T T T T T T T
Linux debug	B M(1 2 3 4 5 oth) R(C* J R) X
Linux64 opt	B M(1 2 3 4 5 oth) R(C J R) X T T T T T T T T T
Linux64 debug	B M(1 2 3 4 5 oth) R(C J R) X
OS X opt	M(1 2 3 4 5 oth) R(C J R RGL) X T T T T T T T T
OS X debug	B M(1 2 3 4 5) R(C J R RGL) X
OS X64 opt	B M(1 2 3 4 5 oth) R(C J R) T T T T T T T T
OS X64 debug	B M(1 2 3 4 5 oth) R(C J R)
Win opt	B M(1 2 3 4 5 oth) R(C J R) X T T T T T T T T
Win debug	B M(1 2 3 4 5 oth) R(C J R) X
WinXP opt	T T T T T T T T

Then every night, we…

- do all the same builds again
  - full clobber, not dependent
- same changeset across platforms
- create L10n repacks
  - 80+ locales, per OS
- create “nightly” updates
  - someone running yesterday’s nightly build will be prompted to update to tonight’s nightly

TryServer

- Land changes on TryServer first before landing on mozilla-central**
- For best results, refresh your local repo to latest **before** using TryServer.
- Use the ❤TryChooser❤ syntax!(http://mzl.la/i1wj8t)
- Don’t compare your results to other TryServer jobs.
  - Instead compare to results for production code you are based on – where your change is the only difference.

`** Unless you have coverage through a project branch`

What can releng do for you?

- Re-trigger build/test
- Run multiple perf tests
- Cancel running jobs
  - Very important. Specially for the TryServer
- File a bug in mozilla.org:RelEng
  - RelEng working to make this self-service
- Do NOT re-submit to TryServer for missing jobs
  - TryServer will rebuild everything again
  - It will take longer to get results
  - It will slow down TryServer for others and yourself

Where and when to land changes?

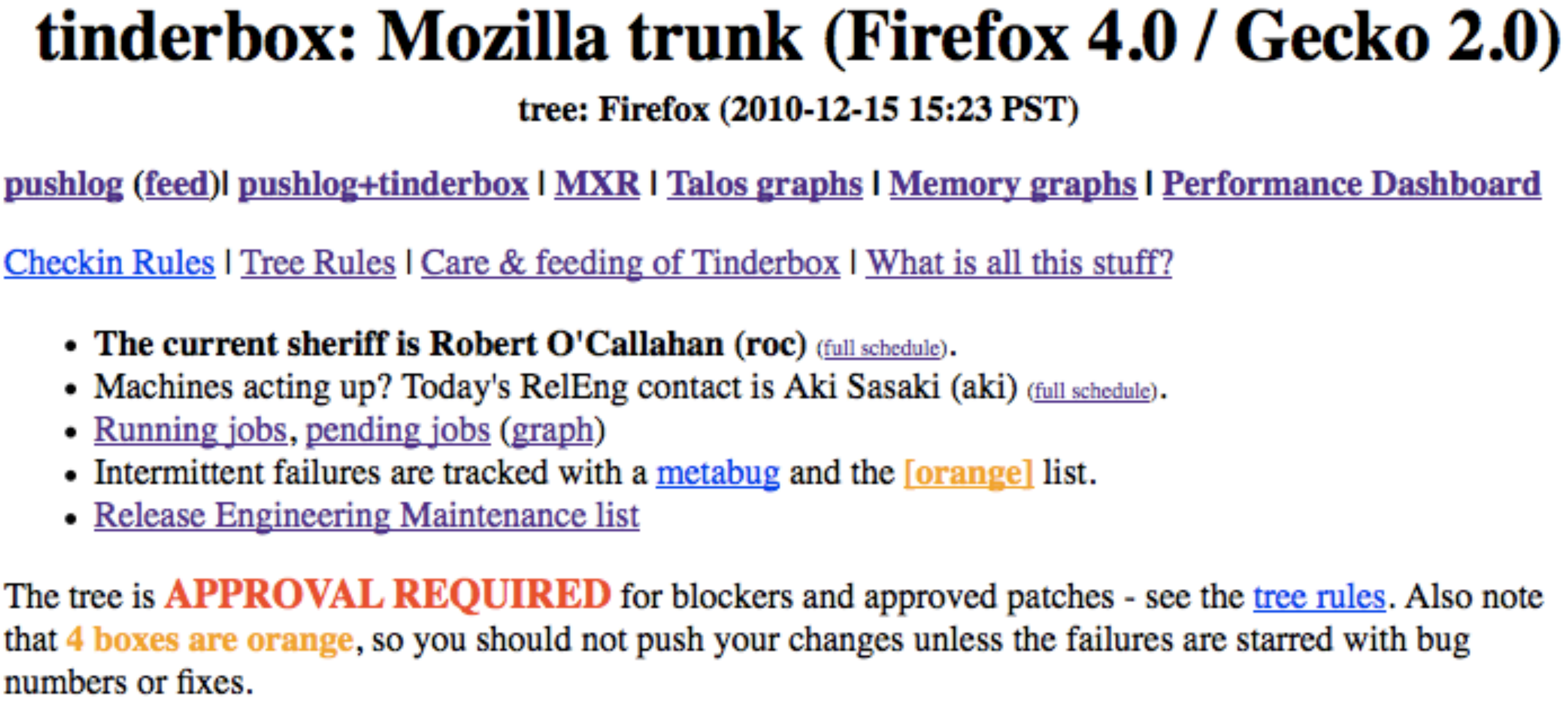
- release repos
  - mozilla-central, mozilla-1.9.2, mozilla-1.9.1
- project repos
  - ~~addonsmgr~~, electrolysis, jaegermonkey, ~~lorentz~~, places, tracemonkey, mobile-browser
- rentable repos
  - birch, maple & cedar
- tryserver repo

NOTE: if you need a new project branch, let us know

Where and when to land changes?

- If working on a project branch, land there first and make sure there are no regressions
    - e.g. tracemonkey
- Land on mozilla-central after checking with **Sheriff** and **Tree rules**
- If also needed in specific release, get approval in bug, and then land in mozilla-1.9.1/2 as directed by Release Driver.

Tree Status



tinderbox: Mozilla trunk (Firefox 4.0 / Gecko 2.0)

tree: Firefox (2010-12-15 15:23 PST)

pushlog (feed)| pushlog+tinderbox | MXR | Talos graphs | Memory graphs | Performance Dashboard

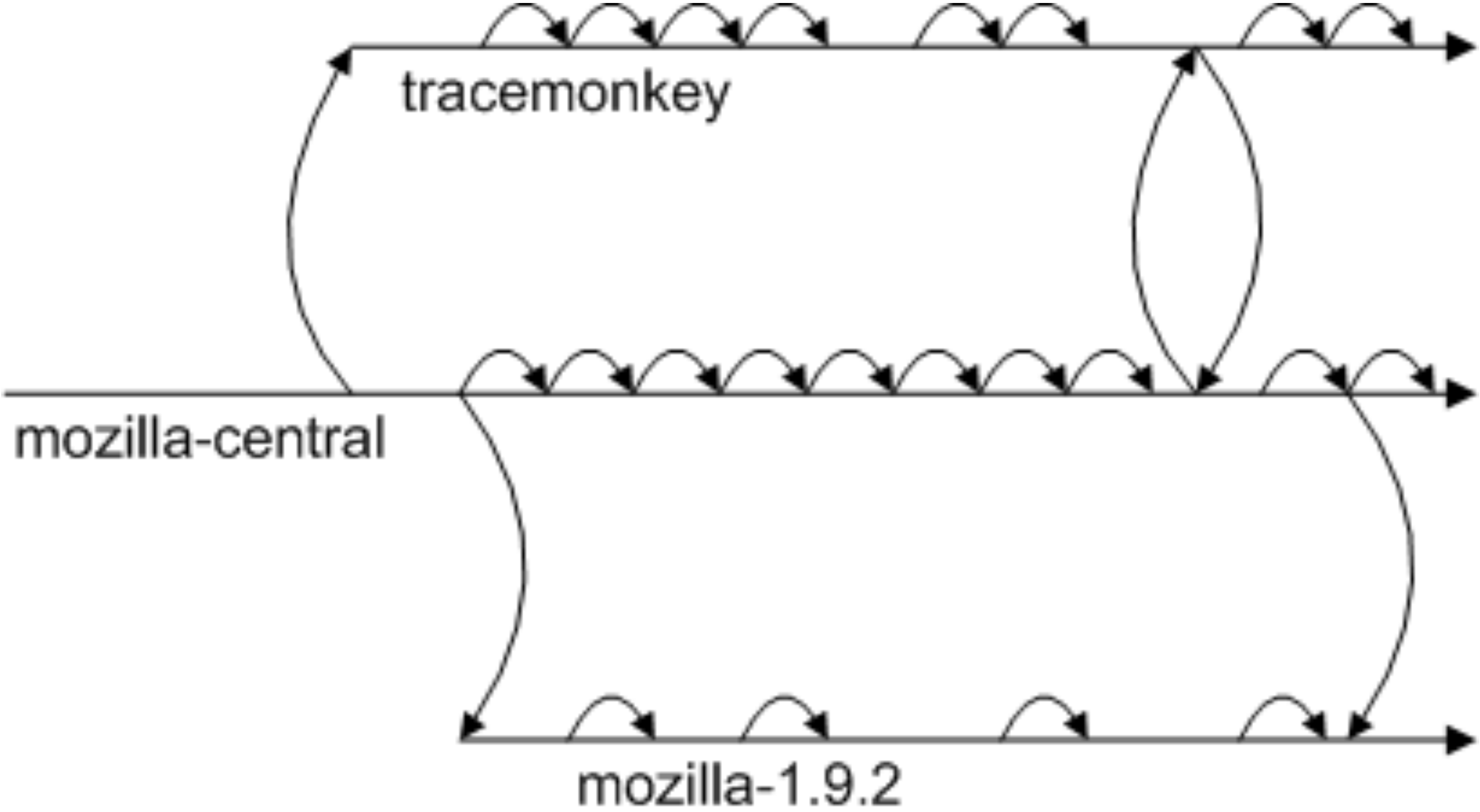
Checkin Rules | Tree Rules | Care & feeding of Tinderbox | What is all this stuff?

- **The current sheriff is Robert O'Callahan (roc)** (full schedule).
- Machines acting up? Today's RelEng contact is Aki Sasaki (aki) (full schedule).
- Running jobs, pending jobs (graph)
- Intermittent failures are tracked with a metabug and the [orange] list.
- Release Engineering Maintenance list

The tree is **APPROVAL REQUIRED** for blockers and approved patches - see the tree rules. Also note that **4 boxes are orange**, so you should not push your changes unless the failures are starred with bug numbers or fixes.

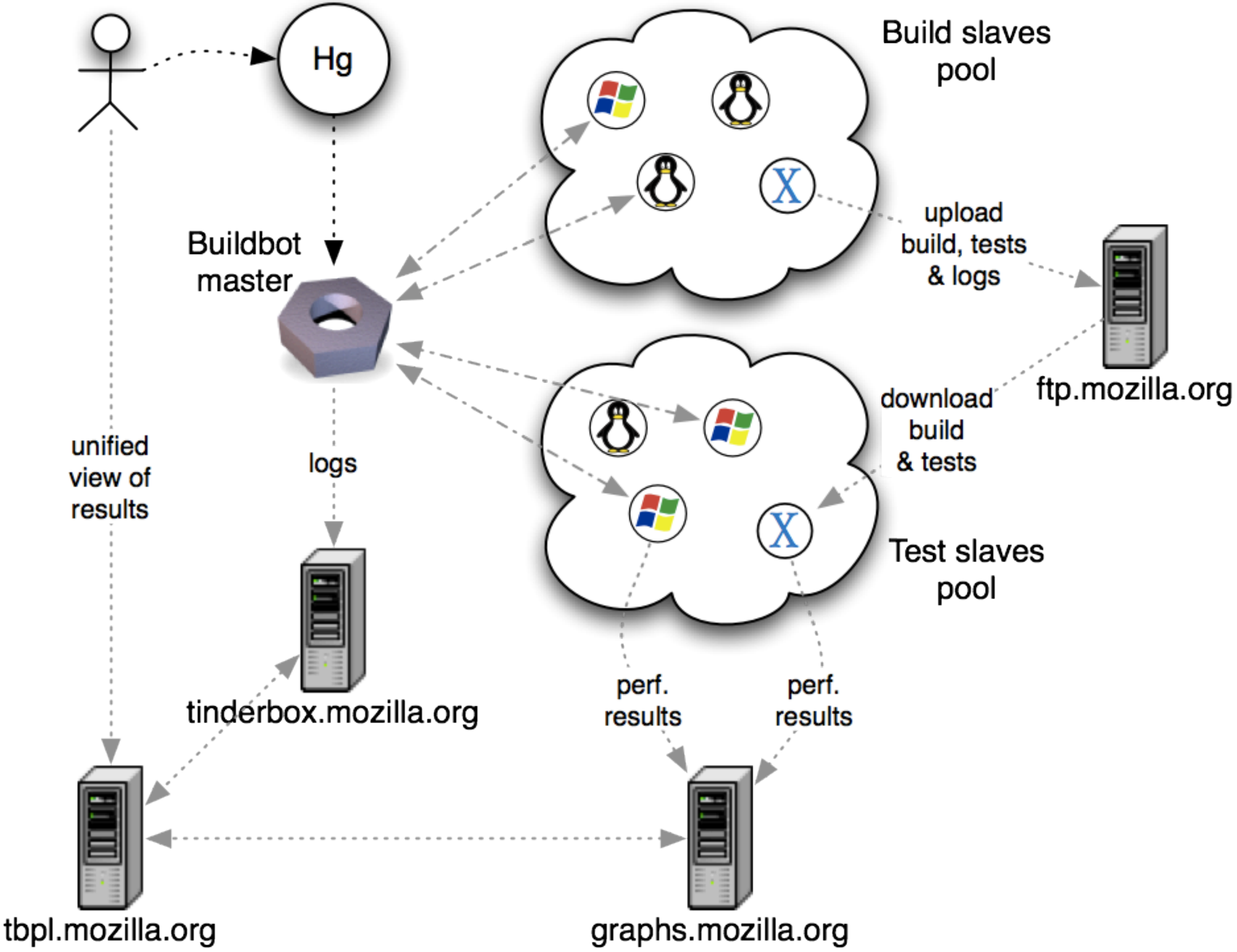
http://tinderbox.mozilla.org/showbuilds.cgi?tree=Firefox

Where and when to land changes?



Tracking work across branches

- File bugs for all work going to release branches, and get approval
- Use the correct branch flags!
- Followon work should be in linked bug.
    - Link to original bug
    - Use “regression” keyword
    - This helps release-drivers track regressions across branches, and helps prevent firedrill releases.


Hg
Build slaves pool
Buildbot master
upload build, tests & logs
ftp.mozilla.org
download build & tests
unified view of results
logs
Test slaves pool
tinderbox.mozilla.org
perf. results
perf. results
tbpl.mozilla.org
graphs.mozilla.org

Current terminology

- Tinderbox server / Tinderbox waterfall
- Tinderbox PushLog / TBPL
- Buildbot master, pool-o-slaves, pods
- TryServer
- Builds
    - depend, clobber, nightly, beta, RCs, release, debug, opt/PGO
    - L10N repacks, partner repacks
- Unittest suites
    - mochitest1-5, reftest, xpcshell, crashtest, browserchrome, a11y
- Talos/performance suites, GraphServer
    - Tp4, nochrome, jss, dirty, cold, Tsvg, Tsspider, Twinopen
- Updates, AUS
    - partial, complete, nightly, beta, release, minor, major

Current terminology

- Project branch
- Tree rules, “When is it ok to checkin?”
- “is the tree green”? (http://isthetreegreen.com)
- Release Driver
- “wait times”
- “end to end cycle time”
- “Queue collapsing”
- “Tree stable timer”

Obsolete terminology

- Tinderboxen / tinderboxes / tinderbox client
- Hourly builds
- Unittest machines
- L10n machines
- build-and-unittest

Where can I get more help?

- Developers have "Sheriff of the day"
  - Schedule is on tinderbox
  - ReleaseEngineering now have "Buildduty" person to handle escalations
- Bugs in mozilla.org:ReleaseEngineering are best.
- irc: #build
- email release@mozilla.com

Q&A