

Why a distributed SCM?

Why forking is good

Ryan Anderson
rda@google.com

Outline

- 1 Why Distributed
- 2 What is Git?
 - Impact on Kernel development
- 3 Technical Concepts
 - Getting started
 - Sharing code
 - Other stuff

Why Distributed?

- Key idea behind Free/Open Source
 - Users can be developers, if they want to.
- What this means
 - Everyone can use, change, compile, and learn from the software
- Where this falls down:
 - With centralized systems, only “committers” can commit
 - “Priviledged caste”

Digression: Features and systems

- Two major varieties
 - Centralized
 - Central, shared, repository
 - Users need permission to get commit access
 - CVS, Subversion (SVN), Perforce, ClearCase
 - Decentralized
 - No central repository
 - Every source tree can be independent (frequently)
 - Convention: A special, central, repository exists
 - Many open source projects are moving to this
 - Arch / Bazaar, Bazaar-NG, SVK , BitKeeper, Git

How does a decentralized system fix this?

- Everyone can:
 - view the full history
 - commit changes
 - create branches
 - do anything they want, basically.
 - develop without interference from other people's changes.
- Basically, there is no privileged caste
 - for SCM features
 - Still no guarantee that “well-known” people will take your changes.

Basic stuff

- Source: <http://www.kernel.org/pub/software/scm/git/>
- Packages: Look for git-core
- History: <http://www.kernel.org/git/?p=git/git.git;a=summary>
- Small projects using this:
 - Linux
 - <http://www.kernel.org/git/>
 - Wine
 - x.org
 - freedesktop.org

A brief history of Git

- 2005 April 6 - Public development begins
- 2005 April 18 - 1st multiple branch merge
- 2005 April 29 - Patches applied at 6.7/second (Kernel)
- 2005 June 16 - Linux 2.6.12 released
- 2005 July 26 - New maintainer (Junio Hamano) takes over
- 2005 Dec 21 - v1.0 released
- 2006 April 18 - v1.3.0 released.
- 2006 June 10 - v1.4.0
 - Everything since 2.6.12-rc2 tracked.
- Insanely fast development
- Very mature, already

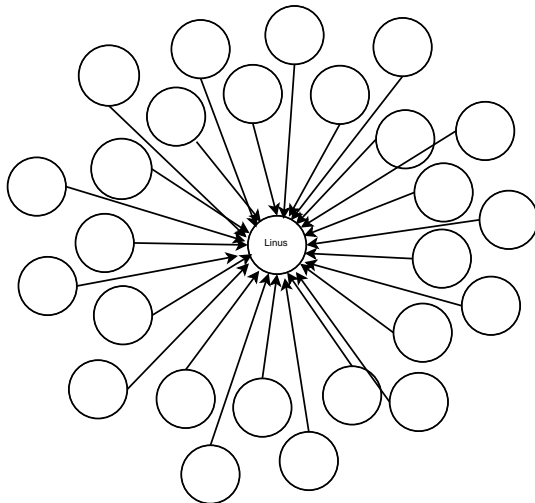
Some stats

- Since April 2005, 33256 revisions.
- 2038 individual contributors
- 73 people committing directly into various Git repositories.

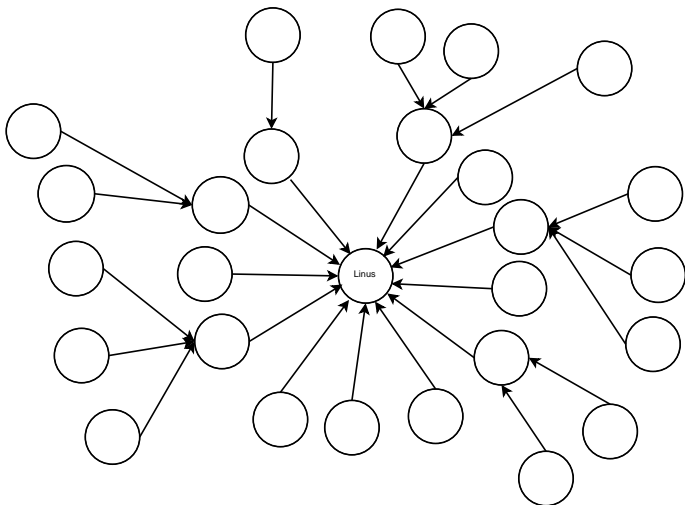
The real impact...

- Linux has never had a centralized SCM tool
- Linus scales poorly
- 2038 people emailing Linus means a lot less than 2038 people getting patches applied.

Linux before



Linux after



Top Linux contributors

Who	Committed	Authored	Ratio
Linus Torvalds & akpm	12415	1471	12%
David S. Miller	3031	766	25%
Jeff Garzik	2429	660	27%
Greg Kroah-Hartman	1879	170	9%
Russell King	1482	570	38%
Paul Mackerras	1328	362	27%
Jaroslav Kysela	1141	62	5%
James Bottomley	1049	189	18%
Mauro Carvalho Chehab	851	250	29%
Ralf Baechle	750	615	82%
Tony Luck	600	139	23%

Concepts

- Content-addressable filesystem
- 4 types of objects
 - Blob - A file
 - Tree - The state of the repository
 - Commit - The state at a given point in time
 - Contains a tree
 - 0, 1, or more parent commits
 - Author information
 - Committer information
 - Tag - GPG signed reference to a commit

Concepts

- The Index
 - Tracks the current state of the directory
 - Well, at least the state Git thinks the directory is in.

Importing a new project

- Importing
 - `tar xzf project.tar.gz`
 - `cd project`
 - `git init-db`
 - `git add .`
 - `git commit`
 - Give a commit message
- Creates a directory
 - `.git/`
 - Stores all the repository metadata

Making changes

- Editing existing files
 - `$EDITOR file1 file2 file3`
 - `git commit -a`
- Or...
 - `$EDITOR file1 file2 file3`
 - `git commit file1`
 - `git commit file3`

Making changes

- Adding a new file
 - \$EDITOR newfile
 - git add newfile
 - git commit
- Removing a file
 - git rm oldfile
 - git commit
- Moving/renaming a file
 - git mv oldfile newfile
 - git commit
 - This should work just like “mv”

Viewing changes

- What'd you do to the working directory?
 - `git status`
- Viewing the history of changes
 - Log:
 - All: `git log`
 - A range: `git log ce5b6e7..HEAD`
 - (or): `git log ce5b6e7..`
 - What was changed?
 - All: `git whatchanged -p`
 - A range: `git whatchanged -p ce5b6e7..HEAD`
 - (or): `git whatchanged -p ce5b6e7..`

Viewing changes (diffs)

- Changes to the working tree: `git diff`
- Changes to the index: `git diff HEAD`
- Changes between arbitrary things: `git diff ce5b6e7 70827b1`

Viewing changes

But this stuff is, well, blah...

Viewing changes (better)

Maybe that newfangled X11 thing can be used

- gitk
- gitweb
- gitview
- qgit

Viewing changes (better)

Maybe that newfangled X11 thing can be used

- gitk
- gitweb
- gitview
- qgit

Sharing your code

- HTTP (no special server code)
- SSH
- git-daemon
 - Bandwidth-efficient updating
 - (Not so CPU-efficient)

Getting a copy of a tree

- `git clone $URL`
- `git clone git://git.kernel.org/pub/scm/git/git.git`

Pulling others' changes

- `git pull`
- `git pull $URL`
- `git pull git://git.kernel.org/pub/scm/git/git.git`
- `git pull $REMOTE`
 - `ls .git/remotes/`
- “git pull” grabs changes and merges them into your local working tree

Sharing your changes

- Using ssh: `git push host:path/`
- For web access
 - Needs git installed
 - `chmod +x .git/hooks/post-update`
- WebDAV works

Other tools

- Extracting into patches: `git format-patch`
- Patch-bombing:
 - `git format-patch`
 - `git send-email`¹
 - (Use `man`, `patch-bomb` yourself first!)
 - Or maybe: `git imap-send`
- `git-cvsserver`
 - Yes, you can run a CVS server against a git backend.

¹Install git-email

Other tools

- git bisect
 - git bisect start
 - git bisect good \$GOODVERSION
 - git bisect bad \$BADVERSION

Questions

Summary

- Source: <http://www.kernel.org/pub/software/scm/git/>
- Why the name?
 - "I'm an egotistical bastard, so I name all my projects after myself. First Linux, now git." – Linus