PERCONA
Performance Consulting Experts

Hg

Percona Company meeting
2007, Egypt

Vadim
Tkachenko

Version Control System

- Why?
  - No need to discuss ☺
- Need to fit our needs
  - Fully distributed environment

History

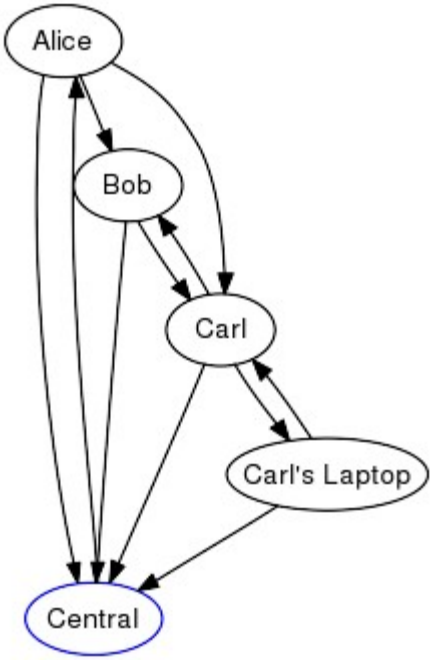
- Dinosaurs
  - RCS, SCCS
    - File level
- Historic
  - CVS (Client-server, non atomic)
- Nowadays
  - Client server – Subversion
  - Distributed

Distributed systems

- BitKeeper (MySQL, Linux Kernel in young ages)
  - Powerful, fast, but $$$$ + $
- Bzr (**Bazaar**)
  - Slow (subjective)
- Git
  - By Linus, Linux kernel in current stage
  - Fast, *nix only, 139 commands
- Hg (Mercurial)
  - Fast enough, almost easy of use ☺
- SVK (built over SVN, limited usage)

What's difference

- Subversion
  - Client – Server
  - Single Server, Full History + repository info on server
- Distributed
  - Peer-to-peer
  - Info is stored locally



A Mercurial Network

Why not SVN

- Commit from development dir goes to server
  - Unstable “main” tree
- “Branch for feature”
  - Does not work for us, each developer has its own web-dir, problematic merge between two different branches
- “Branch per developer”
  - Does not work at all
  - Subversion does not store info about last merge to main tree
  - Impossible to maintain with count of developers > 1

SVN

- Subversion is OK:
  - For single developer
  - Not active projects – Sphinx configs
  - Branches are under control (see single developer)

Why HG

- BitKeeper
  - Expensive
- Git
  - Only *nix, not for Windows users
- Bzr
  - Was slow (clear Python) on simple operations
- SVK (currently used for Boardreader.com)
  - Not clear distributed, only with link to svn
  - Monsters Perl scripts

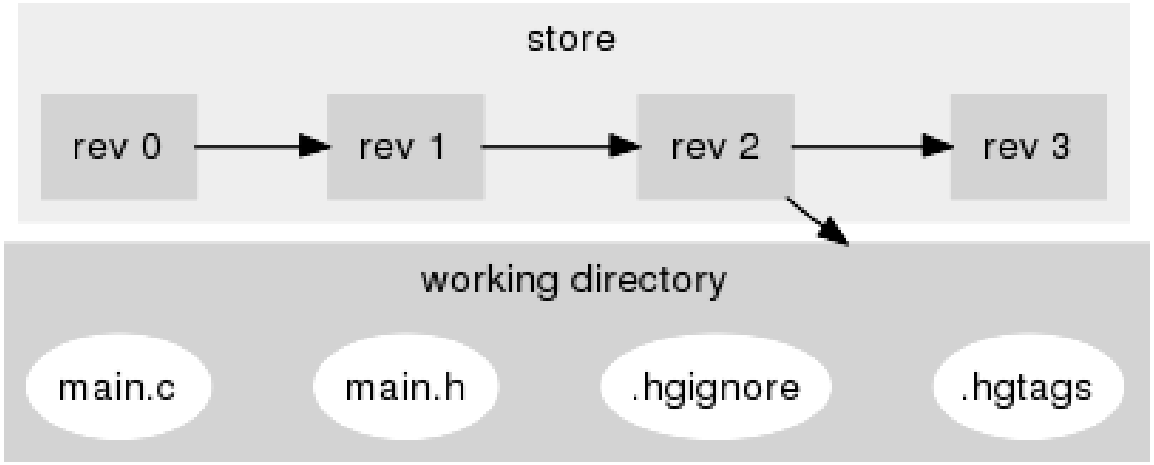
So why HG

- Fast
  - Python, but critical parts on C
- Works on ssh:// http:// file://
- Works on Linux, Windows, MacOS
  - Others you may need
- Good documentation
- Gets popular
  - Mozilla

HG terms

- Repository
- Workdir
- Changeset
- Revision
- Tip
- Head
- Merge

Repository & working dir

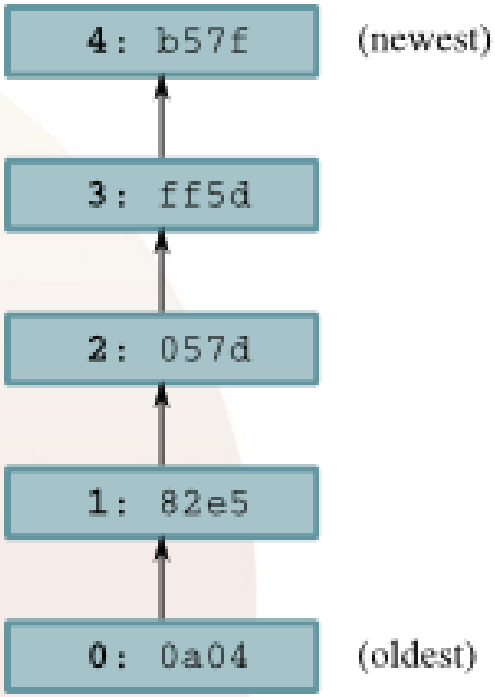


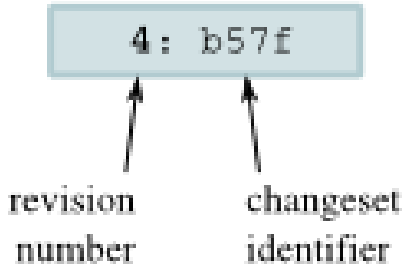
The store contains the **complete** history of the project.
NO central server

The working directory contains a copy of the project's files at a given point in time (eg rev 2), ready for editing.

Changeset & revision

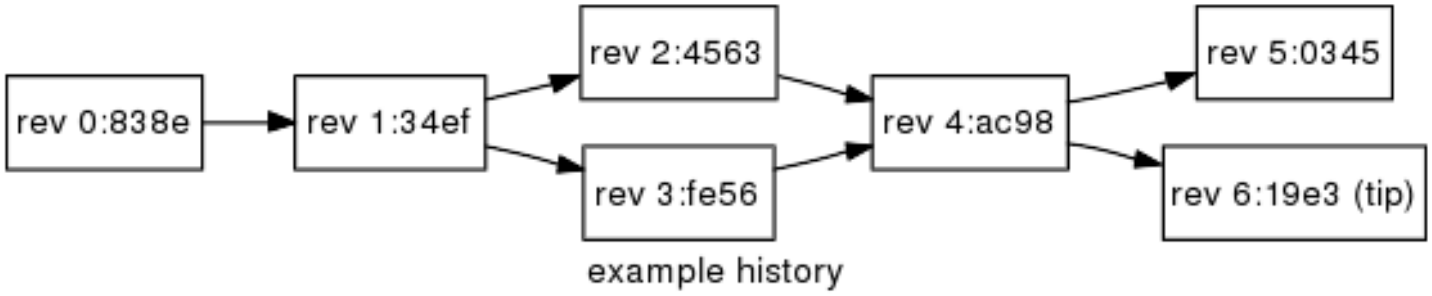
- Commited changes to multiple files are single atomic **changesets**
- Changeset identifier is REVISION
- Two marks : local number and global ID
- Local number may be different for different repositories





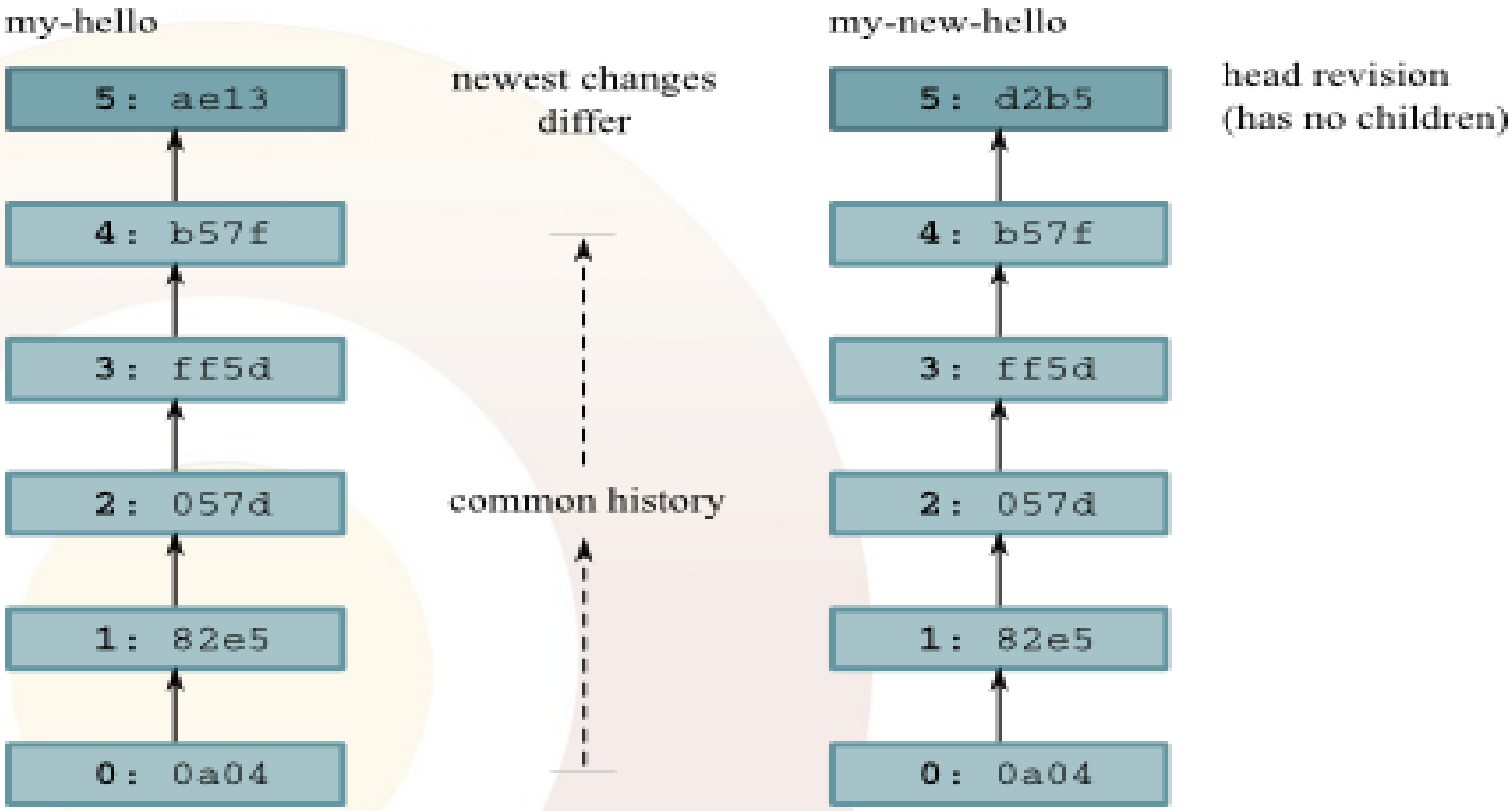
Tip & Head

- HEAD –changeset without childs
- TIP – HEAD with biggest rev N

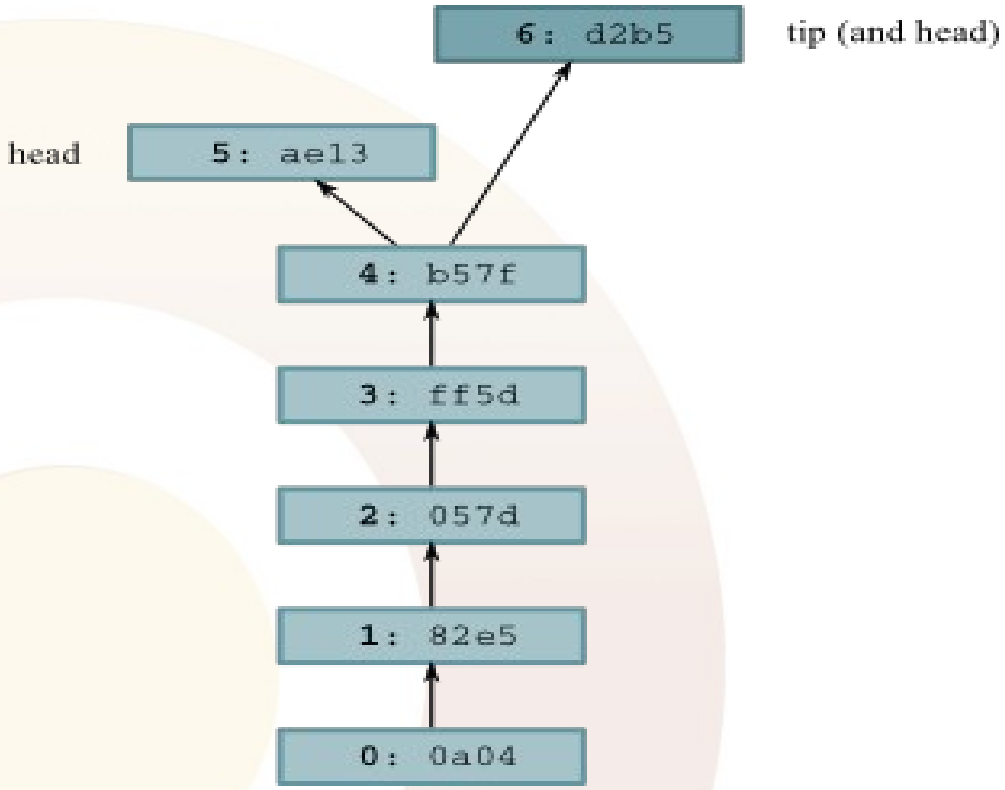


example history

Merge

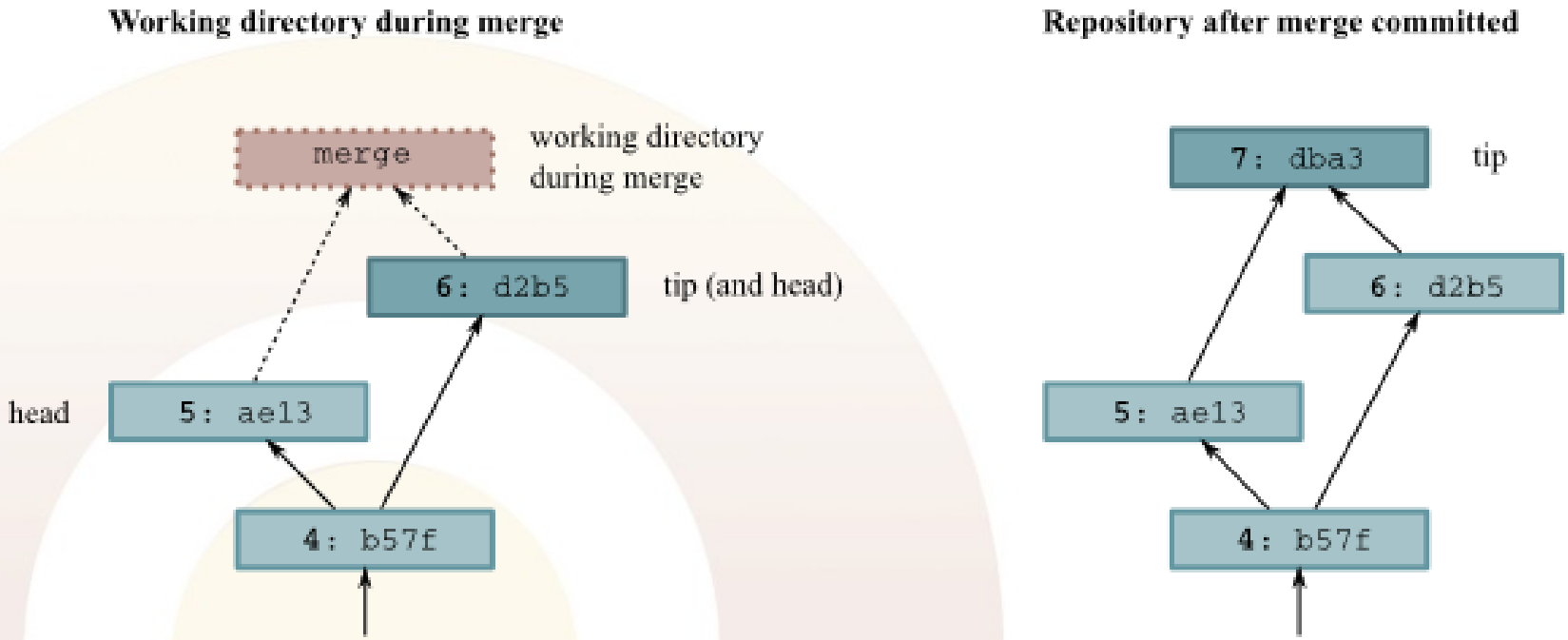


Merge



- Merge from my-hello into my-new-hello
- Local rev# in my-new-hello changed

Merge



- Merge produces new changes
- Changes need to be committed and new rev# assigned

Merge - conflicts

- Merge early – merge often
- Otherwise – painful

HG start to use

- Use built-in help ☺
  - Hg help
- Create repo
  - Hg init – create repo in localdir
  - Hg add – add all files into repo
- Clone repo
  - Hg clone
  - Hg clone ssh://server//path/to/repo
  - Hg clone http[s]://www/path
  - Hg clone /local/path

HG workflow

- Hg commit
  - Fix changes in LOCAL repo
- Hg pull – get changes from default parent dir (remember from clone)
  - Hg pull PATH (ssh / http / file)
- Hg push – push changes to default parent
  - Hg push PATH (ssh / http / file)
- Pull / Push to specified DIR allows to make changes into local repos, ommiting “main” tree

HG workflow

- Hg pull
  - Updates only STORE info, not workdir
- Hg update
- Or hg pull –u
  - Equal to hg pull; hg update

HG workflow

- Hg pull
  - Often requires merge operation
- Hg merge
- Hg commit –m “manual merge”
- Workaround
  - Hg fetch – extension, does all three-in-one

HG more commands

- Hg log – see changes
  - Hg log –rN – see changes for specified rev#
  - Hg log –rN –p – see patch of rev#
- Hg diff
  - Get patches for specified changes
- Hg annotate
  - See who and when made changes to file

HG proposed setup

- Central Repository
- Pull / push from cental repo
- Several branches per developer
  - Hg clone ssh://centalrepo /developmenttree
  - Hg clone ssh://centalrepo /hotfix
- Development tree NEVER pushed to cental repo until considered stable
  - To exchange changesets between trees use push / pull PATH

HG interesting features

- Hg branch
- Hg branch current-fix
- Hg branch hot-fix
- Switch in current dir
- Did not test it for real cases

Mercurial Queues

- **patch management problem**
- Especially interesting for us as we maintain several patches for MySQL and more incoming
- Help to maintain patches for different version of software
- Area for investigations

RTFM

- http://www.selenic.com/mercurial/wiki/
- Mercurial Wiki
- Excellent book
  - http://hgbook.red-bean.com/hgbook.html
- Questions ?