



# hg(1), or Mercurial

**Jiri Cervenka**

Solaris RPE

Sun Microsystems, Inc.

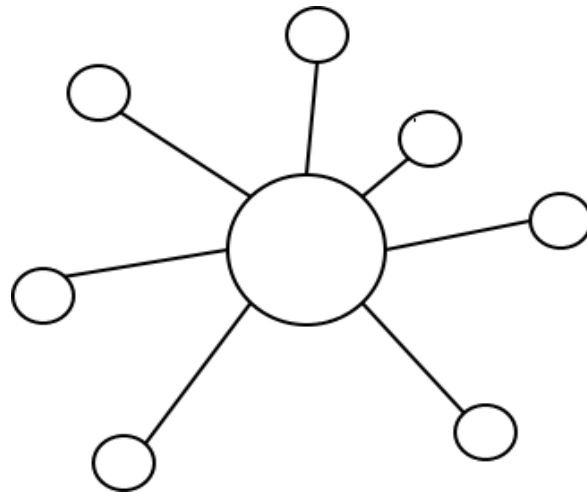
# Agenda

- A bit of context
- Mercurial
  - > Introduction
  - > DEMO
- OpenSolaris SCM

# Playground...

- Software projects with
  - > Large code base
    - # files (e.g. tens of thousands), # revisions
  - > Open development practices
    - Many (e.g. thousands) of developers
    - Physically disconnected
    - Varying levels of network access
- Demands on version control systems

# Centralized approach

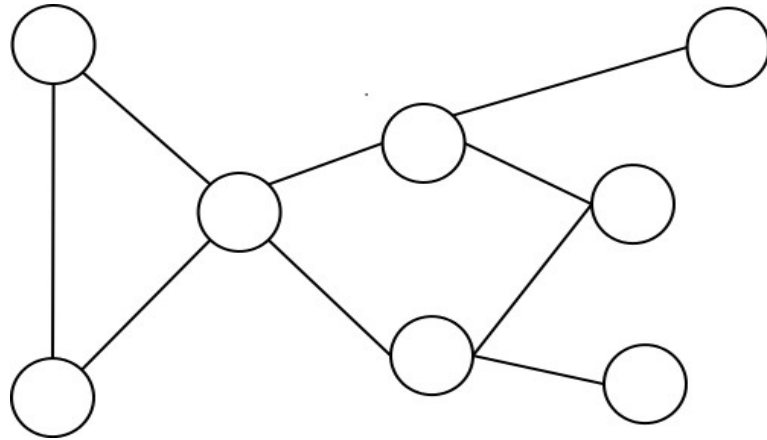


- ....does not scale that well

# Central repository

- Examples CVS, Subversion, ...
- Logic & history kept on server
- Working copy locally
- Repository state is the primary object, usually
- Add-on / ad-hoc methods to delegate responsibility for a subtree
  - > Split and sync, locking
  - > Tend to loose parts of history

# So, let's decentralize



- Repositories are 'equal' – peer to peer

# Distributed version control

- Network for trading changes / patches
- Full repository clones
  - > Work / commit locally
- Physically disconnected
  - > But single logical code stream
- Swap / pick changes, branch, merge
  - > Business as usual
- Surely, “distributed” does not necessarily mean “better”
  - > You still need policies / rules ....maybe more so
  - > Some implementations are better than others

# Examples of distributed VCS

- TeamWare
- BitKeeper
- Arch
- Monotone
- Darcs
- Bazaar NG
- Git / Cogito
- Mercurial
- ...

# Mercurial – Introduction

# Mercurial

- April 2005
  - > Linux – BitKeeper affair ends...
  - > Git 0.1 by Linus Torvalds
  - > Mercurial 0.1 by Matt Mackall
- December 2006
  - > Mercurial 0.9.2 – heading towards 1.0
  - > Used by ~ 30 projects as primary repo (e.g. Xen)
  - > Mirrors for many projects
    - e.g. FreeBSD, Linux

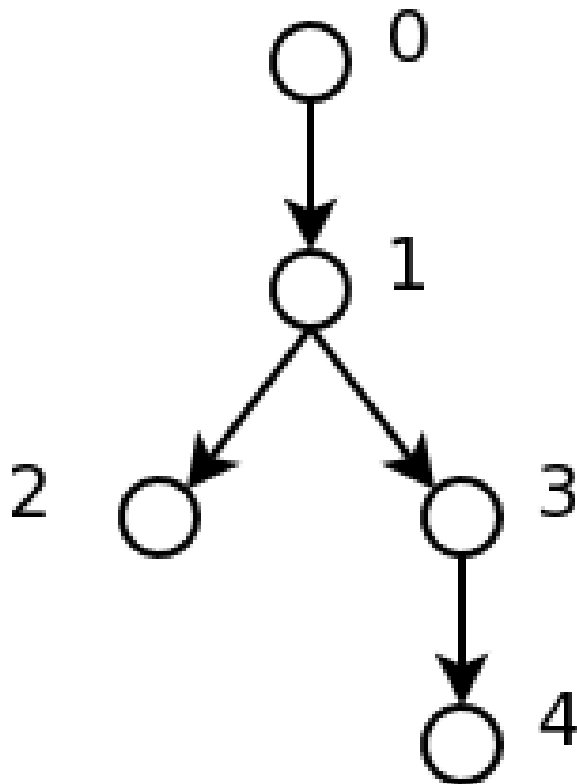
# Mercurial is (or aims to be)

- Small & fast
- Easy to use
- Robust
- Storage efficient
- Extensible
- Portable
  - > Core ~ 14k LOC of Python and ~ 700 LOC of C

# Changeset

- Main object
- Describes code change
- Code diff + some metadata
- Uniquely identified
- Knows who the parent changeset is

# Change history DAG



- Nodes = changesets

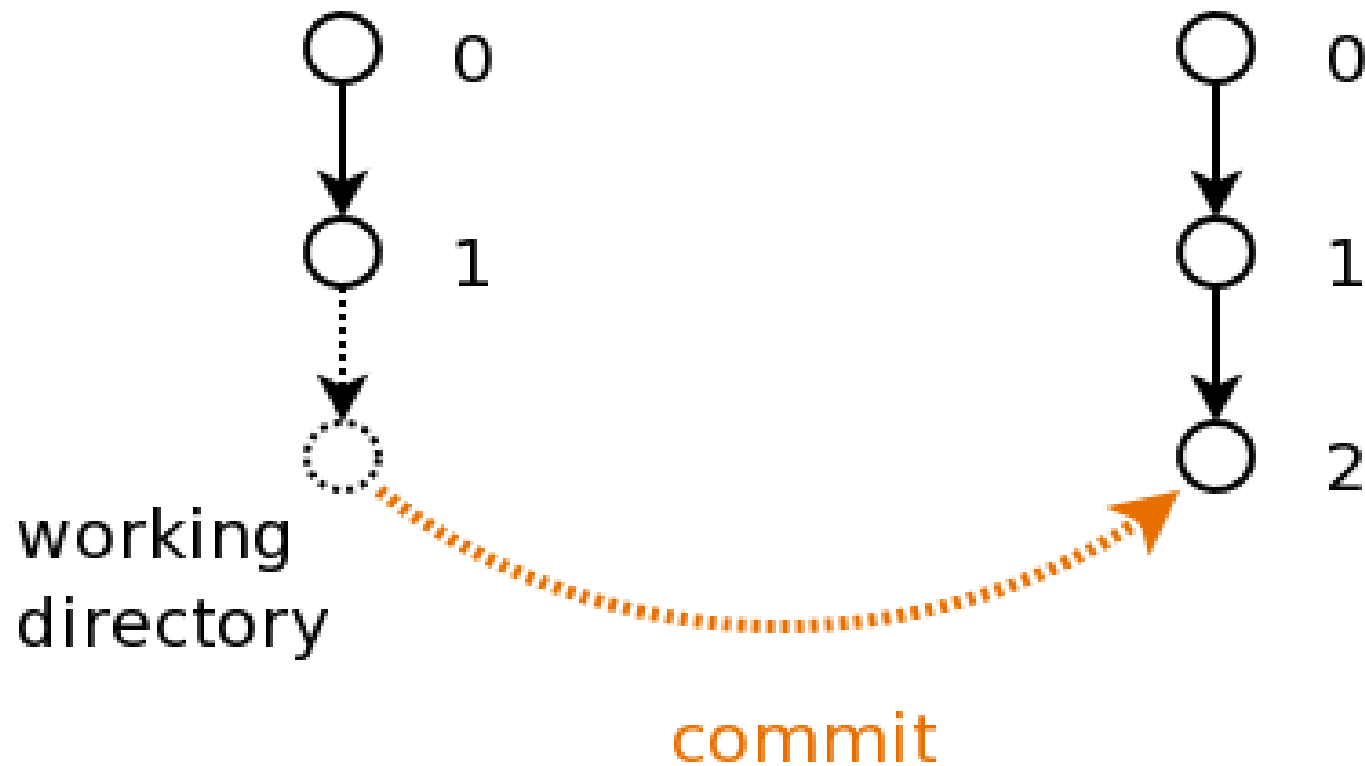
# Changeset - example

- `$ hg log -v`
- `changeset: 3:8bf309ea63726a32cb9aacf4bd960ec12c79d95a`
- `tag: tip`
- `user: crv@lacroix`
- `date: Sun Dec 10 18:02:22 2006 +0100`
- `files: hello_world.c`
- `description:`
- `fix nasty race`

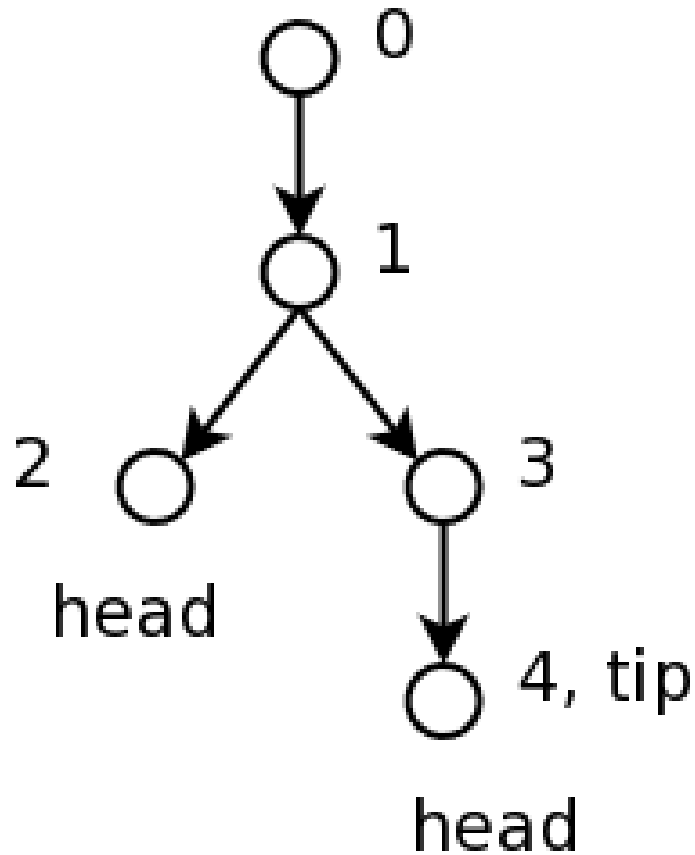
# Changeset Id

- changeset: `3:8bf309ea63726a32cb9aacf4bd960ec12c79d95a`
- “3”
  - > Local revision number
- “8bf309ea63726a32cb9aacf4b...”
  - > Globally unique changeset Id
  - > Currently SHA-1 (i.e. 160 bits)
  - > Used for
    - Identification
    - Integrity check

# Local repository - commit

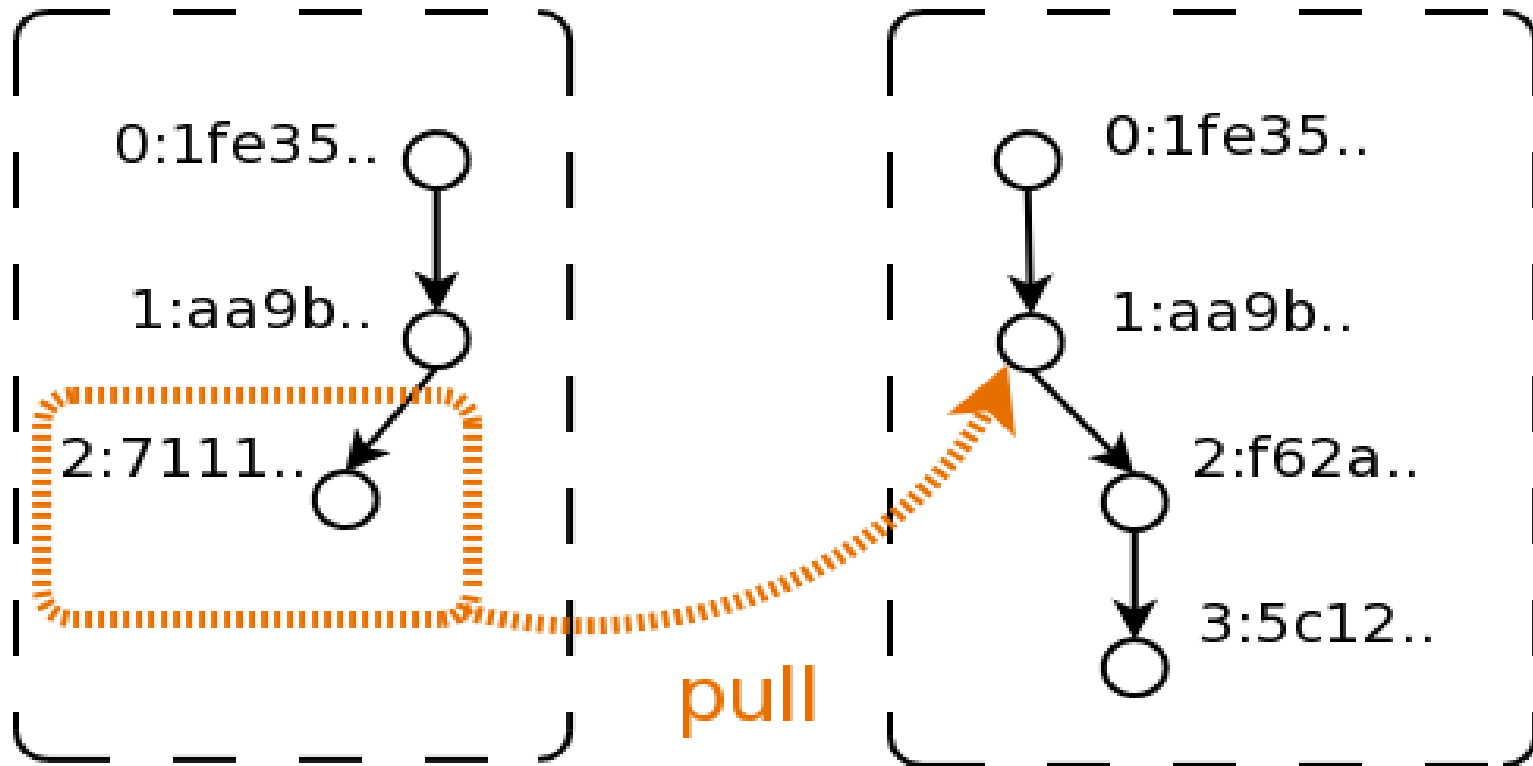


# Branching



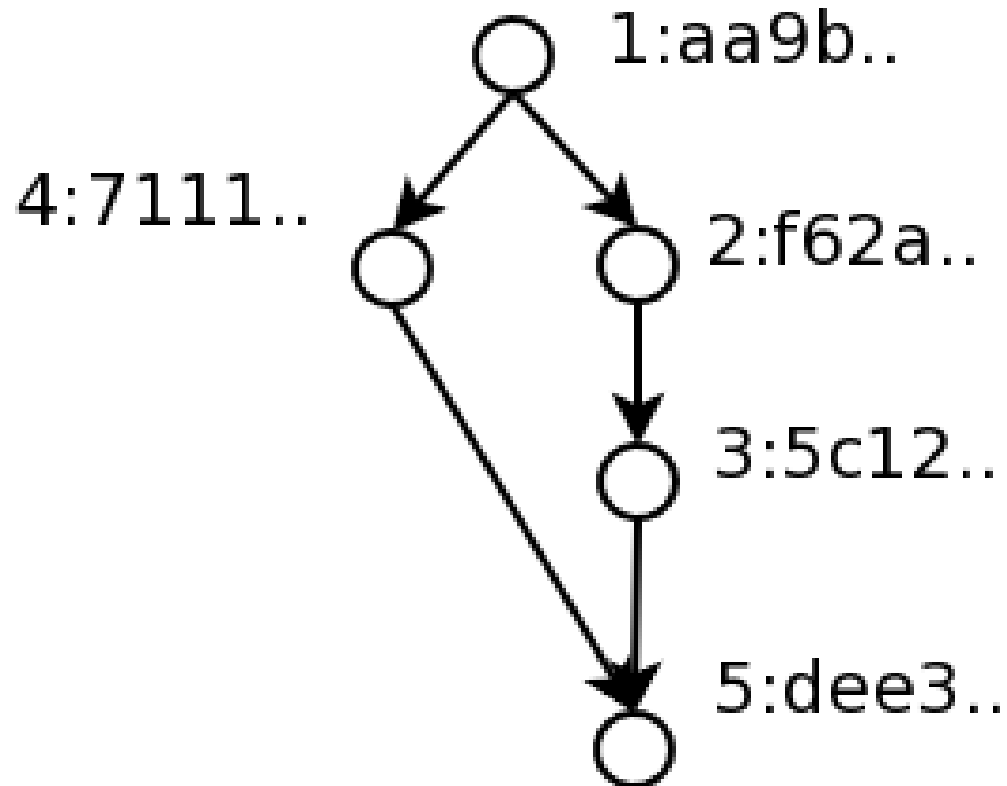
- **Head**
  - > Most recent change on a branch
- **Tip**
  - > Automatic tag for the most recently changed head

# Trading changes – pull / push



- Push is symmetric

# Merging



# Basic commands – no surprises

- Pull / push and clone
  - > Exchange changes, clone repository
  - > Local, SSH, HTTP(S)
- Add / remove / diff / status / commit
  - > Develop locally
- Log / update / merge
  - > Walk / work history tree

# Other

- Diff between repositories
  - > Incoming / outgoing
- Patches
  - > Import / export
- Built-in HTTP server
  - > Serve
- Hooks
- Extensions
  - > MQ – patch queues ala quilt(1)

# Mercurial - DEMO

# Get help

- `$ hg help`
- `<...>`
- `add`            `add the specified files on the next commit`
- `annotate`      `show changeset information per file line`
- `<...>`
- `$ hg add -h`
- `hg add [OPTION]... [FILE]...`
- `add the specified files on the next commit`
- `<...>`
- `options:`
- `-I --include`    `include names matching the given patterns`
- `-X --exclude`    `exclude names matching the given patterns`

# Create repository

- `$ mkdir projX; cd projX`
- `$ hg init`
- `$ ls -a`
- `. .. .hg`
- `$ find .`
- `.`
- `./.hg`
- `./.hg/data`

# Add files under Mercurial control

- `$ touch foo bar`
- `$ hg status`
- `? bar`
- `? foo`
- `$ hg add`
- `adding bar`
- `adding foo`
- `$ hg status`
- `A bar`
- `A foo`

# Commit locally

- `$ hg commit`
- (fill commit message in `$EDITOR`)
- `$ hg log`
- `changeset: 0:b0ff1c82aac1`
- `tag: tip`
- `user: crv@lacroix`
- `date: Sat Dec 09 17:59:53 2006 +0100`
- `summary: ProjX import`

# (Repository layout)

- `$ find .hg -type f`
- `.hg/dirstate`
- `.hg/00manifest.i`
- `.hg/00changelog.i`
- `.hg/undo.dirstate`
- `.hg/undo`
- `.hg/data/bar.i`
- `.hg/data/foo.i`

# Clone repository

- `bollo$ hg clone ssh://projX.org/hg myX`
- requesting all changes
- adding changesets
- adding manifests
- adding file changes
- added 1 changesets with 2 changes to 2 files
- 2 files updated, 0 files merged, 0 files removed, 0 files unresolved
- `bollo$ hg clone myX workY`
- 2 files updated, 0 files merged, 0 files removed, 0 files unresolved
- (local clone uses hard-links)

# Develop X locally

- `bollo$ cd myX`
- `bollo$ uname -a > foo`
- `bollo$ hg status`
- `M foo`
- `bollo$ hg diff`
- `diff -r b0ff1c82aac1 foo`
- `--- a/foo Sat Dec 09 17:59:53 2006 +0100`
- `+++ b/foo Sat Dec 09 19:02:56 2006 +0100`
- `@@ -0,0 +1,1 @@`
- `+SunOS lacroix 5.11 snv_49 i86pc i386 i86pc`
- `bollo$ hg commit -m 'you name foo'`

# Check for updates in parent repo

- `bollo$ hg paths`
- `default = ssh://projX.org/hg`
- `bollo$ hg incoming`
- `searching for changes`
- `changeset: 1:242ea3bedcfa`
- `tag: tip`
- `user: Old.Gregg@lacroix`
- `date: Sat Dec 09 20:34:23 2006 +0100`
- `summary: I name foo`
- (i.e. Parent has 1 new changeset)

# Pull new changes from parent

- `bollo$ hg pull`
- pulling from `ssh://projX.org/hg`
- searching for changes
- adding changesets
- adding manifests
- adding file changes
- added 1 changesets with 1 changes to 1 files (+1 heads)
- (run 'hg heads' to see heads, 'hg merge' to merge)
- **(Note: Working directory not updated)**
- `bollo$ hg incoming`
- searching for changes
- no changes found

# Check what we've got

- `bollo$ hg heads --style compact`
- `2[tip]:0 242ea3bedcfa 2006-12-09 20:34 +0100 Old.Gregg`
- `I name foo`
- 
- `1 883e74eec2ce 2006-12-09 19:03 +0100 bollo`
- `you name foo`
- `bollo$ hg parents --style compact`
- `1 883e74eec2ce 2006-12-09 19:03 +0100 bollo`
- `you name foo`
- (i.e. Working directory is at changeset 883e74eec2ce - local revision 1)

# Check what we've got II

- `bollo$ hg log -r 2 --patch`
- `changeset: 2:242ea3bedcfa`
- `tag: tip`
- `parent: 0:b0ff1c82aac1`
- `user: Old.Gregg@lacroix`
- `date: Sat Dec 09 20:34:23 2006 +0100`
- `summary: I name foo`
- 
- `diff -r b0ff1c82aac1 -r 242ea3bedcfa foo`
- `--- a/foo Sat Dec 09 17:59:53 2006 +0100`
- `+++ b/foo Sat Dec 09 20:34:23 2006 +0100`
- `@@ -0,0 +1,1 @@`
- `+GreggOS blacklake 0.1`

# Merge upstream in

- `bollo$ hg merge`
- (i.e. Merging rev 2 into rev 1, both children of rev 0)
- Automatic merge if possible
- Here, conflicts need manual merge
  - > `hgmerge` will look for external merging programs
  - > If not found, launch `$EDITOR` on
- `<<<<<<< /repos/scratch/myX/foo`
- `SunOS lacroix 5.11 snv_49 i86pc i386 i86pc`
- `=====`
- `GreggOS blacklake 0.1`
- `>>>>>>> /tmp/foo~other.JqoAqX`

# Merge upstream in II

- 0 files updated, 1 files merged, 0 files removed, 0 files unresolved
- (branch merge, don't forget to commit)
- **bollo\$ hg status**
- M foo
- **bollo\$ hg diff**
- `diff -r 883e74eec2ce foo`
- `--- a/foo Sat Dec 09 19:03:20 2006 +0100`
- `+++ b/foo Sat Dec 09 21:42:30 2006 +0100`
- `@@ -1,1 +1,2 @@ SunOS lacroix 5.11 snv_49 i86pc i386 i86`
- `-SunOS lacroix 5.11 snv_49 i86pc i386 i86pc`
- `+cpu0: SunOS lacroix 5.11 snv_49 i86pc i386 i86pc`
- `+cpu1: GreggOS blacklake 0.1`
- **bollo\$ hg commit -m 'two names'**

# Merge upstream in III

- bollo\$ hg parents
- changeset: 3:db684644c6c3
- tag: tip
- parent: 1:883e74eec2ce
- parent: 2:242ea3bedcfa
- user: bollo@lacroix
- date: Sat Dec 09 21:42:43 2006 +0100
- summary: two names

# What are we going to publish?

- bollo\$ hg outgoing ssh://projX.org/hg
- searching for changes
- changeset: 1:883e74eec2ce
- user: bollo@lacroix
- date: Sat Dec 09 19:03:20 2006 +0100
- summary: you name foo
- 
- changeset: 3:db684644c6c3
- tag: tip
- parent: 1:883e74eec2ce
- parent: 2:242ea3bedcfa
- user: bollo@lacroix
- date: Sat Dec 09 21:42:43 2006 +0100
- summary: two names

# Publish your changes

- `bollo$ hg push ssh://projX.org/hg`
- pushing to `../projX`
- searching for changes
- adding changesets
- adding manifests
- adding file changes
- added 2 changesets with 2 changes to 1 files

**(end of demo)**

# Mercurial - Internals

- Manifest
  - > State of repository
  - > List of (filename, version hash)
- Changeset
  - > List of files changed, metadata, manifest version hash
  - > Stored in Changelog
- Revlog – the storage backend
  - > Append only file
- Individual files histories, Manifest and Changelog each stored in a revlog

# Internals – repository layout

- `$ find . -type f`
- `./.hg/dirstate`
- `./.hg/data/hello_world.c.i`
- `./.hg/data/.hgtags.i`
- `./.hg/undo.dirstate`
- `./.hg/00changelog.i`
- `./.hg/00manifest.i`
- `./.hg/undo`
- `./hello_world.c`
- **Two types of objects: dirstate and revlogs**

# Documentation

- Man
  - > hg(1), hgmerge(1)
  - > hgrc(5), hgignore(5)
- Bryan O'Sullivan: Distributed revision control with Mercurial (draft)
  - > <http://www.red-bean.com/~bos/hgbook.pdf>
- Quick Start, Tutorial, FAQ, etc...
  - > <http://www.selenic.com/mercurial>

# Source code management for OpenSolaris

# OpenSolaris Tools Community

- <http://www.opensolaris.org/os/community/tools>
- Discussions / projects such as
  - > Build and install of OpenSolaris
  - > Sun Studio tools
  - > GCC compiler
  - > ...
  - > OpenSolaris Source Code Management
    - <http://www.opensolaris.org/os/community/tools/scm>

# DSCM selection process

- Open process
- Requirements document
  - > <http://www.opensolaris.org/os/community/tools/scm/dscmreqdoc>
- Candidates
  - > Bazaar NG, Git, Mercurial, Monotone, SVK
- Evaluation reports linked from
  - > <http://www.opensolaris.org/os/community/tools/scm/history>
- Mercurial picked in April 2006

# SCM for OpenSolaris projects

- OpenSolaris.org projects offered to use
- 1) Mercurial
  - > The primary SCM system
  - > [http://www.opensolaris.org/os/community/tools/scm/hg\\_help](http://www.opensolaris.org/os/community/tools/scm/hg_help)
- 2) Subversion
  - > For projects requiring centralized approach
  - > [http://www.opensolaris.org/os/community/tools/scm/svn\\_help](http://www.opensolaris.org/os/community/tools/scm/svn_help)

# Mercurial ON repository

- Primary
  - > `ssh://anon@hg.opensolaris.org/hg/onnv/onnv-gate`
- Mirrors
  - > `http://hg.genunix.org/on.hg`
  - > `http://hg.intevation.org/mirrors/opensolaris.org/onnv-gate`

# SCM migration project

- Migrate / convert current TeamWare based tools to work with Mercurial repository
  - > <http://opensolaris.org/os/project/scm-migration>
- Both development and gatekeeping tools
  - > nightly, webrev, wx, ...
  - > backout, buglist, clone\_update, pbcheck, update\_flagdays, ...

**Questions?**

**Thank you!**



# hg(1), or Mercurial

**Jiri Cervenka**

[jiri.cervenka@sun.com](mailto:jiri.cervenka@sun.com)