Version Control with Git and git-svn

Kate Hedstrom
ARSC, UAF

Version Control Software

- **System for managing source files**
  - **For groups of people working on the same code**
  - **When you need to get back last week's version**
- **In the past, I have used RCS, CVS, and SVN, each better than the last**
- **Git was designed for managing the Linux kernel and therefore has these goals:**
  - Fast
  - Support many, many developers
  - Distributed

Distributed?

- Every checkout gives you a copy of the whole repository
- Can compare branches, history while offline
- Can check in your changes to your local repository
- Sharing updates with others is optional

Why change from svn?

- **For our needs as ROMS users and developers, git solves some problems:**
  - Save our own changes
  - Apply patches from the repo one at a time – especially for those waiting months between updates
  - “git format-patch” and “git am” smoother than “diff” and “patch” in the face of conflicts

Getting Started With Git

- **Set up who you are:**

```
% git config --global user.name "you"
% git config --global user.email   \
                    "you@home"
```

- **Get colorful (if you want):**

```
% git config --global color.ui "auto"
```

  - **Without "--global" is for current directory only**

Start a New Repository

- **In the directory with your code:**
  - git init
  - git add .
  - git commit -m “initial message”
- **You now have a .git directory with a database of your files**
- **Revision numbers are SHA1 numbers, same for the same content**

From a Repository

- **From a git url:**
  - git clone <url>
- **Could be another local directory**
- **From an svn url:**
  - git svn clone <url>
- **Default is to suck down the entire history into the database**

Main git commands

- add – add sources to next commit
- commit – check in changes locally
- checkout – change branches
- push – send your changes to a remote site
- pull/fetch – get changes from remote site
- status - find out which files would change on commit
- diff – find out what's different between index and current sandbox
- help

Example

- **Change/add some local files**
  - git add newfile
  - git commit
- **"git add" adds files to the commit list for the next commit**
- **Can selectively add only some of your changes to make logical commits**
  - git commit -a #commits all changes

Svn example

```
% ls /tmp/cpp
branches    tags     trunk
# import whole directory
% svn import /tmp/cpp \
     file:///local/path -m "initial import"
% rm -rf /tmp/cpp
# want to be working in checked-out copy
% svn checkout \
      file:///local/path/cpp/trunk cpp
% cd cpp
[make some changes]
% svn commit
```

Git example

```
% ls /my/src/cpp
cpp.h  cpp.c   Makefile  ...
% cd /my/src/cpp
% git init
# Tell git which files to track
% git add .
% git commit
[make some changes]
% git commit -a
```

Comments on Previous

- No more need to set up the branches, tags, trunk nonsense
- No more need to delete the starting directory and checkout fresh
- Tracked files have to be explicitly added

What about Branches?

- **See the branches:**
  - git branch
- **Make a new branch:**
  - git branch <new> # copy of current
- **Switch to that new branch:**
  - git checkout <new>
- **Both in one:**
  - git branch -b <new>

Seeing History

- git log
- gitk (gui)
- git diff HEAD^
- git log HEAD^^^ or HEAD~3
- git diff b324a87 (SHA1)
- git diff --cached (between index and HEAD)

Index?

- The index is a store of what would be checked in on “commit”
- “git diff” shows difference between index and current sandbox
- “git diff HEAD” shows difference between last checked in and sandbox

Coordination

```
# on midnight
% git clone <URL> roms
% cd roms
[make some changes]
% git commit -a
% git push origin master
```

```
# on cygnus
% git clone …
% cd roms
% git pull
% make
```

- **Coordinate code on multiple systems**
- **Coordinate between multiple programmers**
- **Can be common version or different branches**

Updates

- **An update when two people have changed things can involve:**
  - No conflict because changes are in different places
  - Conflict because two different changes to the same region in the code
- **If there is a conflict this must be resolved by human intervention**
- **One option is to reset (undo)**

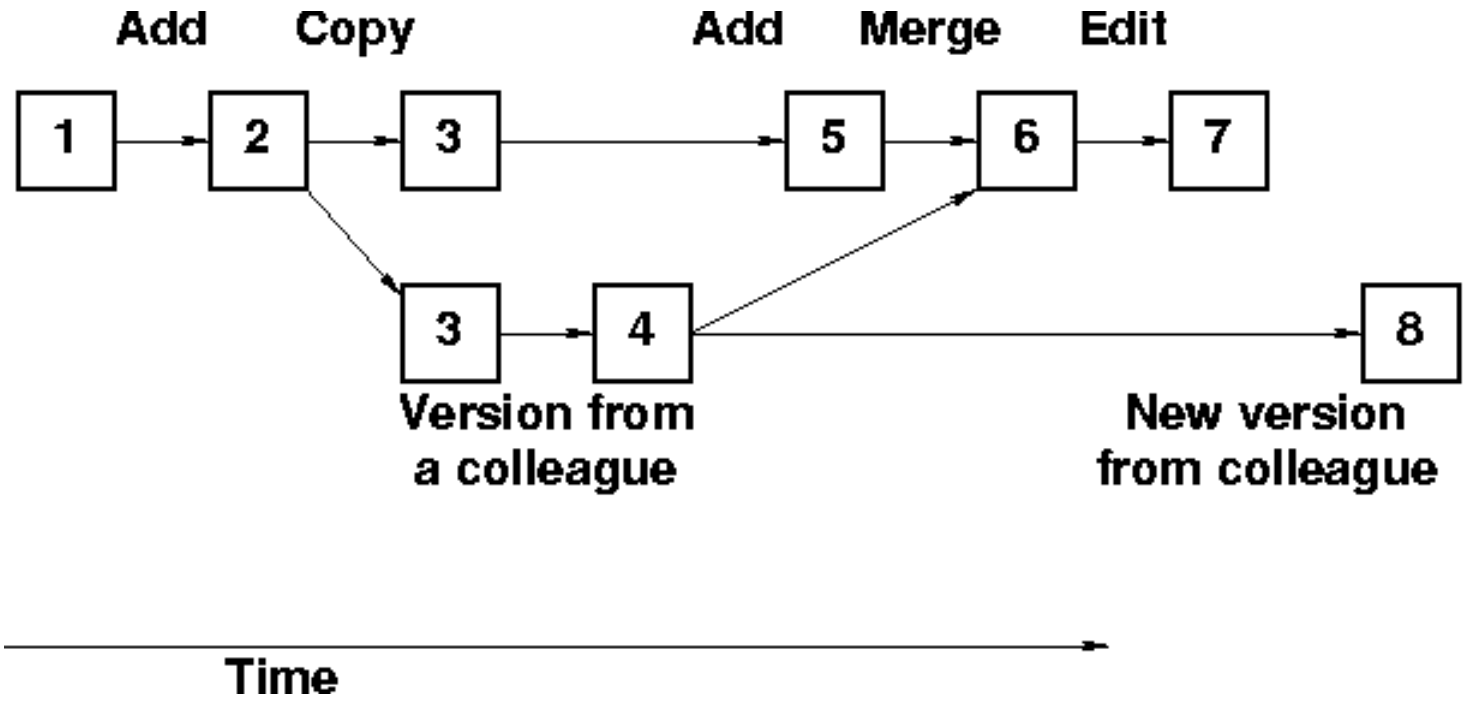
Other git commands

- **delete – no longer need a file**
- **move – rename a file or move to new location**
- **merge – merge changes from another branch**
- **cherry-pick – pick one update from some other branch**
- **remote – register a remote repo**
- **rebase – reorder the history in your local repo**

Revision Numbers

- git uses a database to store the files
- The branch has one revision number to describe that snapshot – it's a SHA1 with 40 characters
- Can see the numbers with "git log"
- Every commit creates a new revision number

Branches



- **Branch can be just for one or a few files or the whole ROMS tree**
- **Rebase can be used to put change 7 after 8**

Conflicts

- **If there is a conflict, git will let you know**
- **The merge failures will look something like:**

```
Clean code before
<<<<<<< HEAD:<file>
My code
=======
New code
>>>>>>> branch:<file>
Clean code after
```

More Conflicts

- **Once you've cleaned up the mess, tell git you're ready:**

```
git add filename
```

- **This will cause git to place the new version into the index**
- **You can instead toss your changes with:**

```
git checkout HEAD filename
```

- **Once all the files are clear (check with "git status") commit the index to the repo:**

```
git commit
```

Git Svn Handshaking

- **Not quite as robust as git alone**
  - git svn clone <url>
  - git svn clone -s -r 1043 <url>
  - git svn rebase # fetch from upstream
  - git svn dcommit # commit to upstream
  - git svn log

Drawbacks?

- **Best with one project per repository (roms, plotting, matlab tools all separate entities)**
- **More rope to hang yourself... and I don't know how to fix a sandbox that's in an odd state**

My Insane Repo Collection

- Bare repository on cygnus (Linux workstation)
- Cloned to each supercomputer via ssh
- Cloned to Enrique's system via ssh
- git-svn only working on Mac laptop
- Mac has git-svn directory, plus clone of cygnus repo, probably soon NCAR git-svn repo

My Branches

- Copy of the svn code
- Copy of a very similar code in the bare cygnus repo
- Copy of the fish branch
- Any other thing I'm working on temporarily

Learn more

- **Version Control with Git, by Jon Loeliger, 2009, O'Reilly**
- **Online at http://git-scm.com/documentation - there are even videos**
- **git help**

Work Along?

- **git config --global user.name “me”**
- **git config --global user.email “me@work”**
- **git config --global color.ui “auto”**
- **Find some code to track...**

In Code Directory

- **git init**
- **git add . # or git add *.f**
- **git commit # -m “message**
- **make a change….**
- **git status**
- **git commit -a**

Ignoring Files

- **Edit .gitignore**
- **git add .gitignore**
- **git diff**
- **git commit**
- **gitk**
- **Can look at .git/config**

Branches

- **git branch hotter**
- **git checkout hotter**
- **Edit some file**
- **git commit -a**
- **git checkout master**
- **Edit same file**
- **git merge hotter**

Conflict? then Clone

- Fix conflict
- git add file
- git commit
- git branch -D hotter
- cd ..
- git clone dir1 dir2
- Check dir2/.git/config