

Distributed Version Control Systems

Ákos Frohner
CERN, IT-DM

- **Blurring the definition of “central” repository**
 - All repositories contain the full change history
 - One can pull or push changesets (sequence of commits)
- **Various workflows**
 - traditional VCS by a dedicated central repository
 - peer interaction between developers
 - Hierarchical workflow (e.g. porting → developer → integrator)
- **Examples**
 - git <http://git-scm.org>
Linux, Perl, GNOME, Samba, X.org, Qt
 - Mercurial <http://www.selenic.com/mercurial>
Mozilla, OpenJDK, OpenSolaris, Xen, Python
 - Bazaar <http://bazaar-vcs.org/>
MySQL, Ubuntu

```
$ cvs co -d LCG-DM-cvs LCG-DM
```

```
# about 20 seconds
```

```
$ du -sh LCG-DM-cvs           # current HEAD
```

```
14M  LCG-DM-cvs
```

```
$ git clone git://lxtank02.cern.ch/LCG-DM
```

```
# about 11 seconds
```

```
$ du -sh LCG-DM/.git         # all history back to 1999
```

```
4.6M  LCG-DM/.git
```

```
$ git branch quotas LCG-DM_R_1_7_2_4  
# switching to local branch  
$ git checkout quotas  
$ $EDITOR VERSION dpm/dpm_procreq.c  
$ git status  
# different comments for each  
$ git commit VERSION  
$ git commit dpm/dpm_procreq.c  
$ git diff LCG-DM_R_1_7_2_4  
$ git diff LCG-DM_R_1_6_11_4
```

patches can be sent to upstream by mail:

```
$ git format-patch LCG-DM_R_1_7_2_4
```

commit changes to CVS*:

```
$ git cherry LCG-DM_R_1_7_2_4 | sed -n 's/^+ //p' \  
| xargs git cvsexportcommit -v -w ../LCG-DM-cvs -p
```

push to upstream via git:

```
$ git push quotas ssh://lxtank02/LCG-DM
```

* the actual commit will not happen, unless you figure out the right flag

- **Browsing history is fast (local)**
- **Easy to branch, easy to merge**
- **Developer/porter/integrator can have local branches**
 - Without commit right to central repository
 - Experimental ideas
 - Long, independent work
- **Local branches**
 - Local history, revision control, commit comments
 - Can be updated from upstream HEAD
 - Can be sent to upstream to inclusion

- **Backup of local copy is the developer's responsibility**
- **Central build requires a public repository**
 - There needs to be a central service, where the official branches are published
- **“Complex” systems: we will not change to it at once**

Hybrid mode is probably the way to go:

- **“Official” repository is in VCS (CVS or SVN)**
- **“Private” repositories are in DVCS (git or Mercurial)**