TehLUG

Client/Server Repository Model SCM Softwares

(Amir Mohammad Saied)

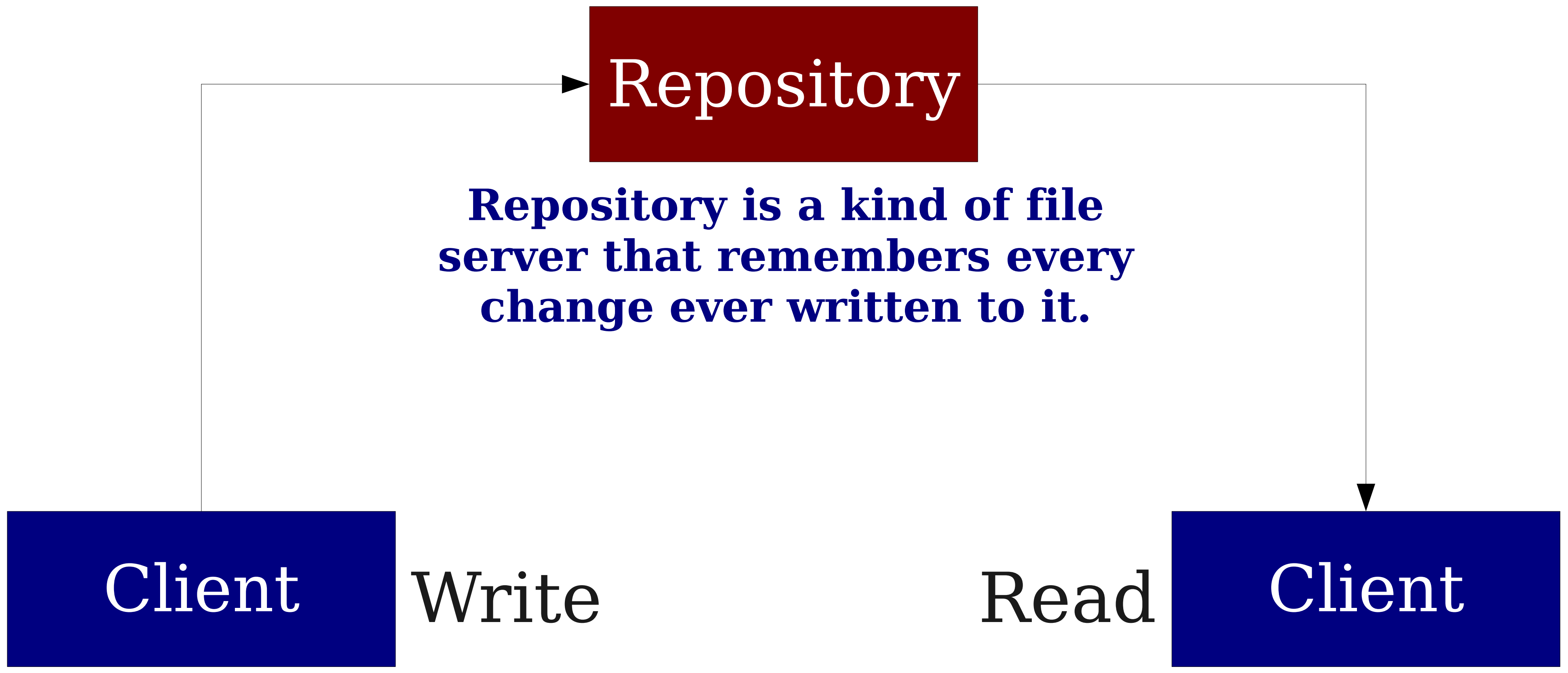
November 7, 2007

Source Code Management
History

- 1972: Original diff algorithm (*Bell Labs*)
- 1972: SCCS (*Bell Labs*)
- Late 1970s: Revision Control System, RCS (*Walter Tichy*)
- Early 1980s: patch (*Larry Wall*)
- 1986: Concurrent Version System, CVS
- Early 2000s: DVCS, BitKeeper and GNU arch.

Source Code Management
Common Vocabulary

- Trunk (Baseline or Mainline)
- Branch
- Check-out (co), Check-in (commit, ci)
- Change, Change list
- Import, Export
- Head
- Repository
- Tag (label)
- Update (sync)



Repository is a kind of file server that remembers every change ever written to it.

Client sees the latest version of tree, but also has the ability to view previous states.

CVS
History

- *Dick Grune*, June 23, 1986
- Module:
  A single project managed by CVS is called a module. Modules stored in repository.
- $CVSROOT
  `:protocol:user@host:path`
- $CVS_RSH
  `ssh` (It's `rsh` by default)
- Don't have $CVSROOT?
  `cvs -d repository`

CVS
Check it out!

```
cvs -z3 -d:pserver:amir@cvs.php.net:/repository checkout -P pear/Validate
```

CVS/

Entries

Lists files and subdirectories this particular CVS co knows about.

Repository

Stores path to the corresponding directory in the repository.

Root

Contains the path to the repository (Overwrites `$CVSROOT`*).*

CVS
More Check-outs

```
cvs checkout -r1.15 src/foo/bar.c
cvs checkout -rRELEASE_0_5 pear/Net_SmartIRC
cvs checkout -D "1 fortnight ago" ls
```

CVS
Update your wc

- `cvs update`
- `cvs update -d`
- **U** *updated successfully*
- **A** *added but not yet committed*
- **R** *removed but not yet committed*
- **M** *modified in your working copy*
- **C** *conflict, requires human intervention*
- **?** *file is not in repository*

CVS
Let the world know what you did

- `cvs commit`
- `$CVSEDITOR` or `$EDITOR`
- Meaningful change notes (please!)
- `cvs add filename`
- `cvs remove filename`
- First remove it from fs, no actual remove though, just a move to Attic subdir in repo.
- CVS won't remove Directories. That's possible only by changing the repository.

Feature	**CVS**	**Subversion**
Repo. files format	RCS Files	BDB, FSFS
Speed	Slow	Faster. More operations for offline mode, but full backup of all work files.
Metadata	Store in file.	Attach bunch of possible named attributes.
FileTypes	Intended for text data storage, so binary stuff (and unicode) requires adjustments.	Does it with no further instructions.
Rollback	Rollback any commits	No rollback, just set a previous good state as the last state (bad commits will remain there).
Transactions	"All or Nothing?" nah, thanks!	Yep, it's called Atomic.

Subversion
History

- *June 2000,* Coding begins
- *August 2001,* Subversion becomes self-hosting
- *2002,* 1st release
- *February 2004,* Release 1.0.0
- *2007,* Release 1.4.5

Subversion
Versus CVS

- Most current CVS features.
- Directories, renames, and file meta-data are versioned.
- Commits are truly atomic (like a database transaction, technically there's no difference).
- Standalone server option.
- Versioning of symbolic links.
- Efficient handling of binary files.
- Parseable output.
- Localized messages.

Subversion
Network Protocols

- `file://`
  Direct repository access to local or network drive.
- `http://`
  Access via WebDAV protocol to Subversion aware Apache server.
- `https://`
  Same as http:// but with SSL encryption.
- `svn://`
  Unauthenticated TCP/IP access via custom protocol to an svnserve server.
- `svn+ssh://`
  Same as svn:// but Authenticated and Encrypted.

Subversion
How to check-out

$$\texttt{svn checkout } \underbrace{\texttt{svn://svn.osp.ir/svnroot/persism/trunk}}_{\text{Repository URL}} \ \underbrace{\texttt{persism}}_{\text{Working Copy}}$$

Subversion
Status of working copy

- `svn status`
- **A** *added but not yet committed.*
- **C** *conflict, require human intervention.*
- **D** *deleted but not yet committed.*
- **M** *modified.*
- **R** *replaced but not yet committed (object is first deleted but another object of the same name is added).*
- **X** *external.*
- **?** *object is not under control.*
- **~** *the kind of object in wd and repo is different.*
- **I** *the object is not under version-control but Subversion is told to ignore it.*
- **!** *object is under version-control but is missing.*
- `svn update` *will refetch 'em.*
- `svn revert` <*file*> *will restore a missing file.*

Subversion
Send changes from working copy

- `svn commit`
- `-m "- Log Message"`

Subversion
The log messages

- `svn log file(s)`
  one or more files separated by spaces.
- `svn log -r 5:10`
  logs 5 through 10 in chronological order.
- `svn log -r 10:5`

logs 5 through 10 in reverse order.

- `svn log -r 7`
  log for revision 7.

Subversion
diff

- `svn diff`
  You'll see the difference between your working-copy and the cached copy in .svn.
- `svn diff -r 4:5 foo.bar`
  Two revisions would directly compared.
- `svn diff -r {date}`
  Revision at start of the date

Subversion
Revisions magic words!

- HEAD
  The latest revision in the repository.
- BASE
  The revision number of object in working copy.
- COMMITTED
  The most recent revision <= BASE which an item changed.
- PREV
  COMMITTED – 1

```
svn diff -r PREV:COMMITTED foo.bar
```

Subversion
The Ultimate Resource (Free!)

http://svnbook.red-bean.com/