Version Control Systems(VCS)

Kiran Vasudev[1]

[1]Hochschule Bonn-Rhein-Sieg

March 13, 2018

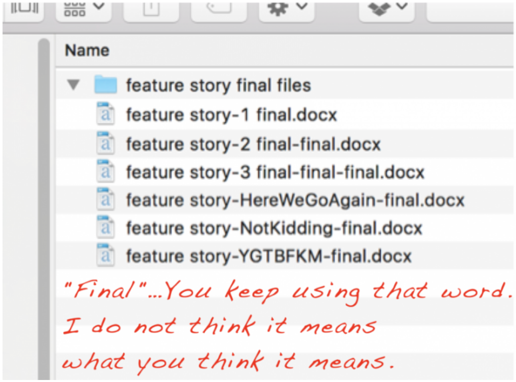
Outline

What is it?

- A method to manage file changes over time. These files are stored in a central location.
- A backup when things go wrong.
- A software that helps team members collaborate with minimum disruptions.

Why should we use it ?

- A complete long-term change history of every file.
- Branching and merging
- Ability to trace each change made
- Allows multiple developers to work simultaneously



Name

feature story final files

feature story-1 final.docx

feature story-2 final-final.docx

feature story-3 final-final-final.docx

feature story-HereWeGoAgain-final.docx

feature story-NotKidding-final.docx

feature story-YGTBFKM-final.docx

"Final"...You keep using that word. I do not think it means what you think it means.

Figure: Why we need Version Control[1]

Types of VCS

Centralized VCS

- Centralized VCS keeps the history of changes on a central server.
- Designed with the intent that there is only "One True Source"
- If you want to make a copy of your data, you have to copy/paste it, literally.
- Sourceforge.net uses this type of versioning.
- **Example:** SVN (Subversion)

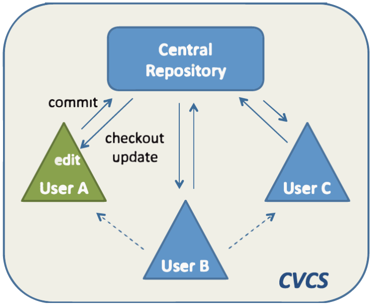


Figure: Centralized Version System [2]

Types of VCS

Decentralized VCS

- In distributed VCS, everyone has a local copy of the entire works history
- Each repository is as good as the other ie. each repository acts as a "True Source".
- Robust to central server crashes
- Your local copy is a repository, and you can commit to it and get all benefits of source control - Offline Source Control.
- Mozilla Firefox uses this kind of versioning.
- **Example:** Almost all open source projects use this type of version control.

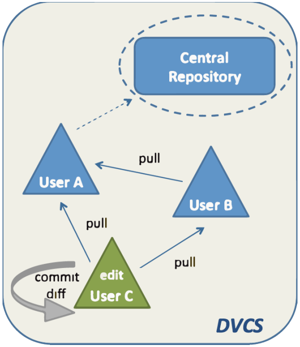


Figure: Distributed Version System [2]

Pros and Cons I

Git vs SVN

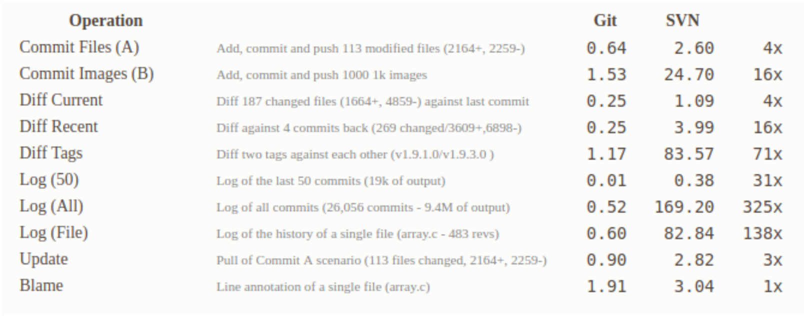
Pros of using SVN

- Subversion has better GUIs than Git
- Working through versions in Git is tougher than SVN. Git uses SHA-1 hashes while SVN uses sequential revision numbers.

Pros of using Git

- Git is much faster than SVN as they do not have to keep communicating with a central server when changes are made.
- Very secure. Every file and commit is checksummed.
- The Git branching model is much better than any other VCS'.
- Git repository file formats are simple and hence repair and corruption is rare. Space requirements are small
- As it is a DVCS, scalability is not an issue.

Pros and Cons II
Git vs SVN

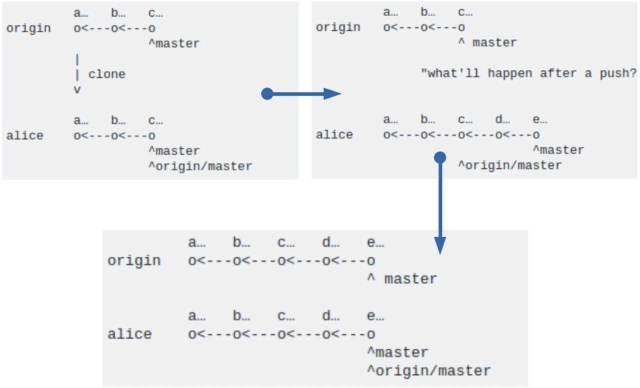


Operation		Git	SVN	
Commit Files (A)	Add, commit and push 113 modified files (2164+, 2259-)	0.64	2.60	4x
Commit Images (B)	Add, commit and push 1000 1k images	1.53	24.70	16x
Diff Current	Diff 187 changed files (1664+, 4859-) against last commit	0.25	1.09	4x
Diff Recent	Diff against 4 commits back (269 changed/3609+,6898-)	0.25	3.99	16x
Diff Tags	Diff two tags against each other (v1.9.1.0/v1.9.3.0)	1.17	83.57	71x
Log (50)	Log of the last 50 commits (19k of output)	0.01	0.38	31x
Log (All)	Log of all commits (26,056 commits - 9.4M of output)	0.52	169.20	325x
Log (File)	Log of the history of a single file (array.c - 483 revs)	0.60	82.84	138x
Update	Pull of Commit A scenario (113 files changed, 2164+, 2259-)	0.90	2.82	3x
Blame	Line annotation of a single file (array.c)	1.91	3.04	1x

Figure: Comparison of Git and SVN speeds[3]

Branching I

Git vs SVN



```
         a...  b...  c...
origin   o<---o<---o
                   ^master
         |
         | clone
         v
         a...  b...  c...
alice    o<---o<---o
                   ^master
                   ^origin/master
```

```
         a...  b...  c...
origin   o<---o<---o
                   ^ master

               "what'll happen after a push?

         a...  b...  c...  d...  e...
alice    o<---o<---o<---o<---o
                             ^master
                   ^origin/master
```

```
         a...  b...  c...  d...  e...
origin   o<---o<---o<---o<---o
                             ^ master

         a...  b...  c...  d...  e...
alice    o<---o<---o<---o<---o
                             ^master
                             ^origin/master
```

Figure: Branching in Git[4]

Branching II

Git vs SVN

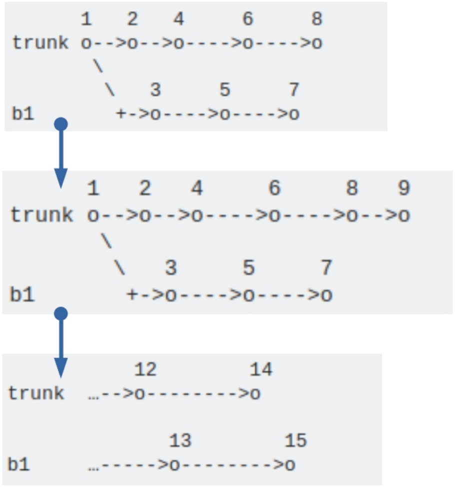


Figure: Branching in SVN[4]

Take home message

- There is not much difference between the most popular VCSs. It ultimately comes down to the use case.
- If there is a need for "Offline Source Control" with really good branching features, Git's fantastic. Open source projects are worked on with the help of Git.
- If there is a need to have a strictly centralized Source Control that is simple and has excellent tooling (at least on Windows), choose SVN.

References

- Naming Conventions
- CVCS and DVCS
- Comparison between Git and SVN speeds
- Branching SVN vs Git