assignment01 - Technical Report about Git

20141718 Lee, Jisu

September 17, 2018

Contents

1 Introduction

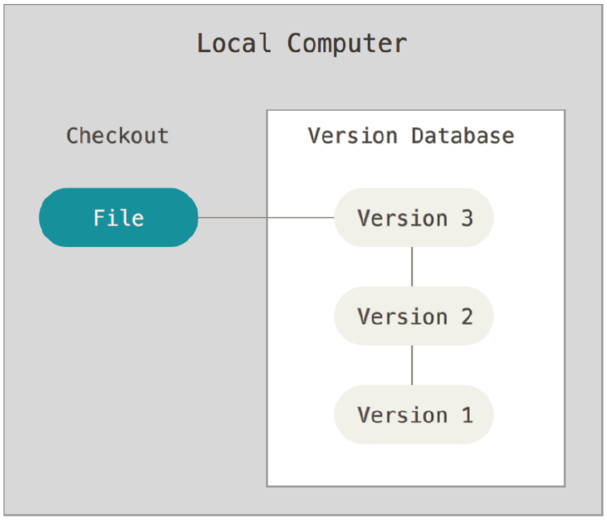
Git is a widely used version control system. In this report, I am going to show how Git has come out into the world among many version control systems and its basic features. I will also introduce github, and advanced git features.

2 What is Git?

2.1 Version Control System

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later. Version control system(VCS) allows you to revert selected files back to a previous state, revert the entire project back to a previous state, compare changes over time, see who last modified something that might be causing a problem, who introduced an issue and when, and more. In a nutshell, when you screw things up, you can easily recover files using VCS, with a very little overhead.[1]

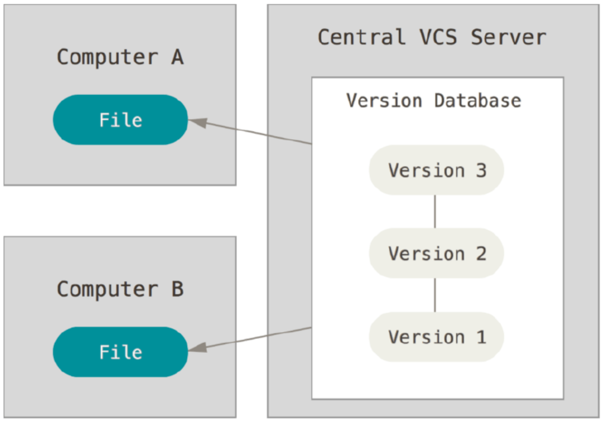
2.1.1 Initial VCS - local VCS



At this system, many versions of a file is stored at local database and a file is re-created by adding up those versions.

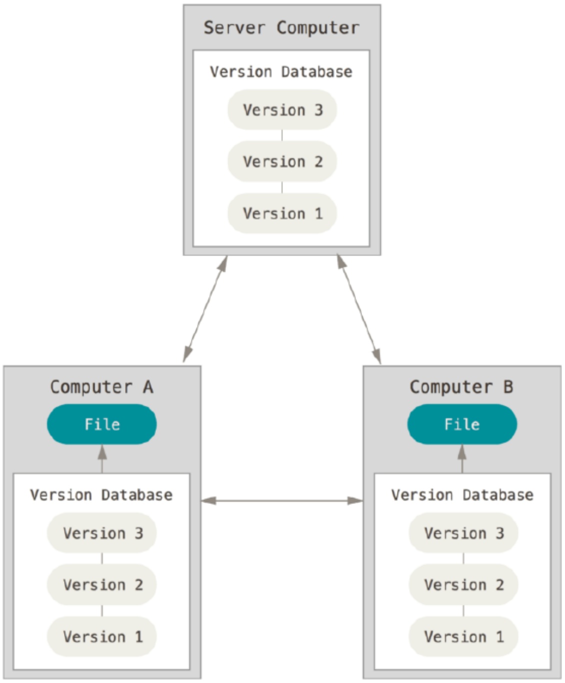
2.1.2 centralized VCS

With local VCS, it is hard to collaborate developers in other systems. Centralized VCS deals with this problem, by providing a single server that contains all the versioned files, and a number of clients that check out files from that central place.[1]



2.1.3 Distributed VCS

However, centralized VCS has some serious problems. The most obvious one is that when server goes down, nobody can collaborate at all. To deal with this problem, distributed VCS stepped in. Clients not just have the snapshot of the files; they fully mirror the repository, so that if any server dies, any of the client repositories can be copied back up to the server to restore it.[1]

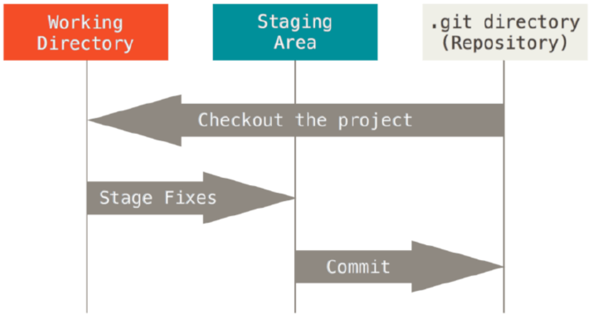


And Git, is distributed VCS.

3 Git in a Nutshell

3.1 The Three States

Git has three main states that files can reside in: committed, modified, and staged:



- Committed means that the data is safely stored in your local database.
- Modified means that you have changed the file but have not committed it to your database yet.
- Staged means that you have marked a modified file in its current version to go into your next commit snapshot.[1]

3.2 Basic Functions

3.2.1 creating git repository

You typically obtain a Git repository in one of two ways:
1. You can take a local directory that is currently not under version control, and turn it into a Git repository, or
2. You can clone an existing Git repository from elsewhere.
In either case, you end up with a Git repository on your local machine, ready for work.[1]
At This report, I will only introduce first way of creating git repository.

Initializing a Repository in an Existing Directory
First, go to the project's directory you want to initialize with Git.
And type:

```
git init
```

3.2.2 recording changes to the repository

If you want to record a state of the proejct, commit the project. You can use git status to check if there's any files to commit.

```
git status
```

There are two types of files: tracked and untracked.
If you want to commit some untracked files, use git add ¡file¿ to track the file.

```
git add <file>
```

Then you can commit files using git commit command.

```
git commit
```

3.2.3 view the commit history

To view the commit history, use git log.

```
git log
```

3.2.4 working with remotes

To see which remote servers you have configured, you can run the git remote command.

```
git remote
```

3.3 Features of Git

3.3.1 Git uses Snapshots

Git takes snapshots of all files at that moment and stores a reference to that snapshot.

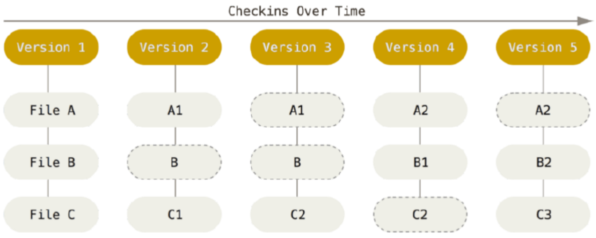


Figure 1: snapshot

Git thinks about its data more like a **stream of snapshots**.[1]

3.3.2 Nearly Every Operation is Local

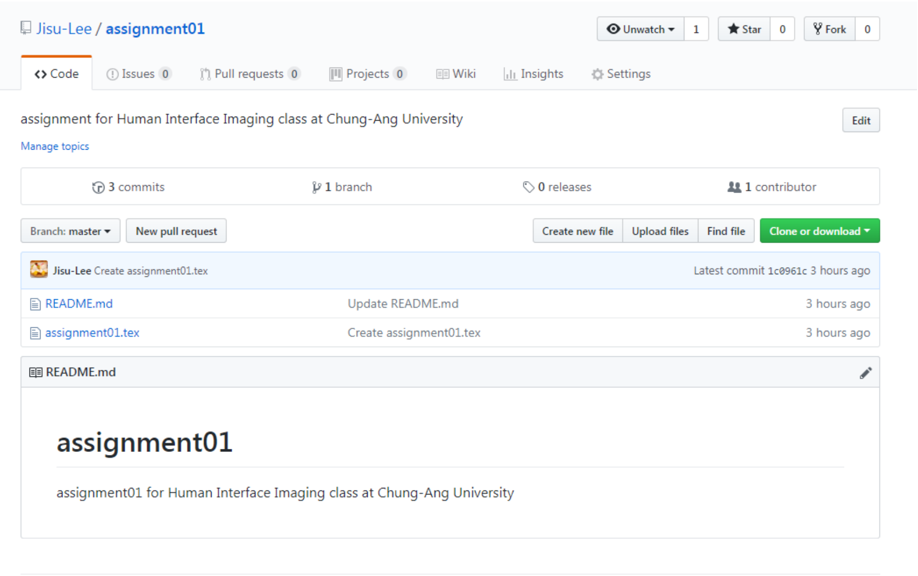
Most operations in Git need only local files and resources to operate. And this makes Git much faster than any other CVCS, where most operations have that network latency overhead. Because you have the entire history of the project right there on your local disk, most operations seem almost instantaneous.[1]

Git Has Integrity Everything in Git is check-summed before it is stored and is then referred to by that checksum. This means it's impossible to change the contents of any file or directory without Git knowing about it. This functionality is built into Git at the lowest levels and is integral to its philosophy. You can't lose information in transit or get file corruption without Git being able to detect it.[1]

4 Conclusion

In this report I introduced briefly about Git. With its powerful speed and fault-tolerant feature, Git is indeed a great version control system to use. Therefore, I conclude Git is very promising for the future.

5 My Github Screen Capture and Link



assignment for Human Interface Imaging class at Chung-Ang University

3 commits | 1 branch | 0 releases | 1 contributor

Jisu-Lee Create assignment01.tex — Latest commit 1c0961c 3 hours ago

README.md	Update README.md	3 hours ago
assignment01.tex	Create assignment01.tex	3 hours ago

assignment01

assignment01 for Human Interface Imaging class at Chung-Ang University

link : https://github.com/Jisu-Lee/assignment01

References

[1] Ben Straub Scott Chacon. *Pro Git*. Apress, 2014.