Version Control Protocols: GIT vs. SVN

CENG-0031 Higher Layer Network Protocols

**by**

Mehmet Harun Gülen

Department of Computer Engineering

Cukurova University

hgulen@cu.edu.tr

Advised by

Assoc. Prof. Dr. Fatih Abut

October 11, 2023

**ABSTRACT**

Thanks to systems that allow team members working on a software project to work on the correct version, team members know that their work is safe and this leads to increased motivation.

Developers working on large and small software projects or systems that are constantly changing, have become very clear after a while to follow what, when, by whom and for what reason their changes were made. Software tools made to track changes are called version tracking systems. In fact, they are nothing but files stored in the recording medium of the device that has undergone changes. These files can be software projects, the configuration of a router or switch created for a network, or a website.

1 Introduction

There are two types of version tracking systems here. These systems are divided into two as Centralized and Disturubted systems. Here, I will explain SVN as an example of the first Centralized systems. We will examine the Git system, which is one of the Disturubted version control systems, which is released in order to keep up with the changes in the software development concept and we will explain the differences between them.

Subversion(SVN) is open-source and comes under the Apache License

Likewise, Git is an open source and free system.

2 SVN



All change history is kept in SVN Server. The working copy of the project is located on the developer machine. Because there is only the working copy on the developer computer, it takes up less space and the first copy is uploaded to the local machine very quickly. It allows small changes to be sent to the server quickly. Operations done here.

**Check-out**: It is the process of removing the read-only property of the modified file at the operating system level and notifying the mark to the SVN server.

**Commit or Check-in** : It is the process of sending changes to the server.

**Update**: The process of pulling the changes uploaded to the server.

**Lock**: It is a blocking process to prevent changes to a file on the server.

**Merge**: It allows merging in any conflicting situation.

3 Git

Git considers the whole of the changed files as streams of snapshots. It is this approach that makes Git different from other version control systems.

For this reason, in order to work on a project on Git, an exact copy of the repository running on the server is included in the local workspace, including the change history. This causes more space to be consumed in the local work area.

**4 Operation**

|  |  |
| --- | --- |
| **Centralized (SVN)** | **Distributed(Git)** |
| GET LATEST VERSIYON | FORK, CLONE OR COPY |
| GET | PULL |
| MERGE | MERGE |
| CHECK-IN = COMMİT | STAGE-COMMIT-PUSH |
| CHECK-OUT | PULL REQUEST –MERGE REQUEST (FORK) |

**5- Differences**

|  |  |  |
| --- | --- | --- |
| **Scope** | **Centralized (SVN)** | **Distributed(Git)** |
| Storage | Only working copy on local machine | Full repository is copied to local machine with file history. |
| Speed | Allows rapid manipulation of atomic changes | Processing of changes takes time as they are duplicated by steps such as commit stage push. |
| Offline-work | Difficult to work without central computer | You can work independently not connected to lan/wan |
| Integrity | No | Everything in Git is checksummed 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. |
| Authentication | Inclueded | Protocol is the lack of authentication |
| Set up time | Easy to set up | Not easy to set up. It also requires firewall access to port 9418 |

**6 Conclusion**

Both protocols have pros and cons for software developers. Since Git is a protocol that came out after SVN, the Git protocol emerged as a result of the experiences and new needs that emerged while using it in SNV. But first of all, we need to keep more space locally to solve the need for offline work. Here, the decision of the project manager is very important. However, nowadays, these platforms are actually easy to transition, so developers can decide according to their own needs.

**Reference**

<https://git-scm.com/>

https://en.wikipedia.org/wiki/Git