There are two types of version control: centralized and distributed.

Centralized version control

Centralized version control systems store a single “central” copy of a user's project on a server and users need to commit their changes to this central copy. One can pull the files that he needs, but he never has a full copy of his project locally. Subversion (SVN) and Perforce are two of these kinds following this approach.

Distributed version control

With distributed version control systems (DVCS), one doesn’t rely on a central server to store all the versions of a project’s files. Instead, he clones a copy of a repository locally so that he has the full history of the project. Two common distributed version control systems are **Git and Mercurial**.

There are both pros and cons for two of the above mentioned approaches. While a centralized version control system avoids distributing the files to local systems when needed, it might seem a bottleneck to a developer since all the tasks need to go through a central repository. For any reason, if that centralized server is down, everyone will be in great trouble.

For distributed version control systems, after pulling from the central repo, one can work locally without thinking about central management in the server that can easily eliminate the bottleneck of centralization.