

UNDERSTANDING OF SVN STRUCTURE

SVN is a software versioning and a revision control system distributed under a free license. Developers use Subversion to maintain current and historical versions of files such as source code, web pages, and documentation. Its goal is to be a mostly-compatible successor to the widely used Concurrent Versions System.

During the process of upload our group's code to the SVN, gained some knowledge of the inner mechanism of SVN.

The basic structure of SVN is as follows.

The complex SVN system is actually made of 3 parts: user, network layers and repository.

1. Users actually: It is the client server, which is the SVN client programs, like TortoiseSVN, and the plugin Subclipse.
2. Network layer: It is the transition media from the client and the data storage center.
3. Repository: Repository is the central store of data in the subversion system. It is actually the database for the users' information. And Berkeley DB and

FSFS, which we haven't explained in the lecture, are the two methods that help realize the storage function of the SVN system.

Berkley DB

Berkley DB is a computer software library that provides a high-performance embedded database for key/value data.

Original development of Subversion used the Berkeley DB package. Subversion has some limitations with Berkeley DB usage when a program that accesses the database crashes or terminates forcibly. No data loss or corruption occurs, but the repository is offline while Berkeley DB replays the journal and cleans up any outstanding locks. When using Berkeley DB repository, the safest way to use it is by a single server process running as one user, instead of through a shared filesystem.

FSFS (short for *Fast Secure File System*)

FSFS repository stores a revision tree in a single file, and so all of a repository's revisions can be found in a single subdirectory full of numbered files.. FSFS is a new version store center. It even does not need a database to store the data. It is relatively a new store system that is faster than Berkeley DB when large sum of files needed to be uploaded, but many of its functions have not be tested much enough while Berkeley DB is more stable and mature.

Some more comparisons can be get from the following table:

Feature	Berkeley DB	FSFS
	very; crashes and permission problems can leave the database	
Sensitivity to interruptions	“wedged”, requiring journaled recovery procedures.	quite insensitive.
Usable from a read-only mount	no	yes
Platform-independent storage	no	yes
Usable over network filesystems	no	yes
Repository size	slightly larger	slightly smaller
Scalability: number of revision trees	database; no problems	some older native filesystems don't scale well with thousands of entries in

a single directory.

Scalability:

directories with

many files

slower

faster

Speed: checking

out latest code

faster

slower

Speed: large

slower, but work is spread

faster, but finalization delay

commits

throughout commit

may cause client timeouts

Group

permissions

sensitive to user umask problems;

works around umask

handling

best if accessed by only one user.

problems

Code maturity

in use since 2001

in use since 2004

The choice of the two kinds of repository is a trade-off between speed and reliability. Users need to make a decision according to their needs.

Source: <http://toyhouse.cc/profiles/blogs/understanding-of-svn-structure>