A Short Note on Details of Various Version Controllers and My version controller(GIT)

Submitted By,

Harsha K

15IT211

NITK Surathkal

Submitted To,

Raksha Nadigar

Assistant Lecturer

NITK Surathkal

Submitted On

23.01.2018

Overview

This Documentation is about the various version controllers available online and a note on my choice of version controller.

About Version Controller

A component of software configuration management, version control, also known as revision control or source control,is the management of changes to documents, computer programs, large web sites, and other collections of information. Changes are usually identified by a number or letter code, termed the "revision number", "revision level", or simply "revision". For example, an initial set of files is "revision 1". When the first change is made, the resulting set is "revision 2", and so on. Each revision is associated with a timestamp and the person making the change. Revisions can be compared, restored, and with some types of files, merged.

List of various platform where version controller is available

1.Local Data Model.

2.Client Server Model.

3.Distributed Model.

1.Local Data Model

It is something like all the developer should use same file system.It comes as Open source.

2.Client Server Model

In the client-server model, developers use a shared single repository.It come as both Open source and proprietary.

3.Distributed Model

In the distributed approach, each developer works directly with his or her own local repository, and changes are shared between repositories as a separate step.It comes as open source as well as distributed.

Various version controllers under each Platform

Revision Control System (RCS)

It stores the latest version and backward deltas for fastest access to the trunk tip compared to SCCS and an improved user interface,at the cost of slow branch tip access and missing support for included/excluded deltas.This comes under Local data Model.Revision Control System (RCS) is an early version control system (VCS). It can be thought of as a set of UNIX commands that allow multiple users to develop and maintain program code or documents. With RCS, users can make their own revisions of a document, commit changes, and merge them together. RCS was originally developed for programs but is also useful for text documents or configuration files that are frequently revised.RCS is only operated on single file system. It has no way of working with an entire project, so it does not support atomic commits affecting multiple files. Although it provides branching for individual files, the version syntax is cumbersome. Instead of using branches, many teams just use the built-in locking mechanism and work on a single *head* branch.

Working of **RCS**

RCS revolves around the usage of "revision groups" or sets of files that have been checked-in via the "co" (checkout) and "ci" (check-in) commands. By default, a checked-in file is removed and replaced with a ",v" file (so foo.rb when checked in becomes foo.rb,v) which can then be checked out by anyone with access to the revision group. RCS files (again, files with the extension ",v") reflect the main file with addition metadata on its first lines. Once checked in, RCS stores revisions in a tree structure that can be followed so that a user can revert a file to a previous form if necessary.

Concurrent Version System(CVS):(Open Source)

CVS is a version control system, an important component of Source Configuration Management (SCM). Using it, you can record the history of sources files, and documents. It fills a similar role to the free software RCS, PRCS, and Aegis packages.

CVS is a production quality system in wide use around the world, including many free software projects.

While CVS stores individual file history in the same format as RCS, it offers the following significant advantages over RCS:

It can run scripts which you can supply to log CVS operations or enforce site-specific polices.

Client/server CVS enables developers scattered by geography or slow modems to function as a single team. The version history is stored on a single central server and the client machines have a copy of all the files that the developers are working on.

Therefore, the network between the client and the server must be up to perform CVS operations (such as checkins or updates) but need not be up to edit or manipulate the current versions of the files. Clients can perform all the same operations which are available locally.

In cases where several developers or teams want to each maintain their own version of the files, because of geography and/or policy, CVS's vendor branches can import a version from another team (even if they don't use CVS), and then CVS can merge the changes from the vendor branch with the latest files if that is what is desired.

Unreserved checkouts, allowing more than one developer to work on the same files at the same time.CVS provides a flexible modules database that provides a symbolic mapping of names to components of a larger software distribution.

It applies names to collections of directories and files. A single command can manipulate the entire collection.CVS servers run on most unix variants, and clients for Windows NT/95, OS/2 and VMS are also available. CVS will also operate in what is sometimes called server mode against local repositories on Windows 95/NT.

Proprietary client server data model

Microsoft announced the release of a software as a service offering of Visual Studio on Microsoft Azure platform; at the time, Microsoft called it Visual Studio Online. Previously announced as Team Foundation Services, it expands over Team Foundation Server by making it available on the Internet and implementing a rolling release model. Customers could use Azure portal to subscribe to Visual Studio Online. Subscribers receive a hosted Git-compatible version control system, a load-testing service, a telemetry service and an in-browser code editor codenamed "Monaco". During the *Connect(); 2015* developer event on 18 November 2015, Microsoft announced that the service name is changed to **Visual Studio Team Services.**

Microsoft offers Basic, Professional, and Advanced subscription plans for Team Services. The Basic plan is free of charge for up to five users. Users with an MSDN subscription of Visual Studio can be added to a plan with no additional charge.

Distributed Model (Open Source)

In the distributed approach, each developer works directly with his or her own local repository, and changes are shared between repositories as a separate step.

BitKeeper is a software tool for distributed revision control of computer source code. Originally proprietary software, it was released as open-source software under the Apache 2.0 license on 9 May 2016.BitKeeper is produced by BitMover Inc., a privately held company based in Los Gatos, California and owned by its CEO, Larry McVoy, who had previously designed TeamWare.

Mercurial is a distributed revision-control tool for software developers. It is supported on Microsoft Windows and Unix-like systems, such as FreeBSD, macOS and Linux.

Mercurial's major design goals include high performance and scalability, decentralized, fully distributed collaborative development, robust handling of both plain text and binary files, and advanced branching and merging capabilities, while remaining conceptually simple.[3] It includes an integrated web-interface. Mercurial has also taken steps to ease the transition for users of other version control systems, particularly Subversion. Mercurial is primarily a command-line driven program, but graphical user interface extensions are available, e.g. TortoiseHg, and several IDEs offer support for version control with Mercurial. All of Mercurial's operations are invoked as arguments to its driver program hg (a reference to Hg - the chemical symbol of the element mercury).

I am using **GIT controller** for the software version controller.

Git is a version control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source code management in software development,but it can be used to keep track of changes in any set of files. As a distributed revision control system it is aimed at speed,data integrity,and support for distributed, non-linear workflows.

Git was created by Linus Torvalds in 2005 for development of the Linux kernel, with other kernel developers contributing to its initial development.Its current maintainer since 2005 is Junio Hamano.

As with most other distributed version control systems, and unlike most client–server systems, every Git directory on every computer is a full-fledged repository with complete history and full version tracking abilities, independent of network access or a central server.

Git is free software distributed under the terms of the GNU General Public License version 2.

Characteristics

Strong support for non-linear development.

Distributed development.

Compatibility with existent systems and protocols.

Efficient handling of large projects.

Cryptographic authentication of history.

Toolkit-based design.

Pluggable merge strategies.

Garbage accumulates until collected.

Periodic explicit object packing.