

SCM Evaluation

In choosing a SCM for this client, the primary requirements were: 1) It works well in a distributed environment; 2) There is a server available for Solaris; 3) There are clients for Solaris, Windows and the Mac. 4) There is a GUI interface for Windows; 5) The administrative and hardware costs are acceptable. Secondary requirements included a WWW interface for examining documents and some form of change request tracking.

After considering several different systems, the Perforce SCM tool was selected as the most likely candidate. I installed the server on one system, the client on a few others, and used it for some simple projects. It worked quite well in those tests. While I encountered problems, the Perforce tech support group resolved them quite quickly, including providing client libraries for a new platform on short notice. Based on this, I recommend that this client install Perforce on one of their servers, and start using it on a test basis for parts of the existing web content.

Systems Considered

A search - including several web search engines and querying developers using these systems - turned up several systems to check on. They are presented, with a quick overview, in alphabetical order.

BitKeeper

BitKeeper would have been the most promising of the systems, but was not yet available for production use. BitKeeper designers targeted OSS projects. The designers assume distributed developers, with the number of developers running from a few to thousands.

ClearCase

The most capable system considered was ClearCase. It is also very expensive in terms of administration and hardware costs. The consensus of the people using ClearCase was that it generally required both a dedicated person and a dedicated system. These costs are unacceptably high for a SCM for this client.

CVS

CVS was the second most promising system examined, especially considering the pricing. However, I could not find a Windows 95 client. Further, the *module* system CVS uses for labeling parts of a source tree complicates Unix client configuration. As a final strike, the job tracking system consists of programmatic hooks to connect CVS to an external job tracking system.

Perforce

Perforce is the most promising of the systems considered, meeting all of the primary and secondary requirements. I installed and used the server and client software for some small projects for evaluation purposes. The details on this can be found below.

PRCS

PRCS does not have support for a distributed work environment, and is thus totally unsuitable for this client. Version 2.0 is supposed to support a server that remote clients would use to reach the source code.

VSS

Visual Source Safe has marginal support for a distributed work environment. Support for clients other than Windows is from third parties, which is a rich source for version conflicts. I heard numerous reports of VSS losing files.

Evaluation of Perforce

Obviously, the first step was to install Perforce. They distribute the production software with a *default* license that allows two users and two *workspaces*. The installation consisted of copying the two binaries into place, and arranging for the server software to run at boot time. Configuring the Unix client software consisted of setting a few environment variables to tell it where the server was, and what my preferences are. This took no more than a few minutes, total.

For the next step, I took a small project - just a few files - and put it under the control of Perforce. I then continued working on the project, using the Perforce commands where appropriate to notify the Perforce system about the state of the work. While this wasn't unobtrusive - the very nature of an SCM requires some intrusion - it was about as painless as I could imagine it being. When I needed to edit a file, I issued one command. If I forgot that step, the system prevented me from writing on the file, as the file was write-protected at the system level. After informing Perforce that I wanted to edit the file, it was writeable. When I finished editing and testing the file, I submitted the change to Perforce. It started an editor (that I specified at configuration time) on a form that had fields to state the files that had changed, and what the changes did. This was the least painful SCM system I've ever used.

For the next step, I moved to a system elsewhere on the internet, and installed the client there. I then repeated the above process, running the client and server on systems separated by two firewalls. Excepting the initial setup - which involved copying multiple files - the response was not noticeably slower than doing the work remotely.

As a last test, I changed work modes. I added the files on the remote server to the local workspace, and started working on them there. After I was done with a change, I submitted the change to Perforce, moved back to the remote system, and synchronized that workspace with Perforce. While not quite as painless as doing things on only one system, it still worked well. I don't see that this would have been different from trying to work in this way without an SCM.

To test the suitability of the Web interface to Perforce - known as *WebKeeper*, I installed that Apache module on an existing web server. This proved more difficult, as it required software for a platform they didn't support. I went to Perforce's tech support - via electronic mail - and asked about this. After discussing the various options, they provided a new version of the library about 24 hours after we identified the problem. This is typical of my interactions with the Perforce tech support group - timely, accurate response that solved the problem.

WebKeeper worked as advertised. It might take a little bit of work to make it fulfill all our needs. However, if this proves difficult, or impossible, this is not a critical requirement.

Recommendation

This evaluation shows Perforce as an excellent fit for this project. I recommend setting up a small project that involves two people on two platforms under Perforce, as described in the main body of the document.