

Comparison: Perforce and Subversion

Perforce
& Subversion

Overview

This document compares Perforce (version 2008.1) with the open source Software Configuration Management (SCM) tool Subversion (version 1.5.2.) It contrasts some of the major feature differences and provides head-to-head metrics for typical operations such as branching, merging, and check-ins and check-outs, along with a general comparison of the effects of scaling on both systems.

Executive Summary

Attribute	Perforce	Subversion
Branching and Merging	Perforce automatically tracks the history of all branch operations with its advanced and mature merge tracking mechanisms.	Subversion offers branching, and provides basic merge history tracking in the destination file's attributes. Subversion's merge algorithms do not cover all merge scenarios.
Distributed Development	The Perforce Proxy offers a caching solution for remote users that can be deployed limitlessly, with minimal administrative overhead, and at no extra cost. It is designed, developed, and maintained in-house by Perforce's core products engineers.	Remote slave servers can be synchronized from a centralized master server. Synchronization mechanism requires the coding of event-based scripts.
Scalability and Performance	Perforce's architecture is proven and continues to scale well up to and beyond many terabytes and tens of millions of versioned files. Architected for speed, Perforce scales linearly and is currently deployed in environments with 10,000+ users.	Subversion depends heavily on CPU-intensive operations that do not scale well when workloads increase.
Defect Tracking	Perforce provides a basic defect tracking system called jobs, and integrates with leading third-party defect tracking systems.	Not included with Subversion; users need to use a separate defect tracking solution.
Integration with related tools	Perforce integrates with leading IDEs such as Eclipse and Visual Studio, build tools, Microsoft Office, Adobe Photoshop, Maya, and other third party tools.	Integrations are available via the open source community.
System Administration and Support	Provided by Perforce SCM experts and relied upon by over 300,000 users.	Available at additional cost from third parties.

Branching and Merging

Subversion offers codeline branching using a simple copy command to create a branch, and keeps a basic history of the operations attached to each file individually in file-keyed attribute lists. Subversion merges are slow and historically have been problematic and prone to merge errors. Common merging scenarios between sibling and distantly related branches are often not possible. In these scenarios users must revert to performing manual merges. Bi-directional and divergent merging is not supported.

The Perforce Inter-File Branching™ model is powerful and flexible, capable of branching thousands of files rapidly while retaining a complete branch and merge history. Instead of manually tracking all changes across branches, users can rely upon Perforce to merge file changes across multiple branches automatically and with fewer conflicts to resolve. A built-in graphical tool, Revision Graph, displays the detailed branching history of each file for easy visualization of code propagation.

Perforce inherently understands the ancestral relationship files share between branches. Once a change has been integrated from an originating branch into a target branch, there is no ambiguity that the change exists in both branches. Therefore, on subsequent integrations from the target back to the originating branch, users will only be prompted to integrate changes that have occurred directly in the target branch. This kind of integration pattern is referred to as bi-directional or ladder merging.

Divergent branching supports the case where changes or features introduced directly into a source branch are to be ignored or skipped while being merged into a target branch. In effect, an integration record is created but the target file or files remain unchanged. This type of integration is known as an Ignore record in Perforce. The source and target branches remain different even though the integration history indicates there are no pending merges.

Distributed Development

With Subversion, remote developers work directly against local slave servers. All slaves are configured to synchronize their data stores from a centralized master. Event-based scripts triggered by updates in the master notify each registered remote slave to synchronize its archive. Scripts need to be created and then deployed to both slave and master servers.

Distributed development with Perforce doesn't require any additional process overheads or additional cost. Perforce proxies at remote locations support Perforce's distributed architecture. The Perforce Proxy caches and serves files to users at remote locations, thereby reducing traffic across slower WAN links. All users, local or remote, connect to the same central depot, and look at the same project files. Any number of proxies can be deployed, are invisible to end users, and require minimal administrative attention.

Scalability and Performance

With Subversion, the binary delta differencing scheme used to version both binary and text files consumes a large amount of CPU time on the client and server. This is especially problematic when the files are large and there are many concurrent users competing for the same resources.

Perforce is architected first and foremost for speed. The reverse delta RCS format for text and gzip compression on binary files means users don't need to wait for check-ins and other standard SCM operations. Perforce scales linearly, so there is no performance penalty imposed by the number of revisions or the size of any given file(s). The Perforce server has been deployed successfully in environments with thousands of users.

Benchmarks

Perforce easily outperforms Subversion in all equivalent SCM operations during comparative benchmarking tests. The tests results are detailed below.

Test Data

The combined client/server machine on which the tests were run had the following specifications:

- Opteron 246, single core processors, running at 2 GHz
- Total Memory: 8 GB
- Operating System: CentOS 4.3
- Storage: 8-way RAID10 using Standard Maxtor DiamondMax SATA2

Test Results

Performance results for the Perforce and Subversion comparative benchmarks are presented in the tables below. Response times were measured in minutes and seconds.

The dataset for Table 1 below represents the complete Linux kernel, consisting of 24,353 files and consuming approximately 327,292 MB of disk space.

The dataset for Table 2 below comprises five copies of the Linux kernel tree. This dataset totals 121,767 files and is 1.6 GB in size.

Table 1: Performance results for one copy of the complete Linux kernel

Operation	Perforce (2008.1)	Subversion (1.5.2)
Added complete Linux kernel (data described above)	1m 6.681s	1m 8.265s
Lightweight branch ¹ of entire kernel tree	0m 3.940s	0m 0.072s
Syncing branch to desktop	0m 12.382s	0m 58.054s
Heavyweight branch ² of entire kernel tree	0m 11.062s	2m 7.131s
File change detection in 611 edited and 15 new files	0m 5.605s	0m 20.152s
Submitting 611 edited and 15 new files	0m 2.688s	0m 52.855s
Checking in 7 AVI files (1.3GB total size)	2m 22.042s	16m 5.675
Syncing 7 AVI files to desktop (1.3 GB total size)	0m 22.073s	6m 10.005s
Checking in change to two binary files (800 MB total size)	1m 18.601s	0m 44.949s
Checking in large change to a binary file	0m 59.650s	1m 21.450s
Auto-merging 611 changed files into child branch (merge and checked in)	0m 10.686s	1m 12.297s
Auto-merging 611 changed files from child to sibling branch (merge and checked in)	0m 8.918s	0m 43.265s

¹ "Lightweight branch" only branches files on the server. No files are transferred to the client.

² "Heavyweight branch" branches all files on the server and then transfers them to the client.

Table 2: Performance results for five copies of the Linux kernel tree

Operation	Perforce (2008.1)	Subversion (1.5.2)
Adding all 5 kernel trees to source control	7m 38.155s	118m 10.367s
Lightweight branching ¹ of entire tree	0m 26.498s	0m 3.065s
Heavyweight branching ² of entire tree	2m 53.466s	53m 2.597s
Syncing 121,767 files to desktop (1.6 GB)	1m 51.649s	9m 31.325s

¹ "Lightweight branch" only branches files on the server. No files are transferred to the client.

² "Heavyweight branch" branches all files on the server and then transfers them to the client.

Defect Tracking

Subversion does not include defect tracking, but a limited number of third-party integrations are available.

Perforce provides a basic defect tracking system called jobs. A job typically represents an enhancement request or a bug to be fixed. Job definitions are customizable to support Perforce workflow. Jobs can also work with several third-party defect tracking systems.

Data entered into a Perforce job is automatically replicated in the defect tracking tool, and vice versa. Perforce can tell you whether or not a bug fix is present in any particular codeline, regardless of the codeline in which the bug fix originated.

Integration with Related Tools

The open source community supports different GUIs in Subversion, as well as plug-ins for working within popular IDEs.

Perforce has a mature multiplatform GUI, and there are plug-ins for leading IDEs. Perforce also integrates with popular applications including:

- IDEs
- Web and graphical tools
- Software build tools
- Microsoft Office
- API tools
- Merge and diff tools

System Administration and Support

Subversion requires administrators to make several decisions before implementation. For example, administrators must decide whether to configure a web server or to use a local text-based configuration file mechanism for user authentication and authorization, and whether to use a file-based system or a database for storing versioned files.

Subversion offers full accessibility to the source code, but the administrator must invest the time and effort to get familiarized with the codebase. Support for Subversion is available commercially, through third parties.

Perforce functions the same regardless of platform, and provides a built-in security mechanism with many levels of access that can be applied to any subset of the archived files—from the entire depot to the individual file. Versioned files are stored the same on all platforms.

Expert and responsive technical support is a hallmark of Perforce and full technical support is included during an evaluation. Perforce believes that when you buy an SCM system, you're not only putting your faith in the software, but also in the technical support you expect to receive.

Additional Resources

Evaluating Perforce

More than 300,000 users at 5,000 companies rely on Perforce for software configuration management. Perforce encourages prospective customers to judge for themselves during a typical 45-day trial evaluation. And because the quality of a technical support organization is better experienced than described, get started by visiting: www.perforce.com/svn

Scheduling a Demo of Perforce

To learn more about Perforce, schedule an interactive demo tailored to your requirements.

Visit: www.perforce.com/svndemo

Migrating to Perforce

For more information on related software including conversion utilities for migrating from Subversion to Perforce, please visit: www.perforce.com/perforce/loadsupp.html#conv

Perforce Consulting Services has experience assisting customers with migrations from various SCM systems. For more information, please visit: www.perforce.com/consulting

Additional SCM comparisons available at <http://www.perforce.com/comparisons>



North America

Perforce Software Inc.
2320 Blanding Ave
Alameda, CA 94501
USA
Phone: +1 510.864.7400
info@perforce.com

Europe

Perforce Software UK Ltd.
West Forest Gate
Wellington Road
Wokingham
Berkshire RG40 2AQ
UK
Phone: +44 (0) 845 345 0116
uk@perforce.com

Australia

Perforce Software Pty. Ltd.
Suite 3, Level 10
221 Miller Street
North Sydney
NSW 2060
AUSTRALIA
Phone: +61 (0)2 8912-4600
au@perforce.com