

Comparison:

# Perforce and Microsoft Team Foundation Server (TFS)

## **Perforce 2011.1 and Microsoft Team Foundation Server (TFS) 2010**

This document compares Perforce (version 2011.1) with Microsoft TFS (version 2010). Read this comparison to:

- Understand Perforce and TFS's major feature differences
- See head-to-head metrics for operations like branching, merging, check-ins, and checkouts
- Get a general comparison of the effects of scaling on both systems

# Table of Contents

|  |    |
|--|----|
| Executive Summary                          | 1  |
| Overview                                   | 1  |
| Branching, Merging, and Release Management | 2  |
| TFS  | 2  |
| Basic Branching and Merging                | 2  |
| Inefficient Merges Lead to Conflicts       | 2  |
| Ignored Merge History Loses Data           | 2  |
| Indirect Merging Not Supported in TFS      | 4  |
| Branching Model and Guidance               | 4  |
| Perforce                                   | 5  |
| Basic Branching and Merging                | 5  |
| Branching Model and Guidance               | 5  |
| Workspace Management                       | 6  |
| TFS  | 6  |
| Perforce                                   | 6  |
| Visual Tools                               | 6  |
| TFS  | 6  |
| Perforce                                   | 6  |
| Integration with Related Tools             | 8  |
| TFS  | 8  |
| Perforce                                   | 8  |
| Administration and Support                 | 8  |
| TFS  | 8  |
| Perforce                                   | 8  |
| Distributed Development                    | 9  |
| TFS  | 9  |
| Perforce                                   | 9  |
| Scalability and Performance                | 9  |
| TFS  | 9  |
| Perforce                                   | 10 |
| Benchmarks                                 | 10 |
| Test Platform                              | 10 |
| Test Data                                  | 10 |
| Test Results                               | 10 |
| Learn More                                 | 12 |
| Evaluating Perforce                        | 12 |
| Scheduling a Demo of Perforce              | 12 |
| Migrating to Perforce                      | 12 |

## Executive Summary

The choice of software version management system profoundly affects those involved in digital asset management, from software developers to artists to managers. An effective software version management system is one that:

- Provides a full history of the evolution of digital assets
- Enables parallel development and concurrent team activity
- Helps the entire team work more efficiently
- Meets modern development and scalability challenges
- Offers fast, flexible, and reliable service

While Microsoft Team Foundation Server (TFS) has several other Application Lifecycle Management (ALM) components, including task tracking and

build management, this paper will concentrate on the versioning component of TFS.

This document compares Perforce (version 2011.1) with Microsoft Team Foundation Server (version 2010), contrasting major feature differences and providing head-to-head metrics for common operations such as branching, merging, check-ins and checkouts, along with a general comparison of the effects of scaling on both systems.

The analysis in this paper suggests that Perforce is a more productive and robust software version management system with significant advantages in scalability and total cost of ownership. Perforce handles basic and advanced software version management operations more reliably, provides innovative tools, is simpler to install and scale, and does not dictate workflow and tool selection.

## Overview

### Usability and Collaboration

| Attribute   | Team Foundation Server (TFS)  | Perforce   |
|---|---|--|
| <b>Branching, Merging, and Release Management</b> | TFS offers basic branching and merging. TFS's lack of support for advanced merging hampers productivity. Branching framework and release model mostly designated by convention. | Perforce automatically tracks the history of all branch operations with its advanced merge tracking mechanisms. Streams provide best-practices branching framework. Granular triggers and broker available for guidance.   |
| <b>Workspace Management</b>                       | Changelists are not created until work is submitted. Workspace views are simple 1:1 mappings.   | Changelists can be used to organize work in a workspace. Workspace views allow more flexible selection of working files.   |
| <b>Visual Tools</b>                               | TFS clients available in Visual Studio, Eclipse, and a web client. A simple branch graph is provided.   | Full-featured multiplatform standalone visual client available, as well as fully supported integrations for Visual Studio and Eclipse. Third-party integrations support several other IDEs. Visualization tools include Stream Graph, MergeQuest, Revision Graph, Time-lapse View, and more. |
| <b>Integration with Related Tools</b>             | TFS includes several Application Lifecycle Management (ALM) components. Integrations are available from third parties. APIs available for .NET, Java, and web services..        | Perforce follows a best-of-breed approach and offers fully supported integrations with leading defect tracking and build tools. Third-party integrations are available for tools in many other categories. Integration tools include the Defect Tracking Gateway and several APIs.           |

## Administration and Scalability

| Attribute                   | Team Foundation Server (TFS)   | Perforce  |
|-----------------------------|--|---|
| Administration and Support  | Supported by Microsoft. Administration requires knowledge of several components including TFS, SQL Server, IIS, and SharePoint. All server components are Windows only. Upgrade procedures can be complex.   | World-class technical support, training, and professional services provided by Perforce. Simple and consistent deployment results in lower administration costs. Excellent multiplatform support. Simple upgrade procedures with good interoperability across versions.             |
| Distributed Development     | Proxy servers cache file content at remote sites.  | P4Sandbox offers connection-independent versioning and private local branching. Proxy servers offer a file caching solution for remote users with minimal administrative overhead. Replica servers provide a full copy of server data at remote locations for read-only operations. |
| Scalability and Performance | Scaling TFS may require a complex topology with multiple clustered data and application tiers. The data tier can be partitioned across sites, but no replication is available. Performance benchmarks indicate that TFS is slower at branching, checkout, and other common operations. | Perforce scales up to many terabytes and tens of millions of versioned files. Perforce is currently deployed in environments with 10,000+ users and heavy automation. Proxies, Brokers, replica servers, and P4Sandbox provide powerful and flexible deployment architectures.      |

## Branching, Merging, and Release Management

### TFS

#### Basic Branching and Merging

Although TFS offers basic branching and merging, its merge algorithms do not support common merge scenarios such as indirect merges, and sometimes create merge problems. The lack of advanced merge support can decrease productivity, as developers waste time fixing merge problems and adjusting their workflows to work around the limitations. Similarly, uncertainty about merging results will slow the adoption of automated release management processes.

#### Inefficient Merges Lead to Conflicts

As a simple example of a case where TFS merging does not work properly, consider the scenario illustrated in Figure 1. In this scenario, a file was branched immediately after creation, several edits were made on both branches, and some changes from the child branch

(DEV) were merged back (into MAIN). In Step 7, we try to merge the latest changes from the parent (MAIN) to the child (DEV). TFS does not properly consider prior merge credit from Step 5, leading to an unnecessary merge conflict. Although TFS does not report the selected base, it is most likely selecting the wrong base for the merge.

#### Ignored Merge History Loses Data

TFS's merge engine also causes data loss by failing to properly consider prior merge history. Consider the scenario illustrated in Figure 2.

In this scenario, a file was branched immediately after creation. A change was made to the file in the parent branch (MAIN) and merged into the child branch (DEV), but the change was ignored. A change was then made on the child branch. When merging this change back to the parent in Step 6, TFS overwrites the first line of the file. In reality, the only unique change on the child branch was the introduction of the new line at the end of the file, so removing the edit on the first line is unexpected.

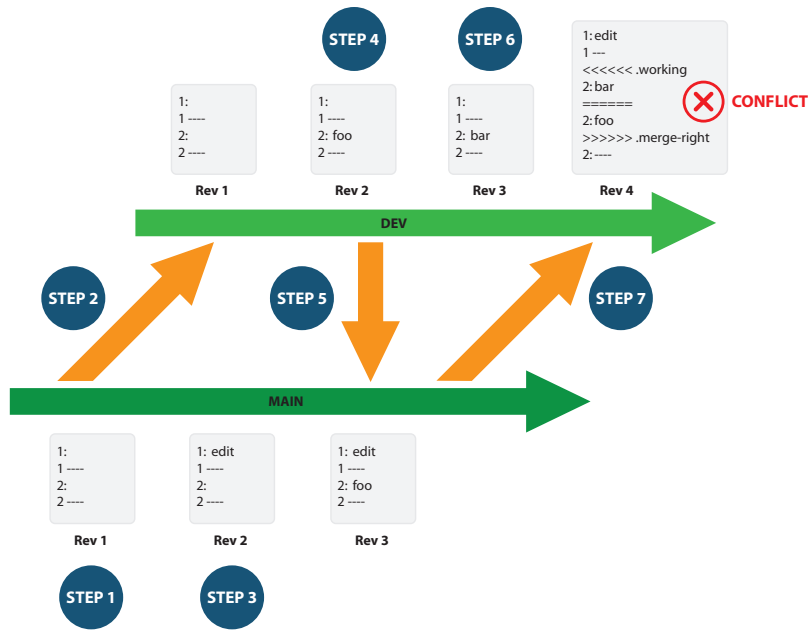


Figure 1: A simple merge scenario in which TFS causes a merge conflict

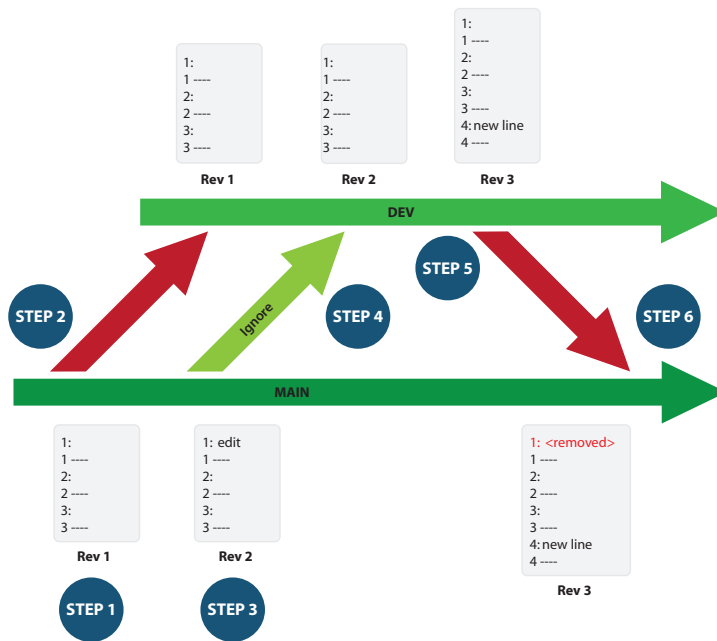


Figure 2: A merge scenario in which TFS causes data loss due to ignoring merge history

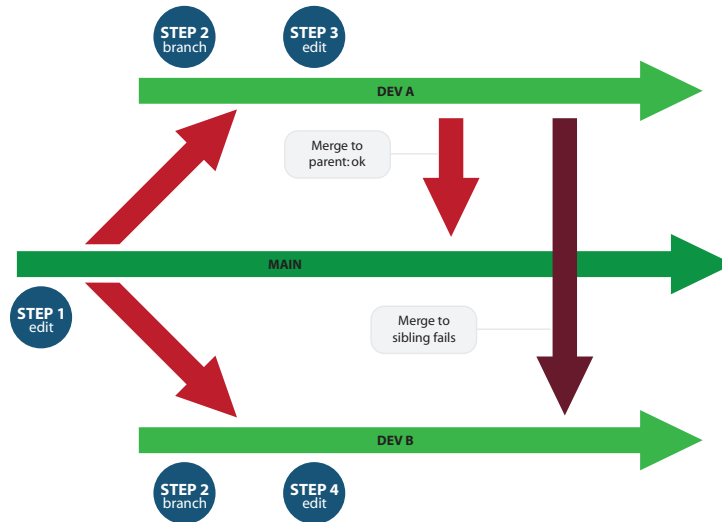


Figure 3: TFS cannot handle indirect merges

### Indirect Merging Not Supported in TFS

Indirect merging (merging between sibling branches) is common in advanced environments. Although not a best practice, indirect merging is sometimes necessary to quickly propagate a bug fix between indirectly related branches. Indirect merging also provides flexibility for moving work from one task branch to another.

TFS does not support indirect merging. Attempting a merge between sibling branches gives an error, as shown in Figure 3. Indirect merging can be forced using the baseless option, but that ignores prior indirect merge history, greatly increasing the chances of merge conflicts.

### Branching Model and Guidance

Development and release management models are built by convention, usually using a known directory structure for different branch types. Branch graph visualization is available in TFS as shown in Figure, but it does not indicate the relative stability of different branches or the intended merge pathways. For instance, there is no indication that the rel-1.0 branch is more stable than the my-task branch, and no indication whether the rel-1.0 branch should accept a change from its parent. Also, no visual indicator exists when merges are pending between branches.

TFS lacks some of the usual tools to guide a custom workflow for branching and release management tasks, such as hooks or triggers that respond to SCM events. TFS does have some workflow capabilities in its other components. For example, the TFS build system can enforce a policy that prevents check-ins unless the code builds successfully. TFS also has an event system that can be used for post-processing of SCM events, such as email notifications of check-ins.

#### Jam Branch Hierarchy

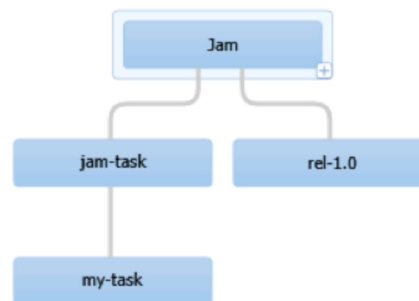


Figure 4: TFS Branch Graph showing the branch hierarchy for the Jam project

# Perforce

## Basic Branching and Merging

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 (including the method of resolving merges). Instead of manually tracking all changes across branches, users can rely upon Perforce to merge file changes across multiple branches automatically with fewer conflicts to resolve. This enables a variety of development scenarios such as client-specific versions, experimental branches, personal or task branches, and the classic release branching patterns. A built-in graphical tool, Revision Graph, displays the detailed branching history of each file for easy visualization of code propagation.

Perforce's merge engine is both robust and flexible. The most complex merge scenarios are supported, including refactoring, indirect merges, and handling non-content changes. Because of this robustness, automation can confidently be incorporated into most branching and release management workflows.

In the three merging scenarios described earlier, Perforce performs the merges correctly:

- In the scenario described in Figure 1, Perforce selects the correct base which results in a merge with no conflicts.
- In the scenario described in Figure 2, Perforce correctly considers prior merge history and merges cleanly.
- In the scenario described in Figure 3, Perforce is able to seamlessly merge between the sibling branches using the indirect merge history.

## Branching Model and Guidance

Perforce Streams provide a lightweight but powerful branching model. Using streams, a product architect defines the relationship between streams, the modules that compose a product, and the direction of change (merges) between streams. This information simplifies and automates many routine user operations.

When using either streams or regular branching, Perforce offers several tools to guide and enforce policy. Triggers and access control are granular, while the Perforce Broker provides command filtering.

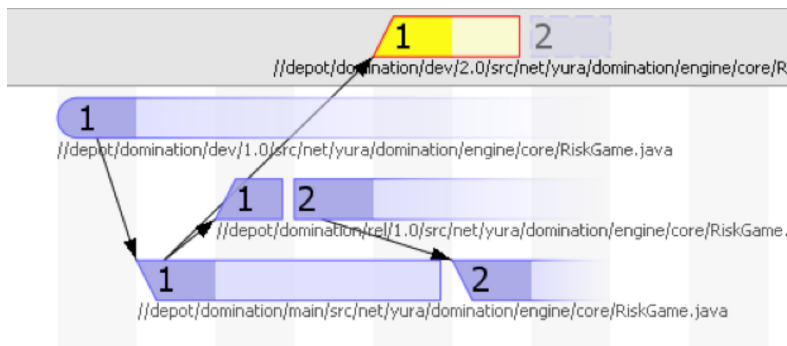


Figure 5: Perforce Revision Graph

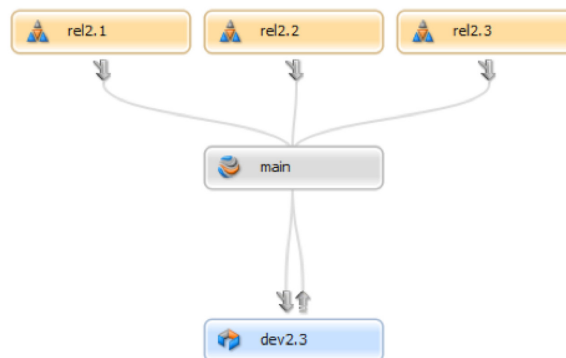


Figure 6: Perforce Stream Graph

## Workspace Management

### TFS

TFS workspaces consist of simple 1:1 mappings between repository directories and the local file system. New workspaces can be created using an existing workspace as a template.

TFS's changesets are roughly equivalent to Perforce changelists, representing atomic units of work and sync points in the repository. However, TFS does not allow the creation of pending changesets to organize work. Changesets cannot be created until check-in (Shelvesets can be used to shelve pending work, but submitting a shelveset checks in the file content as stored in the shelf).

### Perforce

Perforce workspace views are flexible and granular, allowing the selection, exclusion, or remapping of files. For example, a Perforce workspace could exclude all build artifacts (files with a .obj extension) and move region-specific translation files to a different folder for localized builds.

Perforce workspaces can be generated automatically when working with streams, allowing team leads to ensure all developers use the correct set of files. Users can also create Perforce workspaces manually or use existing workspaces as templates.

Users can create pending changelists to effectively organize work on several tasks in a workspace. This simple feature helps users avoid submitting unrelated files by mistake.

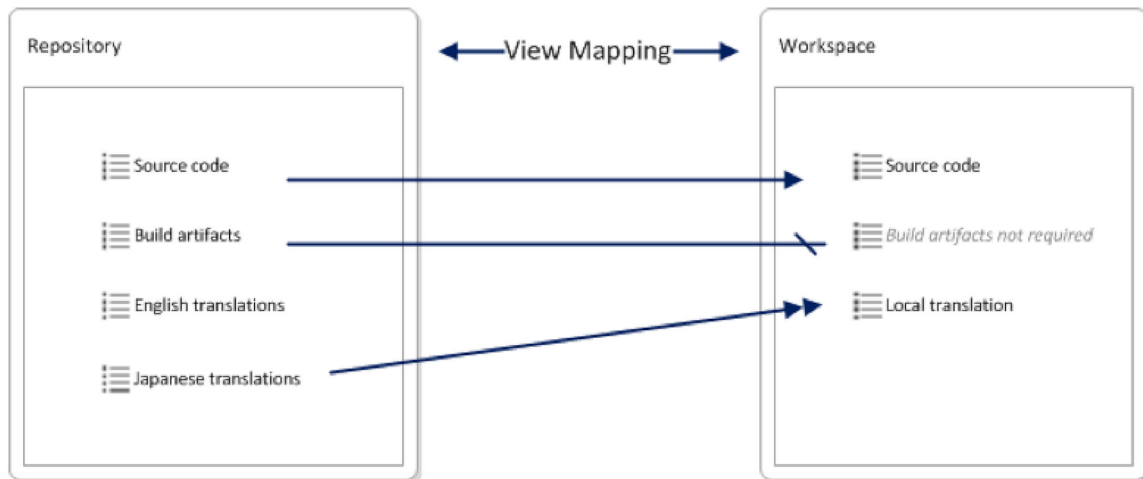


Figure 7: A flexible workspace view in Perforce

## Visual Tools

### TFS

TFS's visual tools are available in Visual Studio and Eclipse. Besides the Branch Graph (Figure 4), TFS provides other techniques for displaying the branching history of a changeset (Figures 8 and 9). However, these tools do not show the full lifecycle of a file like Perforce's Revision Graph (Figure 6).

### Perforce

P4V, Perforce's multiplatform visual client, provides consistent visual tools for all major platforms that are generally more feature-rich than their TFS equivalents. For instance, Perforce's annotation tool includes lifecycle indicators and can show a subset of file history.

Innovative visualizations including the Stream Graph and HTML5 applets enhance productivity. Besides P4V, fully supported integrations are available for Visual Studio and Eclipse. Third-party integrations support several other IDEs.

### Tracking Changeset 139

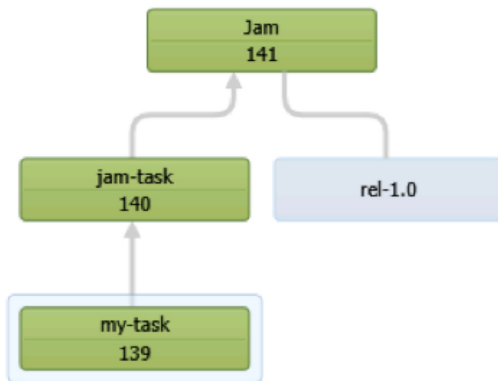


Figure 8: Branch history of a changeset (hierarchy) in TFS

### Tracking Changeset 139

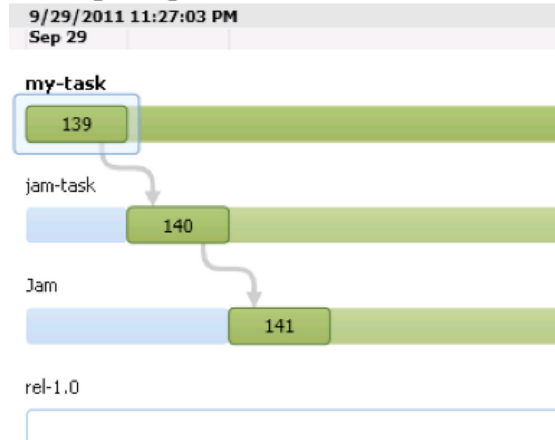


Figure 9: Branch history of a changeset (timeline) in TFS

```

Tracking Changeset 139  History - my-task  Tracking Changeset 139  History - ReadMe.txt  History - main  ReadMe.txt;C115 (Annotated) X H
5 Administrator 9/28/2011 =====
                               CONSOLE APPLICATION : Jam Project Overview
                               =====
AppWizard has created this Jam application for you.

113 Administrator 9/29/2011 my comment
114 Administrator 9/29/2011 a change
113 Administrator 9/29/2011
5 Administrator 9/28/2011 This file contains a summary of what you will find in each of the files that
                               make up your Jam application.

115 Administrator 9/29/2011 more change
5 Administrator 9/28/2011
                               Jam.vcxproj
                               This is the main project file for VC++ projects generated using an Application Wizard.
                               It contains information about the version of Visual C++ that generated the file, and
                               information about the platforms, configurations, and project features selected with the
                               Application Wizard.
  
```

Figure 10: TFS Time Lapse View

```

140 earl
1 earl
353 earl # include "oan.h"
...
/*
 * compile.c - compile parsed jam statements
 *
 * External routines:
 *
 * compile_foreach() - compile the "for x in y" statement
 * compile_if() - compile 'if' rule
 * compile_include() - support for 'include' - call include() on file
217 earl * compile_local() - declare local variables
370 earl ... * compile_local() - declare (and set) local variables
382 earl ... * compile_null() - do nothing -- a stub for parsing
1 earl ... * compile_rule() - compile a single user defined rule
... * compile_rules() - compile a chain of rules
  
```

Figure 11: Perforce Time-lapse View

## Integration with Related Tools

TFS and Perforce follow radically different philosophies in this area. TFS, as part of the Visual Studio Team System, provides a largely monolithic ALM package. Perforce practices a best-of-breed strategy, providing the best versioning solution possible and integrating with powerful tools in other categories.

### TFS

TFS includes other ALM components, including a build management system, test agents, reporting, and SharePoint for collaboration portals. Indeed, every new TFS project requires the selection of a process template.

TFS integrates with Microsoft Excel and Project for work item tracking. Custom tools provide basic check-in/checkout features in other Microsoft Office products.

Third-party integrations are available for non-Microsoft tools. TFS provides APIs for the .NET environment, Java, and web services.

### Perforce

Perforce uses a best-of-breed approach, providing powerful SCM and integrating with leading tools in other categories. Although using a single solution for ALM may seem simpler, preselecting one tool that works well for all teams proves challenging. Assembling an ALM package from best-of-breed products provides the flexibility to change and adapt processes over time.

Integration tools include the Defect Tracking Gateway and fully supported APIs for C/C++, Perl, Python, Ruby, PHP, Java, iOS, and Android. Perforce also integrates with popular applications in several categories including:

- IDEs
- Web and graphical tools
- Software build tools
- Microsoft Office (versioning for Word, Excel, PowerPoint, and Project files)
- Merge and diff tools

## Administration and Support

Installing and administering a Perforce Server is significantly simpler than TFS, resulting in a lower total cost of ownership.

### TFS

TFS is a complex application with several components.<sup>1</sup> A TFS instance includes SQL Server, IIS, and TFS itself, and may include SharePoint and other modules. Architecturally, TFS is minimally divided into data and application tiers with several databases and optional SharePoint and reporting tiers. Depending on deployment size, TFS may run on one to several servers. The requirement for product components distributed over multiple servers adds to the training, maintenance, and diagnostic responsibilities of system administrators.

Installing and maintaining TFS requires expertise in all the components and a considerable investment in planning the deployment architecture. Backup and recovery procedures must be implemented and tested for each tier and component. Upgrades may be complex and time-consuming; for instance, upgrading the application tier usually involves uninstalling the old version of TFS and then installing the new version.

Scaling out a TFS deployment requires expertise in several areas, including SQL Server, IIS, SharePoint, hardware and network load balancing, and clustering.

The TFS server runs only on Windows, although clients are available for other platforms. TFS is supported by Microsoft.

### Perforce

Perforce imposes minimal administrative overhead. Perforce deployment is simple and consistent. Working in a multiplatform environment poses no problem, and Perforce has excellent support for mixed server and client versions. Upgrade procedures are simple and fast, often requiring nothing but replacing the server binary and running a single upgrade command.

Scaling out a Perforce deployment requires very little additional expertise. Proxy servers and brokers require little maintenance, while P4Sandbox is entirely independent of the central server. Replicated servers require a small set of configuration changes, but are otherwise identical to a regular Perforce server. The Perforce server runs on all major platforms.

Expert and responsive technical support is a hallmark of Perforce, and full technical support is included during an evaluation. Perforce believes that customers purchasing a software version management system put their faith in both the software and in the technical support team. Because support is better experienced than described,

<sup>1</sup> <http://msdn.microsoft.com/en-us/library/ms252473.aspx>

Perforce encourages prospective customers to judge for themselves during a 45-day trial evaluation. Perforce also offers a full range of training and professional services. Perforce provides technical support for past releases for an extended duration, and has a defined policy that provides 12 months' minimum notice before an older product is retired.

## Distributed Development

### TFS

TFS offers simple proxy servers that cache file contents to support remote sites or build farms.

### Perforce

Distributed development with Perforce is supported by several tools. These tools are typically transparent to the end user.

P4Sandbox supports work without a connection to the central server, private local branching, and fast local operations.

Perforce proxies at remote locations support Perforce's distributed architecture. The Perforce Proxy caches and

serves files to users at remote locations, reducing traffic across slower WAN links. All users, local or remote, connect to the same central depot and look at the same project files. The Perforce Proxy requires minimal administrative attention.

Replicated Perforce servers provide completely local read-only operations for remote users. As a large percentage of Perforce operations are read-only, using a local replica offers a significant performance benefit.

## Scalability and Performance

### TFS

Perforce outperforms TFS on several performance benchmarks in a single server configuration. (See Benchmarks). More importantly, scaling TFS to support a large user base, distributed work, or heavy data volume can be a daunting task. Examples of scaled-out TFS deployments include multiple application and data tiers, each running in clusters, with several independent data tiers for different sites.<sup>2</sup>

Despite the complexity of a TFS deployment, simple data replication is not supported.

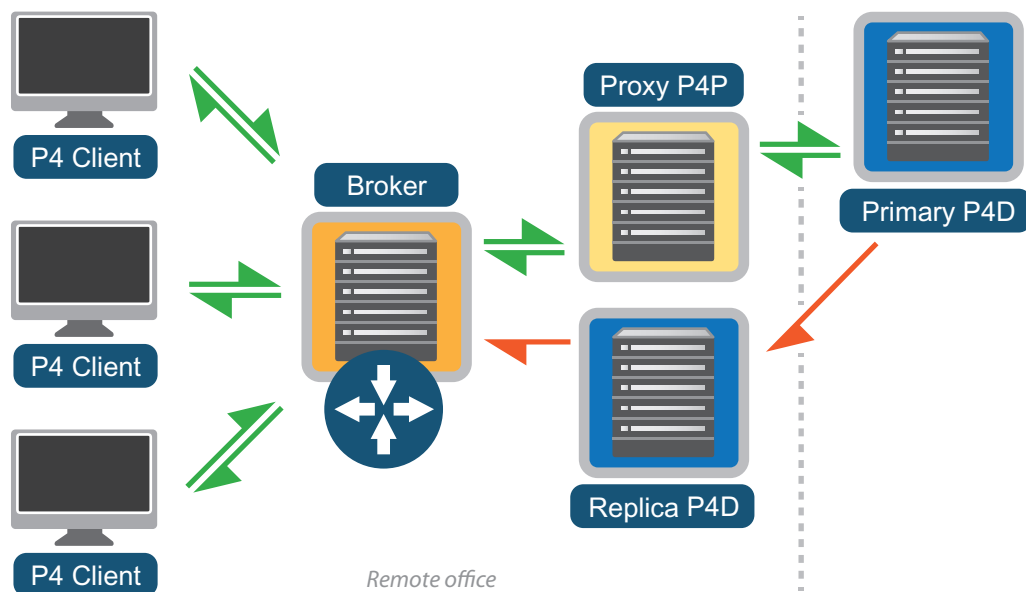


Figure 12: Perforce's distributed architecture

<sup>2</sup> <http://msdn.microsoft.com/en-us/library/ms400682.aspx>

## Perforce

Perforce is built primarily for speed. 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 several thousand users, terabytes of versioned content, and millions of revisions. Perforce's deployment architecture now includes proxies, brokers, replicas and P4Sandbox, and these tools can be tailored to satisfy demanding environments. Replicated servers are particularly useful for supporting automated processes such as aggressive continuous integration; the performance burden of such processes is shifted entirely away from the production server. Tools available for scaling out a Perforce deployment require very little extra configuration in comparison to TFS.

## Benchmarks

Perforce outperforms TFS in common operations such as branching, checking out files, and tagging files. The test results are detailed below.

## Test Results

Table 1: Results for equivalent versioning operations on a small dataset (all time in seconds)

| Operation                           | TFS (2010) | Perforce (2011.1) |
|-------------------------------------|------------|-------------------|
| Add tree                            | 16.4       | 17.1              |
| Lightweight branch                  | 10.3       | 3.2               |
| Checkout tree                       | 14.1       | 1.2               |
| Delete tree                         | 9.3        | 5.0               |
| Label/tag tree                      | 4.4        | 1.2               |
| Merge edits to entire tree          | 20.5       | 31.1              |
| Preview merge (no merge necessary)  | 1.8        | 0.1               |
| Sync entire workspace               | 5.4        | 2.3               |
| Preview sync (no updates necessary) | 0.5        | < 0.1             |
| Restore deleted tree                | 16.9       | 21.8              |

## Test Platform

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

- Amazon EC2 high memory quadruple extra large instance
- Windows Server 2008 R2 Datacenter SP1 64-bit
- 26 ECUs, 8 cores, 68GB RAM
- P4D/NTX64/2011.1
- TFS 2010

## Test Data

The test data for the small run consisted of the Apache web server and Tomcat application server source trees, totaling 46MB and 4,480 files. The medium dataset was four copies of the small dataset. The large dataset comprised 207 large binary files, totaling 632MB in size.

Table 2: Results for equivalent versioning operations on a medium dataset (all time in seconds)

| Operation                           | TFS (2010) | Perforce (2011.1) |
|-------------------------------------|------------|-------------------|
| Add tree                            | 53.6       | 77.1              |
| Lightweight branch                  | 35.3       | 11.3              |
| Checkout tree                       | 59.8       | 6.1               |
| Delete tree                         | 38.3       | 15.0              |
| Label/tag tree                      | 9.6        | 4.5               |
| Merge edits to entire tree          | 67.4       | 109.6             |
| Preview merge (no merge necessary)  | 4.4        | 0.3               |
| Sync entire workspace               | 23.0       | 9.9               |
| Preview sync (no updates necessary) | 0.9        | 0.1               |
| Restore deleted tree                | 56.5       | 76.1              |

Table 3: Results for equivalent versioning operations on a large (binary) dataset (all time in seconds)

| Operation                           | TFS (2010) | Perforce (2011.1) |
|-------------------------------------|------------|-------------------|
| Add tree                            | 47.1       | 46.6              |
| Lightweight branch                  | 1.6        | 0.2               |
| Checkout tree                       | 2.3        | 5.0               |
| Delete tree                         | 6.3        | 0.2               |
| Label/tag tree                      | 0.5        | 0.2               |
| Merge edits to entire tree          | 6.2        | 9.3               |
| Preview merge (no merge necessary)  | 0.9        | 0.3               |
| Sync entire workspace               | 5.6        | 9.3               |
| Preview sync (no updates necessary) | 0.3        | 0.1               |
| Restore deleted tree                | 4.1        | 49.6              |

## Learn More

### Evaluating Perforce

More than 400,000 users at 5,500 companies rely on Perforce for enterprise version management. Perforce encourages prospective customers to judge for themselves during a typical 45-day trial evaluation. Free technical support is included with your evaluation. Get started: [perforce.com/trial](http://perforce.com/trial)

### Scheduling a Demo of Perforce

To learn more about Perforce, schedule an interactive demo tailored to your requirements: [perforce.com/product/demos](http://perforce.com/product/demos)

### Migrating to Perforce

Perforce Consulting Services has experience assisting customers with migrations from various software version management systems. For more information, visit: [perforce.com/consulting](http://perforce.com/consulting)

perforce.com



**North America**  
Perforce Software Inc.  
2320 Blanding Ave  
Alameda, CA 94501  
USA  
Phone: +1 510.864.7400  
[info@perforce.com](mailto:info@perforce.com)

**Europe**  
Perforce Software UK Ltd.  
West Forest Gate  
Wellington Road  
Wokingham  
Berkshire RG40 2AT  
UK  
Phone: +44 (0) 845 345 0116  
[uk@perforce.com](mailto: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](mailto:au@perforce.com)