PERFORCE

Comparison: Perforce and IBM Rational Team Concert

Perforce & IBM RTC

Perforce 2011.1 and IBM Rational Team Concert 3.0.1

This document compares Perforce (version 2011.1) with IBM Rational Team Concert (version 3.0.1)

- Understand Perforce and IBM Rational Team Concert's major feature differences
- Consider the benefits of integrating Perforce with RTC for a flexible ALM solution
- Get a general comparison of the effects of scaling on both systems

Table of Contents

Executive Summary

The choice of version management system (also known as software configuration management, or SCM, in the software industry) profoundly affects those involved in digital asset management, from software developers to artists to managers. An effective 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 IBM Rational Team Concert (RTC) has several other Application Lifecycle Management (ALM) components, including task tracking and build management, this paper will concentrate on the versioning component of RTC.

This document compares Perforce (version 2011.1) with RTC (version 3.0.1). It contrasts major differences in usage, administration, and deployment.

The analysis in this paper suggests that Perforce is a more scalable and flexible version management system that is easier to deploy and maintain. To some extent that conclusion is to be expected, as RTC is intended to be a project and process management system. RTC is a much more complex product than Perforce. The version management component of RTC is not the major focus of the product, and indeed other version management systems, including Perforce, can be plugged into RTC to provide source control while still integrating with RTC task and build management. [1]

In other words, the major goal of RTC is providing project and process management. The individual components of RTC can be replaced as necessary to provide better segment-specific functionality. If the project management aspects of RTC are appealing, then a better solution is to integrate Perforce into the RTC framework.

[1] http://www.ibm.com/developerworks/rational/library/10/integrate-perforce-with-rational-team-concert/index.html

Overview

Attribute	Rational Team Concert (RTC)	Perforce
Release and Process Management	• RTC enforces a streams-based workflow for SCM, and the use of the overall RTC product is based on process management templates.	• Streams provide a powerful yet flexible SCM workflow. • Perforce can be used without streams when an alternative model is appropriate.
Interfaces and Extensibility	• RTC provides rich clients for Eclipse, Visual Studio, and the web client. • The command-line client provides a subset of features. Java and REST APIs are available.	• Perforce provides a standalone GUI as well as rich integrations with Eclipse and Visual Studio. • The command-line client is full featured • APIs are available for C++, Java, and several other languages.
Administration and Support	• Supported by IBM. • Administration requires knowledge of several components including RTC, an application server, and a database. • Upgrade procedures can be complex.	• Technical support, training, and professional services provided by Perforce. • Simple and consistent deployment results in lower administration costs. • Simple upgrade procedures with good interoperability across versions.
Distributed Development	• Forward or reverse accelerator HTTP proxies may improve performance over a WAN, but require extra configuration and dedicated workspaces.	• P4Sandbox offers private repositories for offline work, local branching, and flexible workflows. • P4Proxy offers 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 RTC may require a complex topology with multiple servers and databases. • A single server can be sized to support up to 2,000 users. [2]	• Perforce is deployed in environments with 15,000+ users, millions of files, terabytes of data, and heavy automation. • Perforce Proxy, Broker, replica servers, and Sandbox provide powerful and flexible deployment architectures.

[2] https://jazz.net/library/article/551/

Release and Process Management

RTC

RTC has a fairly rigid data model for SCM.[3] Digital assets are grouped into components for organization and sharing. Every user gets a private stream (server side workspace), and the local working copy is known as a sandbox. Users generally promote or deliver changes to a backing stream. A flow model indicates the preferred way for changes to propagate between streams.

RTC does not support alternative models when a streams-based approach is not applicable. For example, when using the SCM system as a document repository for non-technical users, the streams model offers little value and adds complexity. Or consider a team that uses the SCM system to store multimedia assets. This team may have very specialized procedures and client programs for their data.

Many aspects of working in RTC are bound by a per-project process or workflow, which can be based on a template and customized. For example, preventing a user from submitting a change without linking to a defect requires modifying the project workflow and role (security) settings.[4] In Perforce, such a simple rule can be quickly enforced with a trigger.

Perforce

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.

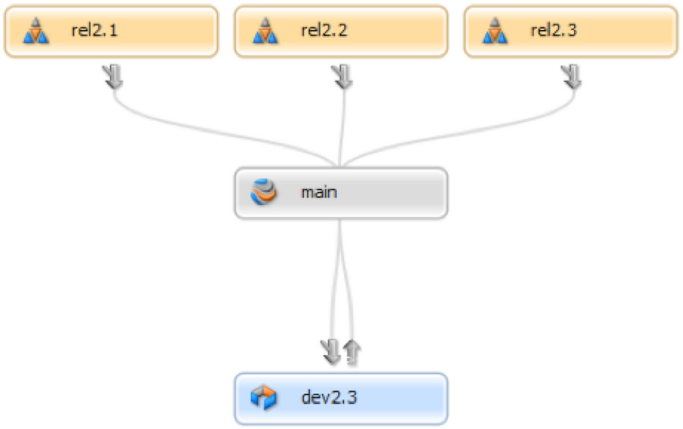

Figure 1: Perforce Stream Graph

The streams model provides guidance and indicates the recommended way to handle concurrent development. However, it is not a strict workflow and the guidelines can be bypassed when necessary.

[3] https://jazz.net/library/content/docs/source-control/faq/index.html#scm_diagram

[4] https://jazz.net/library/article/292

Additionally, Perforce is (and can be) used successfully without streams, providing even more flexibility when alternative workflows and processes are appropriate. Perforce supports Inter-File Branching, which tracks the history and evolution of digital assets down to the file level.[5] Streams build on this branching foundation to provide a collaboration model based on the mainline branching technique.[6] If this model is not desired or appropriate for a particular team, Perforce branching can support any alternative workflow, while still tracking complex merge history, including indirect branching and refactoring. In the example of a team using Perforce as a document repository, there is no enforced model to work around.

Interfaces and Extensibility

RTC

RTC provides rich clients for Eclipse, Visual Studio, and the web client. The command-line client provides a subset of features; responsiveness can be problematic as it is a Java program. Java and REST APIs are available.

Perforce

Perforce provides a standalone GUI as well as rich integrations with Eclipse and Visual Studio. The command-line client is full featured. APIs are available for C++, Java, Perl, Python, Ruby, PHP, Objective-C, and JavaScript.

Administration and Support

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

RTC

RTC is a complex application with several components.[7] An RTC instance includes an application server (usually Apache Tomcat or WebSphere), a database, and RTC itself, and may include Tivoli, integrations and synchronizers, and other components. Depending on deployment size, RTC 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 RTC 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 server and component.

RTC is supported by IBM.

Perforce

Perforce imposes minimal administrative overhead. Perforce deployment is simple and consistent. Upgrade procedures are simple and fast, often requiring nothing but replacing the server binary and running a single upgrade command.

Expert and responsive technical support is a hallmark of Perforce. Aside from technical support, Perforce also offers a full range of training and professional services, including eLearning.

---

[5] http://www.perforce.com/customers/white_papers/interfile_branching

[6] The mainline branching model is based on observed best practices and described fully in Practical Perforce.

[7] https://jazz.net/library/article/496

Distributed Development

RTC

RTC offers no overt support for distributed development. Rather, RTC repositories can share source changes (as well as issues and other artifacts).[8]

HTTP proxy technologies (both forward and reverse acceleration) can be used to improve performance over a WAN.[9] However, at best these technologies will cache file content, not metadata. Switching a working copy (RTC sandbox) is not transparent to the end user; creating a new sandbox is recommended when using a proxy.

Perforce

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

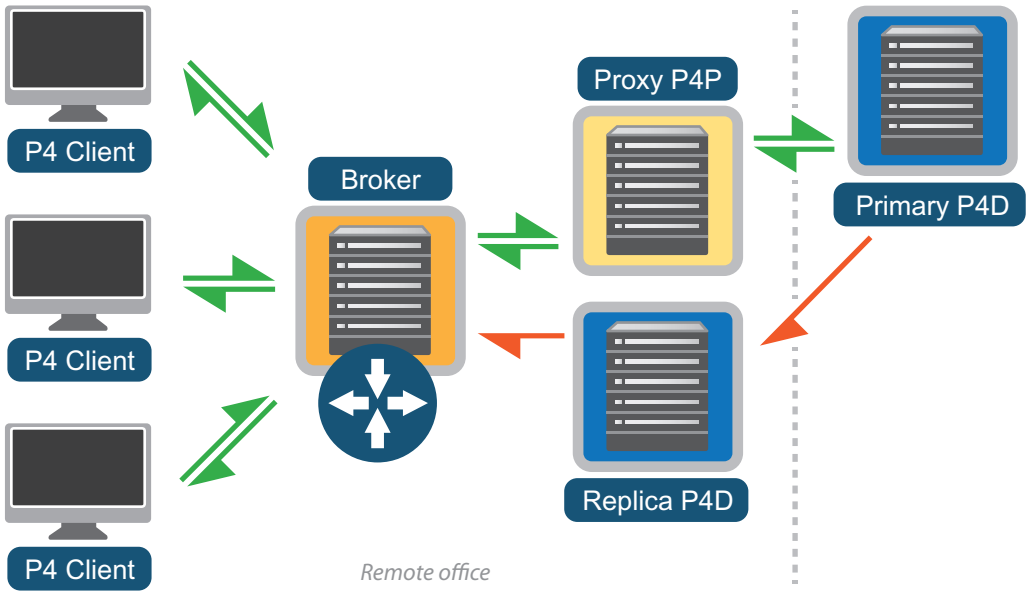

Figure 2: Perforce's distributed architecture

P4Sandbox provides an independent repository for a single user, allowing users to take full advantage of all SCM features in isolation. Thus P4Sandbox supports working without a connection to the central server, creating private local branches, and fast local operations including task switching. P4Sandbox manages the periodic synchronization with the central server, making it easy to learn and transparent to most users.

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.

[8] https://jazz.net/library/article/535/

[9] https://jazz.net/library/article/325/

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

RTC

For even a simple deployment for a small user base, RTC requires 4 GB of RAM. [10] Even at scale, RTC is only sized for approximately 2,000 users, with additional constraints on the number of files per component. [11]

Scaling RTC to support a large user base, distributed work, or heavy data volume can be a daunting task. Examples of scaled-out RTC deployments include multiple pieces of server hardware, application servers, and databases. [12]

Despite the complexity of an RTC deployment, simple data replication is not supported.

Perforce

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.

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.

[10] https://jazz.net/library/article/496

[11] https://jazz.net/library/article/551

[12] http://publib.boulder.ibm.com/infocenter/clmhelp/v3r0/topic/com.ibm.jazz.install.doc/topics/c_topology_example_C.html

Additional Resources

Evaluating Perforce

More than 380,000 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. And because the quality of a technical support organization is better experienced than described, get started by visiting: **http://perforce.com/trial**

Scheduling a Demo of Perforce

To learn more about Perforce, schedule an interactive demo tailored to your requirements.
Visit: **http://perforce.com/product/demos**

Migrating to Perforce

For more information on related software including conversion utilities for migrating to Perforce, see:
http://perforce.com/product/components/migrating_perforce_existing_scm_systems

Perforce Consulting Services has experience assisting customers with migrations from various version management systems: **http://perforce.com/consulting**

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 2AT
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