Perforce Versioning Engine on VMware vSphere®

December 2012

DEPLOYMENT AND TECHNICAL CONSIDERATIONS GUIDE
Table of Contents

Introduction ................................................................................................................................................. 3
VMware and Perforce Overview .................................................................................................................. 3
  Perforce Overview ...................................................................................................................................... 3
  VMware vSphere ........................................................................................................................................ 3
Architecture and Deployment Strategy ......................................................................................................... 5
Testing Process and Results ....................................................................................................................... 6
  Testing Methodology .................................................................................................................................. 6
  Workload Used .......................................................................................................................................... 7
  Hardware and Software Configuration ...................................................................................................... 7
  Results Observed ....................................................................................................................................... 10
Conclusions .................................................................................................................................................. 13
For More Information ................................................................................................................................. 14
Licensing .................................................................................................................................................... 14
Technical Support ...................................................................................................................................... 14
Resources .................................................................................................................................................. 15
Introduction

This document provides information to users who want to run Perforce Versioning Engine (referred to as “Perforce VE” in this paper) on VMware vSphere®. The results of recent testing done jointly by VMware® and Perforce characterize the performance and functionality of Perforce VE on VMware. Finally, a set of best practices are described for utilizing the two product sets together in your datacenter.

VMware and Perforce Overview

This section describes the VMware and Perforce products at a high level and explores the value proposition of using them together.

Perforce Overview

Perforce Software Version Management is one of the most popular products used today for version control and source code management for a wide variety of software assets. It is in use at many major institutions for controlling the versioning and releases of many different artifacts, including the software development content and assets that are used for digital game development.

Users can check-in their original files, and any further changes to their software in all forms, to Perforce VE or servers. They or others can then check those versioned items out to make further refinements to them in a safe way. Perforce software is essential for the smooth functioning of any team of developers or designers that require their product to be stored in different versions in its development and deployment lifecycle.

The architecture of the Perforce system allows it to be deployed in many different configurations to suit different customer needs. Some customers depend on one “central instance” of Perforce VE to control all of their enterprise version management needs. Other customers dedicate a unique Perforce VE to each development project or to each department within the company. There are cache servers that complement the main Perforce VE so that some work can be offloaded from it. These architectural approaches are described further in the Architecture and Deployment Strategy Section below.

VMware vSphere

VMware’s leading virtualization solutions provide multiple benefits to IT administrators and users. VMware virtualization creates a layer of abstraction between the resources required by an application and operating system, and the underlying hardware that provides those resources. A summary of the value of this abstraction layer includes the following:

- **Consolidation**: VMware technology allows multiple application servers to be consolidated onto one physical server, with little or no decrease in overall performance.
- **Ease of provisioning**: VMware virtualization encapsulates an application into an image that can be duplicated or moved, greatly reducing the cost of application provisioning and deployment.

- **Manageability**: Virtual machines may be moved from server to server with no downtime using VMware vSphere® vMotion®. vMotion simplifies common operations such as hardware maintenance and it reduces planned downtime.

- **Availability**: Unplanned downtime can be reduced and higher service levels can be provided to an application. VMware vSphere® High Availability (HA) ensures that in the case of an unplanned hardware failure, any affected virtual machines are restarted on another host in a VMware cluster.

Figure 1. VMware vSphere virtual infrastructure
Architecture and Deployment Strategy

Perforce VE is comprised of a database, one or more depots, and a set of files used for monitoring and backup/recovery procedures.

The depots contain the actual versioned file content. Several depots can be created to contain different types of data with different storage requirements, or simply to provide a convenient organization.

Perforce VE should have three file systems for the components shown above. Maintaining three file systems helps prevent data loss in the event of disk failure, and it enables advanced deployment mechanisms with simple, near-zero downtime backups.

<table>
<thead>
<tr>
<th>Volume</th>
<th>Sample location</th>
<th>Contents</th>
<th>Performance considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metadata</td>
<td>/metadata</td>
<td>P4ROOT with database files</td>
<td>• Optimize I/O for random read/write. Vulnerable to high latency or low bandwidth.</td>
</tr>
<tr>
<td>Logs</td>
<td>/logs</td>
<td>Server logs and active journal</td>
<td>• High performance demands.</td>
</tr>
<tr>
<td>Depot Data</td>
<td>/depotdata</td>
<td>Archive files</td>
<td>• Typically more sequential read/write. Potentially very large amount of data.</td>
</tr>
</tbody>
</table>

Perforce VE may be deployed on several platforms including Windows® and popular Linux® distributions. For the best performance a 64-bit platform is recommended.

Backup and recovery procedures are well documented and a white paper on High Availability and Disaster Recovery is available. See the Resource section later in the document for more information.

The Perforce VE executable is a single binary, p4d, which normally runs as a background process and spawns additional processes and threads to service requests.
Several additional components are available to support remote teams, high data volume, and distributed development, as shown below.

**Figure 3. Perforce VE deployment architecture**

**Testing Process and Results**

To characterize the performance of Perforce VE on VMware vSphere, a series of performance tests were carried out by Perforce. The configuration was tested and the results are summarized below.

**Testing Methodology**

The primary objective of this testing was to determine Perforce VE performance characteristics and to show that Perforce can run resiliently on vSphere and can scale up to large numbers of concurrent users. The testing performed concentrated mainly on performance characterization and did not validate the operation of VMware advanced features including VMware vMotion, VMware HA, VMware vSphere® Distributed Resource Scheduler™, and VMware vSphere® Fault Tolerance (FT) with Perforce deployments in a virtual environment.
Workload Used

The testing performed used the browse benchmark created by the Perforce Software Performance Lab. This benchmark can be configured to simulate varied numbers of Perforce users. Users browsing a Perforce repository are simulated by the browse benchmark’s browsechild application.

The browsechild application continuously browses a Perforce repository tree down many different random paths. Because the browsechild application continuously browses, it simulates the actions of many Perforce users working on tasks, Perforce or otherwise, and thinking time. Many separate instances of the browsechild application can run on several client machines, thereby simulating a very large number of Perforce users. The resulting simulated load on Perforce VE can be significant.

Identical suites of the browse benchmark that varied the number of browsechild instances were run with Perforce VE deployed on both virtual and native machines with nearly identical configurations. The number of cores was also varied similarly on both the virtual and native machines, maintaining the nearly identical configurations between virtual and native.

The suites used the reference11 dataset distributed by the Perforce Software Performance Lab. The reference11 dataset contains definitions for Perforce users, groups, and protections modeled after those at a Perforce production site.

The results from the browse benchmark suites were then evaluated to determine the performance overhead attributable to deploying Perforce VE in a virtual machine on vSphere, and to approximate the number of licensed Perforce users that a configuration might support while delivering a consistent user experience.

Hardware and Software Configuration

Figure 4 provides an overview of the hardware and software configuration used for testing Perforce VE performance.

Figure 4. Perforce VE performance testing configuration
### vSphere Host Configuration

The following table details the vSphere host configuration used for testing Perforce VE performance within vSphere.

**Table 2. vSphere Host Configuration**

<table>
<thead>
<tr>
<th>Host machine</th>
<th>Configuration</th>
</tr>
</thead>
</table>
| Server       | - HP® Proliant™ DL580 G7  
              | - 4 Intel® X7542 2.66 GHz six-core CPUs (24 cores total)  
              | - 512 GB of physical memory  
              | - 2 1 Gbps Ethernet ports  
| Storage      | - 3 146 GB RAID 1+0 LUNs on 6 146 GB 15K RPM SAS drives  
              | - VMFS-5 used for all LUNs  
| VMware       | - vSphere 5.0.0-469512 |

### Perforce VE Virtual Machine Configuration

The table below details the configuration of the virtual machine on which Perforce VE was deployed for testing Perforce performance within vSphere.

**Table 3. Perforce VE virtual machine configuration**

<table>
<thead>
<tr>
<th>Virtual machine</th>
<th>Configuration</th>
</tr>
</thead>
</table>
| Server          | - 4, 8 vCPUs  
                  | - 128 GB of memory  
                  | - 1 1 Gbps Ethernet port  
| Storage         | - 3 146 GB RAID 1+0 LUNs on 6 146 GB 15K RPM SAS drives  
                  | - 32 GB /partition on LUN 2  
                  | - 135.5 GB /db partition on LUN 3  
                  | - 1 GB /boot partition on LUN 2  
| Operating system| - SUSE® Linux Enterprise Server 11 SP1  
                  | - VMware vmxnet3 virtual NIC driver version 1.0.36.0  
| Perforce        | - Perforce Versioning Engine LINUX26X86_64/2010.2/322263 |

### Perforce VE Native Machine Configuration

For a valid performance comparison between Perforce VE on a virtual machine running on vSphere and on a native machine, the same physical HP Proliant DL580 G7 server was used for both configurations. There were two notable differences in the native machine configuration.

First, the native machine was configured with only 128 GB of physical memory so that its memory footprint matched that of the virtual machine within vSphere. The reduced memory was achieved by physically removing 384 GB of memory from the HP DL580 G7 in the native machine configuration.

Second, the operating system was booted from a USB flash drive in the native machine configuration. The MBR of LUN 1 was used by the vSphere installation, so the USB flash drive was used as an alternative for booting the operating system in the native machine configuration.
The following table details the configuration of the native machine on which Perforce VE was deployed for comparing Perforce performance with that on vSphere.

Table 4. Perforce VE native machine configuration

<table>
<thead>
<tr>
<th>Native machine</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>- HP DL580 G7</td>
</tr>
<tr>
<td></td>
<td>- 4 Intel X7542 2.66 GHz six-core CPUs (24 cores total)</td>
</tr>
<tr>
<td></td>
<td>- 8 Hotplug CPUs</td>
</tr>
<tr>
<td></td>
<td>- 128 GB of physical memory</td>
</tr>
<tr>
<td></td>
<td>- 1 1 Gbps Ethernet port</td>
</tr>
<tr>
<td>Storage</td>
<td>- 2 146 GB RAID 1+0 LUNs on</td>
</tr>
<tr>
<td></td>
<td>- 4 146 GB 15K RPM SAS drives</td>
</tr>
<tr>
<td></td>
<td>- 32 GB /partition on LUN 4</td>
</tr>
<tr>
<td></td>
<td>- 146 GB /db partition on LUN 3</td>
</tr>
<tr>
<td></td>
<td>- 1 GB /boot partition on USB flash drive</td>
</tr>
<tr>
<td>Operating system</td>
<td>- SUSE Linux Enterprise Server 11 SP1</td>
</tr>
<tr>
<td>Perforce</td>
<td>- Perforce Versioning Engine LINUX26X86_64/2010.2/322263</td>
</tr>
</tbody>
</table>

browsechild Client Machine Configuration

The browsechild application is a lightweight P4API C++ application. The browsechild application continuously browses a Perforce repository tree down many different random paths. Because the browsechild application continuously browses, it simulates the actions of many Perforce users working on tasks, Perforce or otherwise, and thinking. Since the browsechild application is lightweight, many separate instances can run on just a few client machines, simulating a very large number of Perforce users connecting from a very large number of client machines.

The browsechild instances communicate with Perforce VE using TCP/IP. For this performance testing, Perforce VE and client machines were connected using a single dedicated and isolated 1 Gb network switch. No disk I/O activity results from a browsechild execution, other than reading the ~1 MB browsechild application from /tmp on the client machine, and perhaps reading from some libraries.

For both the virtual and native machine configurations on which Perforce VE was deployed, the browsechild client machines were always native configurations.

The following table details the configuration of a typical native machine on which the browsechild instances were executed for testing Perforce performance.

Table 5. browsechild client machine configuration

<table>
<thead>
<tr>
<th>browsechild client machine</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client</td>
<td>- Dell 2950</td>
</tr>
<tr>
<td></td>
<td>- 2 Intel X5450 3.00 GHz quad-core CPUs (8 cores total)</td>
</tr>
<tr>
<td></td>
<td>- 32 GB of physical memory</td>
</tr>
<tr>
<td></td>
<td>- 1 1 Gbps Ethernet port</td>
</tr>
<tr>
<td>Storage</td>
<td>- 2 146 GB 15K RPM SAS drives</td>
</tr>
<tr>
<td></td>
<td>- 4 GB /tmp partition</td>
</tr>
</tbody>
</table>
Results Observed

The figures below show the best execution time of the browse benchmark for the various configurations. The results for four and eight cores are presented as these configurations might be typical of the virtual machines on which Perforce VE is deployed within vSphere.

Figure 5. Browse benchmark on four cores

Results Observed

<table>
<thead>
<tr>
<th>browsechild client machine</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system</td>
<td>• SUSE Linux Enterprise Server 10 SP3</td>
</tr>
<tr>
<td>Perforce</td>
<td>• browsechild LINUX26X86_64/2010.2/279478</td>
</tr>
</tbody>
</table>

The figures below show the best execution time of the browse benchmark for the various configurations. The results for four and eight cores are presented as these configurations might be typical of the virtual machines on which Perforce VE is deployed within vSphere.
The following table details the performance overhead attributable to deploying Perforce VE on a virtual machine within vSphere for all configurations shown in the figures above. Results in the virtual configuration of using more CPU sockets with a smaller number of cores per socket are highlighted.

Table 6. Performance overhead attributable to deploying Perforce VE on a virtual machine

<table>
<thead>
<tr>
<th>Total cores</th>
<th>Browsing children</th>
<th>Cores per socket</th>
<th>Number of sockets</th>
<th>Virtual run time</th>
<th>Native run time</th>
<th>Percentage overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>12.27</td>
<td>12.59</td>
<td>-2.54</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>12.99</td>
<td>11.44</td>
<td>13.55</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>13.59</td>
<td>14.03</td>
<td>-3.14</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>15.52</td>
<td>11.72</td>
<td>32.42</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>1</td>
<td>4</td>
<td>22.51</td>
<td>22.56</td>
<td>-0.22</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>4</td>
<td>1</td>
<td>22.82</td>
<td>21.77</td>
<td>4.82</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>1</td>
<td>4</td>
<td>51.95</td>
<td>51.83</td>
<td>0.23</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>4</td>
<td>1</td>
<td>53.13</td>
<td>49.62</td>
<td>7.07</td>
</tr>
<tr>
<td>4</td>
<td>32</td>
<td>1</td>
<td>4</td>
<td>107.62</td>
<td>95.41</td>
<td>12.80</td>
</tr>
<tr>
<td>4</td>
<td>32</td>
<td>4</td>
<td>1</td>
<td>111.46</td>
<td>103.14</td>
<td>8.07</td>
</tr>
<tr>
<td>4</td>
<td>64</td>
<td>1</td>
<td>4</td>
<td>202.35</td>
<td>184.43</td>
<td>9.72</td>
</tr>
<tr>
<td>4</td>
<td>64</td>
<td>4</td>
<td>1</td>
<td>201.74</td>
<td>194.71</td>
<td>3.61</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>12.18</td>
<td>11.67</td>
<td>4.37</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>12.48</td>
<td>11.57</td>
<td>7.87</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>13.26</td>
<td>12.05</td>
<td>10.04</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>13.97</td>
<td>11.75</td>
<td>18.89</td>
</tr>
</tbody>
</table>
There are several potential variables that can affect the number of licensed Perforce users that a configuration can support with a consistent user experience. Some of the variables might include:

- Ratio of licensed users to "active" users
- How "active" is a user
- If Secure Sockets Layer (SSL) is used at the Perforce layer. (If so, the number of "active" users that a given configuration can support might be significantly reduced.)
- Speed of the CPUs
- Speed of the I/O subsystem
- Site type as either "metadata heavy" or "file content heavy"
- If the interactive users are competing with heavy automation

There can also be other variables to consider. In light of these potential variables, determining the number of licensed Perforce users that a configuration can support with a consistent user experience – from observing the performance of an artificial benchmark such as the browse benchmark used in this performance testing – is at best an approximation.

For this performance testing, an approximation of the number of licensed Perforce users that a configuration can support with a consistent user experience might be generalized as:

> Each continuously browsing child in the browse benchmark using the reference11 dataset with users, groups, and protections modeled after those at a production site approximates 250 licensed Perforce users.

Relative to the figures shown earlier in this section, this generalization can be used to approximate the number of licensed Perforce users that a virtual machine of a given configuration might support with a consistent user experience.

In the figure that shows the results of the browse benchmark on four cores, the user experience begins to degrade when there are between four and eight continuously browsing children. Therefore, the generalization approximates that a virtual machine with four cores might support 1,000 licensed Perforce users with a consistent user experience.
In the figure that shows the results of the browse benchmark on eight cores, the user experience begins to degrade when there are between eight and sixteen continuously browsing children. Therefore, the generalization approximates that a virtual machine with eight cores might support 2,000 licensed Perforce users with a consistent user experience.

Conclusions

The performance testing results show that deploying Perforce VE on a virtual machine within VMware vSphere performs well. Furthermore, it has potential to reduce cost and increase service levels.
For More Information

Perforce is now making it easy for everyone to take advantage of enterprise version management. The company is headquartered in Alameda, California, with international operations in the United Kingdom, Canada and Australia.

For more information, visit www.perforce.com.

Licensing

There are no relevant exceptions for licensing Perforce on a virtualized platform. A free version of Perforce is available that supports 20 users and 20 workspaces with unlimited files. A commercial license can be purchased to support unlimited users, workspaces, and files.

Technical Support

Contact Perforce Technical Support by email (support@perforce.com) or call one of our offices:

**United States**
Monday to Friday
8:00 A.M. to 6:00 P.M., San Francisco
Phone: +1 510.864.7400
Fax: +1 510.864.5340

**United Kingdom**
Monday to Friday
8:00 A.M. to 5:00 P.M., London
Phone: +44 (0) 845 345 0116
Fax: +44 (0) 845 345 0117

**Australia**
Monday to Friday
8:00 A.M. to 6:00 P.M., Sydney
Phone: +61 (0)2 8912-4600
Fax: +61 (0)2 9929-5590
Resources

For more information about the VMware and Perforce products and solutions discussed in this guide, view the links and references below.

**VMware**

Perforce Versioning Engine, P4D is located within the online VMware Solution Exchange (VSX) at: https://solutionexchange.vmware.com/store/products/the-perforce-versioning-engine-p4d.

**Deployment References**

- VMware vSphere:
- VMware Support and Downloads Web site:
  https://www.vmware.com/support/
- VMware Technology Network (Community) Web site:
  http://communities.vmware.com/community/vmtn
- Performance Best Practices for VMware vSphere 5.0:
- What’s New in VMware vCenter Site Recovery Manager 5.0:

**General Information**

- VMware web site:
  www.vmware.com
- Featured VMware Documentation Sets:
  http://www.vmware.com/support/pubs/
- VMware Licensing Help Center:
  http://www.vmware.com/support/licensing/
- VMware Product Podcasts:
  http://www.vmware.com/technical-resources/podcasts/
- Community, VMware Knowledge Base:
  http://communities.vmware.com/community/vmtn/resources/knowledgebase
- VMware Support Insider:
  http://blogs.vmware.com/kbtv/
- VMware TV:
  http://www.youtube.com/user/vmwaretv
- VMworld TV:
  http://www.youtube.com/user/VMworldTV
• VMware KB TV (external):
  http://www.youtube.com/user/VMwareKB

Perforce

Deployment Best Practices

Knowledge Base articles
• Planning Your Perforce Server Installation:
  http://kb.perforce.com/article/77
• General Performance Recommendations:
  http://kb.perforce.com/article/931
• Recommended Server Hardware Configurations:
  http://kb.perforce.com/article/5

User Conference presentations
• Scaling Servers and Storage for Film Assets:
  http://www.perforce.com/user-conferences/2011/pixar-presentations
• A Perforce Server Dream Machine:

General Information
• Perforce Web site:
  http://www.perforce.com
• Perforce Knowledge Base:
  http://kb.perforce.com
• Perforce System Administrator’s Guide:
• Administration Knowledge Base:
  https://kb.perforce.com/article/498/-admin-tasks
• High Availability and Disaster Recovery Solutions for Perforce: