Technical Report

Deployment and Implementation Guide: Perforce Software on NetApp Clustered Data ONTAP

NetApp and Perforce
April 2013 | TR-4164

Abstract

This technical report describes in detail how to deploy and implement Perforce® software on NetApp® clustered Data ONTAP®. It provides information on best practices, performance of Perforce across NetApp storage protocols, deployment options, and the benefits and storage efficiencies that Data ONTAP offers in a Perforce environment.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

Perforce software is one of the leaders in software configuration management. It is widely deployed by engineering and development teams from small to large enterprises to manage and track different revisions of software components as they are being developed. The value proposition of Perforce is that it has been built to provide maximum control with minimum interference. It revolves around developers and not the other way around. It is also a proven solution for distributed software development because Perforce provides data caching and replication with no special network requirements. Key features of Perforce include software version management for complex projects, workspace management, collaboration, enhanced workflows audit trails, security, access control, and the ability to version any type of data.

Clustered Data ONTAP is NetApp's next-generation storage solution that extends the core strengths of the NetApp Unified Storage Architecture, integrated data protection, and storage efficiency with the addition of massive scalability, increased performance, integrated tiered storage, improved operational efficiency, and a single management system. This third-generation, clustered storage solution provides a foundation for continuous business operation and improved flexibility.

NetApp clustered storage solutions combined with Perforce provide an infrastructure and practices that enable collaboration among distributed development groups, simplify data management for faster development cycles, and improve engineering productivity. Furthermore, NetApp's integrated data protection solutions minimize or eliminate data loss and accelerate recovery of Perforce critical data in the event of hardware failure or catastrophic events.

Perforce is a strong NetApp partner. We collaborate in offering our joint customers a development platform that they can trust. This jointly published paper is an example of this collaboration. The benchmarks and performance results in this paper were conducted by the engineers at Perforce Performance Labs. In addition, FlexClone® and deduplication validation testing was conducted jointly with Perforce engineers.

1.1 Scope

This technical report provides the reader information that can assist in the deployment and implementation of Perforce software on NetApp storage. It contains information relating to performance of Perforce across NetApp storage protocols, deployment options and planning, and NetApp storage efficiencies that are beneficial in Perforce environments. This report will not cover the backup and recovery of Perforce on Data ONTAP. That will be covered in a separate document. Refer to NetApp library for the link.

1.2 Intended Audience

This document is intended for use by individuals who are responsible for architecting, designing, managing, and supporting Perforce applications on NetApp storage. It assumes that readers have some familiarity with NetApp clustered storage solutions and Perforce.

2 Perforce Architecture Overview

Perforce has been designed as a distributed and scalable application. The architecture is multitiered and built using TCP/IP. It is composed of client applications and Perforce versioning service, which manages a database and shared file repositories with data caching, replication, and Git connectivity available as part of the deployment. The client communicates with the Perforce service using TCP/IP.

This section is a high-level description and explanation of the uses of each of these components. For detailed information, refer to the Perforce documentation section on the Perforce website. Figure 1 is a graphical view of the Perforce architecture.

Figure 1) Perforce architecture.
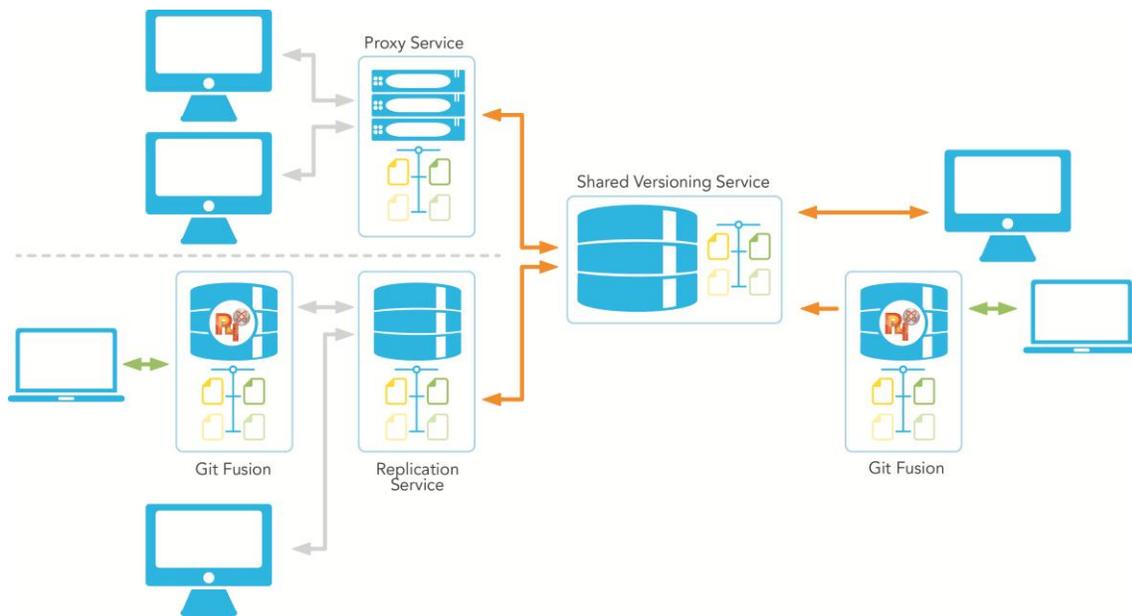
2.1   Perforce Server

The core of the Perforce application is the Perforce versioning service. It manages shared file repositories, referred to as depots, and maintains a database that tracks metadata information related to files in the depots. The Perforce service, p4d, sits on one host, and client workspaces can be anywhere on the network or around the world. Remote sites are supported with data caching and replication. Each command run by a user results in a child p4d process on a Linux® or UNIX® server machine or a thread within the p4s service on a Windows Server® machine. Only the Perforce service accesses the database directly. Accesses to the shared file repositories are done using the Perforce client applications.

2.2   Perforce Client Applications

There are several Perforce client applications that communicate with the Perforce versioning service, including:

- P4, the Perforce command-line client
- P4V, the Perforce visual client
- Git Fusion, a service that allows connectivity for Git clients
- Integrations, or plug-ins customized to work with IDEs and other software using the Perforce application programming interface (P4API)

The Perforce client applications are responsible for transferring files between the shared file repositories to an individual user's workstation, referred to as a *client workspace*. This workspace contains a local copy of files in the depot. These applications allow developers to check in and out files, manage conflicts, create branches, track bugs and change requests, and more.

2.3   Database (Metadata)

The *database files* store *metadata*, including changelists, file attributes, client specs, branch specs, and other data concerning the history and present state of the versioned files. Each database file contains a binary-encoded database table.

2.4 Depots (Versioned Files Repositories)

The depots contain the contents of files versioned in Perforce. Files in the depot are referred to as versioned files. The metadata associated with these files is stored in the Perforce database. Depots can be placed on separate physical storage or network storage for greater flexibility.

2.5 Journal

The journal is a transaction log of updates to the database since creation of the last snapshot, referred to as a "checkpoint" at a certain point in time. It is truncated whenever a checkpoint is done and the previous journal is renamed. Truncation of journal can also be performed without a checkpoint. A database that has been corrupted can be restored from the most recent saved checkpoint, all of the truncated journals since the checkpoint was created, and the current live journal.

2.6 Checkpoint

A checkpoint is a text file that contains all the information necessary to recreate the Perforce database. Checkpoints are a key part of backup and recovery planning and usually reside on a separate storage from the database. The checkpoint is similar to a journal except that all of the transactions are included in the checkpoint in order to create the full database.

2.7 Logs

The *logs* contain commands executed, warnings, errors, and other information to diagnose or investigate problems with the Perforce instance. The level of details for the logs is configurable within Perforce. Log files can also contain auditing information related to user file accesses. Every time a user accesses a file, a record is stored in the audit log file. Having adequate disk space is important when logging is activated.

2.8 Proxy Services

In distributed environments in which the bandwidth can be limited in remote sites, the Perforce proxy services improve performance by caching frequently transmitted file revisions between Perforce applications and the versioning service.

2.9 Replication Services

Perforce offers replication services that allow for duplication of server data from one Perforce server to another server. Replication is unidirectional, and replica servers are read only. The benefits of using Perforce replication include:

- Provides warm standby server
- Reduces load and downtime on primary server
- Provides support for build farms

2.10 Data Flow

In a typical scenario, a developer defines a client workspace, specifying which parts of the repository should be available for local work. The workspace is then synced to transfer files from Perforce to the local work area. The developer then operates on files, performing a series of adds, edits, deletes, merges, and other activities. Developers notify Perforce of their actions either before each action or prior to submit.

All Perforce operations communicate with the versioning service over TCP/IP. Operations will hold read and/or write locks on the Perforce database until metadata is read or updated. The journal and logs are also updated appropriately. Figure 2 shows the data flow in a Perforce environment.

Figure 2) Perforce data flow.
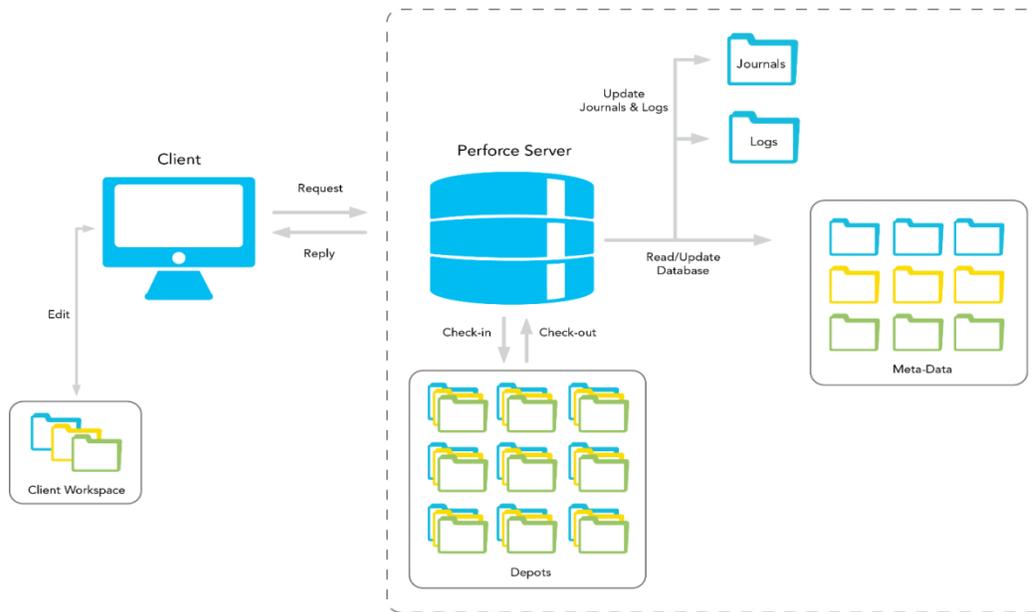
3   Value of NetApp Clustered Data ONTAP in a Perforce Environment

There are numerous challenges in engineering development. Each iteration of the development lifecycle brings with it an enormous amount of data and ever-increasing demands for storage and storage management. Today's environment is marked by increasing pressures to reduce time to market, lower costs, and shorten software development lifecycles. Lack of a reliable, scalable, high-performance storage infrastructure can adversely affect efforts to meet these goals. In addition, the nature of software development itself is changing, with software development projects spanning globally distributed development teams. As a result, 24/7 availability is critical—a factor with which many development shops are still grappling.

NetApp clustered Data ONTAP enables software development organizations to couple multiple storage systems together into a single integrated cluster that can support parts or the entire development infrastructure. The cluster is managed as a single entity, and data can be easily moved between different storage systems transparently to both Perforce and software developers. The clustered Data ONTAP solution offers the following key advantages for Perforce customers:

- Accelerates development with an always-on infrastructure
- Reduces IT costs by managing the entire Perforce storage infrastructure as a single entity
- Decreases risks

3.1   Accelerate Development

In development environments, project requirements can change rapidly, and new projects can increase data and storage requirements. The ability to quickly provision or move data nondisruptively to accommodate these changes is important. Clustered Data ONTAP features such as global namespace, single management system, and nondisruptive volume move can assist in facilitating these requirements.

Dynamically Rebalance Development Workloads

With traditional storage systems, if the Perforce database, depot, and/or journal requires additional storage performance, other projects must be moved off of that storage system, or the data itself must be moved to a system that can provide the additional resources to support the higher performance needs. In

either case, the results are an interruption in data availability and new pathnames for the affected projects.

With clustered Data ONTAP, multiple storage systems are integrated into a single cluster under a global namespace, and projects can be moved between systems transparently to both end users and the Perforce application. With this powerful capability, projects that affect the performance of "hot" data can be transparently moved to other systems, or the data itself can be moved to another system within the cluster. In either case, the load is dynamically rebalanced without disrupting either developers or the Perforce application, optimizing performance access for the data and helping to accelerate project development.

Dynamically Resolve Network Congestion

Clustered Data ONTAP systems also provide the ability to dynamically resolve network congestion issues. With traditional systems, if one of the physical ports to a storage system is congested, the only recourse is to physically move developers to an alternate port. This approach unfortunately disrupts developer access.

With clustered Data ONTAP systems, administrators can simply move the logical interface (LIF) from the congested physical port to another LIF on a less congested port. This action is transparent to developers and to Perforce, again helping to maximize developer productivity and accelerate the development process.

Always-On Infrastructure

Clustered Data ONTAP enables an always-on storage infrastructure that supports development work 7/24/365. For example, when a storage system for Perforce is upgraded, serviced, or retired, the projects on that system can be temporarily moved to other nodes in the system and then rebalanced after completion of the service or upgrade event, without disruption to users.

Clones for Every Developer

NetApp FlexClone thin-cloning technology enables instant, space-efficient clones of production or test data. These clones provide substantial storage savings because they consume space only for blocks that have been changed. With FlexClone, each developer can have a private virtual copy of production or test data for use in that developer's test and development activities.

Instant Backups

Traditional approaches to backups of Perforce data can take substantial time; in the case of tape-based backups, Perforce might be unavailable for a long time. Lengthy disruptions can significantly affect the development process. With NetApp Snapshot™ technology, backups can be done in minutes, eliminating long backup windows.

3.2 Scale Infrastructure, Not Operation Expense

Fast-growing development organizations often find that as the amount of Perforce data increases over time, operating expenses similarly increase. This issue is particularly acute with direct-attached storage (DAS). NetApp clustered Data ONTAP systems minimize the administrative activities and costs associated with growing infrastructures.

Single Point of Administration

A cluster of NetApp systems is managed as an integrated entity, not as a set of independent storage systems. The global management interface enables administrators to move projects between systems and to manage all Perforce storage in the cluster from any node. With this approach, the incremental management burden remains relatively flat as additional systems are added to the cluster.

Transparent Data Movement Simplifies Management

With traditional systems, moving data from one system to another is management intensive. Users need to be informed of the upcoming interruption, administrators need to copy the data, and users must then be informed of the new pathnames. There is also a risk of error throughout this process.

With clustered Data ONTAP, there's no interruption to manage, no user communications are required, and no changes to developer processes are necessary.

Optimize Costs with Tiered Storage

A NetApp cluster can include nodes of varying levels of performance, as well as different types of disks (FC, SATA, and SAS) with varying cost, capacity, and performance characteristics. With clustered Data ONTAP, old Perforce projects can be transparently moved to lower tiers of storage, optimizing costs without affecting developer access.

Unified Storage with Native Protocols

Each system in a NetApp cluster supports native file and SAN protocols, including NFS, CIFS, FC, iSCSI, and FCoE. This approach eliminates the server overhead associated with protocol emulators for NFS or CIFS, and it also eliminates the management complexity associated with deployment and maintenance of these protocol emulators. NetApp is the only vendor in the industry to offer unified storage at scale.

Entire Development Infrastructure on a Single Cluster

With unified storage across a broad range of controllers and storage types, a single integrated NetApp cluster can effectively support the storage infrastructure for Perforce and also build farm software and other downstream workflows. This approach significantly simplifies management and reduces operating costs.

3.3   Reduce Risks

Your software is important intellectual property, and clustered Data ONTAP systems provides powerful data protection capabilities and new levels of flexibility to adapt to change.

Easily Scale

Whether you have a small development shop that might experience rapid growth or a huge shop with continuing growth, a NetApp cluster makes it easy to grow seamlessly as your development efforts expand.

Frequent Backups

With NetApp Snapshot technology enabling quick backups, it is convenient to make frequent backups, protecting more of your developers' work and enabling much more aggressive recovery point objectives.

Simplified Disaster Protection

With traditional storage infrastructures, the complexity of the environment and the available data mirroring solutions make disaster recovery (DR) solutions difficult or impractical to implement.

NetApp SnapMirror® replication technology makes DR practical using an easy-to-implement, robust mirroring solution. Implementing DR can significantly reduce risks for your organization and protect your data in case of a catastrophic event.

4   Perforce Benchmarks

Perforce benchmarks were used to conduct the study across NetApp storage protocols: FC, iSCSI, NFS, and a split configuration in which the Perforce metadata was on iSCSI and depots were on NFS.

The Perforce Performance Lab ran a series of benchmarks against a NetApp clustered Data ONTAP test environment. The same series of benchmarks was performed against each of the NetApp storage protocols. After each set of benchmarks was completed, the test environment was configured for the subsequent protocol. Other than protocol-specific configuration setting, the test environment remained exactly the same. This provided an apple-to-apple performance comparison under the various storage protocols. The Perforce benchmarks used included:

- Browse
- Branchsubmit
- Branchsubmit-edit
- Sync
- Deltas

4.1 Browse Benchmark

The browse benchmark involves a single P4D server and multiple browsechild client machines. Each browsechild instance launched places a load on the server by executing commands that simulate the operational characteristics of the Perforce P4V client. Depending on the configuration, this test can be potentially CPU and network intensive on the P4D server. Varying the settings for the benchmark configuration provides information that gauges how well a computer handles a particular load.

To provide meaningful information about a P4D server environment, the server is preloaded with a sizable database composed of a deep-level repository structure. The reference01 dataset is used for this purpose. The reference01 metadata requires approximately 34GB of disk space in the P4ROOT directory (using the 2012.1 release).

The browse benchmark is directed by the browse script, which resides on the same machine as the P4D server. The browse script deploys and launches instances of the browsechild program on each of a set of user-defined client machines. Each browsechild instance issues a series of commands that repeatedly drills down random paths through the repository without delaying between each browse. A single browsechild instance simulates numerous P4V users, each of which typically thinks about what to browse next. Each browsechild instance is lightweight and requires little CPU time on the client machine. As the benchmark session completes, a log analysis reports statistical information related to the run. The following is a sample output from the browse benchmark's runme phase:

> *Summarizing server log...*
> *64 children over 2 hosts browsed for 199 seconds.*
> *32 children ran concurrently from 2012/06/28 09:59:18 to 2012/06/28 10:00:17.*
> *maximum dirs/second: 9146 at 2012/06/28 09:58:58 (12 concurrent children).*
> *maximum fstat/second: 9147 at 2012/06/28 09:58:57 (11 concurrent children).*
> *maximum filelog/second: 833 at 2012/06/28 09:58:57 (11 concurrent children).*
> *65536 total browses browsed 65456 unique filepaths.*
> *Removing child files from hosts... done.*

4.2 Branchsubmit

The Perforce server (P4D) synchronizes access to its metadata through the use of file locks on the db.* tables. For Perforce tasks that only need to read portions of the metadata, P4D takes read locks on only those db.* tables containing metadata needed by the task. If a task needs to update the metadata within a db.* table, P4D takes a write lock on the db.* table. A read lock on a db.* table can be shared with other read locks on the same db.* table, but a write lock on a db.* table is exclusive of all other locks on that db.* table. In general, P4D minimizes the duration that a write lock is held on a db.* table.

One exception that can result in P4D holding a write lock on a db.* table for an extended duration is the commit portion of a large changelist submission's dm-CommitSubmit phase. Since the commit portion

must be atomic, P4D holds write locks on several important db.* tables for the duration of the commit portion of a changelist submission's dm-CommitSubmit phase. The write locks held block all other tasks that need access to the same tables. It is important that the commit portion of a changelist submission's dm-CommitSubmit phase execute as quickly as possible so that the write locks are released, making the db.* tables available for access to the waiting tasks.

The Branchsubmit benchmark exercises dm-CommitSubmit performance. Using the default configuration, 70,000 files are integrated into a new branch; a changelist list is created and then submitted. Statistics from these actions are extracted for the final report.

By default, the Branchsubmit benchmark executes four phases: a "setup" phase, two "runme" phases in sequence, and then a "cleanup" phase. For this evaluation each phase is run separately. During the "setup" phase, the checkpoint replay time is recorded. Checkpoint replay time reveals CPU and I/O performance. After the "setup" phase, the file system cache is invalidated. This forces the first "runme" phase to perform physical I/O from the back-end storage. Then the second "runme" phase is run manually to record warm cache performance. Since the primary focus of this analysis is in warm cache performance, only the second runme phase results are recorded in this report. The following is a sample output from the first runme phase of the branchsubmit benchmark. The exiting time and files/second commit rate are key performance indicators.

```
Starting branchsubmit runme phase...
Branching files... sleeping... executing... sleeping... done.
  compute phase duration: 4926 milliseconds
  files opened: 70000
Creating changelist... sleeping... executing... sleeping... done.
Submitting changelist... sleeping... executing... sleeping... done.
  elapsed time: 8 seconds
  exiting time (including fsyncs): 2 seconds
  commit duration: 4387 milliseconds
  commit rate: 15956 files/second
Obliterating changelist... sleeping... executing... sleeping... done.
Deleting changelist... sleeping... executing... sleeping... done
```

4.3   Branchsubmit-Edit

The Branchsubmit-edit benchmark is essentially the same as the branchsubmit benchmark with the following exceptions:

- The integrate command does not incorporate the -v (virtual) flag, resulting in the creation of a new branch involving the copy of the branched archive files to the client.
- Edits are performed on the integrated depot files prior to submission.

4.4   Sync Benchmark

The Sync benchmark is similar to the browse benchmark in that it involves a single P4D server and multiple children that are launched from separate client servers. The fundamental difference between the two benchmarks is that a sync operation and not a filelog operation is performed at the end of the random browse. Sync performance can be affected by both network speed and I/O performance of the device where the depots (versioned files) reside. The primary metric of interest is the completion time for a specific amount of synced data by a specified number of children. The following is a sample output from the sync benchmark's runme phase:

Running children on hosts... done.
Summarizing server log...
* 64 children over 2 hosts synced for 181 seconds.*
* 19 children ran concurrently from 2012/06/28 09:38:06 to 2012/06/28 09:38:06.*
* maximum dirs/second: 173 at 2012/06/28 09:37:45 (9 concurrent children).*
* maximum sync/second: 25 at 2012/06/28 09:38:00 (16 concurrent children).*
* 2048 total syncs synced 1899 unique filepaths.*
Total of 16.97 GB where synced from 64 children.

4.5   Deltas Benchmark

The Deltas benchmark includes 10 integrate and 12 sync test scenarios that exercise the **read** capabilities of the storage environment. This is accomplished with the "-n" option in conjunction with Perforce integrate and sync commands. This test can be initiated under a cold cache or warm cache scenario. To void the operating system's file system cache, the file system that contains the Perforce metadata tables is unmounted and remounted. However, this does not negate any caching that might occur on the NetApp controller. Results are recorded in seconds.

Table 1) Sample report output for Deltas benchmark.

| Testname	FC	iSCSI	NFS
integrate -n 001	*x.xx*	*x.xx*	*x.xx*
integrate -n 002	*x.xx*	*x.xx*	*x.xx*
integrate -n 003	*x.xx*	*x.xx*	*x.xx*
integrate -n 004	*x.xx*	*x.xx*	*x.xx*
integrate -n 005	*x.xx*	*x.xx*	*x.xx*
integrate -n 006	*x.xx*	*x.xx*	*x.xx*
integrate -n 007	*x.xx*	*x.xx*	*x.xx*

5 Test Environment

This study was conducted using Perforce version 12.1 (P4D Rev. P4D/LINUX26X86_64/2012.1/490371 [2012/07/02] and P4 Rev. P4/LINUX26X86_64/2012.1/490371 [2012/07/02]). Table 2 shows the Perforce metadata database and depot sizes used for each of the benchmarks.

Table 2) Database and depot sizes.

Benchmark	Metadata (DB) Size	Depot (Versioned Files)
Browse	34GB	None
Sync	9GB	6.4GB
Branchsubmit	34GB	None
Branchsubmit-edit	34GB	21GB
Deltas	3 different datasets: 20GB, 25GB, and 65GB	None

Figure 3 shows the network connectivity of the hardware used for the benchmarks. For the NFS and iSCSI communication with the Perforce server, a 1GbE private network (eth1) was used. The client communication with the Perforce server was on a separate 1GbE network (eth0). For the FC communication, a Brocade switch (4GbE) was used between the Perforce server and NetApp storage. A Cisco Nexus® 5010 (10GbE) switch was used as the interconnect between the clustered FAS3270 HA pair.

Figure 3) Test configuration connectivity.
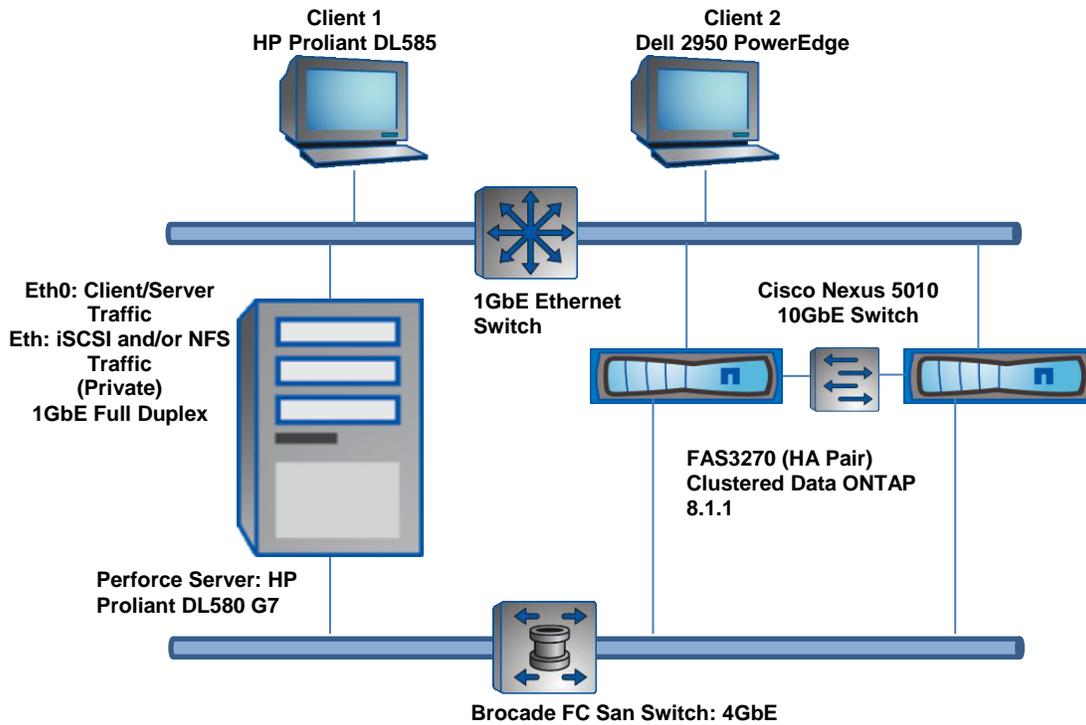
5.1   Perforce Hardware Configuration

The P4D server test machine listed in Table 3 is representative of what might be found at a large Perforce production environment. Although the client test machines are not representative of an individual user's desktop, they are necessary to mimic the high loads that are introduced when numerous users are interacting against the Perforce server environment.

Table 3) P4D server test configuration.

Usage	Machine Specifications	
P4D server	Name	pln4
	Description	Server-class computer
	Model	HP Proliant DL580 G7
	Memory	512GB
	Processors	(4) Intel® Xeon® CPU X7542 @ 2.67GHz (24 cores)

Usage	Machine Specifications	
	Disk drives	(8) 146.8GB 15k SCSI
	Interfaces	(2) 10/1000
	OS	Red Hat Enterprise Linux Server release 6.3 (Santiago)
	Kernel	2.6.32-279.2.1.el6.x86_64
Client test 1	Name	plopt4
	Description	Server-class computer
	Model	HP Proliant DL585
	Memory	32GB
	Processors	(4) AMD Opteron Processor 854 2.8GHz
	Disk drives	(4) 146.8GB 15k SCSI
	Interfaces	(2) 10/1000
	OS	SUSE Linux Enterprise Server 10 (x86_64) - Service Pack 1
	Kernel	2.6.16.46-0.12-smp
Client test 2	Name	plxena
	Description	Server-class computer
	Model	Dell 2950 PowerEdge
	Memory	32GB
	Processors	(2) Quad Core Intel Xeon CPU X5450 @3.00GHz
	Disk drives	(2) 146.8GB 15k SAS (4) 1TB 7.2k SAS
	Interfaces	(2) 10/1000
	OS	SUSE Linux Enterprise Server 10 (x86_64) - Service Pack 3
	Kernel	2.6.16.60-0.54.5-smp

5.2 NetApp Hardware Configuration

NetApp hardware involved a pair of FAS3270 running NetApp clustered Data ONTAP 8.1.1 set up as an HA pair. Volumes were created on a 10-disk aggregate (8 data disks with 2 parity disks). The disks were SAS 15K RPM, 600GB drives. The NetApp storage was dedicated for this testing. The volumes created for the database, journals, depots, and logs resided on one controller head. The NFS mount options used for the Perforce benchmark tests in which Perforce data resided on NFS (unless otherwise indicated) was "nocto, local_lock=all, vers=3".

6   Benchmark Results

This section illustrates the performance of the Perforce benchmarks across NetApp clustered storage protocols. The benchmark test results are from warm cache test runs. While cold cache test runs identify physical I/O performance, most production environments provide a large front-end server capacity, resulting in logical data access from the server's file system cache.

For NFS and the SPLIT (database on SAN and depots on NAS), *local_lock=all and nocto* was specified in the nfs mount options for the NAS volumes. No other customized configurations were used during testing. Essentially, it was a vanilla environment unless otherwise specified (that is, NFS mount options).

6.1   NFS Mount Options Analysis (7-Mode)

During the Perforce on NetApp Data ONTAP operating in 7-Mode performance testing, it was determined that using mount *nocto and local_lock=all* for the Perforce repositories improved NFS performance. Thus, for the testing that required NFS, the database, logs, journals, and/or depots were mounted with the *nocto and local_lock=all*. It is important to note that the *nocto and local_lock=all mount* options should only be used when the mounted data is exclusively accessed by one NFS v3 client machine, in this case, the machine in which the Perforce server is running. Since these mount options affect attribute checking and file locking, respectively, misuse of these options could result in data corruption. For instance, do not use these mount options for client workspaces that are centralized on NetApp storage. In this scenario, the same data can potentially be accessed or modified by two different client machines.

Although the following results are from 7-Mode, this data is still applicable for clustered Data ONTAP. The browse benchmark was used to identify the optimal mount options that will be used for all benchmark test runs during this evaluation. Table 4 shows those results. For these results, lower numbers are better.

Table 4) NFS mount options analysis results (7-Mode).

defaults	local_lock=all	nocto	nocto,local_lock=all
732 sec.	448 sec.	667 sec.	406 sec.

6.2   Browse Results

Figure 4 illustrates the results from the browse benchmark. Lower numbers are better. For the NFS/iSCSI split configuration results, the Perforce database and log files were on iSCSI, and depots and journals were on NFS. The browse benchmark generates a significant amount of logging output during the execution of the test. To maximize performance for the split configuration, the log files were placed on iSCSI.

Note: The NFS client kernel used to get the NFS and split configuration results contained the proposed fix for the NFS client read issue described in the following analysis section.
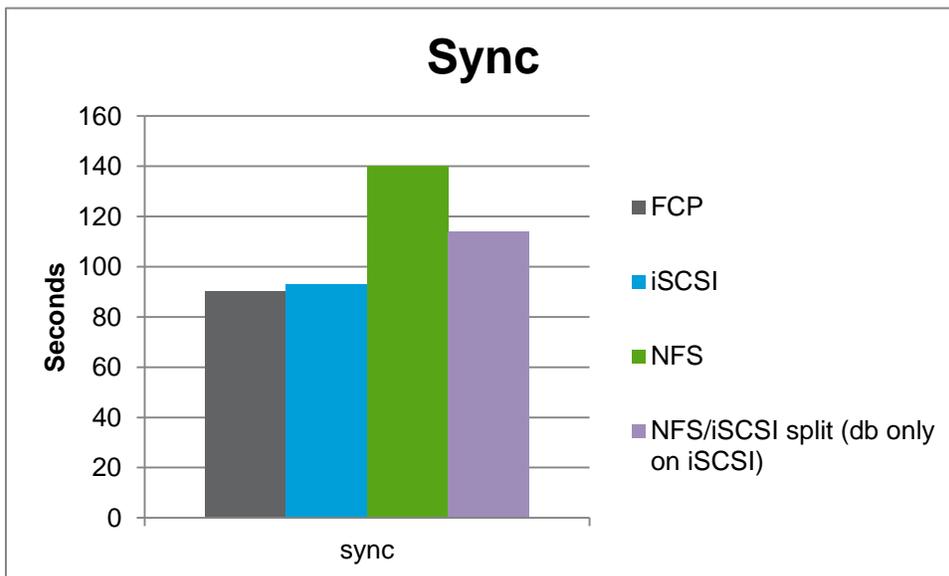
Figure 4) Browse results.
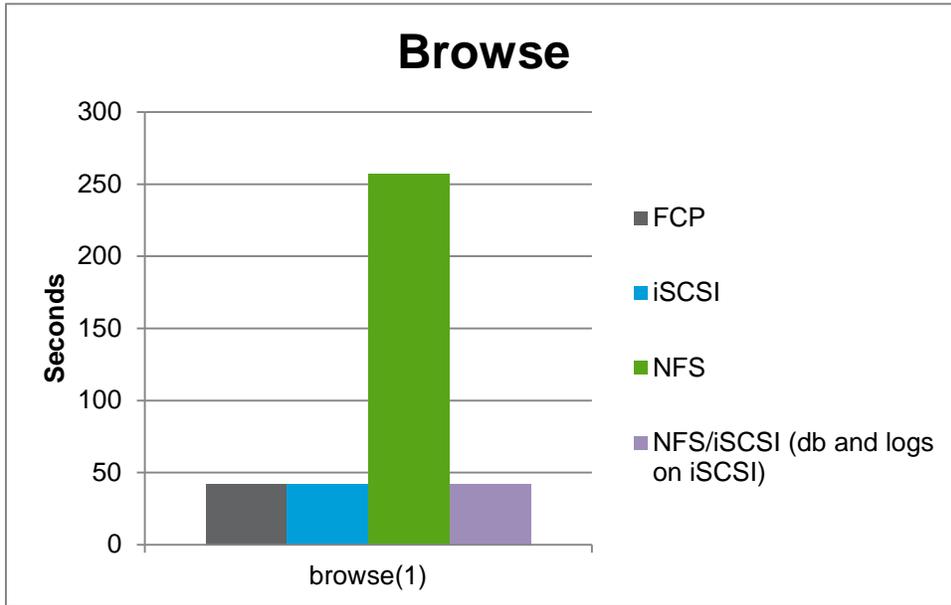
6.3 Sync Results

Figure 5 illustrates the results for the Sync benchmark. For the split configuration, the logs, journals, and depots were placed on NFS and the database on iSCSI. For these results, lower numbers are better.
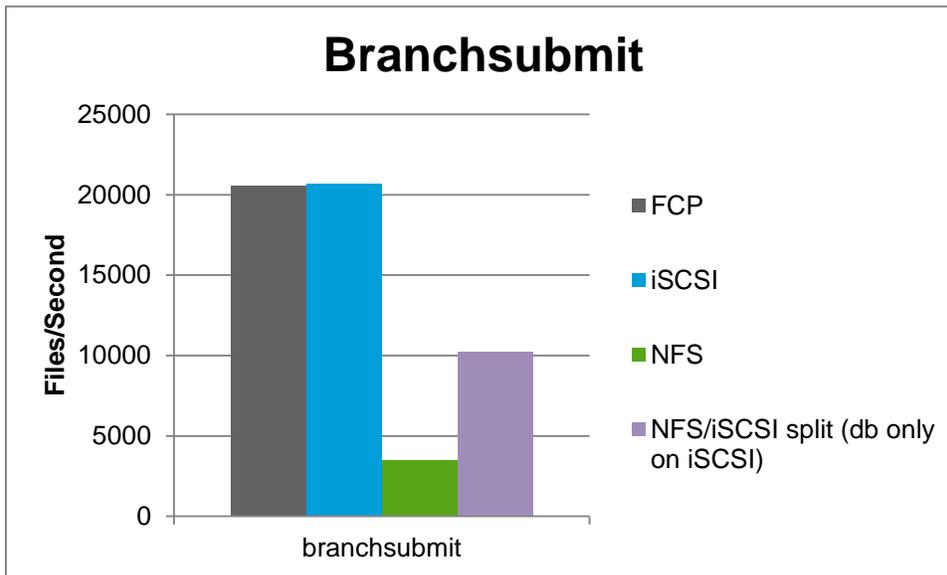
Figure 5) Sync results.

6.4 Branchsubmit Results

Figure 6 illustrates the results for the Branchsubmit results. Results are reported in files per second, and higher numbers are better. This is a single user test performing integration and submits.

Figure 6) Branchsubmit results.

Branchsubmit

6.5   Branchsubmit-Edit

Figure 7 illustrates the Branchsubmit-edit results. Again, these results are reported as files per second, and higher numbers are better. For the Branchsubmit and Branchsubmit-edit tests, journal files are written during write transactions. The results show that placing the journal on iSCSI in a split configuration provided results that were as good as those for FCP and iSCSI. Thus, placing journals on SAN (FC or iSCSI) can improve overall performance.

Figure 7) Branchsubmit-Edit results.

Branchsubmit-Edit
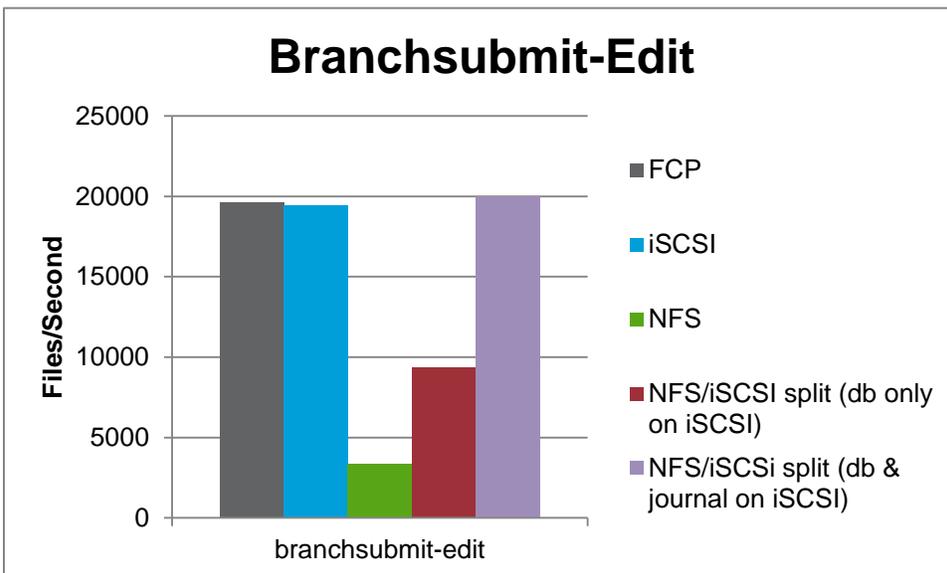
6.6   Deltas Test Results

The following results are from the deltas benchmark. Lower numbers are better. Note that although this benchmark utilizes Perforce integrate and sync commands, they are implemented in **read** mode using the

"-n" option. The Deltas benchmark is an inline serial, single-user command test, compared to the Browse benchmark, which is a multiuser load test.
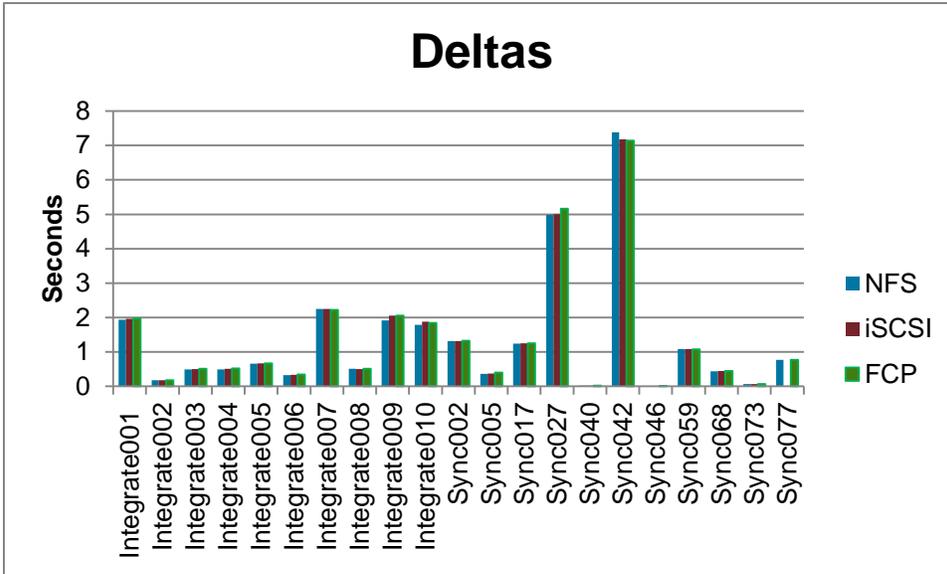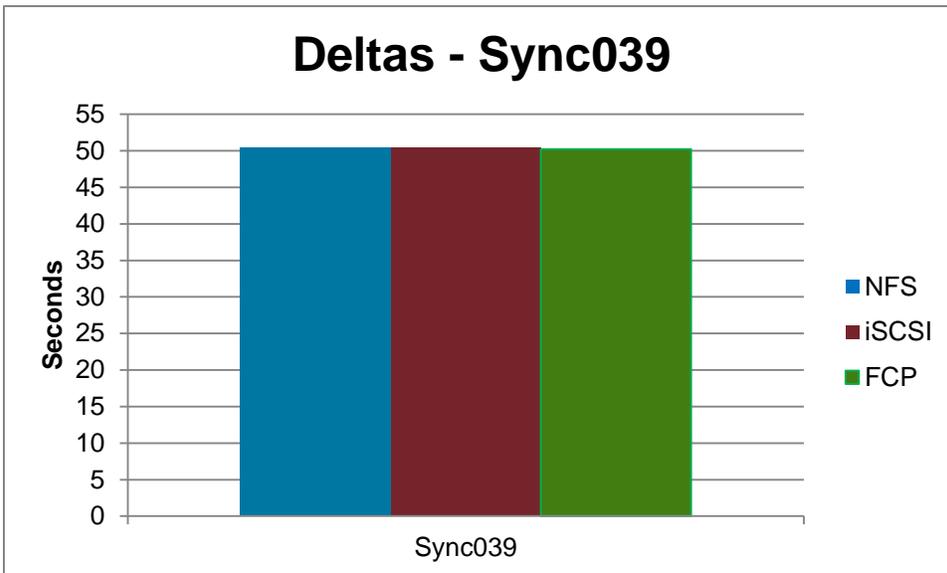
Figure 8) Deltas results.

Figure 9) Deltas Sync039 results.

7   Analysis

When reviewing the test results contained in this report, bear in mind that all of the results reported in this document were obtained from warm cache test runs. Because of this, iSCSI and FCP are able to take advantage of the local server memory (file system cache). Due to the additional overhead of the NFS client, the file system cache advantage becomes less effective, even when implementing *nocto and local_lock* mount options. As a result, there is a significant disparity in most of the test results between iSCSI or FCP and NFS.

7.1 Summary of Performance Benchmark Results

The key takeaways from running the benchmarks include:

- Performance is best when the database, journals, and logs are on FCP and iSCSI protocols, followed by the split configuration, in which the Perforce database resided on iSCSI and the file depots on NFS, and then NFS.
- For the Deltas test, which is a read benchmark, FCP, iSCSI, and NFS are comparable in performance.

7.2 Other Performance Observations

- During this test it was observed that there were issues relating to the Linux NFS client kernel, which is causing the huge performance gap between NFS and FCP/iSCSI protocols. As in the previous section, it has been observed that setting *local_lock=all* causes improvement compared to the defaults. However, in the NFS client kernel, this parameter was being ignored, causing invalidation in the caches and refetching of data from the network storage device. The read issue has been fixed, and the fix will be placed in future release of the Linux kernel. However, the write issue is still being investigated.
- In the split configuration, it has been observed that placing the journals on the FC or iSCSI can bring the results for the split configuration closer to the results for FC and iSCSI. Due to the write issue in the NFS client kernel code, it has affected the results for the split configuration. Thus, placing the journals on an FC or iSCSI LUN will alleviate this issue until this issue has been fixed or resolved.

8 Deployment Options

Perforce database, file depots, journals, and logs can be deployed on NetApp clustered Data ONTAP storage for enhanced flexibility, integrated data protection, and simplified administration. Perforce critical data can be deployed on any NetApp storage protocol: FC SAN, iSCSI, NFS, or a split configuration in which the database and journals are on a SAN and file depots and logs on a NAS. Table 5 highlights the advantages and disadvantages of deploying Perforce critical data on each of NetApp's storage protocols.

Table 5) Pros and cons of Perforce deployment options.

Deployment	Pros	Cons
FC SAN	Performance is fast compared to NAS protocol.	More expensive because it requires a dedicated fiber-optic storage network. Extra management of switches. Backup and recovery are complex and require more steps compared to NAS.
iSCSI	Performance is fast and comparable to FC SAN. More cost efficient than FC SAN because it uses the existing Ethernet infrastructure.	Backup and recovery are complex and require more steps compared to NAS.
NFS	Inexpensive. Simple to manage. Offers ease of use and granularity in backup and recovery operations.	I/O performance is much slower compared to other protocols.

Deployment	Pros	Cons
Split configuration (database on iSCSI or FC and depots on NFS)	Improves Perforce database and journals write performance. Cost efficient if database deployed on iSCSI compared to FC.	Backup and recovery are complex because backups are performed by using two separate protocols while maintaining concurrency between them. However, using NetApp SnapDrive® data management software alleviates this complexity.

9 Planning

Selecting the right storage for Perforce data to address the needs and requirements of a particular deployment is a complex task. The sizing policies are application specific and vary according to the capacity requirements and workload of every project. However, here are some questions to ask potential joint NetApp and Perforce customers to assist in sizing the project and understanding the customer's needs and requirements.

- What are their main storage criteria: performance, data protection, reliability, manageability, and so on?

- If they have an existing Perforce database and file depots, what are the current sizes and their expected rate of growth?

- What is the typical number of Perforce commands run a day?

- How many developers will access the Perforce database and file depots?

- What is their existing infrastructure: platform (servers and storage devices), protocol, and network infrastructure details? Do they have a distributed development environment?

- Is storage to be used solely for Perforce applications?

- What is their current backup and restore mechanism?

The NetApp Unified Storage Architecture offers customers several options for deploying Perforce repositories. Table 6 lists the different NetApp storage protocols to determine which deployment best addresses the challenges of product development, such as performance, heterogeneous access, manageability, cost, and ease of backup and recovery. In terms of performance, FC or iSCSI is the best; however, in terms of manageability, heterogeneous access, and backup and recovery, it might not be the ideal choice. In terms of cost, a NAS solution would be the best choice. However, if the customer wants a balance of all the factors, the best choice is the split configuration in which the database is on iSCSI or FC SAN for performance and the depots, journals, and logs are on NAS for better access, cost, manageability, and backup and recovery. Thus, there are trade-offs when deciding which type of storage to select.

Table 6) Selection criteria across NetApp storage protocols.

Protocol Criterion	FC	iSCSI (Software)	iSCSI and NFS (Split Configuration)	NFS
Performance	Excellent	Excellent	Good	Fair

Protocol / Criterion	FC	iSCSI (Software)	iSCSI and NFS (Split Configuration)	NFS
Direct file access to heterogeneous clients	No	No	Yes	Yes
Manageability	Fair	Fair	Good	Excellent
Cost	Fair	Excellent	Excellent	Excellent
Granularity and ease of backup and recovery	Fair	Fair	Good	Excellent

10 Storage Efficiency

There are two NetApp technologies that can further improve the storage efficiency of Perforce file depots and client workspaces. The use of NetApp deduplication and FlexClone technologies can provide storage savings that can assist in reducing and/or deferring IT costs and improving the productivity of engineers.

10.1 Deduplication

Deduplication is a key feature that can significantly reduce the data footprint of developer workspaces in Perforce environments. NetApp deduplication runs as a background process, and the system can perform any other operation during this process. However, NetApp highly recommends scheduling deduplication when the application is quiescent, possibly at night and on weekends.

Usually developers who work on the same project work on the same set of the files in the depots and would check out a copy of these files (that is, common libraries, sources, binaries) in each of their own workspace areas. This would produce an abundant amount of duplicated data if all developers' workspaces are centralized on NetApp storage. Deduplication removes the redundant blocks, saving storage disk space.

File depots can also benefit from deduplication since branches or differing projects might check in the same set of header files, source code, images, and static libraries that are required for their projects. Deduplication was conducted at Perforce labs on a file depot that contained three copies of large binary data (2GB). Percent savings observed was 79%. **Note:** Deduplication highly depends on the dataset, and results can vary according to your dataset.

Another added benefit of deduplication is when used in conjunction with SnapMirror. Only deduplicated data is replicated at a remote site, which can speed up the transfer and reduce bandwidth utilization.

For more information about NetApp deduplication and compression, see TR-3966: Compression and Deduplication for Data ONTAP 8.1 Operating in Cluster-Mode.

10.2 FlexClone

Developers work with large volumes of data. It can be challenging getting a quick copy of source and prebuilt workspaces. With NetApp FlexClone technology, workspaces can be created in minutes instead of hours.

NetApp FlexClone technology enables instant, space-efficient clones of production or test data to be created easily, minimizing storage requirements, because physical disk space for these clones is required only when data blocks are changed. With NetApp FlexClone, each engineer can have a private virtual

copy of production or test data for use in test and development activities. This also can allow for the Perforce administrators to quickly spin up copies of the production environment for testing purposes.

The foundation of NetApp FlexClone is a writable point-in-time image based on a NetApp Snapshot copy of a volume. The cloned volume stores only the data that changes between the source and clone. The Snapshot copy can already exist, or it can get created automatically as part of the cloning operation. Creating the FlexClone volume takes just a few moments because the copied metadata is very small compared to the actual data. The parent volume can change independently of the FlexClone volume. It can also be split from its parent to create a new standalone volume. Space is used very efficiently, since new disk space is only allocated when there are metadata updates and/or additions to either the parent or the clone volume. While this might sound like magic, this technology has been an integral part of Data ONTAP and deployed for numerous years in productions.

Traditionally, there are two methods for obtaining a Perforce workspace:

- A developer can create a p4 client, populate this client workspace with source code (p4 sync), and build the binaries. This takes an inordinately long time: in one use case, around 45 minutes to download the source code, and another 1–2 hours to build the initial workspace.

- Alternatively, the developer can use a fully populated prebuilt workspace, using hard links back to the existing prebuilt workspace. This approach avoids the build step because it provides prebuilt objects, but is slow and error prone because files are shared among developers (using hard links) as well as with the build daemon. Any user can trash a shared file, and all the other users (including the build daemon) suffer.

Procedure to Create Perforce Workspaces Using FlexClone

FlexClone technology offers an opportunity to provide a developer with a prebuilt workspace in a matter of minutes with much more reliability, because these files are not shared (no hard links) with other workspaces. The generic steps to clone a workspace are as follows:

1. Create volume on NetApp storage system.
2. Make volume accessible using export and mount on host's server.
3. Run "p4 sync" on host to populate the template workspace.
4. Run the build process in the template workspace to generate build artifacts.
5. Create Snapshot copy of volume that contains the prebuilt workspace on NetApp storage system.
6. Create clone from Snapshot copy.
7. Export the cloned volume on NetApp storage system and then mount clone onto host.
8. Change ownership (by this we mean change the ownership of every file/directory in the tree to be owned by the user to which you want to give the volume). Can be done efficiently by creating a script that takes a user ID (uid) as a numeric value and a list of files, which then calls the chown system call for each file, setting the owner to the passed uid. If the number of files or list of files is large, one can use 'xargs' to split the list into multiple invocations of the script.
9. Define a Perforce client to let Perforce know the existence of the new workspace, modifying fields such as Root directory and Owner with user-specific information.
10. Run 'p4 flush' to make the workspace think it has the file content already. (If you are not cloning at the head revision of the data, you can use the template workspace as a revision specifier.) Note: If clone is used only for builds and no edits are needed, then flush is not required.
11. (Optional) Run 'p4 reconcile' to verify that the workspace data is consistent. Ignore any build artifacts that are not kept in the Perforce repository.

Clustered Data ONTAP has support for global namespace, and thus it can provide seamless access to the workspaces for multiple users. The clone can be seen underneath the root volume without the need to explicitly mount the clone on the server.

In a distributed environment, you can use SnapMirror to replicate the Snapshot copies of the built-out workspace on a central storage to the remote sites. Subsequently, clones can be taken of this Snapshot copy at the remote site to provide user workspaces at that site.

Using traditional approaches to create Perforce workspaces can be time-consuming, especially when the workspaces become large. With FlexClone technology, it only takes time for the first full build and for incremental builds. This time is amortized over all the users, because it takes only minutes for each user workspace to be created from this central Snapshot copy of the built-out workspace. This is possible because cloning using FlexClone technology is instantaneous and also has the added advantage that new clones do not take any additional space until changes are made to the workspace.

For more information on NetApp FlexClone technology, refer to TR-3347: A Thorough Introduction to FlexClone.

Considerations

Using FlexClone to create Perforce workspaces can dramatically improve engineering productivity; however, there are some things to consider, which include:

- Cleanup of cloned workspaces in order to not leave numerous abandoned workspaces. Proper way to cleanup requires the following:
  1. Run Perforce 'client –d'.
  2. Remove the contents of workspace directory.
  3. Destroy clone on NetApp storage system.
- If one employs a bisect workflow for continuous integration testing, make sure that the bisect workflows employs the 'p4 sync–p' variant to prevent spamming the server.
- Limited to 32,767 FlexClone copies per volume on Data ONTAP 8.1 and 255 for prior releases of Data ONTAP.
- Certain Perforce operations (that is, flush or sync) require exclusive locks on several important db.* files to preserve atomicity. Commands needing locks on any of the db.* files will be blocked until the command has completed. Running numerous flushes and sync operations can affect performance. However, using NetApp SAN protocols such as FCP and iSCSI for the Perforce database can alleviate this issue.

Perforce software combined with NetApp storage solutions improves engineers' productivity and thus assists in releasing products on time. In addition to using FlexClone to create Perforce workspaces, it can also have other uses to make engineers more efficient, which include testing of product patches, diagnosing an issue using the clone volume without affecting the production environment, and performance testing in which a baseline can be established and subject the clone to various performance parameters.

11 Best Practices

NetApp recommends the following best practices when deploying Perforce on NetApp Data ONTAP storage solutions:

- Create and lay out volumes with recovery and manageability in mind; use separate volumes for Perforce database, depots, journals, and logs. This will make the recovery process easier and possibly assist in diagnosing any performance issue since there is data separation.
- Consider the number of spindles for your volumes. Selection of the number of spindles can affect performance. Setting this number too large or too small can affect overall performance and recovery of disks. Using NetApp default values is a recommended best practice to start. Refer to the Data ONTAP Administration Guide for the defaults.

- When planning disk storage space of volumes or LUNs, reserving space for Snapshot copies must be considered. Refer to TR-3348: Block Management with Data ONTAP FlexVol, FlexClone, and Space Guarantees on how to calculate space reservations for LUNs.

- For best write performance, Perforce metadata and journals should be placed on FC or iSCSI.

- Since performance is comparable for reads across protocols, the file depots could be mounted on NFS volumes for long-term storage scalability and manageability.

- Refer to the NFS mount options analysis, section 6.1 earlier, for NFS v3 options that can help improve performance when deploying Perforce database and depots on NFS.

- If the log level of Perforce application is set to a high value (3 or greater or debug) or the server activity introduces high logging, then this will have a negative impact on NFS performance, again due to the write issue. Thus, placing logs on FC or iSCSI is recommended if performance is the major criterion in the deployment environment.

12 Support Matrix

It is advisable to always check the support matrix for both Perforce and NetApp to determine whether the version of the operating system, hardware, and software required to deploy Perforce and NetApp is supported. For Perforce, refer to the Perforce website for its support matrix. For NetApp, refer to the NetApp Interoperability Matrix for the current qualifications and system requirements for your specific configuration, including the following:

- Operating system version and patch requirements
- HBA models and drivers
- Supported versions of Data ONTAP
- Supported configurations
- SAN booting requirements

13 Conclusions

NetApp continues to offer compelling solutions for Perforce customers with its next generation of storage products. NetApp clustered storage solutions combined with Perforce software provide an infrastructure and best practices that help accelerate product development, improve data protection, and minimize costs. The flexibility and modularity of NetApp unified storage solutions offer high-performance and high-capacity options for development environments. With use of a robust and highly resilient unified storage system such as NetApp, enterprises can be more confident that their Perforce environment remains available 24/7 and does not reduce the productivity of software developers. Innovative technologies such as FlexClone and deduplication further help optimize Perforce environments. NetApp's partnership and collaboration with Perforce help make NetApp's storage solution a safe choice in Perforce environments.

References

- Optimizing Data Storage for Software Development, by Subra Mohan, John Kim, and Erik Mulder: http://www.netapp.com/us/media/wp-7043.pdf
- Case Study: Streamlining Engineering Operations with Perforce SCM, by Subra Mohan: http://www.netapp.com/us/media/wp-7094.pdf
- Data ONTAP Guide https://support.netapp.com/documentation/productlibrary/index.html?productID=30092
- Introducing Perforce http://www.perforce.com/perforce/doc.current/manuals/intro/intro.pdf

- Perforce System Administration Guide
  http://www.perforce.com/perforce/doc.current/manuals/p4sag/p4sag.pdf

Version History

Version	Date	Document Version History
Version 1.0	March 2013	Initial version: Agnes Jacob, NetApp, and Tim Brazil and Randy DeFauw, Perforce

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

Go further, faster®

NetApp®

www.netapp.com