Oracle Database 11g
Direct NFS Client

An Oracle White Paper
July 2007

NOTE:

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

INTRODUCTION

Networked-Attached Storage (NAS) systems have become commonplace in enterprise data centers. This widespread adoption can be credited in large part to the simple storage provisioning and inexpensive connectivity model when compared to block-protocol Storage Area Network technology (e.g. FCP SAN, iSCSI SAN). Emerging NAS technology, such as Clustered NAS, offers high-availability aspects not available with Direct Attached Storage (DAS). Furthermore, the cost of NAS appliances has decreased dramatically in recent years.

NAS appliances and their client systems typically communicate via the Network File System (NFS) protocol. NFS allows client systems to access files over the network as easily as if the underlying storage was directly attached to the client. Client systems use the operating system provided NFS driver to facilitate the communication between the client and the NFS server. While this approach has been successful, drawbacks such as performance degradation and complex configuration requirements have limited the benefits of using NFS and NAS for database storage.

Oracle Database 11g Direct NFS Client integrates the NFS client functionality directly in the Oracle software. Through this integration, Oracle is able to optimize the I/O path between Oracle and the NFS server providing significantly superior performance. In addition, Direct NFS Client simplifies, and in many cases automates, the performance optimization of the NFS client configuration for database workloads.

DIRECT NFS CLIENT OVERVIEW

Standard NFS client software, provided by the operating system, is not optimized for Oracle Database file I/O access patterns. With Oracle Database 11g, you can configure Oracle Database to access NFS V3 NAS devices directly using Oracle Direct NFS Client, rather than using the operating system kernel NFS client. Oracle Database will access files stored on the NFS server directly through the integrated Direct NFS Client eliminating the overhead imposed by the operating system kernel NFS. These files are also accessible via the operating system kernel NFS client thereby allowing seamless administration.

Benefits of Direct NFS Client

Direct NFS Client overcomes many of the challenges associated with using NFS with the Oracle Database. Direct NFS Client outperforms traditional NFS clients, is simple to configure, and provides a standard NFS client implementation across all hardware and operating system platforms.

Direct NFS Client – Performance, Scalability, and High Availability

Direct NFS Client includes two fundamental I/O optimizations to increase throughput and overall performance. First, Direct NFS Client is capable of performing concurrent direct I/O, which bypasses any operating system level caches and eliminates any operating system write-ordering locks. This decreases memory consumption by eliminating scenarios where Oracle data is cached both in the SGA and in the operating system cache and eliminates the kernel mode CPU cost of copying data from the operating system cache into the SGA. Second, Direct NFS Client performs asynchronous I/O, which allows processing to continue while the I/O request is submitted and processed.

Direct NFS Client, therefore, leverages the tight integration with the Oracle Database software to provide unparalleled performance when compared to the operating system kernel NFS clients. Not only does Direct NFS Client outperform traditional NFS, it does so while consuming fewer system resources. The results of a detailed performance analysis are discussed later in this paper.

Oracle Direct NFS Client currently supports up to 4 parallel network paths to provide scalability and high availability. Direct NFS Client delivers optimized performance by automatically load balancing requests across all specified paths. If one network path fails, then Direct NFS Client will reissue commands over any remaining paths – ensuring fault tolerance and high availability.

Direct NFS Client – Cost Savings

Oracle Direct NFS Client uses simple Ethernet for storage connectivity. This eliminates the need for expensive, redundant host bus adaptors (e.g., Fibre Channel HBA) or Fibre Channel switches. Also, since Oracle Direct NFS Client implements multi-path I/O internally, there is no need to configure bonded network interfaces (e.g. EtherChannel, 802.3ad Link Aggregation) for performance or availability. This results in additional cost savings, as most NIC bonding strategies require advanced Ethernet switch support.

Direct NFS Client - Administration Made Easy

In many ways, provisioning storage for Oracle Databases via NFS is easier than with other network storage architectures. For instance, with NFS there is no need for storage-specific device drivers (e.g. Fibre Channel HBA drivers) to purchase, configure and maintain, no host-based volume management or host file system maintenance and most importantly, no raw devices to support. The NAS device provides an optimized file system and the database server administrator simply mounts it on the host. The result is simple file system access that supports all Oracle file types. Oracle Direct NFS Client builds upon that simplicity by making NFS even simpler.

One of the primary challenges of operating system kernel NFS administration is the inconsistency in managing configurations across different platforms. Direct NFS Client eliminates this problem by providing a standard NFS client implementation across all platforms supported by the Oracle Database. This also makes NFS a viable solution even on platforms that don't natively support NFS, e.g. Windows.

NFS is a shared file system, and can therefore support Real Application Cluster (RAC) databases as well as single instance databases. Without Oracle Direct NFS Client, administrators need to pay special attention to the NFS client configuration to ensure a stable environment for RAC databases. Direct NFS Client recognizes when an instance is part of a RAC configuration and automatically optimizes the mount points for RAC, relieving the administrator from manually configuring the NFS parameters. Further, Oracle Direct NFS Client requires a very simple network configuration, as the I/O paths from Oracle Database servers to storage are simple, private, non-routable networks and no NIC bonding is required.

Direct NFS Client Configuration

To use Direct NFS Client, the NFS file systems must first be mounted and available over regular NFS mounts. The mount options used in mounting the file systems are not relevant, as Direct NFS Client manages the configuration after installation. Direct NFS Client can use a new configuration file 'oranfstab' or the mount tab file (/etc/mtab on Linux) to determine the mount point settings for NFS storage devices. It may be preferrable to configure the Direct NFS Client in the mount tab file if you have multiple Oracle installations that use the same NAS devices. Oracle first looks for the mount settings in $ORACLE_HOME/dbs/oranfstab, which specifies the Direct NFS Client settings for a single database. Next, Oracle looks for settings in /etc/oranfstab, which specifies the NFS mounts available to all Oracle databases on that host. Finally, Oracle reads the mount tab file (/etc/mtab on Linux) to identify available NFS mounts. Direct NFS Client will use the first entry found if duplicate entries exist in the configuration files. Example 1 below shows an example entry in the oranfstab. Finally, to enable Direct NFS Client, you must replace the standard Oracle Disk Manager (ODM) library with one that supports Direct NFS Client. Example 2 below highlights the commands to enable the Direct NFS Client ODM library.

Example 1: Sample oranfstab File

```
server:  MyNFSServer1

path:  192.168.1.1
path:  192.168.1.2
path:  192.168.1.3
path:  192.168.1.4
export:  /vol/oradata1 mount: /mnt/oradata1
```

Example 2: Enabling the Direct NFS Client ODM Library

```
prompt> cd $ORACLE_HOME/lib

prompt> cp libodm11.so libodm11.so_stub

prompt> ln –s libnfsodm11.so libodm11.so
```

DIRECT NFS CLIENT – A PERFORMANCE STUDY

In this section, we share the results of a performance comparison between operating system kernel and Oracle Direct NFS Client. The study used both OLTP and DSS workloads in order to assess the performance of Oracle Direct NFS Client under different types of workloads.

Performance Study Overview

The following test systems were configured for this testing:

- **DSS Test System:** For this testing, a 4-socket, dual-core x86_64 compatible database server running Linux was connected to an enterprise-class NAS device via 3 Gigabit Ethernet private, non-routable networks.
- **OLTP Test System:** For this testing, a 2-socket, dual-core x86 compatible database server running Linux was connected to an enterprise-class NAS device via a single Gigabit Ethernet private, non-routable network.

The database software used for both tests was Oracle Database 11g. After mounting the NFS file systems, a database was created and loaded with test data. First, database throughput was measured by the test applications connected to the Oracle database configured to use operating system kernel NFS. After a small amount of reconfiguration, the test applications were once again used to measure the throughput of the Oracle database configured to use Oracle Direct NFS Client.

Performance Study - Database Overview

The database used for the performance study consisted of an Order Entry schema with the following tables:

- **Customers.** The database contained approximately 4 million customer rows in the customer table. This table contains customer-centric data such as a unique customer identifier, mailing address, e-mail contact information and so forth. The customer table was indexed with a unique index on the customer identification column and a non-unique index on the customer last name column. The customer table and index required approximately 3GB of disk space.
- **Orders.** The database contained an orders table with roughly 6 million rows of data. The orders table had a unique composite index on the customer and order identification columns. The orders table and index required a little over 2 GB of disk space.
- **Line Items.** Simulating a customer base with complex transactions, the line item table contained nearly 45 million rows of order line items. The item table had a three-way unique composite index on customer identification, order identification and item identification columns. The item table and index consumed roughly 10 GB of disk space.
- **Product.** This table describes products available to order. Along with such attributes as price and description, there are up to 140 characters available for a detailed product description. There were 1 million products in the product table. The product table was indexed with a unique index on the product identification column. The product table and index required roughly 2 GB of disk space.
- **Warehouse.** This table maintains product levels at the various warehouse locations as well as detailed information about warehouses. The warehouse table is indexed with a unique composite index of two columns. There were 10 million rows in the warehouse table and combined with its index required roughly 10 GB.
- **History.** There was an orders history table with roughly 160 million rows accessed by the DSS-style queries. The history table required approximately 30 GB of disk space.

Performance Study – DSS-Style Performance Analysis

To simulate the I/O pattern most typical of DSS environments, Oracle Parallel Query was used during this test. The workload consisted of queries that required 4 full scans of the orders history table—a workload that scans a little over 640 million rows at a cost of slightly more than 100 GB of physical disk I/O.

The network paths from the Oracle Database server to the storage was first configured with the best possible bonded Ethernet interfaces supported by the hardware at hand. After executing the test to collect the operating system kernel NFS performance data, the network interfaces were re-configured to a simple Ethernet configuration—without NIC bonding. Next, the workload was once again executed and performance data gathered.

Better Scalability

Figure 1 shows the clear benefit of Oracle Direct NFS Client for DSS-style disk I/O. Although both the operating system kernel NFS and Oracle Direct NFS Client both delivered 113 MB/s from a single Gigabit network path, adding the second network path to the storage shows the clear advantage of Oracle Direct NFS Client —which scaled at a rate of 99%. The operating system NFS with bonded NICs on the other hand delivered only 70% scalability. The net effect was approximately 40% better performance with Direct NFS Client than with the operating system kernel NFS. Most importantly, the Direct NFS Client case was much simpler to configure and could have easily been achieved using very inexpensive (e.g. non-managed) Ethernet switches.

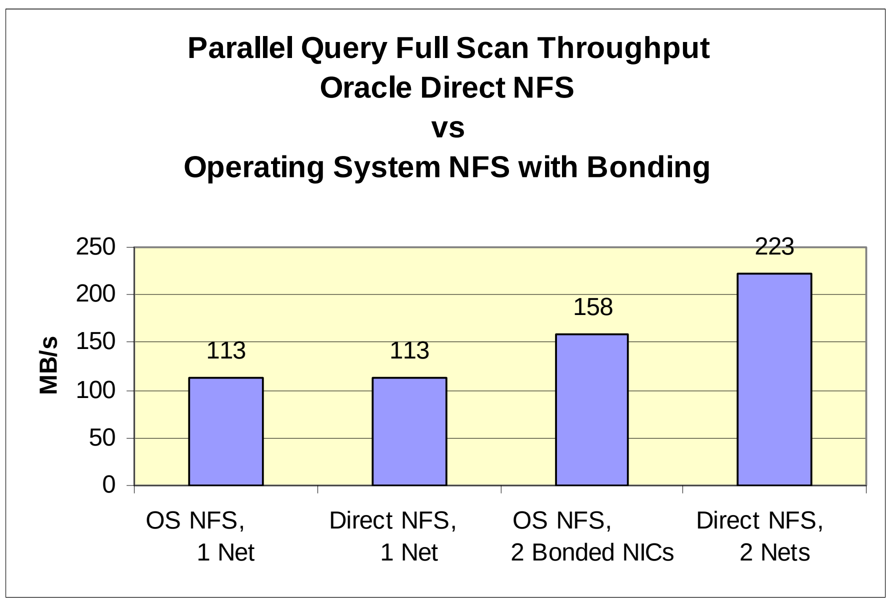

Figure 1: DSS-Style Throughput Comparison—Oracle Direct NFS Client versus Operating System NFS

“Bonding” Across Heterogeneous NIC

As mentioned earlier in this paper, one of the major pitfalls of configuring bonded NICs is the fact that more costly Ethernet switches are required. Another subtle, yet troubling requirement of bonding is that it is generally necessary to configure homogeneous NICs for each leg of the bonded interface. Depending on what hardware and operating system is being used, homogeneous NICs might stipulate both the same manufacturer and/or same host connectivity (e.g., all PCI or all motherboard). That can result in unusable hardware resources. For instance, most industry standard servers come from the factory with dual-port (and sometimes more) Gigabit Ethernet support right on the motherboard. However, it is generally not possible to incorporate these interfaces into a bonded NIC paired with PCI-Express NICs. This, however, is not an issue with Oracle Direct NFS Client.

With Oracle Direct NFS, all network interfaces on a database server can be used for both performance and I/O redundancy, regardless of manufacturer or how they connect to the system (e.g. motherboard, PCI). To illustrate this point, the *oranfstab* file was modified on the test system to include one of the Gigabit Ethernet interfaces on the motherboard. Configured as such, Oracle Direct NFS Client was load-balancing I/O requests to the NAS device across 2 PCI-Express NICs and one NIC on the motherboard. Figure 2 shows how Direct NFS Client was able to exploit the additional bandwidth delivering 97% scalability to achieve full table scan throughput of 329 MB/s.

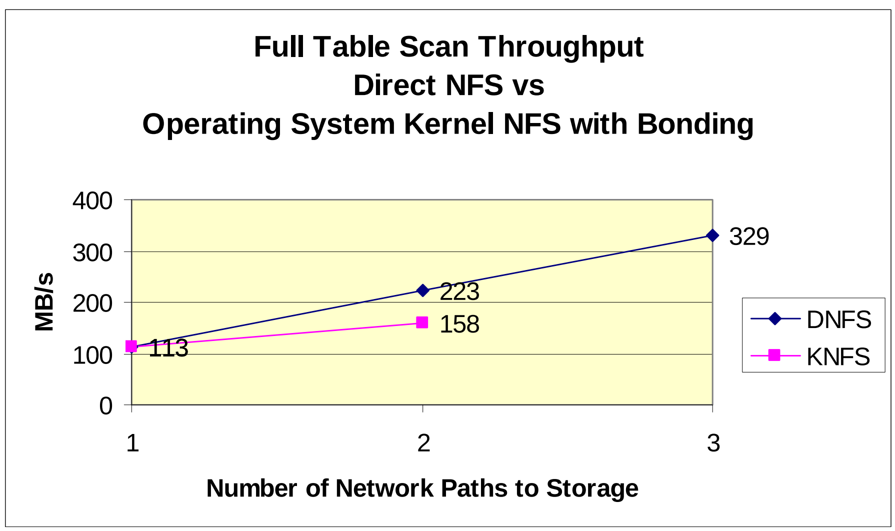

Figure 2: Oracle Direct NFS Client Sequential Read I/O Scalability

Reduced Resource Utilization

One of the main design goals of Oracle Direct NFS Client is improved resource utilization. Figure 3 shows that with a single network path to storage the operating system kernel NFS requires 12.5% more kernel-mode CPU than Oracle Direct NFS Client. This 12.5% overhead is rather unnecessary since both types of NFS exhibited the same 113 MB/s throughput with 1 network path to storage. Moreover, the percentage of processor utilization in system mode leaps 3.6 fold to 32% when going from one to 2 network paths. Using Oracle Direct NFS Client, on the other hand, results in only a 2.9 fold increase when going from one to two networks (e.g. from 8 to 23%) – a 25% improvement over the operating system kernel result.

The test system was configured with a total of 4 network interfaces, including 2 on the motherboard and 2 connected via PCI-Express. In order to reserve one for SQL*Net traffic, a maximum of 3 networks were available for Direct NFS Client testing and 2 for bonded operating system kernel NFS. To that end, Figure 3 also shows that the Direct NFS Client system-mode CPU overhead with 3 network paths to storage was 37%--only 6% more than the 35% cost of 2 network paths with operating system kernel NFS. It may be noted that, with only 6% more overhead, the Direct NFS Client case was delivering 108% more I/O throughput (i.e. 329 vs. 158 MB/s) than the operating system kernel NFS with 2 network paths.

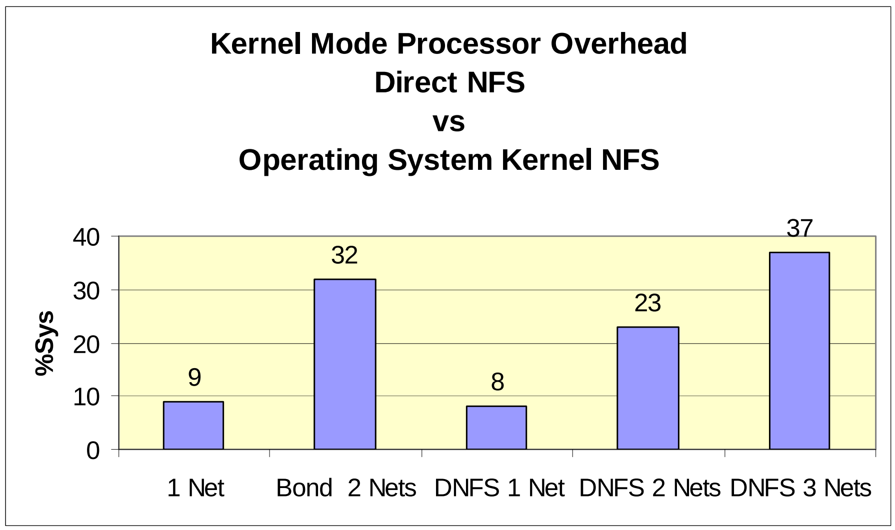

Figure 3: DSS-Style Testing. CPU Cycles Consumed in Kernel Mode.

The improved efficiency of Oracle Direct NFS Client is a very significant benefit. Another way to express this improvement is with the *Throughput to CPU Metric (TCM)*. This metric is calculated by dividing the I/O throughput by the kernel mode processor overhead. Since the ratio is throughput to cost, a larger number represents improvement. Figure 4 shows that Oracle Direct NFS Client offers significant improvement—roughly 85% improvement in TCM terms.

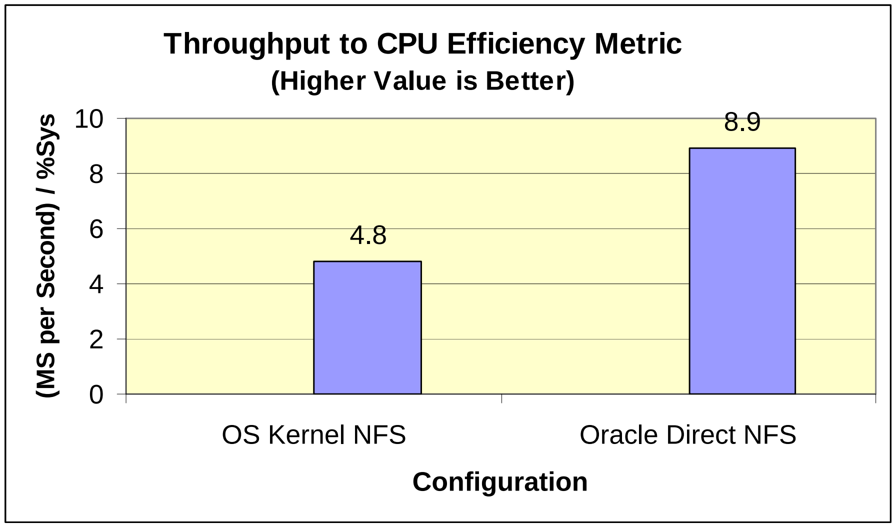

Figure 4: Throughput to CPU Metric Rating of Oracle Direct NFS Client vs Operating System Kernel NFS

Performance Study - OLTP Performance Analysis

In order to compare OLTP-style performance variation between operating system kernel NFS and Oracle Direct NFS Client, the schema described above was accessed by a test workload written in Pro*C. The test consists of connecting a fixed number of sessions to the Oracle Database 11g instance. Each Pro*C client loops through a set of transactions simulating the Order Entry system.

This test is executed in a manner aimed at saturating the database server. The Linux `uptime` command reported a load average of nearly 20 throughout the duration of the run in both the operating system kernel NFS and Oracle Direct NFS Client case. Load average represents the sum of processes running, waiting to run (e.g., runable) or waiting for I/O or IPC divided by the number of CPUs on the server. A load average of nearly 20 is an overwhelming load for the database server. With that said, even a server with some idle processor cycles can have extremely high load averages since processes waiting for disk or network I/O are factored into the equation.

The OLTP workload was the first to saturate the storage in terms of IOPS, but only in the Direct NFS Client case. Without Direct NFS Client, the workload saturated the database server CPU before reaching the throughput limit of the storage. Due to the improved processor efficiencies of Direct NFS Client, the database server consistently exhibited 3% idle processor bandwidth while delivering higher throughput than the CPU-bound case without Direct NFS Client.

Figure 5 shows that the test case configured with Oracle Direct NFS Client was able to deliver 11% more transactional throughput.

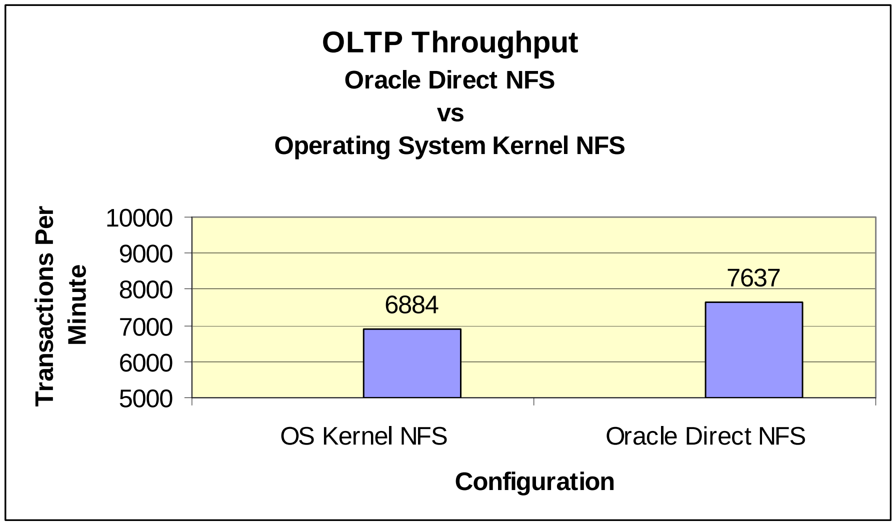

Figure 5: OLTP Throughput Comparison. Oracle Direct NFS Client yields 11% OLTP performance increase.

Figure 6 shows data from the Oracle statspack reports that can be compared to real-world environments. Figure 6 shows that physical I/O was consistently improved by using Oracle Direct NFS Client. As described above, the OLTP test system was a 2-socket dual-core x86 compatible server, therefore physical I/O rates in excess of 5,000 per second are significant.

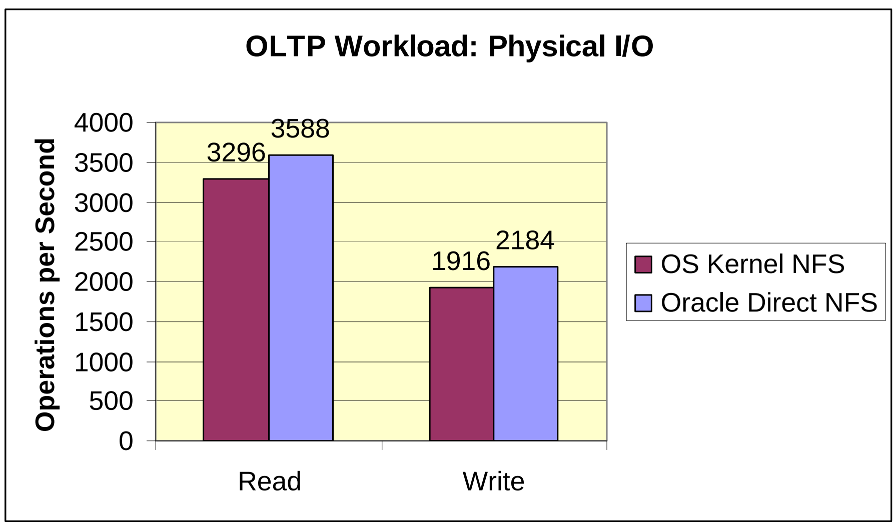

Figure 6: Oracle Statspack Data for Physical Read and Write Operations

Performance Study – Summary Analysis

These performance tests show that the use of Oracle Direct NFS Client improves both throughput and server resource utilization. With Oracle Direct NFS Client both DSS and OLTP workloads benefited from reduced kernel-mode processor utilization. Since DSS workloads demand high-bandwidth I/O, the case for Oracle Direct NFS Client was clearly proven where the gain over operating system kernel NFS was 40%. OLTP also showed improvement. By simply enabling Direct NFS Client, the processor-bound OLTP workload improved by 11%. These performance results are only a portion of the value Oracle Direct NFS Client provides. Oracle Direct NFS Client also improves such features as Direct Path loads with SQL*Loader and External Tables. RMAN also requires high bandwidth I/O so it too will benefit from Oracle Direct NFS Client.

CONCLUSION

Decreasing prices, simplicity, flexibility, and high availability are driving the adoption of Network-Attached Storage (NAS) devices in enterprise data centers. However, performance and management limitations in the Network File System (NFS) protocol, the de facto protocol for NAS devices, limits its effectiveness for database workloads. Oracle Direct NFS Client, a new feature in Oracle Database 11g, integrates the NFS client directly with the Oracle software. Through this tight integration, Direct NFS Client overcomes the problems associated with traditional operating system kernel based NFS clients. Direct NFS Client simplifies management by providing a standard configuration within a unified interface across various hardware and operating system platforms. The tight integration between Direct NFS Client and the Oracle database vastly improves I/O performance and throughput, while reducing system resource utilization. Finally, Direct NFS Client optimizes multiple network paths to not only provide high availability but to achieve near linear scalability by load balancing I/O across all available storage paths.

Oracle Database 11g Direct NFS Client
July 2007
Authors: William Hodak (Oracle), Kevin Closson (HP)

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com