NETAPP TECHNICAL REPORT

NetApp Performance Acceleration Module Oracle OLTP Characterization

Dean Brock, NetApp
January 2009 | TR-3753

A CHARACTERIZATION OF ORACLE DATABASE 11*G* OLTP PERFORMANCE USING NETAPP PERFORMANCE ACCELERATION MODULE TECHNOLOGY

TABLE OF CONTENTS

1 INTRODUCTION

The NetApp® Performance Acceleration Module (PAM) combines hardware and software components that can significantly increase the performance of random read–oriented workloads using NetApp storage. PAM is implemented in Data ONTAP® 7.3 using custom software features, called FlexScale™, combined with intelligent DRAM-based read cache modules added to the storage controllers. Up to five modules can be configured in the PCI Express slots of a storage controller and presented as a single 80GB read cache. The PAM module tested and characterized in this technical report was a PCI Express card providing 16GB of extended cache memory per storage controller. The additional PAM cache is especially valuable for optimizing random read–intensive applications such as database OLTP workloads. Use of PAM modules facilitates reduced I/O latency, greater application concurrency, and better end-user response times. In this study, we used an Oracle® Database 11*g*R1 and OLTP workload running on a standard Red Hat Enterprise Linux® 4 Update 4 server to demonstrate the performance benefits of reduced random-read I/O latencies when using PAM.

2 EXECUTIVE SUMMARY

The objective of this technical report is to provide PAM performance characteristics that enable field engineers and partners to make informed decisions about deploying PAM in their Oracle production environments. To accomplish this objective, the transactional throughput and I/O latency effects of implementing PAM were measured in three dimensions:

- OLTP database size (number of warehouses)
- Oracle SGA size (GB of server RAM)
- Workload generators (number of users)

Overall, we found that enabling PAM on the FAS3140 active-active configuration improved OLTP transaction throughput a minimum of 6% up to a maximum of 35% depending on database, SGA, and workload sizes. The key factor in achieving these OLTP workload performance improvements was the significant reduction in NFS average read latency. Across all three workload dimensions, enabling PAM reduced random-read I/O latency by a minimum of 28% up to a maximum of 40%.

For further details, refer to the results and charts in section 3.

> We recommend that the data outlined herein be used for comparison purposes and the individual data points not be considered as absolutes. It must be recognized that enterprise data systems can vary greatly in terms of complexity, configuration, and application workload, impacting both performance and functionality.

2.1 TEST CONFIGURATIONS

The test bed consisted of an HP ProLiant 580 G5 server and a NetApp FAS3140 active-active configuration storage system with 112 X 144GB 15K RPM FC disks. The server software environment included Red Hat Enterprise Linux version 4 update 4 running Oracle Database 11*g*R1 on the host server and Data ONTAP 7.3 on the storage controllers. Each storage controller was configured with a 16GB PAM hardware component. PAM was enabled or disabled in accordance with the objective for each test. FlexScale is the software that drives the NetApp PAM functionality. Default FlexScale parameters were used for all tests:

flexscale.lopri_blocks	off	no caching of recent writes and large sequential reads
flexscale.max_io_qdepth	512	sets max I/O queue depth (do not change)
flexscale.normal_data_blocks	on	cache user and system data blocks

NetApp FlexShare™ quality-of-service functionality combined with PAM intelligent caching allows you to set caching policies on specified volumes, adding or subtracting from the system-wide mode set with the FlexScale options. For further information about FlexShare, refer to Appendix H. FlexShare priority scheduling was set to “off” for all tests. The database server was configured with two quad-

port Intel® Pro/1000 Gigabit Ethernet NICs. Groups of four GigE links were directly connected to each of the FAS3140 storage controllers. Oracle DNFS (NFSv3 client) storage I/O protocol and automatic load-balancing were used to spread the transaction load evenly across each group of four GigE links. During testing we made sure that hardware resource bottlenecks were avoided in all areas, including database server CPU, physical memory, and network capacity. Additional details of the hardware and network configuration can be found in section 4.

2.2 DATABASE WORKLOAD DESCRIPTION

The test workload simulated an On-Line Transaction Processing (OLTP) system. During the testing, benchmark scripts and executables (workload generators) simulate users driving an OLTP-type load consisting of a steady stream of random-read and random-write operations (approximately 75% reads and 25% writes) using an 8KB request size against the test database. This workload was designed to emulate the real-life activities of a wholesale supplier's order processing system in which inventory is spread across several regional warehouses. The benchmark was run in server-only mode, which means that all user-client processes and the Oracle Database 11*g*R1 instance were executing on the same host system. All user processes were running without "think time" (that is, the users continually submit transactions without simulating any delay between transactions). In terms of measured database throughput, the metric of interest was defined as the number of completed new order transactions processed per minute. Throughout this document, this measurement is referred to as "order entry transactions" ("OET").

The physical size of the database for all tests was approximately 600GB, representing the data storage for 6,000 regional warehouses. The test database was loaded with data and then a Snapshot™ copy created. We used this Snapshot copy to restore the database to its initial state before each test to make sure all tests started with the identical set of database files. Run-to-run variation for identical runs was approximately +/–1%.

The test procedure used for all tests consisted of the following steps:

1. Disable PAM on each storage controller ("options flexscale.enable off").
2. Shut down the Oracle Database instance to clear the SGA.
3. Restore the Snapshot copy of the freshly loaded database volumes.
4. Start up the Oracle Database instance on the server.
5. Execute transactions during a 30-minute "warm-up" time period.
6. Execute transactions during a 30-minute "measured" interval.
7. Enable PAM on each storage controller ("options flexscale.enable on").
8. Repeat steps 2–6.

The data and statistics presented in this paper were recorded during the last 30 minutes (the "measured" interval) of each test.

3 RESULTS SUMMARY

The following charts demonstrate the effect of enabling PAM on OLTP throughput and NFS read latencies depending on the relationship between data access patterns, working-set size, and total cache size. In these charts and discussions, the Oracle system global area (SGA) is the total amount of server RAM reserved for database data buffers, data definitions, redo log buffers, and other structures required by the Oracle Database instance.

3.1 TRANSACTION RATE VS. DATABASE SIZE

For this series of tests the Oracle SGA was held at a constant 12GB, and the workload generators (users) were held constant at 140. We then varied the number of warehouses accessed by these 140 users from a minimum of 1,000 to a maximum of 6,000 in increments of 1,000. The idea was to increase the size of the working set used by the 140 users while keeping the size of the Oracle SGA constant. At 1,000 warehouses the working set is small enough to nearly fit into the combined Oracle SGA and FAS3140 RAM with PAM disabled. Because the random-read operations are served primarily from cache in this case, enabling PAM at this level has less effect. Nonetheless, the additional caching provided by the PAM

card yields a 6% performance boost for a peak performance of 87,731 OETs. As the number of warehouses increases, the size of the working set increases accordingly, making it less likely that a random-read request will be satisfied using the SGA or storage controller RAM. It is just this case where enabling the additional 32GB (16GB per storage controller) of PAM cache can significantly improve performance by improving the "cache hit" rate and lowering the likelihood that random-read operations require a disk access. For example, when using 6,000 warehouses, **we found that OETs increased by 27% compared to using 6,000 warehouses with PAM disabled**.

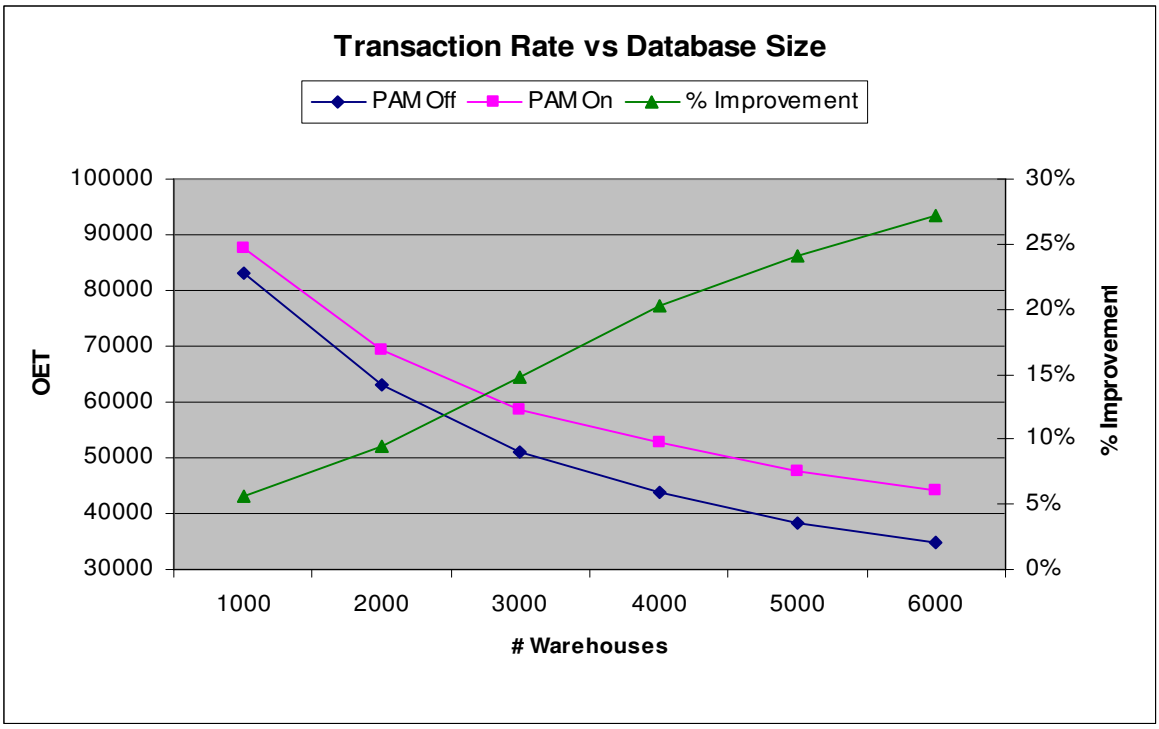


Figure 1) The PAM effect on transaction rate over a range of database sizes.

Reference Figure 2: The OET improvement associated with enabling the PAM module is a direct result of improved NFS average read latencies. In this OLTP workload, the I/O pattern for an OET is 75% random reads and 25% random writes using an 8KB request size. With this workload, the read requests to the FAS3140 are completing 30% to 40% faster with PAM enabled regardless of the number of warehouses in use.

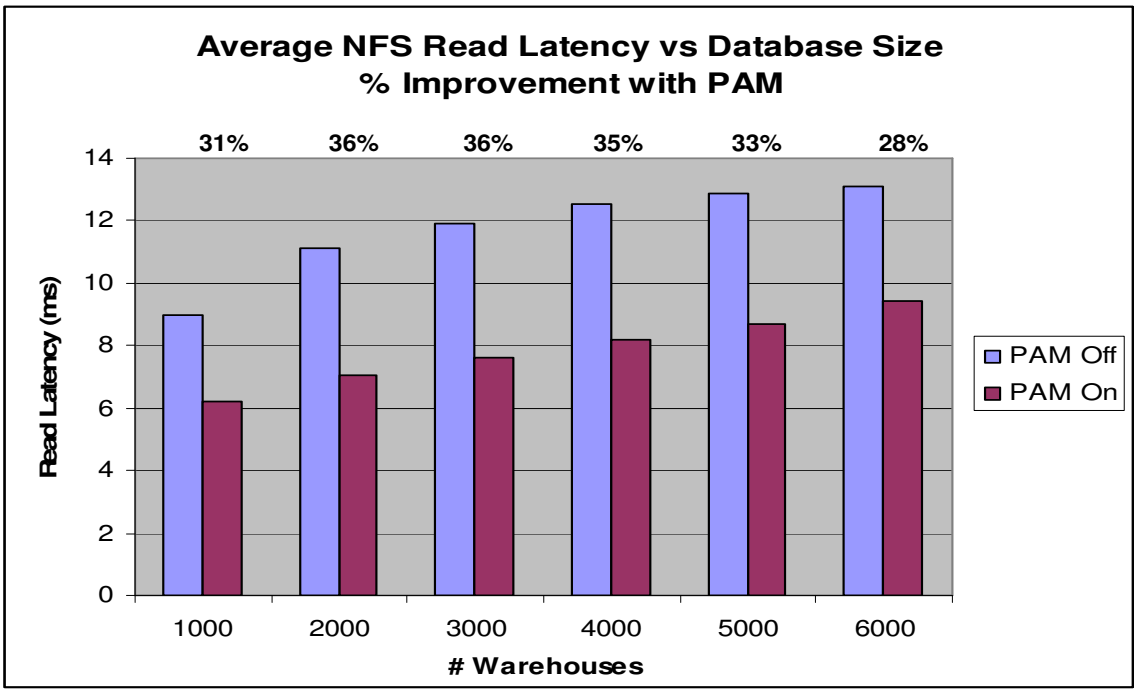


Figure 2) The PAM effect on average NFS read latency over a range of database sizes (lower latencies are better).

Reference Table 1: Other points of interest in evaluating the impact of PAM are database server CPU utilization as well as disk and CPU utilization on the FAS3140 storage controllers. On the database server, the CPU utilization increases as the amount of work (OET) completed increases. On the storage controllers, as the number of warehouses increases, the disk and CPU usage on the FAS3140 increases as there is an increased number of random-read requests that are satisfied from the disks. For a specific working-set size, enabling PAM decreases the disk utilization and improves random-read latency, which allows the system to do more work.

Table 1) The PAM effect on CPU and disk utilization over a range of database sizes.

	Number of Warehouses					
	1,000	**2,000**	**3,000**	**4,000**	**5,000**	**6,000**
Utilization %s for:						
PAM off						
Disks	63%	77%	79%	81%	84%	82%
FAS3140 CPU	40%	47%	48%	47%	48%	48%
Host CPU	89%	81%	74%	69%	67%	63%
PAM on						
Disks	50%	63%	69%	74%	75%	77%
FAS3140 CPU	42%	52%	55%	57%	58%	58%
Host CPU	91%	88%	85%	81%	76%	77%

The consequence of “doing more work” is higher CPU utilization on both the database server and NetApp storage system. That is, PAM improves the overall transaction capacity of the system. The end result is that, for a given load, using the PAM card allows you to make more effective use of your storage resources, resulting in overall lower average read latencies and higher numbers of OETs.

Note: Figure 2 and Table 1 statistics were collected and analyzed for the SGA size and user count performance tests presented in the remainder of this report. However, we found database server CPU utilization as well as FAS3140 disk and CPU utilization were very similar to those described above for all test cases. For this reason, we chose to omit those results.

3.2 TRANSACTION RATE VS. ORACLE SGA SIZE

For this series of tests, we used a fixed database size of 3,000 warehouses and held the number of workload generators (users) constant at 140. We then varied the size of the Oracle SGA database buffer cache from 4GB to 22GB in increments of 2GB.

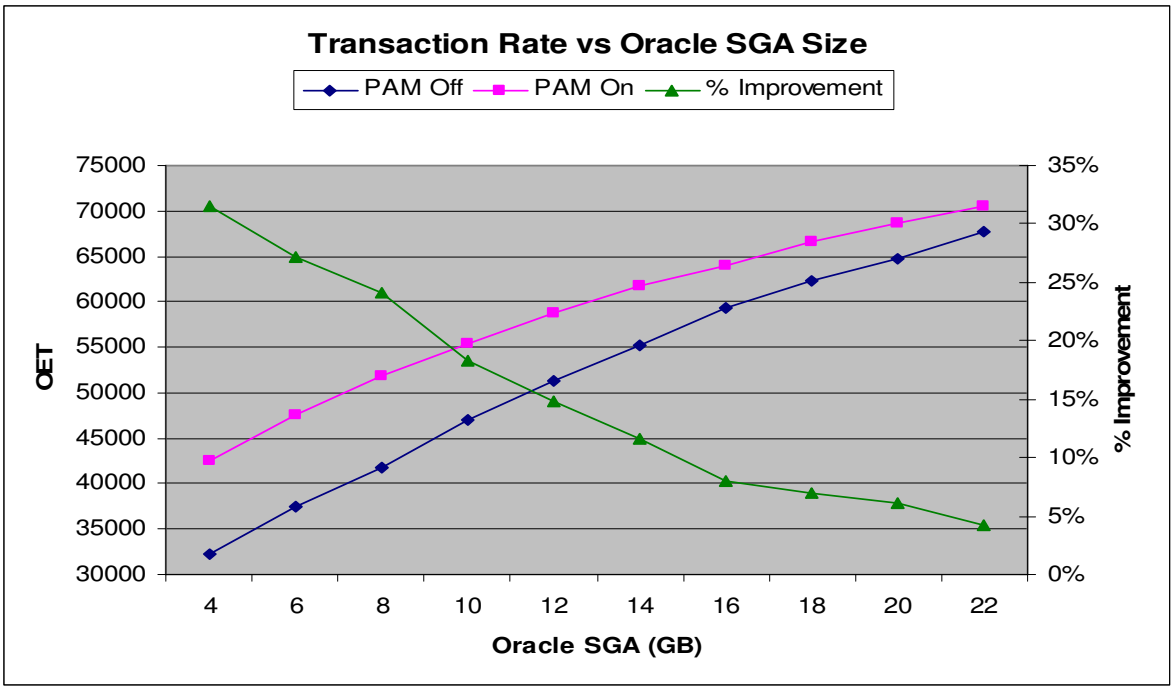


Figure 3) The PAM impact on transaction rate over a range of Oracle Database 11*g* SGA sizes.

With smaller SGA sizes, relatively few database blocks can be cached in the SGA. Consequently, the additional 32GB of read cache provided by the **PAM module has a significant impact, yielding a 32% increase in OET compared to using a 4GB SGA and no PAM cache**. With larger SGA sizes, much of the benchmark working set for this set of tests is generally cached in the host-side Oracle SGA and/or storage controller RAM. Enabling PAM at this level only improves transactional throughput by about 4%.

3.3 TRANSACTION RATE VS. USER COUNT

For this series of tests, we varied the number of workload generators (users) from 40 to 140 in increments of 20 while keeping the database size at 3,000 warehouses and the Oracle SGA set to 12GB.

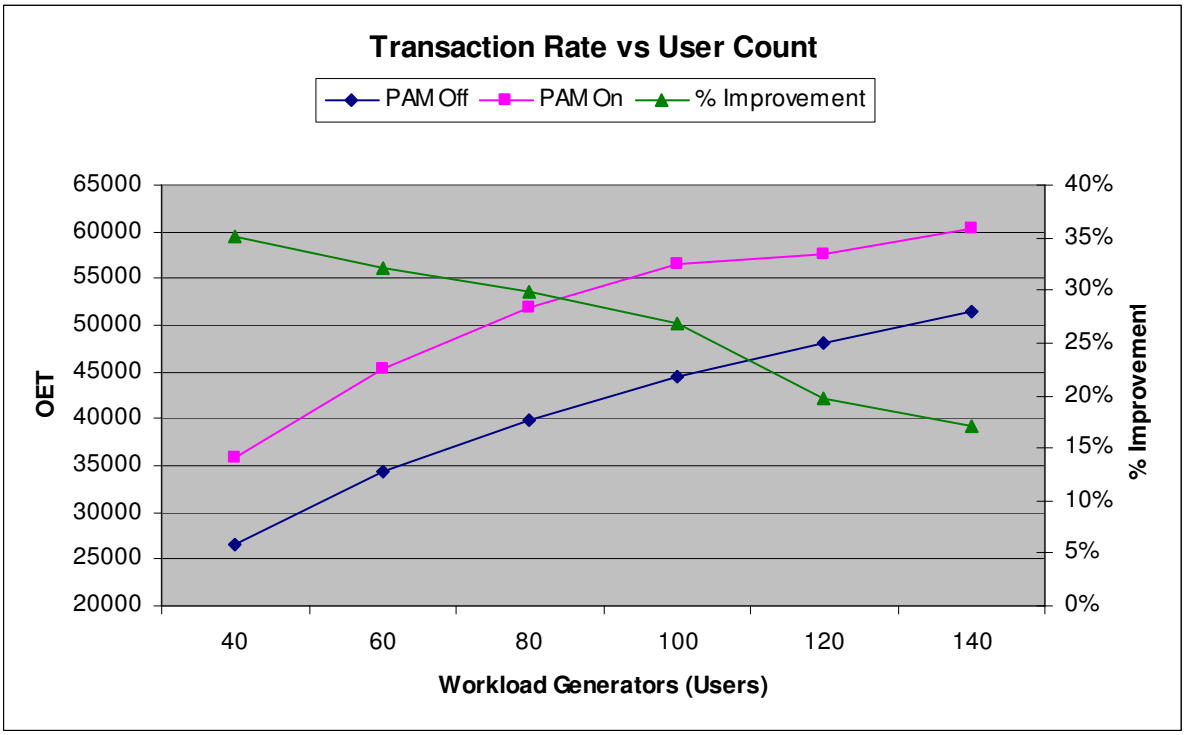
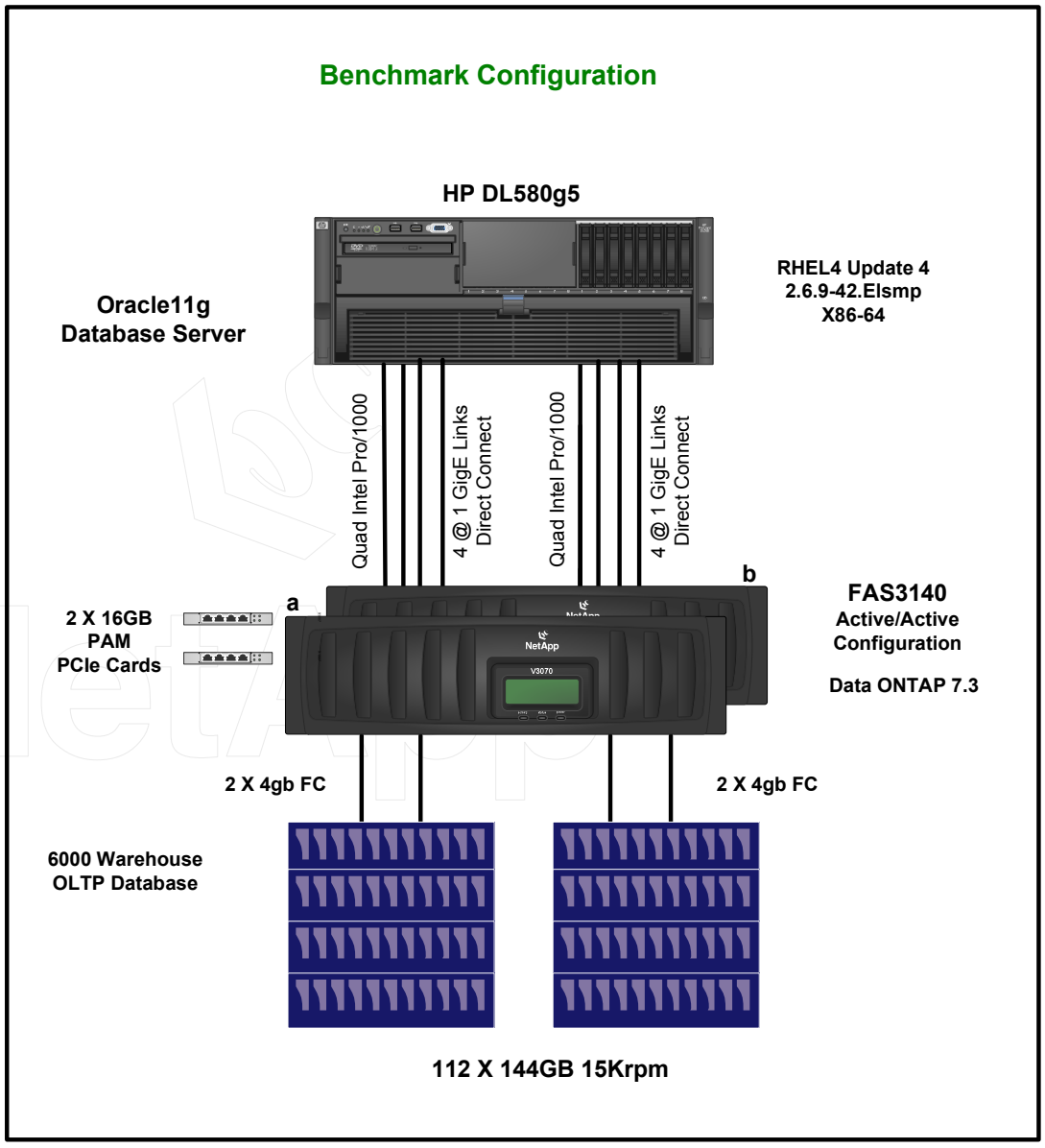


Figure 4) The PAM impact on transaction rate over a range of workload generators.

Recall that for this particular OLTP-type workload, the users execute transactions concurrently and have zero think time: that is, as soon as a transaction completes, the next transaction starts. Since no CPU or bandwidth bottlenecks were in the test configuration, more user processes generate more concurrent I/Os and randomly access a larger current working set. At lower numbers of users, PAM enables most of the working set to reside in cache, thus minimizing disk reads. **The result is a 35% improvement in OET for 40 workload generators when using PAM compared to no PAM.** As the number of users increases, the number of concurrent random-read I/Os increases, resulting in fewer data blocks found in cache. While this reduces the overall effectiveness of the PAM card, we found it still delivered a significant 17% improvement in the number of OETs generated with 140 workers compared to not using a PAM card.

4 TEST ENVIRONMENT DETAILS

Refer to Appendixes A through G for pertinent host, storage system, and database parameter details.



4.1 HOST AND NETWORK CONFIGURATION

The HP DL580g5 contained two quad-core 2.4GHz Xeon processors and 32GB memory. It was configured with two quad-port Intel Pro/1000 Gigabit Ethernet NICs. The link speed was set to 1Gb per second with standard frames enabled (MTU size of 1,500). Oracle Database 11*g*R1 DNFS was configured to enable automatic load balancing across the multiple GigE pathways. All I/O pathways between the host and storage system were direct connected.

4.2 STORAGE PROVISIONING

Each FAS3140 storage controller was configured with a 16GB PAM PCIe card and 56 X 15K RPM FC disks. Disks were allocated as follows on each controller:

3 disks	aggr0: root0
51 disks	aggr1: database storage

2 disks spares

The default aggr0 of three disks was left unchanged and used solely as the FAS3140 storage system's root volume on each controller. The total database storage, aggr1 on each controller, was spread (as evenly as possible in terms of I/O activity) across a total of 102 disks. There were two spare disks remaining after creation of these aggregates. RAID-DP® was used with the default RAID group size of 16 disks. Flexible volumes were created on each aggr1 for data/index files and redo logs. Both data/index and log volumes were exported and mounted on the host system using NFS mount options per the latest NetApp database best practices (see http://now.netapp.com/NOW/knowledge/docs/bpg/db/):

hard,rw,rsize=32768,wsize=32768,bg,vers=3,tcp,actimeo=0,nointr,suid,timeo=600

Except for the NFS options settings noted in Appendix B, all FAS3140 parameters, options, and volume settings were set to default values.

5 CONCLUSION

The goal of this project has been to provide relevant information for making informed, intelligent decisions regarding deployment of NetApp Performance Acceleration Module technology. The results obtained support this objective and clearly demonstrate significant performance improvements enabled by deployment of PAM for the specific workloads tested in this report. From an Oracle OLTP perspective, the Performance Acceleration Module reduces the latency of random-read operations, resulting in faster transaction completion times when I/Os are serviced from the low-latency PAM medium. The results are overall higher transactional throughput and lower disk utilization. However, the extent to which this occurs depends on a number of factors, including the total cache size versus the working-set size and randomness of the reads. PAM will likely enhance the performance of any application characterized by intensive random-read I/O with a working set that mostly fits into the extended cache provided by one or more PAM modules.

APPENDIXES

A PERTINENT RHEL4 PARAMETERS

/etc/sysctl.conf:

```
kernel.shmall=28000000000
kernel.shmmax=28000000000
kernel.shmmni=4096
kernel.msgmax=16384
kernel.msgmni=1024
kernel.sem=4096 512000 1600 2048

net.ipv4.ip_local_port_range=1024 65000
net.ipv4.tcp_sack=0
net.ipv4.tcp_timestamps=0
net.ipv4.tcp_rmem=4096 262144 16777216
net.ipv4.tcp_wmem=4096 262144 16777216

net.core.wmem_default=16777216
net.core.rmem_default=16777216
net.core.wmem_max=16777216
net.core.rmem_max=16777216

sunrpc.tcp_slot_table_entries=128
```

/etc/security/limits.conf:

```
oracle soft nproc 2047
oracle hard nproc 16384
oracle soft nofile 1024
oracle hard nofile 65536
```

B PERTINENT NETAPP STORAGE SYSTEM SETTINGS

```
nfs.v3.enable on
nfs.tcp.enable on
nfs.tcp.recvwindowsize 262144
nfs.tcp.xfersize 65536
```

C MOUNT OPTIONS

NFS Mount Options (/etc/fstab entries)

```
hard,rw,rsize=32768,wsize=32768,bg,vers=3,tcp,actimeo=0,nointr,suid,timeo=600
```

D PATCHES, DRIVERS, AND SOFTWARE

HP DL580g5 Server
RHEL4 Version 2.6.9-42.ELsmp

NetApp FAS3140 Active-Active Configuration Storage System OS
Data ONTAP 7.3.1

Intel PRO/1000 GbE Network Interface
Driver Version e1000 7.5.5-NAPI

E PERTINENT ORACLE PARAMETERS

To maximize the performance of Oracle Database 11*g*R1 on the Red Hat Linux operating system, database server configuration and tuning are important. Disclaimer: No universal Oracle Database

11*g* server tuning parameters apply to all workloads on all platforms; the appropriate parameters will depend on specific factors such as workloads, hardware, OS platform, and application usage.

```
compatible = 11.1.0.0.0
resource_manager_plan = internal_plan
parallel_max_servers = 100
recovery_parallelism = 40
db_writer_processes = 4
db_cache_size = 10G
db_recycle_cache_size = 0
db_16k_cache_size = 2G
db_2k_cache_size = 0
db_4k_cache_size = 0
shared_pool_size = 832M
pga_aggregate_target = 500M
db_block_size = 8192
dml_locks = 500
plsql_optimize_level = 2
log_buffer = 33554432
processes = 1200
sessions = 1325
transactions = 1200
open_cursors = 100
cursor_space_for_time = true
db_block_size = 8192
db_file_multiblock_read_count = 0
disk_asynch_io = true
filesystemio_options = setall
undo_management = auto
undo_retention = 180
_in_memory_undo=false
_undo_autotune=false
undo_tablespace = undo_1
workarea_size_policy = auto
parallel_execution_message_size = 16384
```

F ORACLE DATABASE 11*G*R1 DNFS CONFIGURATION

```
server: fas3140-1
path: xxx.xxx.x.xxx
path: xxx.xxx.x.xxx
path: xxx.xxx.x.xxx
path: xxx.xxx.x.xxx
export: /vol/tpccdat1   mount: /u02/tpccdat1
export: /vol/tpcclog1   mount: /u02/tpcclog1
server: fas3140-2
path: xxx.xxx.x.xxx
path: xxx.xxx.x.xxx
path: xxx.xxx.x.xxx
path: xxx.xxx.x.xxx
export: /vol/tpccdat2   mount: /u02/tpccdat2
export: /vol/tpcclog2   mount: /u02/tpcclog2
```

G PERTINENT HARDWARE DETAILS

Database server: HP 580G5 X86-64 dual Xeon quad-core 2.4 GHz CPUs, with 32GB RAM running Red Hat Enterprise Linux AS release 4 (Nahant Update 4) (Linux version 2.6.9-42.ELsmp) and Oracle Database 11*g*R1 Enterprise version.

Storage system: NetApp FAS3140 active-active configuration running Data ONTAP 7.3. Each storage controller contained 2 dual core AMD CPUs @ 1.8GHz, 8GB cache, 512MB NVRAM, and 56 X 15K RPM 144GB disks accessible using two 4Gb per second FC loops.

H NETAPP FLEXSHARE SOFTWARE

FlexShare is a quality-of-service tool that comes with Data ONTAP 7G storage systems such as the FAS3140. It lets you assign per-volume and per-cache policies and specify the relative priorities of application storage on a volume-by-volume basis. For example, using FlexShare options, Oracle Database "redo" logs could be prioritized differently than normal data or index files. **With FlexShare you can tune how the storage system should prioritize its resources.**

ACKNOWLEDGMENTS

Technical Direction and Advice

Dave Tanis

Lee Dorrier

Paul Updike

Stephen Daniel

References

NetApp Database Best Practices
http://now.netapp.com/NOW/knowledge/docs/bpg/db

NetApp Best Practice Guidelines for Oracle Database 11*g*
http://media.netapp.com/documents/tr-3633.pdf

Oracle Database 11*g* Release 1 Performance: Protocol Comparison on Red Hat Enterprise Linux 5 Update 1
http://media.netapp.com/documents/tr-3700.pdf

Performance Acceleration Module Design and Implemenation Guide (TR-3680)

FlexShare Design and Implementation Guide
http://media.netapp.com/documents/tr-3459.pdf

www.netapp.com