SAME and the HP XP512

An Oracle White Paper
November 2002

SAME and the HP XP512

EXECUTIVE SUMMARY

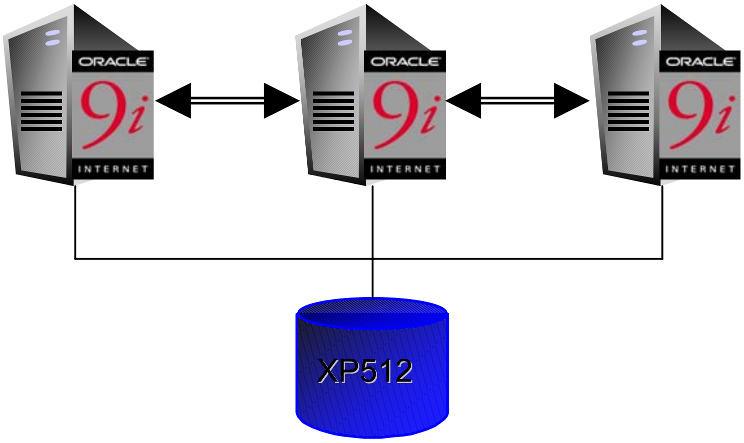
Oracle's Stripe and Mirror Everything (SAME) was first presented at Oracle OpenWorld in 1999 as a way to efficiently and simply allocate data to disks, achieving maximum performance with minimal upfront effort or application knowledge. SAME is in use in many high end, high availability applications in Oracle, increasing performance and reducing management time. However, the benefits of this discipline continue to spark controversy today, with questions about the effectiveness of striping together randomly and sequentially accessing data, and the impacts of the use of caching disk devices.

Our purpose was to evaluate SAME on the HP XP512 Storage Array. We wanted to test the notion that the separation of the log files and datafiles would provide significant performance improvement in the SAME environment. We also wanted to explore some permutations of the XP storage array to see if different underlying disk configurations or variations in LVM stripe sizes would have an impact on the test results. The experimental methodology was to use light and heavy workloads of an OLTP application type for each test case, and to vary the underlying disk configurations to observe the changes. This method was deemed the fairest way to create an environment where the changes in physical layout would be easily observed.

The discussions in this paper assume that the reader is familiar with the concepts described in the SAME paper. This paper, **Optimal Storage Configuration Made Easy,** is available for download from the Oracle Technology Network in the Internet DBA/Performance section at (http://otn.oracle.com/deploy/performance/pdf/opt_storage_conf.pdf).

TEST CONFIGURATION

The first series of tests evaluated a pure SAME environment where all database files and log files were striped across the same set of spindles compared with the modified SAME environment where the logfiles were separated from the rest of the files. These two results were matched against results from a 'traditionally' allocated database. We used a 3-node HP N-Class cluster (see Test Component Configuration) running Real Application Cluster (RAC) connected through a Brocade switch to the XP 512 disk array (see HP XP512 Storage Array). Each of the configurations is broken out in more detail below:



Pure SAME

In this configuration every datafile, log file and control file was striped over the 40-array groups in the XP frame, with a 256 KB LVM stripe size. With the exception of the stripe size, this is the original configuration recommended in the SAME Paper previously published. In a later section of this paper we show the results of testing using a 1MB stripe size.

The designation of "Pure SAME" is a designation we used to differentiate it from "Modified SAME" (described below)

Modified SAME

Following the initial testing, the "Modified SAME" configuration was configured with the data files and control files striped over 32 array groups with a stripe size of 256 KB and the log files were then placed on the remaining 8 array groups. Two of the instances were configured to have the log files spread over 4 array groups, and the third instance had another 4 array groups. For the log files we did not utilize any LVM striping.

Traditional

In this configuration the datafile access pattern (I/Os per second) was examined from the ”Pure SAME” runs using a combination of Oracle and HP utilities. Based upon the I/Os per second, different database objects were allocated sets of array groups in the XP512 storage array. For instance, the object that had the highest I/O’s per second was striped over 8 array groups using a 256 KB stripe size. The redo log files were placed similar to that mentioned for the ”Modified SAME” configuration above.

Due to the configuration of the HP XP512 Storage Array having 4 ACP pairs we decided to stripe the database objects over different ACP’s. At a minimum, the database objects were striped over 2 array groups on 2 separate ACP’s.

It should be noted that even though the workload is used regularly inside Oracle to provide consistent throughput, the initial thoughts of which objects were accessed the most was found to be incorrect after analyzing the data access pattern from the “Pure SAME” configuration.

Application Workload and Methodology

The intent of this testing was to keep the workload constant and measure the effect of the different disk configurations. The experimental methodology was to hold the workload consistent and to vary the underlying disk configurations and observe the changes. The workload was an OLTP type application that generated small, highly random operations against the database. In addition, Oracle processing of the redo logs accounted for a slight variation in the I/O mix. Our workload tests had the following characteristics

The light workload was designed to test the different configurations in a non-I/O bound environment. We determined this by moderating the number of clients to the point that any I/O related wait event was maintained near 10ms. We labeled this as the 10 client run.

The heavy workload stressed the database tier and the HP XP512 storage array to its limits. This workload resulted in an average CPU utilization in excess of 80%, and disk utilization in excess of 90% for each of the database tier machines as reported by the HP Glance utility and. . We labeled this as the 125 client run.

Each client launched was targeted to the same Oracle instance (for example clients launched on client driver machine #1, connected to the database instance on database machine #1), and each client driver was instructed to work against one-third of the available warehouses. This is application partitioning, and is intended to minimize the traffic on the cluster interconnect. The reason for selecting this type of configuration was to stress the XP512 Storage Array as opposed to stressing the cache fusion capabilities of Oracle9i Database Release 2’s Real Application Clusters. We also choose to implement Asynchronous I/O through Oracle as this is the most efficient writing mechanism from the database and kept with our goal of stressing the backend storage.

Workload Runs

Each of the workloads mentioned above was executed 3 times against each of the different storage configurations. Therefore, for each layout a total of 6 runs were performed, 3 times with the light workload and 3 times with the heavy workload. Before each run, the database was restored to ensure that the data was consistent when the workload runs were initiated.

Workload Duration

Each workload run was executed for a total duration of 90 minutes. Before each run, each of the Oracle instances was restarted to ensure the database buffer cache was empty, and that the database shared pool was initialized. Each client driver was given a 20-minute ramp-up time, which was used to start the individual clients on each node and to also warm both the database buffer cache and the database shared pool. The runs then ran for 60 minutes during which time statistics were being captured and then a 10-minute ramp-down time was added to the end of the run. The transactional throughput figures reported below, along with the other statistics reported were taken from the 60-minute run .

RESULTS

The results are analyzed using two key components, the transactional throughput of the different configurations, and two wait event statistics, log_file_sync and db_file_sequential_read,

The time taken for a *log_file_sync* event to occur was used to provide an indication of the total time taken for a log file IO operation to complete.

The *db_file_sequential_read* event that occurs as part of the reading of particular database block(s) from the disk was used as an indicator of a physical read from disk.

The Oracle statistics measured for the total time taken to write a block to disk was not considered since the XP Storage Array was equipped with a 16 GB non-volatile disk cache. That cache would allow blocks to be acknowledged as written without the need for the actual disk operation.

For each test, we analyzed the transactional throughput and wait time based upon the different workload intensities and the different disk configurations.

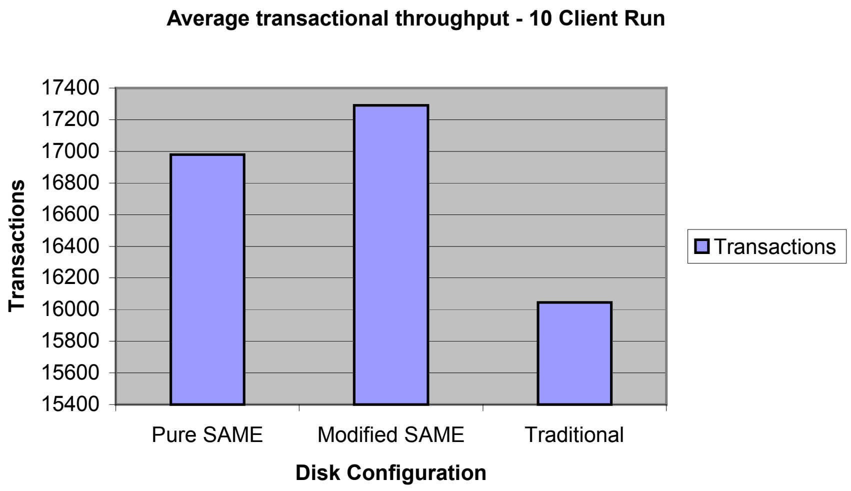


Figure 1: Average transactional throughput - 10 Client Run

The graph above shows the average transactional throughput when running under a load designed to provide a consistently fast commit time. Key here is the performance of "Modified SAME" in relation to the other two tests. Providing a dedicated set of disks for the online redo log files provided an increase in throughput when the XP Storage Array was not saturated.

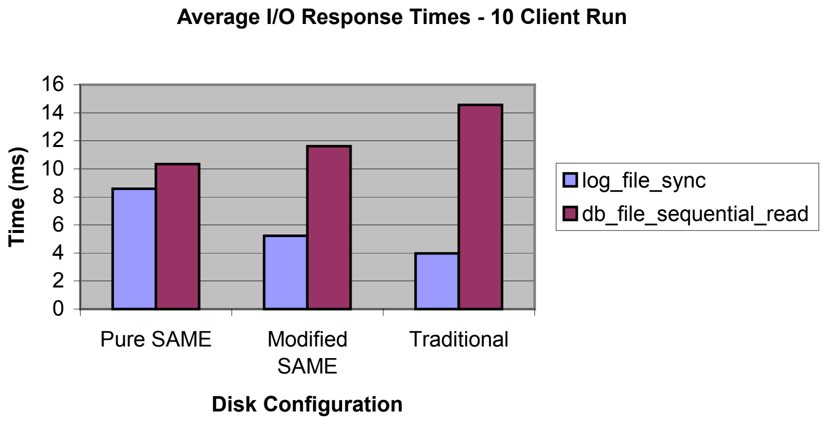


Figure 2: Average I/O Response Times - 10 Client Run

The graph above shows the average I/O response time as measured within the Oracle database. The configurations where a dedicated set of array groups were provided for the online redo log files shows a marked improvement in the response times for the *log_file_sync* event, but due to the reduction in the number of array groups available for the datafiles, the *db_file_sequential_read* operations were higher. The transactional throughput was relatively unchanged because there was no I/O bottleneck.

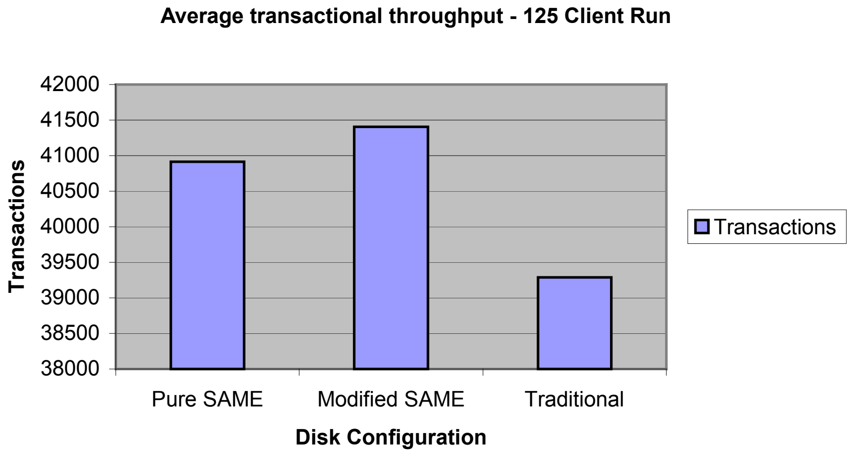


Figure 3: Average transactional throughput - 125 Client Run

The graph above shows the average transactional throughput when running under a load designed to stress both the HP XP512 Storage Array and the HP database hosts. The "Modified SAME" configuration does have a slightly higher throughput in these tests, but by only 1.2% in terms of real transactional throughput and the "Traditional" configuration does lag behind them both but by less than 4%

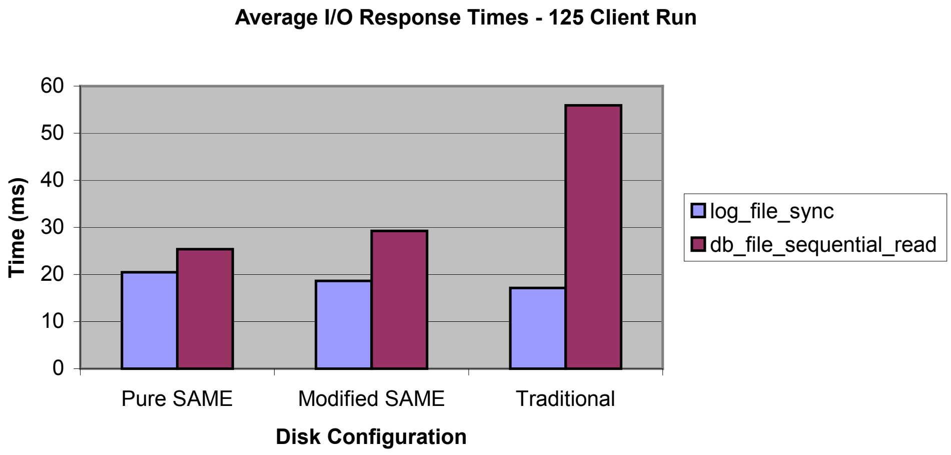


Figure 4: Average I/O Response Times - 125 Client Run

This final graph is also consistent with our pre-test expectations. Waits for I/O operations are fairly evenly balanced from the Pure SAME tests and more skewed for the other two tests due to the uneven distribution of file I/O. Log file sync benefit from the dedicated spindles by approximately 10% while the data read operations are affected by the reduction in spindles. In the case of "Modified SAME" the data read operations were approx 15% worse and the localized hotspots generated in the "Traditional" runs, as confirmed by the analysis of the HP XP512 Storage Array resulted in data read operations being in excess of 120% worse than the "Pure SAME" configuration.

ADDITIONAL CONSIDERATIONS

In further testing we attempted to measure the results of additional configurations against the "Pure SAME" results mentioned before.

There were two additional configurations that are documented below.

1MB Pure SAME

In this configuration every data file, log file and control file was striped over 40 array groups, as in the "Pure SAME" configuration, but this time, a stripe size of 1MB was used in stead of the 256KB stripe size used previously.

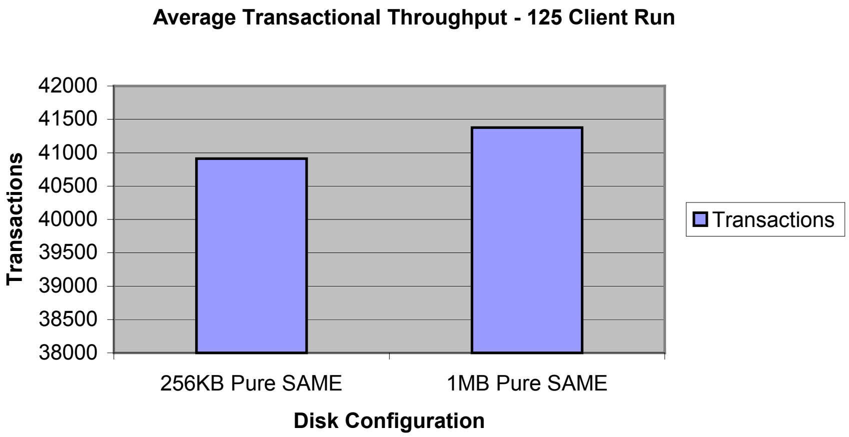


Figure 5: Average Transactional Throughput - 1MB Pure SAME

This graph shows that the variation in transactional throughput between the two stripe sizes when running under a heavy load is approximately 1%. This is mainly due to the fact that the average I/O size is small (8KB), the access is random and the HP XP512 is configured in RAID-01.

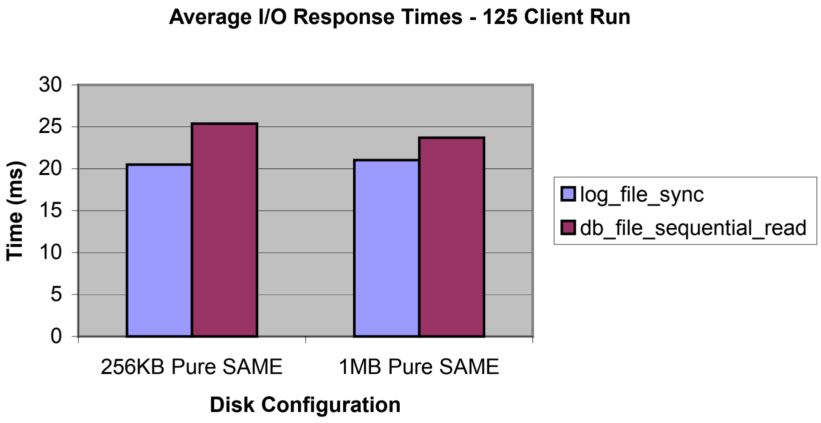


Figure 6: Average I/O Response Time - 1MB Pure SAME

This graph shows that the variation in response times for the two key events is negligible.

Therefore under the heavy OLTP workload, it can be seen that varying the stripe size had little effect on throughput or response time.

LUSE SAME

The final disk configuration used was termed “LUSE SAME”. LUSE is an HP XP512 storage disk configuration where multiple LDEV’s are merged together to present a single physical device to the HP-UX Operating System. In our case all the concatenated LDEVs belong to the same array group. Following the XP512 reconfiguration, the data files, log files and control files were striped over the 40 array groups, with a 256KB stripe size, in a manner similar to that of the “Pure SAME” configuration.

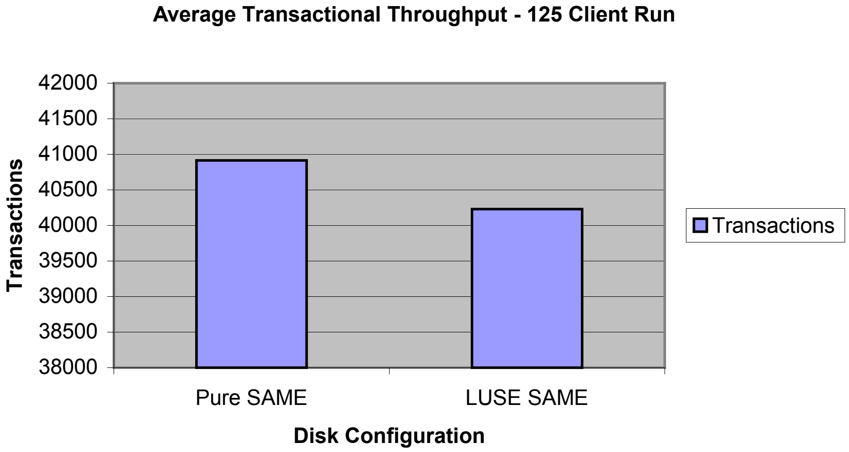


Figure 7: Average Transactional Throughput - LUSE SAME

This graph shows the variation in transactional throughput between the original “Pure SAME” configuration and the “LUSE SAME” configuration when running under a heavy load. The difference in transactional throughput is less than 2% Technically, in the SAME framework, these disk configurations are identical, but the 2% drop is to due to the overhead of managing a LUSE device.

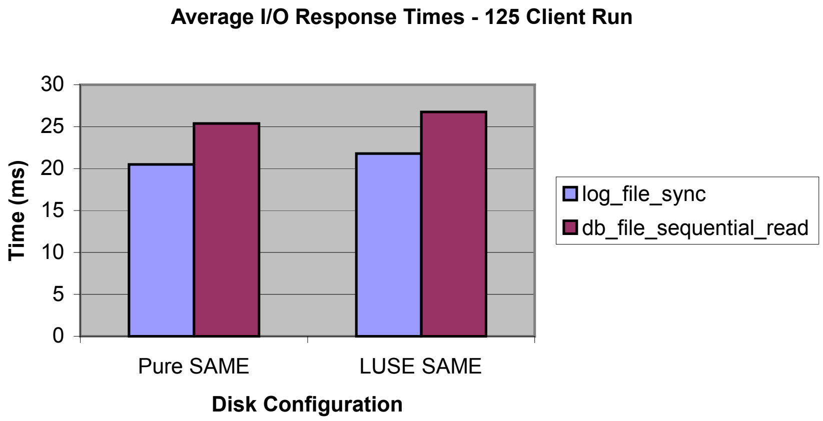


Figure 8: Average I/O Response Time - LUSE SAME

This graph shows that the variation in response time is again negligible.

Once again, under a heavy OLTP workload, it can be seen that the different disk configuration had little effect on throughput or response time

TEST COMPONENT CONFIGURATION

Hardware Configuration

The test environment consisted of a 3-node database tier running Oracle9i Database Release 9.2 and Oracle9i Real Application Clusters, with 3 client machines running the OLTP application to generate a steady workload. The client and database machines were connected via a private gigabit network to ensure the results were not affected by external network operations. The storage array for the test was an XP512 with 4 Channel Host Interface Processor (CHIP) pairs, 4 Array Control Processors (ACP) pairs and 16 GB of non-volatile cache. The XP512 was connected to the 3-node database tier through four HP Brocade Silkworm 2800 16 port fibre channel switches and 4 Host Bus Adaptors (HBAs) on each machine.

N.B. For the particular configuration, the use of the Brocade Silkworm switches was not necessary but was introduced to evaluate the complexity introduced by the SAN.

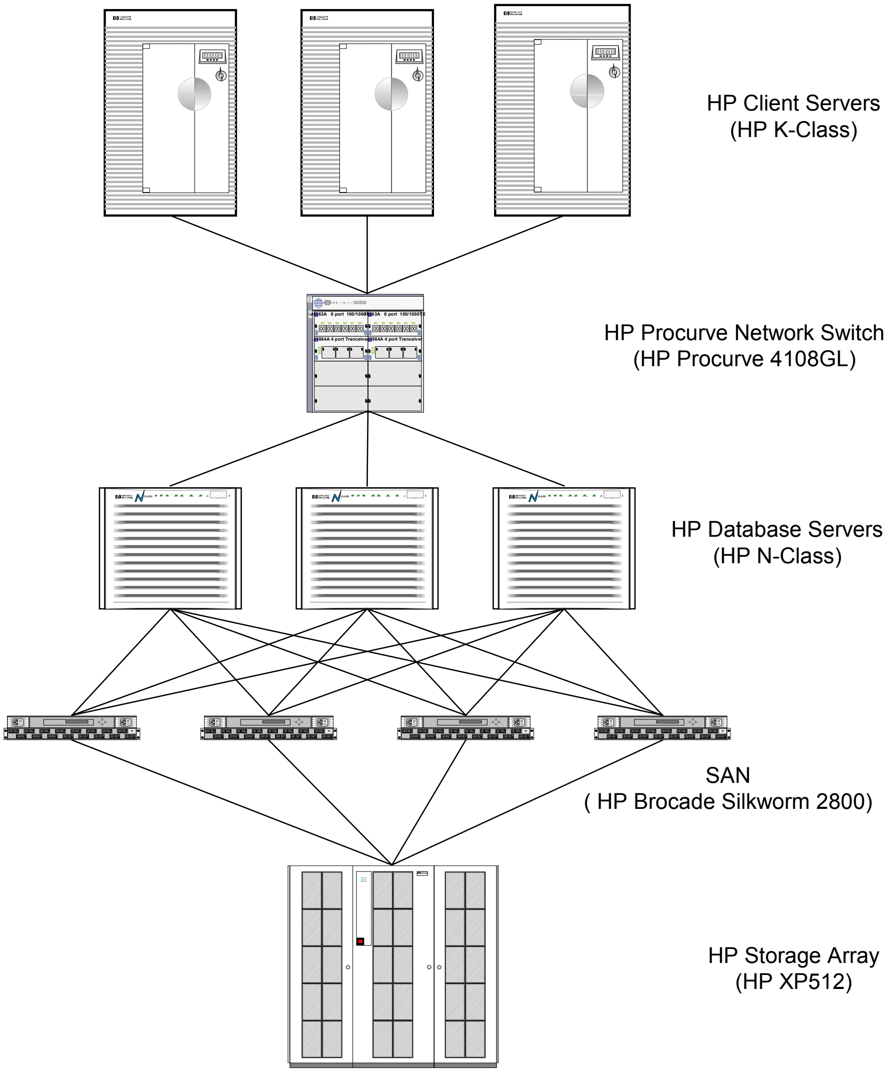


Figure 9: High Level Architecture

HP Client Servers

The client servers for the test were 3 HP K-Class systems with 6 CPUs and 8 GB of memory.

6 x 280 MHz PA-RISC processors

8 GB memory

1 x HP 1000BaseSX for private network to HP Procurve Network Switch

The software installed for the database servers was:

HP-UX version B.11.00

HP OnLineJFS (Advanced VxFS) version B.11.00

1000Base-SX HSC Gigabit Ethernet Driver version B.11.00.11

HP Procurve Network Switch

An HP Procurve 4108GL switch was used for the private network between the HP Client Servers and the HP Database Servers. The switch had 2 x HP J4863A – 6 port 100/1000BaseTX modules and 2 x HP J4864A – 4 port Transceiver with 4 x J4131A – SX modules. The HP J4863A module was used by the 10/100BaseTX heartbeat network and 1000BaseTX private network from the HP Database Server. The 1000BaseSX private network from the HP Client Tier used the J4131A modules.

HP Database Servers

The database server for the test was a 4-node HP N-Class cluster with 8 CPUs and 16 GB memory. For the purpose of the tests, we used only 3 of the nodes in the cluster.

8 x 440 MHz PA-RISC processors

16 GB memory

4 x HP Fibre Channel HBA's for connection to the XP512 storage array

1 x HP Fibre Channel HBA for connection to the VA7100 storage array

1 x HP HyperFabric 4x for cluster interconnect

1 x HP 10/100BaseTX for cluster heartbeat traffic

1 x HP 1000BaseTX for private network to Client Servers

The software installed for the database servers was:

HP-UX version B.11.00

ServiceGuard OPS Edition Bundle version A.11.09

HP OnLineJFS (Advanced VxFS) version B.11.00

HyperFabric 9000/[78]00 version B.11.00.12

PCI Tachyon TL/TS Fibre Channel version B.11.00.10

1000Base-T PCI Gigabit Ethernet Driver version B.11.00.11

Storage Area Network

Four HP Brocade Silkworm 2800 16 port fibre channel switches, with 12 x 1 Gb GigaBit Interface Converter (GBIC) adaptors per switch were used for the test.

Each HP Database Server in the cluster was connected to each Brocade Silkworm switch, and the HP XP512 Storage Array was connected to each switch via 4 CHIP ports.

HP XP512 Storage Array

The HP XP512 Storage Array contains the following components.

4 x CHIP Pairs

4 x ACP Pairs

16 GB Nonvolatile Cache

256 MB Shared Memory

20 x 72 GB (10K rpm) Array Groups – 5 per ACP Pair

24 x 18 GB (10K and 15K rpm) Array Groups – 6 per ACP Pair

The 72 GB array groups were formatted in the Open-L emulation mode, providing 4 Logical Devices (LDEVs) per array group.

The 18 GB array groups were formatted in the Open-E emulation mode, providing 2 LDEVs per array group.

The database used a total of 40 array groups (16 x 72 GB and 24 x 18 GB). One LDEV from each array group was used for the Oracle database. The remaining LDEVs on each array group were s used to hold backups of the database that was restored between each run.

The remaining 4 x 72 GB array groups were used by the individual hosts for Unix File Systems (JFS) supporting the Oracle9i Database Release 9.2 software, database configuration files and archived redo log files.

The 4 CHIP pairs were set to high-speed mode also referred to as high performance mode. Only the first two ports from each CHIP board was used.

All the HP Database Servers had the same view of the XP storage array

Database Environment

HP Database Server

The database version used for the testing was Oracle9i Database Release 2 (9.2.0.1) running Real Application Clusters. For the cluster interconnect we used HP HyperFabric Messaging Protocol (HP/HMP).

Although the cluster was configured as a 4-node cluster, we used 3 nodes of the cluster for the purpose of our testing. The 4th node while still active and part of the HP/MC ServiceGuard cluster did not run an instance of the database.

Oracle9i Database Release 2 Size

The approximate database size is as follow:

DATA components consisting of application data was 245 GB

INDEX components consisting of application index was 39 GB

SYSTEM components consisting of SYSTEM, Rollback, Temporary and Statistics was 82 GB

Redo Log files consisting of online redo log members was 64 GB

The database was based upon the Oracle OLTP workload.

Disk Configuration

The purpose of the test was to verify the effects on the workload throughput based upon various database disk configurations. Before describing the configurations in detailed, a brief description of the HP XP512 storage array is required.

HP's XP platform characteristics overview

The XP Storage Array is built in multiples of 4 disks. These 4 disks are configured into an array group, which simply put is a unit of raw disk capacity.

The 4 disks on an array group are then assigned a RAID level, either RAID-01 (Striped mirroring) or RAID-5 (3 data stripes and 1 distributed parity).

The array group is then assigned an emulation type, which means the raw disk capacity is broken into multiple Logical Devices or LDEVs. For the purpose of the testing, the 73 GB array groups were configured with the OPEN-L emulation modes, and the 18 GB array groups were configured with the Open-E emulation mode.

OPEN-L emulation mode on a 73 GB array group (total raw capacity of 292 GB) running RAID-01, provides 4 LDEVs, each of approx. 36 GB in size.

OPEN-E emulation mode on an 18 GB array group (total raw capacity of 72 GB) running RAID-01, provides 2 LDEVs, each of approx. 14 GB in size.

Once the LDEVs have been created, they are then mapped to the XP CHIP ports. These LDEVs are then mapped to physical devices by the Unix operating system that the Logical Volume Manager could use.

The Logical Volumes were striped across the ACP pairs and array groups in a round-robin fashion. The XP CHIP processors are specialized, they process either an even or odd LUN. A mis-configuration, can impact the I/O performance. We had an even number of odd and even LUNs for each CHIP pair to process. In addition, we applied this approach to striping across the Host Bus Adaptors (HBAs). This was done to minimize any contention in feeding data to the array by manually load balancing the input.

You can download the following paper for more information on the XP512, http://www.hp.com/products1/storage/products/disk_arrays/xpstoragesw/performance/paper.pdf

SUMMARY AND CONCLUSIONS

After testing we found our assertions from the SAME paper held true. Using an internal Oracle application to generate a consistent reproducible high throughput OLTP workload, we were able to draw the conclusions listed below. It should be noted that these conclusions were drawn from measurements based on a specific high through put workload. However, these concepts should be generally applicable to other workloads.

SAME configuration dramatically reduces the complexity of initially configuring your Oracle database to optimize IO utilization.

SAME disk configurations consistently performed better than a more traditional, application based disk layout even with foreknowledge of the application's data access patterns. However, note that there is a risk of creating hot spots with the traditional method.

Isolating Oracle redo log files with their more sequential workload from the remainder of the Oracle datafiles did not produce any significant benefits. Our testing showed an improvement of just over 1% compared to the SAME configuration.

Varying the disk stripe size or exercising the different disk configuration options available within the XP did not produce any significant changes in the test results with this OLTP workload.

APPENDIX A

Statistics Recorded

The following statistics were recorded throughout the duration of each run

The OLTP workload recorded the transactional throughput for all transactions as well as the number of individual transaction executed during the 60 minute run duration.

The Oracle database monitoring tool, statspack, was configured to run every 5 minutes for the full 90-minute duration that the OLTP workload was active for.

Five HP Glance advisor scripts were running, capturing information every minute. The statistics captured included Global CPU Utilization, Memory Utilization, Queue Size (Run Queue, Disk Queue, Memory Queue, etc…), and CPU Utilization by CPU and Network Utilization by Network Interface.

HP provided a special version of the HyperFabric software that allowed the monitoring of the HyperFabric network used by Oracle for Cache Fusion. By default, the HyperFabric software monitors the number of packets being transmitted, but not the size of the packets being transmitted. This information was extracted every minute.

APPENDIX B

XP IO REPORTS[1]

Disk configuration	IO size	IOsec	RT<10	10<RT<50	RT>50	XP Cache Hit	XP Back-end Utilization	
							DB files	Logs
PURE SAME	8KB	6716	31.50%	24.80%	43.70%	15.22%	56.00%	56.00%
MODIFIED SAME	8KB	6873	30.94%	25.33%	43.73%	16.83%	57.40%	23.00%
TRADITIONAL	8KB	6560	31.56%	27.56%	40.88%	18.88%	48.00%	48.00%

Table 1: Light Load (10 Client Run)

The 10-client run is purely a response time exercise because the array was not saturated. At moment this workload generated bursty writes. The Logical Volume Manager (LVM) stripe size was set to 256KB for all the light load runs.

Disk configuration	IO size	IOsec	RT<10	10<RT<50	RT>50	XP Cache Hit	XP Back-end Utilization	
							DB files	Logs
PURE SAME	10KB	11636	19.32%	58.79%	21.89%	4.40%	94%	94%
MODIFIED SAME	8KB	11848	15.49%	45.76%	38.75%	4.00%	92%	40%
TRADITIONAL	8KB	11357	8.73%	29.03%	62.26%	3.38%	11% - 100%	
1MB PURE SAME	10KB	11552	20.20%	59.59%	20.21%	4.64%	94%	94%
LUSE SAME	10KB	11237	18.00%	53.97%	28.03%	4.22%	94%	94%

Table 2: Heavy Load (125 Client Run)

The heavy load put a lot of stress to the XP back-end. The IO throughput is in alignment with the transaction throughput. The “PURE SAME” disk configuration offers the best response time due to the fact that the data was striped across all the available spindles. The Traditional disk configuration showed a wide variation in the XP back end utilization based on data file usage as shown in the figures in the table above. This is consistent with known patterns of disk access for this workload and the known downfall of the traditional disk modeling approach. The LVM stripe size for all the disk configurations but the “1MB PURE SAME” was 256KB.

[1] RT: Response Time in millisecond.

XP Back-end Utilization: % XP total bandwidth consumed for database files and logs.

ACKNOWLEDGEMENTS

Thanks to HP/Baila Ndiaye

SAME and the HP XP512
November 2002

Author: James Viscusi, Andrew Babb
Contributing Authors: Ashish Prabhu, Doug Utzig, Lawrence To, Pradeep Bhat, Ray Dutcher, Ron Weiss, Shari Yamaguchi, Susan Kornberg, Baila Ndiaye

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
www.oracle.com