



Configuring the Oracle Database with VERITAS Software and EMC Storage for Optimal Scalability, Manageability, and Performance

Table of Contents

- I. Introduction
- II. Storage Topics
- III. Configuring Oracle Database with Storage Subsystem
 - u Traditional Approach
 - u New Approach
- IV. Test Configurations and Analysis
- V. Conclusion
- VI. Appendix
 - u Test Configuration Details
 - u EMC Disk Configurations
 - u Step-by-Step Setup
 - u EMC Symmetrix
 - u VERITAS Database Edition for Oracle

I. Introduction

In today's increasingly data intensive, web and e-commerce enabled world, data storage requirements and management of that storage are becoming increasingly important. Databases continue to grow at an incredible rate as businesses store every detail of their customers as they navigate their web sites. Downtime for any reason is unacceptable and must be avoided at all cost as users are only a couple of mouse clicks away from the competition. Management of such a dynamic environment is difficult for all database and system administrators. The ever important priorities of mission critical systems, such as availability, reliability, scalability, and performance, are magnified in today's e-business world.

Today, customers are looking for optimal scalability, manageability, and performance. They are looking for the flexibility to grow their business and adapt to changes from the outside world, and from within. They want to maintain a competitive advantage by deploying resources that will give them an edge over the competition, and thus seeking ways to simplify management of their current infrastructure. And, they are trying to attain the highest level of performance available from the investments they have made in software, hardware, and personnel.

As anyone experienced in this area will attest, performance tuning is a trade-off, often pitting one goal against another. Common questions are:

- u How do I maximize performance while at the same time minimize management overhead?
- u How do I maximize flexibility and minimize cost?
- u How do I maximize throughput and minimize Online Transaction Processing (OLTP) response time?

The goal of this white paper is to provide a list of recommendations that will allow customers to optimize overall scalability, manageability, and performance. Oracle, EMC, and VERITAS understand what customers are asking for as well as the challenges faced in trying to meet their objectives. Moreover, the recommendations provided must be able to handle today's ever-changing computing environment. Availability is also a customer priority. This paper includes basic techniques, but does not go into detail about high availability solutions, as this is a topic of other joint technical white papers.

There are many options for configuring a storage subsystem for a database workload. However, it is not obvious when to use each feature and option. In this paper, we first describe the traditional approach of configuring a storage subsystem for a database workload based on the application type, data file I/O characteristics, and availability requirements.

We then describe a new approach to configuring a storage subsystem for a database workload that is simple, easy to manage, and achieves excellent performance. The idea is to Stripe And Mirror Everything (SAME), or to place Oracle files on striped and mirrored drives.

In our testing, we used EMC hardware mirroring (RAID 1) for data protection and VERITAS software striping (RAID 0) for increased performance. Our test results show that the SAME method is successful and should be considered in most situations.

We recommend the following:

- u Implement hardware mirroring and software striping for availability and performance reasons.
- u Use the SAME method with Oracle.
- u Minimize disk contention by evenly distributing IO across the Symmetrix.
- u Set I/O block size to 1MB. In our test, this demonstrated optimal performance.
- u Set VERITAS stripe depth to 1MB . In our test, this demonstrated optimal performance.
- u Use VERITAS Quick I/O.



There are situations where manageability considerations may make the implementation of SAME difficult; however, this approach can be used in most cases and works well.

Our tests were conducted on a single Oracle Database approximately 50GB in size, striped across 16 to 32 devices.



II. Storage Topics

Mirroring (RAID 1) vs. RAID 5

We recommend RAID 1, or mirroring, over RAID 5 for performance reasons.

Mirroring is the simplest way to achieve data redundancy. For each data drive, there is a second drive that contains exactly the same data. Data is written to both drives. Thus, if one drive fails, the other drive can provide an exact copy of the lost data immediately. RAID 5 protects data through use of parity information. With RAID 5, both data and parity information are striped across all the drives in the RAID group. If a drive fails, original data can be reconstructed from surviving drives using parity information.

EMC requires that all data on a Symmetrix storage subsystem be protected either through hardware mirroring or RAID S (EMC RAID 5). Mirroring requires twice the usable storage capacity for protection but offers better performance than a RAID 5 solution, which typically requires only 10-25% additional storage capacity for protection but may cause some performance degradation. This is especially true in a write intensive environment. RAID 5 can be a viable solution where storage cost may be more important than performance.

Another advantage of mirroring over RAID 5 is that mirror splits can be used to make very fast copies or backups of files.

Striping (RAID 0)

We recommend RAID 0+1, or mirroring with striping, for availability and performance reasons.

Striping (RAID 0) balances the I/O workload across all the disk drives in the stripe set, thus reducing potential hot spots and enhancing performance by spreading data across multiple drives so I/Os can be done in parallel.

The amount of information written to each disk before moving to the next is known as the *stripe depth*. The group of physical drives being striped across is known as the *stripe set* or *striped volume set*. The number of logical devices included in the stripe set is known as the *stripe width*.

In the following figure, we have a stripe set of four physical disks and stripe width of four logical devices. The number of the logical devices does not have to be the same as the number of physical disks.

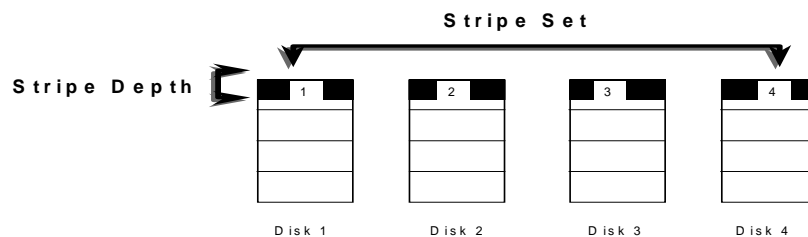


Figure 1: Striping

Storage Subsystem Cache

Use of non-volatile cache in a storage subsystem can also improve performance. When a write request is put into a non-volatile cache during a write operation, the write is instantly confirmed, thereby enabling faster write performance. In a read operation, a storage system

may read ahead or pre-fetch additional data into the non-volatile cache. If the read can be satisfied in the cache (cache hit), or the pre-fetch logic has anticipated the subsequent reads, the operation will complete in a fraction of the elapsed time of an actual disk read.

EMC Symmetrix supports up to 32GB of non-volatile cache. Both random and sequential writes can benefit from cache; however, sequential reads benefit more from cache than random reads. For example, EMC Symmetrix can use cache for sequential read ahead on a per device (hyper-volume) basis. In workloads that are truly random read, the Symmetrix cache hit ratio will be less efficient.

Possible Bottlenecks

In configuring a high performance Oracle database storage solution, there are additional areas where bottlenecks may occur:

- u System - Does the system have enough CPU or memory resources?
- u Operating System - What are the OS limitations (for example, Windows NT has only 256 I/O queue depth)?
- u File System - How does it compare to raw partition performance?
- u I/O Channel - Are there enough I/O controllers?
- u Disk - Are the disks and their interfaces fast enough?
- u Other Applications - What other applications are running on the system?

Any of these areas, in addition to how Oracle files are laid out or which striping method is used, may impact performance significantly. Therefore, it is important to locate the bottlenecks.

In our testing, we tried to minimize these bottlenecks so we could determine the best storage configuration without regard to these complicating factors.



III. Configuring Oracle Database with Storage Subsystem

In this paper, we examine two methods of configuring an Oracle database with the storage subsystem. We then compare the test results and discuss their implications. The first configuration method is the traditional approach of laying out Oracle files based on the application type, data file I/O characteristics, and availability requirements.

The second configuration method is the new Stripe And Mirror Everything (SAME) approach, or striping and mirroring all Oracle files on all available disk drives.

Traditional Approach

The traditional method of configuring the Oracle database and storage offers maximum configuration flexibility and performance tuning, but requires in-depth understanding of the I/O characteristics of Oracle files and applications as well as future growth requirements. We briefly discuss this method using simple categorization of application by either (Online Transaction Processing (OLTP) or Decision Support Systems (DSS).

I/O Profile of Oracle Files

Various I/O operations that Oracle performs and their characteristics can be easily described by considering the logical grouping of data files called tablespace.

System tablespace - System tablespace contains all of the system-related tables. Most of these tables are created during database creation. This tablespace encounters mostly random block reads with minimal writes.

Redo Logs - Oracle recommends 4, 8, or 16 multiplexed online redo log files. Online redo logs are written sequentially for every change made to an Oracle block. Online logs are written by the LGWR process. The size of the write operation varies with the transaction throughput. Online redo logs are read sequentially during recovery. The online redo log file is also read sequentially by the archiver process after it is completely filled.

Archive Logs - The archiver copies the online redo log files into archive files after the online redo log file is filled. Archive files are created dynamically, so they cannot be placed on raw devices. The archiver reads the online redo log file synchronously and writes the archive redo log file asynchronously. These archive log files are read sequentially during backups and media recovery.

Rollback segments - Rollback (undo) segments are used for read consistency, database recovery, and to rollback uncommitted transactions. Normally, 10 to 30 extents (user-specified) are allocated for each rollback segment when they are created. Every transaction sequentially writes to only one extent at any given time. Extents are allocated dynamically and are reused in a circular fashion. If a rollback segment fits in the cache and is sufficiently hot, then it will only be written to disk during a checkpoint. If an incremental checkpoint is used, undo blocks are continuously written to disk.

Temporary tablespace - Temporary tablespace is used mostly for on-disk sort operations. Using direct I/O, reads and writes on this tablespace are mostly sequential.

Control files - Control files are relatively small files that are mostly read during instance startup. Writes to control files happen during log switch and checkpoints, and when any structural change is made to the database.

User tables and indexes - It is difficult to characterize the I/O pattern of the user tables and indexes. In general, OLTP queries perform one or more reads on the index block followed by a read on a table block. However, things get complex when the application nature changes. What is good for normal operation may not be optimal for maintenance operations. For example, during data load, index create, on-disk sort, create table as select, index merge, read/write of LOBS, index range scan, index fast full scan, parallel operations, and backup and recovery operations.

I/O Profile of Applications

Applications have their own I/O characteristics. Depending on the type of application, there are a number of storage and Oracle parameters that can be tuned to optimize overall performance.

For example, in a purely OLTP system where large numbers of users make small (single block) I/O requests, database administrators (DBAs) tend to configure stripe depth of a volume to be a multiple of the Oracle database block size (`db_block_size`). This configuration increases the probability that an I/O request for a data block will be serviced by only one drive, and can probably handle a large number of concurrent users without developing potential disk contention.

Whereas in a purely DSS system, few processes make large I/O requests. It is often efficient to make the stripe depth small, so that many disks in the array will work to return the data faster. Note that when using parallel query, the results are less predictable due to the large degree of parallelism.

It is important to remember that even OLTP systems, which typically perform small random reads and writes, often perform a significant amount of sequential I/O during backup and restore operations, index builds, data loads, or batch jobs. Customizing the stripe depth for purely small random reads can result in significant delays when performing maintenance operations that perform sequential I/O.

Even though the traditional method is complex, it is possible for an expert to configure a storage subsystem, after continuous monitoring and going through many iterations, to get the best possible performance. The traditional method may also work well in certain specialized situations such as a database environment with highly parallel, large sequential I/O operations.

This approach produces a very customized Oracle and storage subsystem setup since each configuration is a function of a specific application and its environment. If the application or environment change, major reconfiguration and re-tuning may be necessary to maintain overall performance.

In our traditional approach testing, we separated online redo logs from the rest of the data files (tablespace). We also did some testing as part of our traditional configuration by separating redo logs, tables, and indexes from system, temporary, and rollback segment tablespaces.

We limited our test result analysis only to the redo log separation from the rest of the data files.

New Approach

The SAME method offers simplicity, manageability, performance, and availability.

This approach is possible because of a combination of technical advances that have occurred over the last few years. These advancements include:

Parallel database execution. This allows scans of large amounts of data to be parallelized across many disks and many I/O requests so that a single disk or process does not become a bottleneck.

Efficient and scalable striping of files using the VERITAS Volume Manager (VxVM). This allows files to be striped over many disks without hurting I/O performance and with minimal overhead on the host.

Efficient file system I/Os using VERITAS Quick I/O. This eliminates the need for using raw devices for performance purposes in a single instance database.

Dynamic re-striping of data. This allows disks to be added and subtracted from a striped volume without having to rebuild the volumes from scratch and while keeping the system available. (We tested the functionality of this feature; however, we did not test the performance.)



Large cache on EMC Symmetrix. A large non-volatile cache reduces the overhead of doing large writes and allows write I/Os to be scheduled in an efficient manner. They also can entirely eliminate disk writes for frequently written database blocks that might otherwise become a hot spot (for example, if actual disk writes are deferred). The large caches also increase the availability of databases by speeding up recovery operations.

Highly reliable mirrored disks with online disk replacement. This allows striping to be used across many disks without the fear that a disk failure will cause data loss, significant downtime, or a complete database restore.

Dynamic load balancing and path failover. Host-based products like EMC PowerPath or VERITAS DMP (Dynamic Multipathing) provide added flexibility, remove potential I/O channel bottlenecks, and enhance database availability.

Advanced features like Symmetrix Dynamic Mirror Service Policy. This allows sequential read operations to be accessed from the M2 or mirror as well as the M1 or standard volume.

The advent of these technical advances has eliminated many of the trade-offs that previously had to be made when configuring a storage subsystem. We can now implement a simple configuration that is both efficient and easy to manage: Stripe And Mirror Everything (SAME).

By mirroring all the disks, the failure of any single disk can be tolerated. By striping all the data, hot disks are avoided and disk throughput is maximized to all tables.

The greatest benefit of SAME is that it is application independent. It will work for any application and will even work in mixed environments. To configure the storage subsystem, one only needs to know the amount of disk storage required and the maximum aggregate I/O bandwidth required by the application. Detailed knowledge of the I/O characteristics of the various transaction types and tables is not required.

SAME does not require constant adjustment as the application and workload evolve. It works well for both random and sequential access patterns. Thus OLTP, batch, data loads, import/export, backup, recovery, and reporting will all perform well.

SAME is designed to take maximum advantage of the aggregate I/O bandwidth of the storage devices. Maximizing I/O bandwidth is becoming increasingly important since disk capacity is increasing at a much higher rate than disk throughput. Also, a parallel operation executed against any table can take advantage of all the I/O bandwidth capacity of the storage subsystem. This allows for fast reporting, loading, reorganization, and indexing operations on any table.

IV. Test Configurations and Analysis

The goal of this paper is to provide general guidelines on how to easily configure storage with an Oracle database without sacrificing availability and performance, using VERITAS software and EMC Symmetrix. We also want to validate our new Stripe and Mirror Everything (SAME) method with test results and compare the results with the more traditional configuration method.

We performed a number of database tests on numerous Oracle/VERITAS/EMC configurations. All the configurations used the same number of physical disks and the same size database. The only difference was how the database files were laid out on the disks. In the “VI. Appendix” section, detailed test configurations and the three main disk configurations are described.

We selected the following three disk configurations because they have the best performance numbers in their category:

Disk configuration 1: Oracle files striped across all 32 drives and their mirror copy on the same 32 drives (SAME with possible disk contention)

Disk configuration 2: Oracle files striped across 16 drives and their mirror copy on the other 16 drives (SAME with no disk contention)

Disk configuration 3: Oracle log and data files in separate volumes on 16 drives and their mirror copy on the other 16 drives (Traditional with no disk contention)

First we compared disk configurations 1 and 2 to see how potential disk contention can affect the database test results. Then, we compared disk configurations 2 and 3 to see the performance difference between the SAME method and the traditional method of data layout.

The database tests included database creation, large sequential read/write, random read/write, and mixed environment with a single user and multi-users.

From the test results, we reached the following general conclusions:

- u The SAME method works.
- u Minimize disk contention by evenly distributing IO across the Symmetrix.
- u I/O block size set to 1MB demonstrated optimal performance.
- u VERITAS stripe depth set at 1MB demonstrated optimal performance.
- u Use VERITAS Quick I/O.

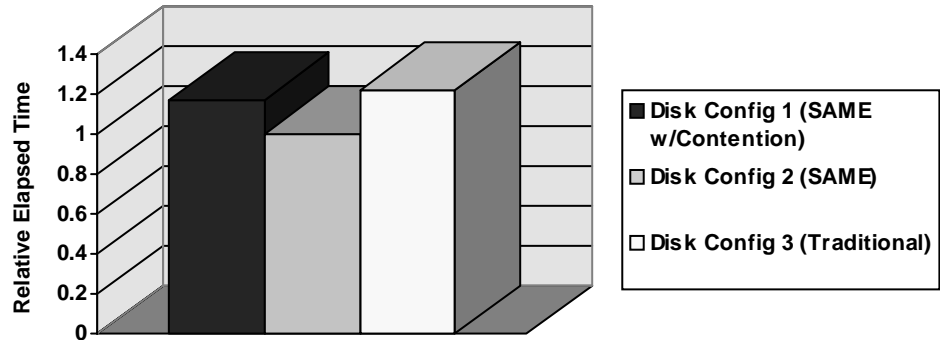
SAME Method Analysis

We excluded disk configuration 1 (32-way stripe for both primary and mirror) as the base configuration for comparison early because disk configuration 2 (16-way stripe for primary and 16 way for mirror) has better performance numbers. See “Disk Contention Analysis” for details.

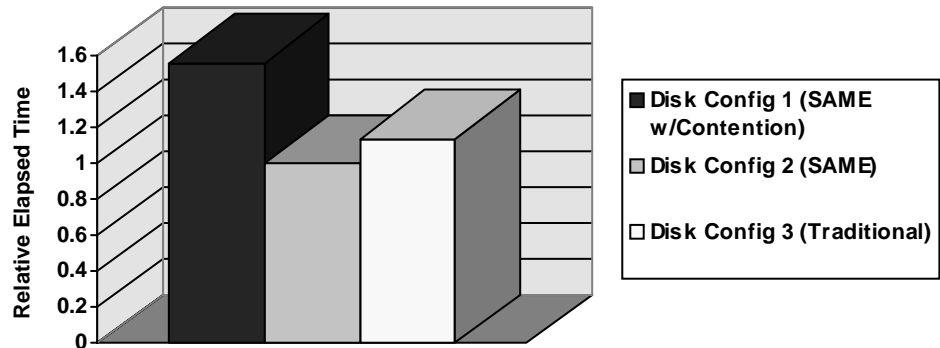
Comparing the test results from disk configuration 2 (SAME approach) and 3 (traditional approach), *SAME approach has equivalent or better results than the traditional approach*. For example, the full index scan time of disk configuration 2 is 22% and 14% faster than that of disk configuration 3 in a large sequential read environment and mixed environment (large sequential read and random read/write), respectively. The two disk configurations have the same amount of data and same number of disks, the only difference was data layout.



Full Table Scan (Large Sequential Read)



Full Table Scan (Mixed I/O Load)



Disk configuration 2 (SAME) also has *equivalent or better test results* than disk configuration 3 (traditional) in the following areas:

- u **Database build operations** - database creation, tablespace creation, table loading, and index creation.
- u **Large sequential read operations** - full table and index scan in single and multi-user environments.
- u **Large sequential write operations** - tablespace and index creation in single and multi-user environments.
- u **Random read write operations** - OLTP test with eight clients.

Our test results show the following benefits of the SAME method:

- u Striping across all disks ensures high disk utilization and performance.
- u Hotspots on individual disks are eliminated.
- u Mirroring data ensures high availability.
- u Overall performance is better.

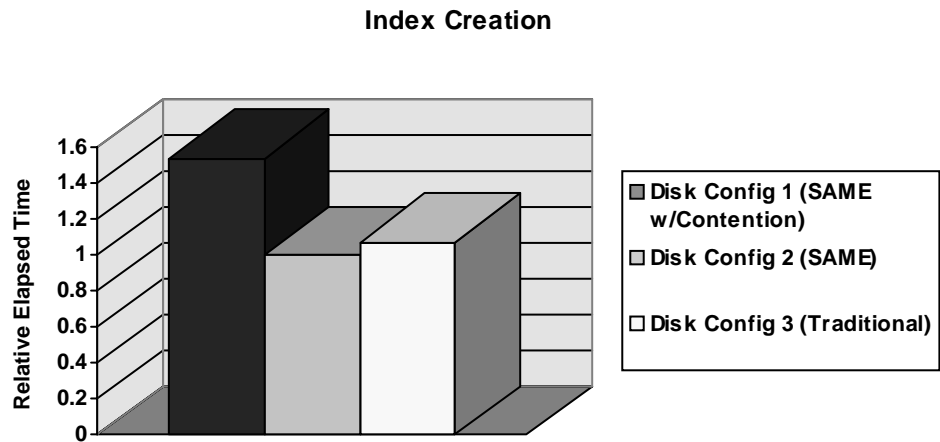


Disk Contention Analysis

We compared placing all the primary and mirrored data on the same disk group (disk configuration 1) and separating the primary and mirrored data into two disk groups (disk configuration 2). Disk configuration 1 can have disk contention due to the possibility of simultaneous access of primary data from one file or stripe set and mirrored data from another file or stripe set on the same disk. Disk configuration 2 does not have this problem. According to test results, disk contention does impact performance.

Full table scan time of disk configuration 2 (SAME) is 17% faster and 43% faster than disk configuration 1 (SAME with contention) in a large sequential read environment and mixed environment (large sequential read and random read/write), respectively.

We also observed that disk configuration 2 index creation time is 54% faster than that of disk configuration 1.



Striping Considerations

In general data striping increases Oracle I/O performance. The results of our testing showed some interesting trends depending on the environment. The variables that we changed during testing included:

- u Stripe depth (VERITAS Volume Manager stripe unit size).
- u I/O block size (size of I/O being striped across multiple disks).
- u Number of users (single user vs. multi-user environment).
- u Type of I/O (read-intensive vs. write-intensive environment).

Oracle I/O Block Size

We observed that for both read and write operations, regardless of the stripe depth, the following configurations gave us the best results:

- u **Write:** `DB_FILE_DIRECT_IO_COUNT` x Oracle block size should equal 1MB. The data showed that when performing a tablespace creation, 1MB I/O block size can be as much as 31% faster than 64K block size.
- u **Read:** `DB_FILE_MULTIBLOCK_READ_COUNT` x Oracle block size should equal 1MB. The data showed that when performing a full table scan, 1MB I/O block size can be as much as 62% faster than 64K block size.



General Recommendation: Our results indicated that in multi-user environments, the stripe depth should be set to 1MB. In our case, we varied the stripe depth between 256KB and 1MB and found that 1MB depth offered the best overall performance. Also, for maximum performance, the I/O block size should be calculated as the stripe depth multiplied by the stripe width. For example, we used stripe widths of 512KB, 1MB, and 2MB with the stripe depth of 1MB across two disks and found that the best performance occurred with the stripe width of 2MB.

Full Table Scan of 17GB Table With Parallel Degree of 4 DB_BLOCK_SIZE=8K

Stripe Depth	Oracle Parameter DB_FILE_MULTIBLOCK_READ_COUNT	Oracle Read I/O Size	Actual Read Size	Elapsed Time
1MB	8	64K	64K	8 min 46 sec
1MB	16	128K	128K	7 min 47 sec
1MB	128	1024K	1024K	6 min 13 sec
1MB	256	2024K	1024K	6 min 13 sec

Note Oracle Read I/O Size is equal to DB_BLOCK_SIZE x DB_FILE_MULTIBLOCK_READ_COUNT.

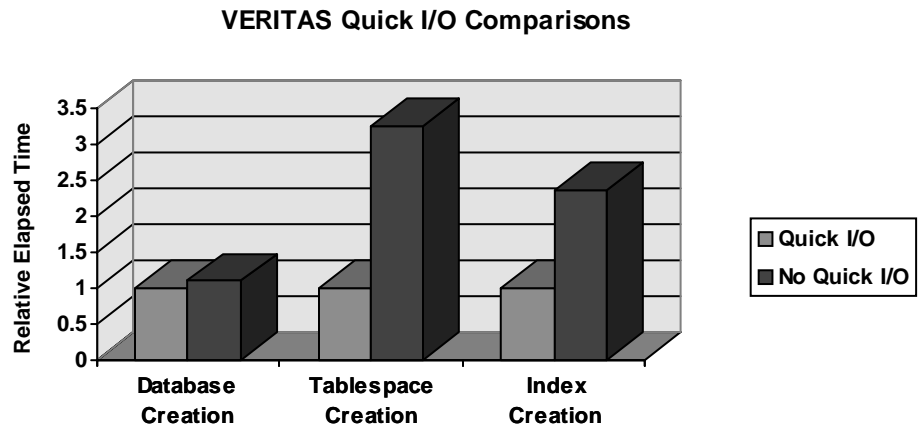
VERITAS Quick I/O Results

VERITAS Quick I/O technology showed dramatic performance improvements in our test configurations, especially in large sequential write environments. This increase is expected as VERITAS Quick I/O was designed to provide performance improvements in write-intensive OLTP (Online Transaction Processing) environments. VERITAS Quick I/O implements a direct I/O write capability that enables large I/Os to bypass typical file system buffering as well as eliminating potential redundant locking between the file system and Oracle databases. The test results demonstrated the benefits of VERITAS Quick I/O.

From the various tests, the most dramatic performance boost was for tablespace creation time. Tablespace creation can be up to 325% faster than a comparable Oracle configuration running on a file system without Quick I/O. Although it is not reflected in this paper, this likely resembled the equivalent performance we would expect if we ran the test using raw I/O. In essence, VERITAS Quick I/O provides the ability to have file system manageability with raw device performance.



The results listed below show the performance increase of running an Oracle database with VERITAS Quick I/O vs. running the same Oracle database on a file system without Quick I/O:



V. Conclusion

Simplicity is good. Our test results demonstrated that when configuring an Oracle database with a storage subsystem, the Stripe And Mirror Everything (SAME) method is effective. We were able to show that the SAME method works as well as, or better than, the traditional method. We also determined specific Oracle, VERITAS, and EMC Symmetrix configuration recommendations, as well as demonstrated improved database performance when using VERITAS Quick I/O. Our general conclusions are:

- u The SAME method works.
- u Minimize disk contention by evenly distributing IO across the Symmetrix.
- u I/O block size set to 1MB demonstrated optimal performance.
- u VERITAS stripe depth set to 1MB demonstrated optimal performance.
- u Use VERITAS Quick I/O.

Hardware and software technology advances have overcome some past limitations.

Oracle, VERITAS, and EMC products include many advanced capabilities, such as parallel database execution, efficient and scalable striping using VERITAS Volume Manager, and large Symmetrix cache, that make the SAME approach possible.

For example, the large EMC non-volatile cache eliminates the need for striping Oracle log files using fine grain striping to achieve very fast sequential I/O. The cache will buffer the write operation and de-stage it to disks at a later time.

The SAME method offers a simple, easy way to configure an Oracle database and storage while providing high data availability and performance. Plus the SAME approach is application and work load independent, so it can be implemented easily in a fast changing computing environment.

A new idea. Today, most Oracle DBAs do not configure an Oracle database and storage based on the SAME method because that is not how they were trained to implement Oracle and storage. Then, there is the issue of losing control and flexibility of a manually tuned storage configuration. However, database and storage solutions are getting bigger and more complex, just as the storage environment is changing at ever faster Internet speed. The SAME method offers a new and innovative way of correcting this problem.

In this paper, we described how we implemented the SAME configuration with the Oracle/VERITAS/EMC Symmetrix combination and obtained very good database performance results. Therefore, with the right storage software and hardware solution, we recommend that Oracle DBAs consider the new SAME approach.

VI. Appendix

Test Configuration Details

Hardware Details

SUN E3000 (Sun4u SPARC, Ultra-Enterprise)

4 x 248Mhz CPUs

2GB RAM

2026MB Swap

FC64-1063 EMC-S 64 bit FDDI controller

EMC Symmetrix 3630-36 (5265 Microcode)

2GB cache

32 x 36GB physical disks

32 disks were divided into 4 equal sections called hyper-volumes, with each hyper-volume of size 8.2GB. Only the first hyper-volume from each disk is used for testing eliminating the possibility of contention

The third and fourth hyper-volumes were used for backup of the database.

Configuration of hyper-volumes and mirroring is done at the EMC level (Operating System sees each hyper-volume as a separate disk, such as c2t15d0) and the hyper-volumes used for mirroring are not seen by the Operating System.

Software Details

Solaris 2.6

JNI Fibre Channel SCSI HBA driver (64 bit) - version 2.2.3.EMC

VERITAS Database Edition for Oracle - 2.0.2

VERITAS File System (VxFS) - 3.3.2

VERITAS Volume Manager (VxVM) - 2.5.6

VERITAS Quick I/O - 3.3.2

Database Details

Oracle8i Release 2 (8.1.6)

Database used for testing is an extract of Oracle Accounts Receivables application schema and the total size of the database is 50 GB including system, rollback segment and temporary tablespaces.

System tablespace	:500MB
3 x redo logs	:100MB (each)
Rollback tablespace	:2000MB
Tools tablespace	:200MB
Temp tablespace	:10GB



Data tablespace	:30 GB
Index tablespace	:10 GB

EMC Disk Configurations

Disk Configuration 1 (1x32)

VxFS with Quick I/O

Stripe depth (unit) 256K - 1MB, Stripe width 1x32 (stripe Oracle files over 32 drives) (mirror on the same 32 drives)

Disk Configuration 2 (1x16)

VxFS with Quick I/O

Stripe depth (unit) 256K - 1MB, Stripe width 1x16 for everything (mirror on the other 16 drives)

Disk Configuration 3 (1x2,1x2,1x12)

VxFS with Quick I/O

Stripe depth (unit) 256K - 1MB, Stripe width 1x2 for redo log1, 1x2 for redo log2, 12 for everything else (mirror on the other 16 drives)

Step-by-Step Setup

1. Install and configure the EMC Symmetrix	Contact EMC Systems Engineer assigned to the customer site to develop a Symmetrix configuration optimal to your environment. Refer to the recommendations listed in “Figure A” of the appendix.
2. Install JNI fibre adapter hardware for EMC	<p>In our test environment, we used JNI 64-bit S-Bus fibre adapter (FC64-1063-EMC-S 64-bit adapter with integrated non-OFC optical interface short wave).</p> <p>Install the adapter according to hardware installation guide.</p> <p>Reboot the host</p> <p>(command line <i>>reboot -- -r</i>) (boot prompt <i>ok>boot -r</i>)</p> <p>After reboot, verify presence of the adapter card with the following command:</p> <p><i>prtconf -v grep fcaw</i></p> <p>Response should be: fcaw (driver not installed)</p>



<p>3. Configure JNI device driver software</p>	<p>Install the JNI device driver package from the supplied CD-ROM.</p> <p>Note The CD-ROM normally contains the driver package for EMC. If it does not, you can download it from http://www.jni.com/fibre/ftp/Released/Symmetrix/Solaris/v2.2.3.EMC. Select “JNIfcawEMC.pkg” and save the link. The driver version we used in our test is 2.2.3.EMC.</p> <p>The driver package is installed using:</p> <pre>pkgadd -d JNIfcawEMC.pkg</pre> <p>After successful installation of the driver, verify the device driver:</p> <pre>prtconf -v grep fcaw</pre> <p>Response should be</p> <pre>fcaw, instance #0</pre> <p>At this point, the operating system sees the JNI adapter card, but not the target drives (EMC disks/hyper-volumes).</p>
<p>4. Install EMC Control Center (ECC):</p>	<p>Customer selection of hyper-volumes will require:</p> <ul style="list-style-type: none"> u ECC Symmetrix Manager, version 4.01 u ECC Symmetrix Disk Reallocation, version 4.01 <p>Reference the ECC Product manuals for installation and configuration instructions</p>



5. Identify devices to be used in striped volume sets:	<p>Customers should identify devices to be used in stripe sets, in a pattern such that IO will be evenly distributed across as many of the backend physical disks and disk directors (controllers) as possible:</p> <ul style="list-style-type: none">u Start the EMC Symmetrix manageru Open the Disk Director Configuration Viewu All types of Symmetrix devices will be visible. Customers should select only M1 (standard) devices. The portion of the disk device represented in yellow denotes the type of Symmetrix device.u Select standard devices (blue icons) so that they are evenly distributed across as many of the backend disk directors as possible and across physical disk drives. <p>For Example: In figure A (below), devices 0D, 0F, 11, and 13 are located across physical disks and across all available disk controllers. In our tests, two disk directors provided a total of eight backend adapter ports. Stripe sets in this environment should be multiples of 8 volumes in size to evenly distribute IO activity across the backend of the Symmetrix. Should sixteen ports be available, stripe set increments of 16 volumes would be recommended.</p> <p>Customers should also select devices for a stripe set according to a “wide then deep” pattern. First select devices across all available backend disk directors (wide) before selecting devices from the same director (deep). For large stripe sets where striping deep is required, select the devices in a C, E, D, F pattern to take advantage of the dual processors on the disk directors.</p> <p>For Example, in Figure A there is one disk director (DA02) represented. The Symmetrix supports two to eight directors, depending on the model. Devices should be selected across the available directors (DA02, DA03, DA04, etc.) before selecting devices down a director (DA02: C, E, D, or F).</p>
---	---

6. Assignment of Devices to host channels:	<p>The devices that were identified in step 3B, now must be evenly distributed across the channels connecting to the host.</p> <ul style="list-style-type: none">u In the EMC Symmetrix Manager GUI, Open the Controls menu; select SDR (Disk Reallocation) and open.u Enter the customer passwordu From the Symmetrix Disk Reallocation Window the unassigned Standard devices will be available in a pool at the bottom of the screen. <i>Figure C</i> is an example of this window.u Select the devices chosen in step 3B and evenly distribute them across the channels attached to the host. Specific device numbers should be dragged from the available pool and dropped across the available host channels. <p>In our prior example we selected four devices, 0D, 0F, 11, and 13. If possible, assigning each of these devices to a separate host channel will provide an even distribution for IO. If there are two channels available, then 0D/0F should be assigned to one channel and 11/13 to another.</p> <ul style="list-style-type: none">u Assign a Target and LUN address for each deviceu When all of the devices have been allocated, open the controls menu, select SDR and commit. <p>NOTE: EMC Systems Engineers can also pre-assign devices at the time of the Symmetrix installation if the customer requests it.</p>
---	--



<p>4. Configure host and kernel settings</p>	<p>Edit <code>/kernel/drv/sd.conf</code> file and make entries for EMC disks (LUNs)</p> <p>For example:</p> <pre>name="sd" class="scsi" target=15 lun=0; name="sd" class="scsi" target=15 lun=1; ... and so on for all LUNs and targets</pre> <p>Reboot the host. During reboot, you may see warnings like:</p> <pre>Corrupt label - wrong magic number</pre> <p>To fix the above warning for the EMC disks, invoke the format command, select EMC disks, and label them (except gatekeeper disks).</p> <p>To enable the necessary scsi fibre commands, we used the following EMC related kernel parameters in our test environment (set in <code>/etc/system</code> file):</p> <pre>set sd:sd_max_throttle=20 (default is 256, but in most EMC configurations, it is set to 20) set scsi_options = 0x7F8 set sd:sd_io_time = 0x78</pre> <p>For more information, see the <i>EMC Fibre Channel Product Guide</i>.</p> <p>Note I/O transfer size from host to EMC Symmetrix is controlled by kernel parameter 'maxphys' with a default value of 128K. If an I/O block size exceeds the default value, the I/O request will be broken up into more than one requests - each request does not exceed the default size.</p> <p>Please refer the following JNI file for further details: http://www.jni.com/fibre/ftp/Released/Symmetrix/Solaris/v2.2.3.EMC/release.txt. See Section 1.2, "Raw I/O tuning Recommendations." Set default value to 1MB.</p> <p>At this point, the operating system sees the EMC drives.</p>
---	---



<p>5. Install VERITAS software packages</p>	<p>Install VERITAS Database Edition 2.0.2 for Oracle (Solaris 2.6). Please refer to the <i>VERITAS Database Edition for Oracle Installation Guide</i> for complete details.</p> <p>As part of VERITAS software installation, we installed the following components:</p> <p>VRTSdbes VERITAS Database Edition for Oracle - 2.0.2 VRTSdbpat VERITAS Database Edition for Oracle VxFS Patch -2.0.2 VRTSqio VERITAS file device interface - 3.3.2 VRTSvxfs VERITAS File System - 3.3.2 VRTSvmsa VERITAS Volume Manager Storage Administrator -3.0.2 VRTSvsma VERITAS Storage Manager Agent - 3.2.2.1 VRTSvxva VERITAS Volume Manager Visual Administrator - 2.5 VRTSvxvm VERITAS Volume Manager - 2.5.6 VRTSvxvm.2 VERITAS Volume Manager, Binaries 3.0.1 VRTSvmman VERITAS Volume Manager (manual pages) - 2.5.6</p> <p>VERITAS related kernel parameters in our test environment (set in /etc/system file)</p> <p>* vxvm_START (do not remove)</p> <p>forceload: drv/vxdmp forceload: drv/vxio forceload: drv/vxspec</p> <p>*vxvm_END (do not remove)</p> <p>* vxfs_START -- do not remove the following lines:</p> <p>* VxFS requires a stack size greater than the * default 8K. * The following values allow the kernel stack * size for all threads to be increased to 16K.</p> <p>set lwp_default_stksize=0x4000 set rpcmod:svc_run_stksize=0x4000</p> <p>* vxfs_END</p> <p>We are now ready to add EMC drives into VERITAS Volume Manager control.</p>
--	---



<p>6. Create a volume</p>	<p>Start vmsa GUI tool for VERITAS storage configuration</p> <pre>/opt/VRTSvmsa/bin/vmsa</pre> <p>The vmsa tool has easy interface to add and initialize the disks. Once all EMC drives are initialized (under some disk group such as emcdg), we can create a volume of desired properties. In our SAME method, we created a volume called emcvol1 with the following layout: striped, 16 columns, Stripe Unit Size of 256K (Configuration 1).</p> <p>Click on the “Assign disks” button and select 16 disks. Then, click on the “Add file system” button and select VxFS FS type.</p> <p>The above VERITAS step can be done at the command level as well.</p> <p>Now, we should have one volume striped across a desired number of disks with selected Stripe Unit Size, with a VxFS file system created and mounted (for example, the mount point is /emcfs_data).</p> <p>Note Oracle and VERITAS recommend using the SAME volume for the Oracle database only.</p>
<p>7. Enable Quick I/O and datafile creation</p>	<p>By default, the mounted vxfs file system is not Quick I/O enabled. To add Quick I/O support, pre-allocate database files using:</p> <pre>/usr/sbin/qiomkfile -s 500M /emcfs_data/system.dbf</pre> <p>This command creates two files, the actual file (.system.dbf) of specified size, and a soft link file (system.dbf) pointing to the actual file:</p> <pre>> ls -al /emcfs_data/*system.dbf -rw-rw-r-- 1 oracle dba 524320768 Mar 8 17:47 .system.dbf lrwxrwxrwx 1 oracle dba 23 Feb 28 17:21 system.dbf > .system.dbf::cdev:vxfs:</pre> <p>Datfiles referred by the database should be the soft link file (system.dbf), not the actual file (.system.dbf). Pre-allocate all datfiles before being used in tablespace creation including redo log files</p> <p>When using pre-allocated Quick I/O files, you do not need to specify the size of the datafile (for example, Create tablespace xyz datafile '/emcfs_data/system.dbf' reuse. In this case, Oracle will determine datafile size and create the tablespace. Please note that specifying size 0 will give an ORA-1257 error)</p> <p><i>Oracle I/O related init.ora parameters we set in our test are:</i></p> <pre>u db_block_size=4096 # default 2048 u db_file_multiblock_read_count=256 # default 8 # set to 256 to have multiblock read I/O of 1M size (256*4096) u db_file_direct_io_count=256 # default 64 # set to 256 to have datafile initialization I/O of 1M size (256*4096)</pre>



Figure A

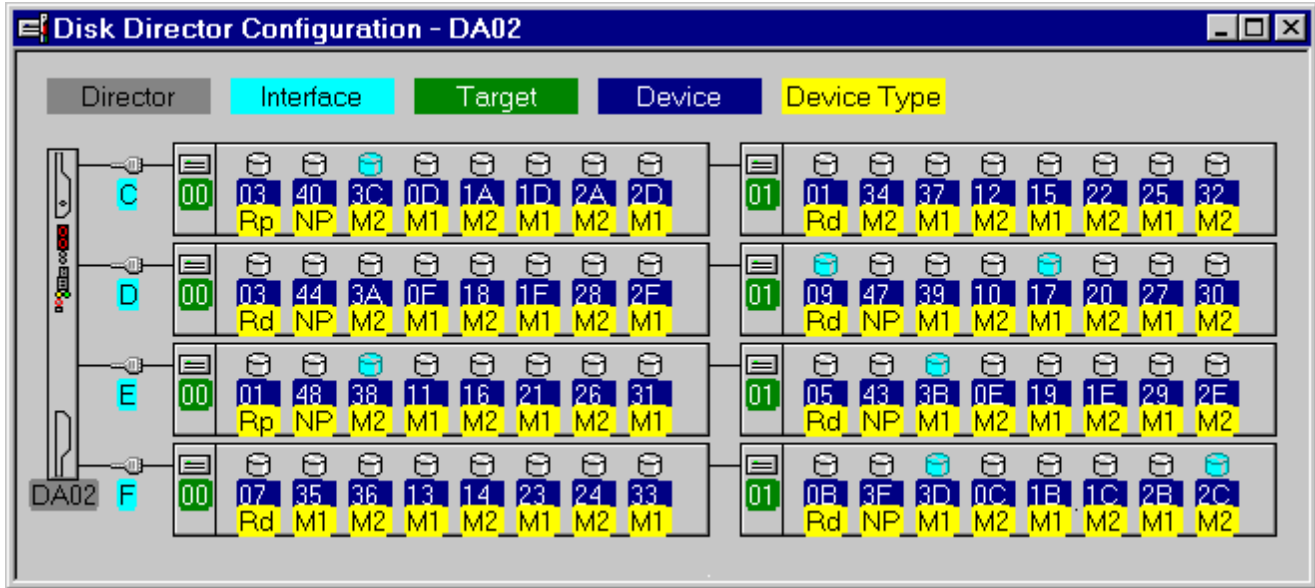


Figure B

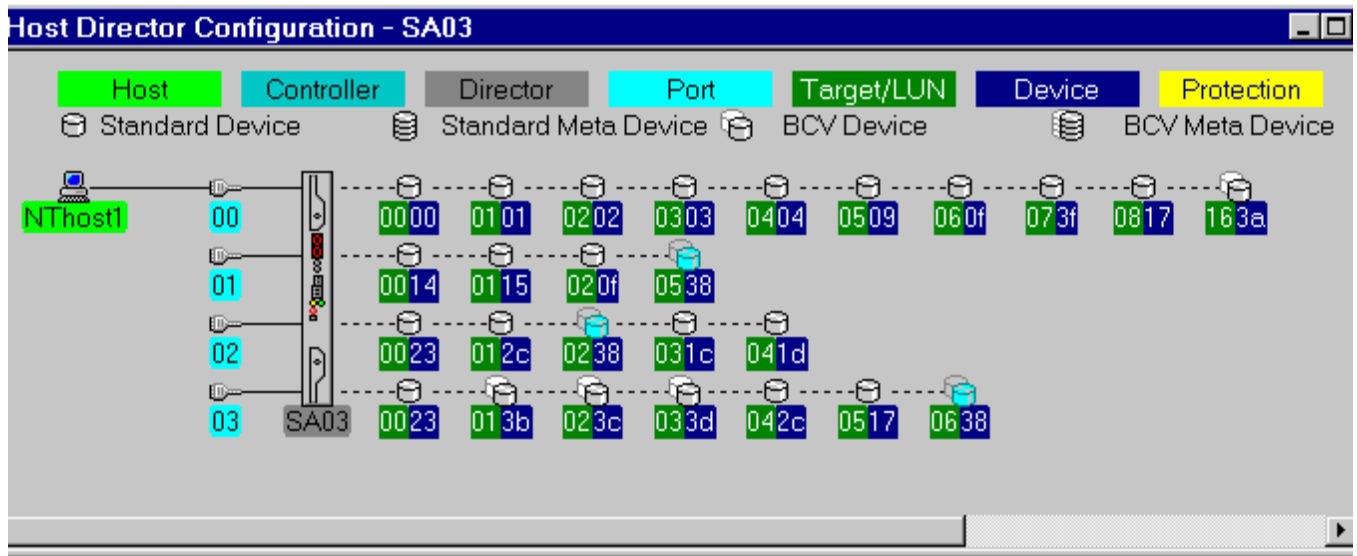
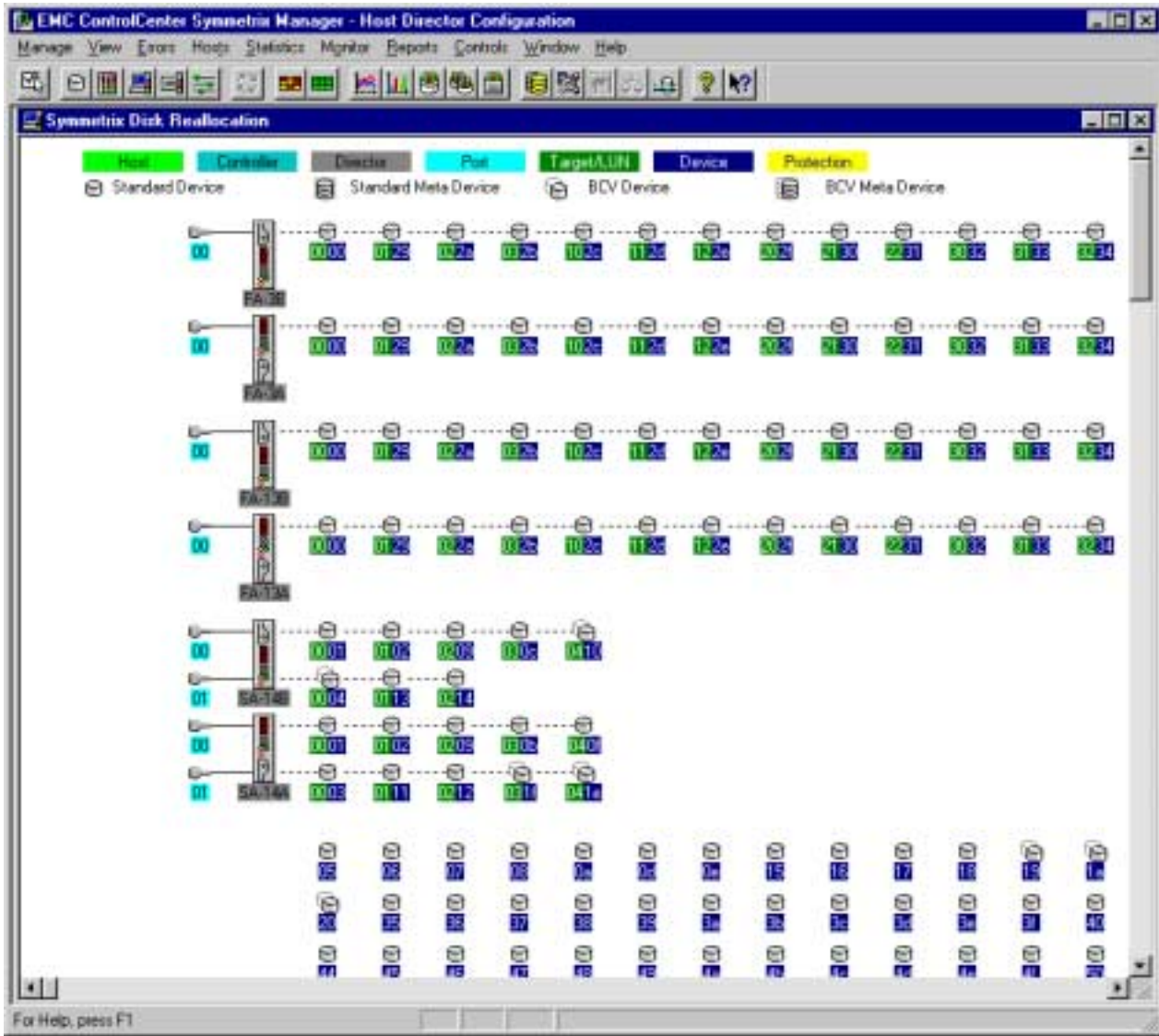


Figure C



EMC Solutions:

The following is a brief description of the EMC products discussed in this document. More detailed product information can be obtained from the EMC company website:

<http://www.emc.com/products/products.htm>

EMC Symmetrix: The Symmetrix series Enterprise Storage systems provide a shared repository for a company's most valuable resource—its information. Symmetrix systems provide the industry's highest performance, availability, and scalable capacity with unique information protection, management, and sharing capabilities for all major systems, mainframe, and other environments.

EMC Control Center: EMC Control Center (ECC) is a centralized, server-resident software application for managing an EMC Enterprise Storage configuration from a single console. ECC provides extensive user management of storage components across the Enterprise Storage network including monitoring, configuration, control, tuning and planning capabilities.

In this document we discussed two components of ECC:

EMC Control Center Symmetrix Manager: Symmetrix manager is the foundation of the ECC solution, providing convenient access to internal configuration, operational status and real-time performance information.

EMC Control Center Symmetrix Disk Reallocation: Symmetrix Disk Reallocation (SDR) gives an administrator the ability to reassign existing Symmetrix open systems logical volumes between Fibre and SCSI adapters, Target/LUN addresses or open system hosts. Using a drag-and-drop window SDR facilitates workload balancing by allowing volumes to be evenly distributed across the Symmetrix.

EMC PowerPath: PowerPath is a server-resident software product from EMC that works with a Symmetrix storage system to deliver intelligent I/O path management. With EMC PowerPath, administrators can improve the server's ability to manage heavy storage loads through continuous, intelligent I/O balancing. PowerPath automatically configures multiple paths and dynamically tunes for performance as workloads change. PowerPath also adds to the high-availability capabilities of the Symmetrix storage system by automatically detecting — and recovering from — server-to-storage path failures

VERITAS Database Edition for Oracle

VERITAS Database Edition for Oracle product information can be obtained from VERITAS company's web site:

<http://www.veritas.com/us/products/>

VERITAS Database Edition includes the following major components.

VERITAS File System

The VERITAS File System (referred to as VxFS) is an extent-based, intent logging file system intended for use in environments that deal with large volumes of data and that require high file system performance, availability, and manageability. VxFS also provides enhancements that make file systems more viable in database environments.



VERITAS Quick I/O

Databases can run on either file systems or raw devices. Database administrators often create their databases on file systems because it makes common administrative tasks (such as moving, copying, and backing up) easier. However, running databases on most file systems significantly reduces database performance.

When performance is an issue, database administrators create their databases on raw devices. VxFS with Quick I/O presents regular, preallocated files as raw character devices to the application. Using Quick I/O, you can enjoy the management advantages of databases created on file systems and achieve the same performance as databases created on raw devices.

More specifically, Quick I/O's ability to access regular files as raw devices improves database performance by:

u **Supporting Kernel Asynchronous I/O**

Asynchronous I/O is form of I/O that performs non-blocking system level reads and writes, which enables the system to handle multiple I/O requests simultaneously. Operating systems such as Solaris provide kernel support for asynchronous I/O on raw devices, but not on regular files. As a result, even if the database server is capable of using asynchronous I/O, it cannot issue asynchronous I/O requests when the database runs on file systems. Lack of asynchronous I/O significantly degrades performance. Quick I/O lets the database server take advantage of kernel-supported asynchronous I/O on file system files accessed using the Quick I/O interface.

u **Supporting Direct I/O and Avoiding Double Buffering**

Most database servers maintain their own buffer cache and do not need the system buffer cache. Database data cached in the system buffer is therefore redundant and results in wasted memory and extra system CPU utilization. By supporting direct I/O, Quick I/O eliminates double buffering. Data is copied directly between DBMS cache and disk, which lowers CPU utilization and frees up memory that can then be used by the database server buffer cache to further improve transaction processing throughput.

u **Avoiding Kernel Write Locks**

When database I/O is performed using the write() system call, each system call acquires and releases a write lock inside the kernel. This lock prevents multiple simultaneous write operations on the same file. Because Oracle databases implement their own locks for managing concurrent access to files, per file writer locks unnecessarily serialize I/O operations. Quick I/O bypasses file system locking and lets the database server control data access.

VERITAS Volume Manager

The VERITAS Volume Manager (referred to as VxVM) builds virtual devices called volumes on top of physical disks. Volumes are accessed by a file system, a database, or other applications in the same way physical disk partitions would be accessed.

Using volumes, VxVM provides the following administrative benefits for databases:

- u Spanning of multiple disks—eliminates media size limitations.
- u Striping—increases throughput and bandwidth.
- u Mirroring or RAID-5—increases data availability.
- u Online relayout—allows online volume layout changes to improve database performance. As databases grow and usage patterns change, online relayout lets you change volumes to a different layout. This is accomplished online and in place. Use online relayout to change the redundancy or performance characteristics of the storage, such as data organization (RAID levels), the number of columns for RAID-5 and striped volumes, and stripe unit size.
- u Hot-relocation—automatically restores data redundancy in mirrored and RAID 5 volumes when a disk fails.
- u Fast volume resynchronization—ensures that all mirrors contain exactly the same data and that the data and parity.
- u Volume snapshots— allows backup of volumes based on disk mirroring.
- u Dynamic multipathing (DMP)—allows for transparent failover, load sharing, and hot plugging of SCSI devices.
- u Free space pool management—simplifies administration and provides flexible use of available hardware.
- u Online administration—allows configuration changes without system or database down time.



