EMC²

Performance and Tuning Considerations for Running Oracle Databases on EMC Storage

January 1998

C722

Table of Contents

Executive Summary

The use of an EMC Symmetrix® Enterprise Storage System as the underlying storage solution for Oracle database applications provides several performance enhancements compared to standard storage solutions. This paper provides an overview of the Symmetrix architecture and then explains how it enhances performance. The purpose of this paper is to demystify the factors that need to be considered when configuring and tuning an Oracle database running on a system that uses EMC Enterprise Storage™.

Symmetrix has a large cache that is leveraged to increase overall system performance. With the EMC prefetch algorithms, faster reads are achieved by finding more data in memory (cache) and with RAID 1 dual-access to data enabled. Faster writes occur because once a write is placed in nonvolatile cache, the OS does not have to wait until the data is written to disk. The overall system performance is further enhanced by moving the burden of mirror maintenance from the OS to the Symmetrix.

The configuration of a Symmetrix is flexible and yet can be complex at times. Being able to map logical devices back to physical disks requires navigating several layers of indirection. Although not required for good performance, it is important to understand the configuration in this level of detail when setting up striping.

The question, "to stripe or not stripe" is another area of confusion in that Symmetrix does device distribution to increase throughput. In most Symmetrix configurations, the multiple devices per host channel are spread across the many disks on the back-end. This allows those reads, which are not in cache, to be satisfied by a larger number of spindles. However, since most installations have a high cache hit rate, this is often not an issue. The results will vary depending on the type of work load, access methods and overall application architecture.

The cache in the disk subsystem removes some of the limitations of JBOD (just a bunch of disks). With a large cache available in an EMC disk array, there are some conventional Oracle tuning practices which may no longer be necessary to achieve similar performance gains.

Thousands of production Oracle applications use EMC storage as a component of the overall solution. This combination has led to many frequent questions about how to best tune and configure Oracle to run on Symmetrix for optimal performance. This paper is intended to be a starting point for sharing information and an understanding of answers to these questions.

Purpose

The intent of this white paper is to:

- Provide a general overview of the Symmetrix architecture and components with a perspective on performance.
- Help improve understanding of how an EMC Symmetrix system can increase database performance.
- Demystify the configuration issues that might impact performance and maximize effectiveness.
- Provide some basic configuration and tuning guidelines for Oracle and EMC users.

Audience

This white paper has been written to address experienced Oracle developers, DBAs and system administrators who are familiar with EMC but not experts on Symmetrix integrated cached disk arrays. It is assumed that the concepts of hardware and storage principals are understood.

Disclaimer

This paper has been written to reflect the information known to the author at this time and is not a definitive authority on the subject of Oracle performance and tuning. There are many books available on Oracle performance and tuning (Appendix 1 is only a partial list).

This paper will be updated and re-released as more research and feedback from readers is incorporated. Your feedback as to how this can be improved is appreciated. Please email your comments and constructive suggestions to: Manning_Paul @isus.emc.com.

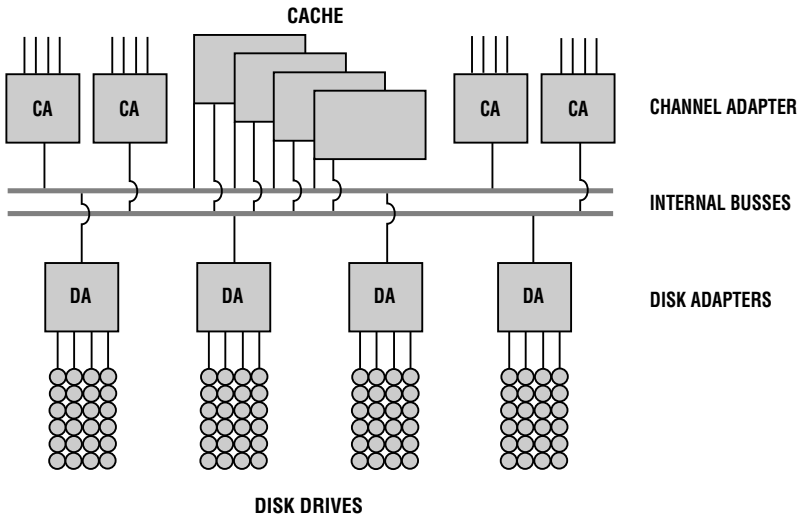
Symmetrix Architecture

Symmetrix is a line of EMC intelligent storage systems. These storage systems have several components that help provide reliable high performance access to data. Diagram 1 shows most of these components. A brief description of the following components and their function is provided as a high level overview:

- Cache
- Internal Busses
- Disk Adapters
- Channel Adapters
- Battery Backup Power
- Automatic Phone-Home Support Module

At the center of this architecture is the **cache**, which is central to enhancing performance. It provides shared memory for the storage subsystem. This cache is nonvolatile, in that it has battery backup and is dual-ported. All memory boards are connected to both of the system's two **internal buses**. These internal buses are the backbone of the system and provide access to other components within the disk array. The cache integrates what is often referred to as the front-end and backend of the Symmetrix.

Diagram 1. Symmetrix Architecture



On the **backend** are **disk adapters** (DAs), sometimes called disk directors, which manage all I/O from cache to the physical disks. These DAs are paired with neighboring DAs to provide redundant access paths that can be used in the event of either a bus or DA failure.

Off of each disk director there are four channels to which up to six **disks** can be attached. The disks can be 4GB, 9GB, 18GB or 23GB physical drives.

On the **front-end** the **Channel Adapters** (CAs), also referred to as SCSI Adapters, manage the interface of data (I/O) requests going to and from each host system. EMC supports SCSI fast and wide differential (FWD), Ultra SCSI and Fibre Channel Arbitrated Loop (FC-AL) for Open Systems and Windows NT connections, and ESCON with bus and tag connections for Mainframe systems. Each CA card provides multiple cable connections. Ultra SCSI and FWD CA cards have four connections per CA while FC-AL has two connections per CA card. Remote Link Directors (RLDs) can be installed on the front-end and provide interconnect to a remote Symmetrix for Symmetrix Remote Data Facility capability.

Other components not shown in the diagram, but that provide an important function are the **battery backup** and the **Automatic Phone-Home Support Module**. If the Symmetrix loses power, its batteries will provide power for up to fifteen minutes. After three minutes of no power, the Symmetrix will close all CA traffic and begin an orderly shutdown. The cache is destaged to disk and all disk mirrors are synchronized before the system shuts off.

The microcode running in cache continuously runs tests and checks for problems, which if detected, are reported to the EMC Customer Service Center via a phone-home function. This feature enables problems to be corrected before they cause data unavailability. Microcode runs on each disk drive and measures the distance between the disk platter and the disk head. If the system detects the head is getting too close to the platter, it logs an error and phones home so that a replacement can be installed before a head crash occurs.

One example of this phone-home feature having a significant financial impact is a case where a data center air conditioner failed in the middle of the night. The problem was detected by the Symmetrix environmental tests. After the box phoned EMC support and logged an error, the System Administrator was called at home and informed of the problem. That call saved that particular system and many other systems in the data center from having a meltdown.

It is important to be familiar with these components of a Symmetrix in order to understand how this integrated cached disk array can improve system performance.

EMC Performance Gains

The cache and microcode that runs in cache are the two main performance accelerators. In a typical Oracle environment where EMC storage has replaced standard disks, overall performance has increased on average 20-30 percent. Performance increase will vary depending on the type workload, the percent busy and the amount of tuning which has been done on the system prior to disk system upgrade.

Below are some ways in which Symmetrix enables enhanced performance.

1. **Off loading the host system** of the task of mirror maintenance. Symmetrix presents the host-based system devices that are protected (with RAID 1 or RAID-S) in a way which is transparent to that host system. In short, Symmetrix takes care of redundant protection without the host-based system having to expend any additional CPU cycles. This allows the host system to do more for applications and leave the busy work to the disk subsystem. This may not be a noticeable amount of performance gain on a system that is not busy. However, it makes a big difference on a system that is CPU-bound. The performance is further accelerated when the Symmetrix is managing third mirror or remote mirrors, providing greater flexibility in further off loading the cycles needed to run

batch processing and backups. This topic is covered in greater detail as part of an EMC presentation on business continuance and will not be addressed here.

2. **Faster Writes —** When a write request is put into a nonvolatile cache the write is instantly confirmed, thereby enabling faster write performance. The Symmetrix then destages the cache and synchronizes the mirrors while the host system is going on to its next task. Although the amount of cache in a Symmetrix is quite large, it is also important to note that this is not an unlimited resource. There will come a point in some work loads (like database loads or massive index creates) where cache becomes saturated and the Symmetrix will have to defer some writes to flush cache in order to enable faster writes. However, in most situations the cache hit rate for writes is very high and performance is greatly enhanced by this feature.

3. **Faster Reads —** It is generally known that reads will be much faster if found in memory instead of on disk. This applies to both shared global area (SGA) and Symmetrix cache. With a large amount of cache there is a good chance that recent writes of data may still be in cache. However, the real read performance gain with Symmetrix comes from the prefetch logic. This is the process of anticipating what might be needed in cache so that it will be there when requested. The prefetch algorithms which run in cache on Symmetrix are designed to maximize the cache hit rate for reads.

4. **Dual-Access with RAID 1 —** When data is read from a mirrored device, Symmetrix may get the data from either the primary (M1) or the secondary (M2) copies of the data and reduce disk contention, which leads to queuing and disk wait time. Contention that is caused by conflicting disk I/O operations accessing the same disk can be avoided by using the alternate mirror, resulting in a large performance increase. In the case of a large sequential read, the system can also get alternate blocks from both M1 and M2 devices and fill up a buffer faster than reading from a single drive, again increasing the throughput rate.

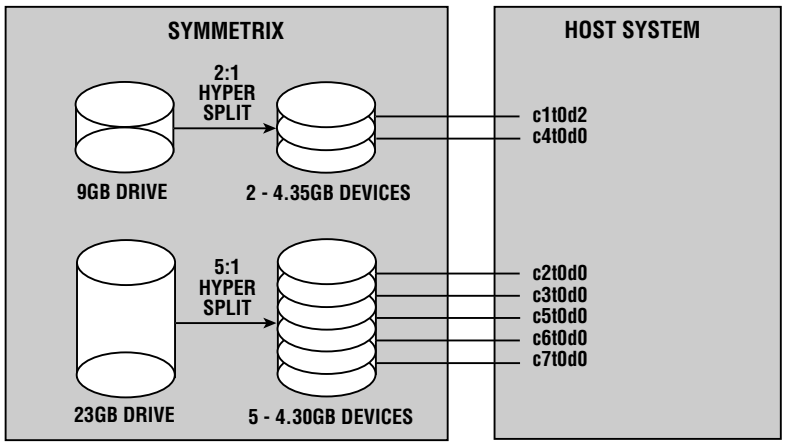
5. **Symmetrix device distribution —** The standard configuration of Symmetrix has multiple disk drives per channel that are spread out across the backend with the mirrors (M1 and M2) located on different DAs. This configuration allows the large I/O requests to be satisfied by multiple processors and data channels within the Symmetrix to minimize channel contention and maximize throughput. This approach is done for the same reason one might use host-based striping. The net result is if the cache does not satisfy the I/O request, the wait is dependent on how long it takes to move a certain amount of data to or from disk. The spreading out of the number of independent arms engaged to address this movement will speed it up.

Physical-to-Logical Mapping

Before tuning an Oracle database residing on EMC storage, it is important to understand the mapping of physical-to-logical devices. To clearly understand which logical devices are mapped on which physical disk, one needs to navigate their way through the logical volume mapping and then find their way through the Symmetrix configuration to locate the physical disk drives.

Most Symmetrix systems today are populated with either 9GB or 23GB drives. These numbers do not in fact represent the true amount of useable disk space. In the case of the Seagate 9GB drive, there is about 8.7GB of addressable and useable capacity. In the example depicted in Diagram 2, the OS is presented a set of devices that are a little over 4GB each. The Symmetrix configuration creates hyper-volumes which are logical subsets of a bigger disk. With the 9GB drive, configuring a 2:1 hyper split presents two 4.35GB devices for each physical drive. In the case of the 23GB drives, configuring a 5:1 hyper split provides five 4.3 GB devices.

Diagram 2. Physical-to-logical mapping.



The Symmetrix configuration determines:

- Number of hyper splits per physical device
- Level of protection (RAID 1 or RAID-S)
- Number of devices to be seen on each channel
- The Target ID and LUN assigned to each device it presents to each host channel
- The mapping of all Symmetrix devices seen and unseen by the OS (mirrors and gatekeepers)

In the case of shared disks in a cluster setup, the configuration can present a given device to several host channels. This enables a cluster to share disks without having to share channels or cables. This increases simplicity, performance and reliability and avoids cabling snarls.

At boot up time, the OS will scan all of its I/O channels and assign and construct entries in its physical device tables for each drive it finds on each controller. Each Symmetrix device will show up as a standard disk and is assigned a physical address such as /dev/rdsk/c1t0d2. This example translates into a raw device with

- Controller 1 (c1)
- Target 0 (t0)
- LUN 2 (d2)

Using a system management utility, the system administrator can then mount these devices as part of a file system or use a Logical Volume Manager to address them as part of a volume group.

The Logical Volume Manager is needed if striping is to be done at the host-system level. Volume groups are assembled and logical volumes are created as subsets of each volume group.

To avoid confusion often associated with these layers, which need to be understood when mapping logical devices to physical drives, it is a good idea to be consistent with how components are referred to in each layer. As a matter of practice, always refer to the physical drives as HDAs or "spindles." Refer to hyper-volumes as "devices," which are seen as the same entity from both the Symmetrix and OS perspective. Reserve the term "logical" for logical volumes at the LVM layer or file system level.

Physical	Middle	Logical
HDA	hyper	volume
spindle	device	logical
drive		

Table 1. Terms for components

Knowing the physical-to-logical mapping is important to understand when tuning a system. It is particularly important if you are setting up OS striping in that you do not want to stripe across four channels, thinking these are all separate drives, and find out later that the Symmetrix configuration has those devices on the same physical drive or backend channel.

To stripe or not to stripe

The question of striping is a subject of hot debate with seemingly no correct answer. This is partly due to the fact that striping can either help or hurt performance, depending on the workload and how well a database might already be partitioned. In general, the need to do OS striping on a Symmetrix has been reduced by the cache, and intelligent algorithms help eliminate bottlenecks and achieve what OS striping enables (faster I/Os).

What is striping?

Striping is defined as a series of disks or devices that are shared by multiple objects/files. An example might be placing four disks into a group and then putting files or raw partitions across all four drives. After writing a certain number of blocks to one device, the system moves to the next device, thereby spreading out data across four spindles. Reads from that file or raw partition can get data from multiple locations at one time and are not limited by the I/O rate of a single device.

Stripe Width/Stripe Set - The number of devices/ drives across which one stripes.
Stripe Depth/Stripe Size - The amount of data written to a device before moving to the next.

In most cases, the striping is managed by a host-based Logical Volume Manager (LVM), though in some cases it is done through explicit partitioning at the Oracle database level.

Why does one stripe?

Two main objectives can be achieved by striping:

- Avoid I/O hot spots and individual physical drive limitations by spreading all I/Os across multiple channels and devices.
- Enhance overall throughput for large sequential read and write environments by reading and writing the multiple spindles all at once.

OS Striping with Symmetrix

Symmetrix disk arrays are configured to distribute devices on the backend, in that disks on that same channel are mapped to different DAs. Therefore, the need to stripe is already partially satisfied. The second feature that enhances performance, with or without striping, is the fast writes enabled by the nonvolatile cache. The Symmetrix cache is leveraged with striped data in the same way it is used with unstriped data. However, there is a slight disadvantage in that the prefetch logic is not able to guess as well what the next track to be read is if striping is used as it does when striping is not in place. This disadvantage is small in most environments and is easily offset by the overhead cost of managing the striping at the OS level. In summary, in most cases, there is little difference in overall performance whether or not OS striping is used with a Symmetrix.

Striping with Data Warehousing

The kind of work load found in data warehousing environments is typified by large scale queries that do full table scans. During database builds and loads there are many large scale operations, such as create database, create tables and create index for large sequential tables. In this type of work load, performance is limited by how fast the entire contents of the database can be processed. However, these large sequential reads are the type of workload on which the Symmetrix prefetch algorithms achieve the highest read cache hit rate. Striping can speed up single threaded operations, but at the same time can reduce the efficiency of the hit rate prefetch logic. If data warehouse load speed is more important than standard performance, then striping makes sense. If loads are not constrained, than the Symmetrix prefetch will enable faster reads than OS striping.

Striping considerations

Some additional considerations are:

- Once striping is set up, it is more difficult to reconfigure than unstriped environments. This results in additional time the system administrator needs to devote to setting up and maintaining the striped environment.
- Although the DBA may not have to worry about database partitioning with striping, it is harder to reallocate and reassign storage.
- There is an overhead associated with managing the striping which should be considered. This impact is far higher on a busy system than on an idle one.
- Striping with 3rd mirrors is complex to setup and thereby increases the complexity of managing EMC TimeFinder and SRDF.

There have been some detailed studies on performance relative to varying stripe size and width, but the conclusions seem to vary with work load. In general, if there are multiple simultaneous sequential reads to volumes on a given set of drives, striping is likely to decrease performance, compared to not striping. In all other cases, striping should improve or provide the same performance as not striping.

As for a few rules of thumb:

1. When doing striping, it is important to use increments available on the Symmetrix. For 32 drives, you should consider a stripe set of 4 to 8. Be symmetrical to avoid creating a roving hot spot.
2. It makes sense to use a stripe size that is a multiple of block size, as set in the init.ora file and not fragments of that size. Also, always start with at least 2X the block size.
3. A good starting point is a 4-way stripe set with 64K stripe size. An upward limit is an 8-way stripe with a 256K stripe size.
4. Avoid a stripe width of greater than 8 on a Symmetrix. Risk of contention goes up and performance gains seem to level off.

As for the answer to the "stripe or not stripe" question, it is analogous to whether loading passengers onto a commercial airline is quicker if loaded row-by-row from back to front versus using the zone approach to load, where passengers with window seats board first, followed by center seats and then lastly the isle seat passengers. The answer is that both are about the same. The only way to speed up this process is by taking carry-on luggage away from boarding passengers. The reason this analogy is fitting to a Symmetrix is that the disk front-ended by an intelligent cache speeds up the performance in much the same way that reducing carry-on luggage does in airports. Less time is spent waiting for something to be placed in or pulled from storage.

Detecting Bottlenecks

Tools for monitoring performance

In general, the most important source of information for monitoring performance is a collection of data from both OS and Symmetrix tools. On the host-system side, the SAR utility and various OS-specific add-ons (like Glance) help determine the average number of I/Os per second and average wait time on a channel, device or physical device. Useful command line options for SAR are -d and -u. The ability to collect this data over a long period of time as well as instantaneously are important criteria for monitoring performance.

On the Symmetrix side, the best tool available to the general user is Symmetrix Manager for Open Systems. This tool is useful for understanding the configuration of a Symmetrix as well as providing performance monitoring features. It has a GUI interface that allows the display of many important I/O statistics simultaneously in a graphical view. Some of these metrics include cache hit rates for read and write, percent of cache in use, and average throughput rates for the Symmetrix. It also allows drill down to look at specific areas that might be a bottleneck. Specifically, it allows viewing I/O rates for a given CA, DA or physical drive.

This kind of information is essential to identify where performance might be constricted or bottlenecked. Equally important is that once a change or tune is done, specific areas need to be checked to confirm that there has been an increase in performance. If this is not the case, determine which other areas might need further diagnosis.

Oracle-Specific Issues

When a DBA has many years of Oracle performance and tuning experience there is a certain amount of conventional wisdom acquired that needs to be rethought once a Symmetrix is configured as part of the hardware solution. This conventional wisdom is the result of working around limitations inherent to standard disk. However, once cache is introduced and intelligent algorithms are used to manage I/O, some of this thinking needs to be revisited, as the reasons for taking certain actions may no longer be relevant.

1. **Use many small devices** to enhance performance. This makes sense if one is limited by single-drive contention or file-system overhead. However, in a cached disk this delay is not a factor.
2. **Separate log files from data and index files.** With nonvolatile cache, fast reads are not going to be limited by how fast log information can be written to disk. If cache is not saturated, the write will be acknowledged as soon as it is received and the normal delay associated with a single spindle will not be an issue.
3. **Stripe because that is what we have always done.** There are some cases where this will make no difference while in other types of work loads, striping may actually degrade performance. There is no universal answer to this issue, but it is worth reviewing why striping was done in the first place, what the impact to the OS is, and what gain can be attributed to it.

The Basic Rules

1. **Start simple —** Treat the EMC Symmetrix at first as if it were any other set of disks. Digest the performance gained there, then begin to add complexity and learn the layout needed to identify the tool set and start further tuning.
2. **More spindles and cache is better —** The ability to get top performance from a system will depend on the ability to avoid contention and to evenly spread throughput. Having enough cache on a Symmetrix is important to prefetch and higher cache hit rates.
3. **Bigger is not always better —** Do not confuse capacity and performance. Often one must be compromised to achieve the other. Doubling the capacity of storage does not speed up the performance. Capacity is measured in GB while performance is measured in I/Os per second.
4. **Look at the "big picture" —** Rarely will any one component of the overall solution be the sole limiting factor when database performance is slow. Learn the lay of the land before getting too focused on where the problem is located. Look at the whole picture and apply efforts across the entire environment.
5. **Trade offs will have to be made —** There is no perfect solution and there is always going to be more than one solution. One has to choose between two things which cannot be had at the same time.
6. **You are only as fast as your slowest link —** Great database tuning will not get around a poorly tuned application - Address tuning from the perspective of all parts of a system, both software and hardware: storage, host system, database, application, networks and workload hours.
7. **Tuning is both an art and a science —** Tuning needs to be done at all levels to ensure overall success. It requires practice and constant learning.
8. **Faster is NEVER fast enough —** Do not become the victim of your own success. Be aware that once the average response time is cut in half, the next goal will be to do it again. Build this law of diminishing returns into your users' expectations
9. **Performance should NEVER be the only consideration. —** Availability, reliability and manageability should be very high on the list of concerns, as are price of ownership (not to be confused with purchase price alone) and scalability (head room to grow).

Appendix 1 - Books and papers on Oracle performance tuning

This list is by no means a complete listing of all books available on this subject. However, it provides a sampling of some which are frequently found in the libraries of many DBAs and mentioned in papers on this subject. It is important to note that there is limited, if any, description of the use of the cached disk array in any of these materials.

Oracle7 for UNIX Performance Tuning Tips (Oracle part number A22535-2)

Oracle for Sun Performance Tuning Tips (Oracle Part Number A22554) 11 page white paper

Oracle & UNIX Tuning by Ahmed Alomari (Prentice Hall PTR, ISBN 0-12-849167-4)

Oracle DBA Handbook by Kevin Loney (Oracle Press, ISBN 0-07-882182-1)

Oracle SQL High-Performance Tuning by Guy Harrison

Data Warehouse Performance Management Techniques by Andrew Holdsworth
(Oracle Services, Advanced Technologies 2/96)