

## RESILIENT LOW-COST STORAGE

*Juan Loaiza, Oracle*  
*Sue Lee, Oracle*

### **ABSTRACT**

The advent of low-cost ATA disk based storage arrays and low-cost storage networks together with the introduction of Oracle Database 10g has made it possible to create a Database Storage Grid that has very low cost and excellent performance and availability. This paper describes the theoretical and practical aspects of building a Database Storage Grid. It also discusses the databases and environments that are ideal candidates for low-cost storage.

### **INTRODUCTION**

The traditional Oracle database configuration consists of a monolithic server connected to a monolithic storage array. In recent years, Oracle customers have steadily replaced their monolithic SMP servers with clusters of low-cost servers running a low-cost operating system, Linux. The catalyst for this trend is Oracle's Real Application Cluster technology, which enables a database to be implemented on a cluster of servers with scalable performance, high availability, and a much lower cost.

Typically, the costliest component after the host servers is the storage. In many cases, a costly monolithic storage server can be replaced with a grid of low-cost storage arrays in a very similar fashion to what is being done with server grids. A low-cost storage grid must provide performance, management functionality, and high-availability that is comparable to today's prevailing storage solutions to be considered for production use. It must do this at a lower price to compel users to reconsider their approach to storage.

We begin by describing low-cost storage arrays and the Database Storage Grid. We explain how Oracle implements the functionality needed to create a storage grid. We then discuss the minimum features that a low-cost storage array must have to be part of a Database Storage Grid. Low-cost storage has both benefits and limitations; we discuss the databases and functions for which low-cost storage can be most successfully used. We describe the performance characteristics of low-cost storage, based on our own extensive evaluations and benchmarks. We then describe how to successfully deploy low-cost storage, given your availability, performance, capacity, and cost goals. We conclude by describing the Oracle Resilient Low-Cost Storage Initiative.

### **DATABASE STORAGE GRID**

The basic building block for a low-cost Database Storage Grid is a simple, modular, storage array. It consists of 1 or 2 controllers that manage a single shelf of 12 to 16 ATA disks. The controllers connect the storage array to a SAN or LAN network via 1 to 4 Fibre Channel, iSCSI, or NAS connections.

A grid of database servers accesses a grid of low-cost storage arrays via SAN or LAN switches, as shown in Figure 1. Each low-cost storage array can be shared by multiple databases. We call this grid of low-cost storage arrays a Database Storage Grid.

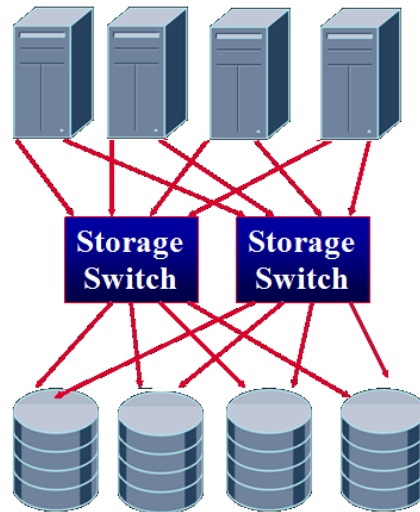


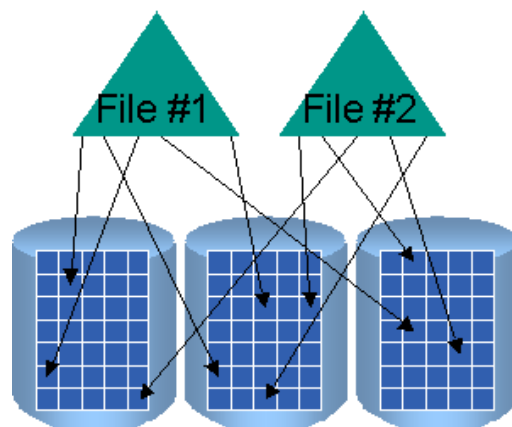
Figure 1 Low-Cost Storage Grid

A Database Storage Grid has the following characteristics:

- Access Transparency. A database server cannot tell how many storage arrays are in the grid. In fact, it is not aware that its data is stored on a grid rather than a monolithic storage array.
- Automatic Load Balancing. The data from each database is evenly distributed across the storage arrays. Data accesses are also evenly distributed; there are no hot spots.
- Unlimited Transparent Scalability. Storage arrays and disks can be easily added or removed. When the storage configuration changes, data is automatically rebalanced for uniform storage and access distribution.
- Data Protection. Data is protected against disk failures, storage array failures, disasters, and human errors.

### USING AUTOMATIC STORAGE MANAGEMENT

Automatic Storage Management (ASM), introduced in Oracle 10g, is a key component of the Database Storage Grid. ASM provides a simple storage management interface that is consistent across all server and storage platforms. The administrator specifies the disks within the Database Storage Grid that ASM should manage. ASM partitions the disk space into 1 megabyte storage units and combines these units to form database files. For each database file, ASM chooses storage units from all disks across multiple storage arrays to ensure that data storage and load is evenly distributed. Thus, as shown in Figure 2, each disk and storage array contains the data of multiple database files and each database file is stored on multiple disks and storage arrays.



## Figure 2 ASM File Striping

With ASM, storage arrays can be easily added to or removed from the Database Storage Grid. Unlike traditional RAID-0 striping techniques, ASM does not distribute data using a rigid formula that requires data to be re-striped after each storage configuration change. Instead, a file's storage units are tracked using database oriented indexing techniques. This indexing allows disk and storage array changes to be made easily and efficiently with no database downtime. To ensure uniform data storage and access distribution, ASM automatically rebalances files as the database is running after storage is added or removed from the grid.

Protection against hardware failures is a key feature of ASM. An administrator specifies disks that fail together because they share components such as a controller or network connection by placing them into a common Failure Group. In a Database Storage Grid, a failure group is typically a set of disks within a storage array. For storage arrays with multiple controllers where each disk is accessed through only one controller, a failure group consists of disks served by a single controller.

ASM protects data from storage array and disk failures by mirroring data across Failure Groups. When a failure occurs, ASM re-mirrors the data across alternate Failure Groups. Failures are therefore transparent to the database. In addition, the window of vulnerability from a second failure is small because a new mirror is created immediately and automatically after the failure.

### USING DATA GUARD, FLASHBACK, AND RMAN

Oracle also protects the data against other types of failures. Data Guard protects against disasters and data corruption by automatically maintaining a second copy of the database. Flashback protects against human errors. Flashback provides snapshot capabilities, allowing an administrator to rewind a table or the entire database to a specific moment in time before the error occurred. RMAN provides full and incremental backup to disk or tape for archival and protection against corruptions. All of these Oracle features provide the same functionality on a Database Storage Grid as on traditional monolithic storage. Because this advanced functionality is implemented in the database and not in the storage arrays, it works transparently across the storage arrays in the Database Storage Grid. Advanced functionality is not limited to the data that fits within a single array, as is normally the case with traditional storage configurations.

### HIGH AVAILABILITY ON LOW-COST STORAGE

Some of the functionality provided by ASM, Data Guard, Flashback, and RMAN is available in monolithic storage arrays. Supporting this functionality is one factor that drives up the complexity and cost of monolithic storage arrays. Low-cost storage arrays, on the other hand, do not typically support this functionality, especially across storage arrays.

Oracle's storage management functions are the key to achieving high availability on low-cost components. Low-cost storage arrays are reliable, but not as reliable as monolithic storage arrays. However, Oracle's management capabilities make a grid of low-cost storage arrays extremely reliable. The grid can be constructed such that it has no single points of failure. Failures or bugs on a storage array are contained to the storage array, preventing the problem from affecting other arrays in the Storage Grid. A Database Storage Grid does not depend on flawless execution from its component storage arrays. Instead, it is designed to tolerate the failure of individual storage arrays.

### REQUIREMENTS FOR LOW-COST STORAGE ARRAYS

A low-cost storage array must implement a core set of features in order to be used in a Database Storage Grid.

#### NETWORKED STORAGE

Networked storage enables high performance and availability in a grid configuration. Therefore, the storage array must be attached to a network via Fibre Channel, iSCSI, or as a NAS device, allowing it to be accessed by multiple database servers or multiple instances from a single database server.

#### INTER-OPERABILITY CERTIFICATION

Since many customers will implement the Database Storage Grid into an existing SAN or LAN network, it's important that the storage array work with existing HBAs, vendors, and operating systems. Even though a storage array may be built to Fibre Channel specifications, our experience has shown that it may not be compatible with other Fibre

Channel conforming switches, HBAs, or drivers. Therefore, it is important that the storage array be officially certified against the switch, HBA, host operating system, and driver that you are planning on using.

### **REMOTE MANAGEMENT AND FAILURE ALERTS**

While Oracle will protect the data from storage array or disk failures, the administrator must be promptly informed of the failure to avoid hitting the window of vulnerability when mirroring is no longer possible because the number of total failures is too large.

The storage array must automatically alert the administrator about any hardware failures. It must allow failure-prone components such as disks, fans, and power supplies to be hot-swapped. In addition, since the administrator is responsible for managing a grid of storage arrays, it must provide remote management and monitoring tools.

### **HIGH AVAILABILITY**

It is important to note that because Oracle can mirror (or even triple or quadruple mirror) the data, that extreme availability of an individual storage array is beneficial, but not necessary. Instead, the administrator should consider availability of the Database Storage Grid as a whole, focusing on failures that effect large number of hosts or storage arrays, such as connectivity failures.

For instance, the Storage Grid should be able to tolerate a switch failure. If storage arrays are only connected to one switch, then a failure of that switch will result in failure to access those storage arrays. Similarly, if the host is connected by only one HBA or to only one switch, then it will be unable to access the Storage Grid in the event of a failure of its HBA or switch. Therefore, for very high availability, the Storage Grid should be implemented with redundant switches and redundant host HBAs.

### **AVAILABILITY FROM A HIGHER PERSPECTIVE**

At a certain point, investing in extreme high availability of the Database Storage Grid may be less strategic than investing in alternate approaches to failure protection. For example, triple storage redundancy is often not cost effective. Standby databases, implemented by Data Guard, can provide increased protection against failures that are typically more common than double component failures. Standby databases, in addition to protecting against site and Storage Grid outages, protect against human errors, data corruptions, and software defects.

If you don't want to invest in a standby database, consider triple mirroring the database logs and control file, and placing your archive logs on an independent storage subsystem. These files are relatively small in terms of storage space. By triple mirroring these files you can ensure that your database can be recovered from a double disk failure with no data loss.

### **AN EXAMPLE: APPLE XSERVE RAID**

One example of a low-cost storage array that fits the requirements described above is Apple's Xserve RAID. This storage array contains 2 independent controllers with a battery-backed cache. Each controller manages up to 7 250 GB or 400 GB ATA disks and is connected to the SAN with a 2 Gbit Fibre Channel connection. The storage array can be monitored and managed remotely and alerts the administrator of failures by email or pager. While it contains redundant cooling and power supplies, a controller does not fail-over to the other in the event of a controller or network failure. The Xserve RAID supports hosts running Linux, Windows, and MacOS.

The Apple Xserve RAID's list price for 5.6 TB is \$12,999, or \$2.32 per GB.

Its performance is excellent. For a workload of 8KB random reads, it can sustain 1100 I/Os per second (IOPS). For a multi-user sequential workload, it can sustain 266 MBps of read throughput.

Other examples of low-cost storage arrays include:

- EMC/Dell AX100
- Engenio Vail 2822, resold as IBM DS4100 (formerly FAS*t*T100) and StorageTek Blade Controller 210
- Hewlett-Packard MSA1500
- NetApp FAS250

## **TRADE-OFFS**

Before purchasing and deploying a Database Storage Grid, it is important to understand how low-cost storage arrays compare to traditional storage arrays.

### **ATA VS FIBRE CHANNEL DISKS**

A key component of low-cost storage is the disk. Most traditional storage arrays use Fibre Channel or SCSI disks. Low-cost storage, on the other hand, uses ATA or serial ATA (SATA) disks. ATA disks lack command queuing, resulting in lower performance for random I/Os. However, their sequential I/O performance is very good and hence is what ATA disks are typically promoted for.

From a cost perspective, ATA disks are much lower in cost than Fibre disks. From a cost per gigabyte perspective, their cost difference is even more pronounced; ATA disks have excellent areal density, allowing them to provide capacity at a much more reasonable price. In some configurations, the principal motivator for determining the number of disks is performance rather than capacity. In these cases, the excess capacity can be used to improve overall availability by creating extra mirrors of data or an on-disk backup.

The primary concern for most ATA disk users is its reliability. The mean-time-between-failures (MTBF) is lower for ATA disks than Fibre disks: 500K hours (58 years) versus 1.5M hours. The MTBF of an ATA disk may be even lower if it is used continuously at a high rate. Note, however, that most database storage is configured for a maximum I/O rate, so this very high continuous utilization doesn't typically occur.

Consequently, a key issue for low-cost storage is compensating for higher MTBF rates. As discussed previously, mirroring hides disk failures. To deal with the higher failure rates, you will need to budget for more replacement disks when using ATA disks. Even after adding the price of some extra disks, the overall storage price is still typically lower than for Fibre Channel disks since the ATA MTBF of 58 years is still very good.

### **OTHER COST CONSIDERATIONS**

There are several other costs to consider when evaluating the cost of Database Storage Grid as a whole. Redundant network ports and switches may add to the overall cost. Because storage is implemented as a grid of storage arrays, more switch ports may also need to be purchased. The warranty period of ATA disks is typically lower than Fibre disks; hence an extended warranty may need to be added. You can often save money on support for a Storage Grid since the failure of a single array is masked by the database and thus will not bring down the overall grid. Therefore, lower cost 5x8 support might be purchased instead of 24x7 support. Finally, the number of disk trays affects the total cost because each disk tray consumes additional rack space and power. ATA disk based arrays require fewer trays for capacity intensive applications. Fibre disk based arrays require fewer trays for random I/O intensive applications.

## **PERFORMANCE**

The performance of any storage array is very dependent on its workload.

### **RANDOM I/O WORKLOAD**

A random I/O workload consists of read and write I/Os that are the size of the database disk block (typically 8KB) and are randomly distributed across the entire storage array. Because the database buffer cache absorbs most of the locality in the access pattern of the application, the storage array cache's hit rate is extremely low. Therefore, this workload primarily reflects the performance of the disks themselves.

A Fibre disk offers about twice the random read I/O performance as an ATA disk. A 7200 rpm Fibre disk can sustain about 150 IOPS whereas an ATA disk of the same speed can sustain about 80 IOPS. 150 IOPS with 8KB I/Os results in a total transfer rate of 1.2 MBps. Consequently, for this workload, the aggregate performance of a storage array is not channel bound and is equal to the sum of its disks' random I/O performance.

The random write performance, in theory, is slightly lower than the random read performance. In practice, however, it is much more dependent on the implementation of the storage array. A well-implemented controller with a battery-backed cache can lazily issue writes to the disks in order, resulting in better performance because the time spent seeking is minimized. In fact, for a low-cost storage array implementing a disk elevator algorithm, the write performance can be almost 4 times faster than its read performance.

Even though the per disk performance of an ATA array is lower than a Fibre disk array's, the price performance may be comparable. The same performance can be achieved by doubling the number of ATA disks, but because ATA disks are much cheaper, the resulting price is similar. For example, the Apple Xserve RAID costs \$11.90 per IOPS. A mid-range Fibre-based array consisting of one head and 105 disks costs \$11.75 per IOPS. Furthermore, note that buying twice as many ATA disks will result in four or more times as much storage space because of the high density of ATA disks.

### STORAGE ARRAY CACHING

Most low-cost storage array controllers have a non-volatile cache. The cache is critical if the controller is protecting the data with RAID-5. In all other cases, its benefit for reads is marginal because most access locality is captured by the database's buffer cache.

On the other hand, caching can help the performance of writes a lot, particularly if the controller implements a disk elevator algorithm for flushing the data, as discussed previously. If a low-cost storage array contains multiple controllers, each with their own write cache, then by default, all write data is typically mirrored across the caches so that in the event of a controller failure, another controller can flush the cached write data to disk. The mirroring of the write cache often greatly impairs performance. In our evaluation, the overhead of write cache mirroring degraded write performance by 50% on some storage arrays.

For some low-cost storage arrays, write cache mirroring can be disabled. The database can recover from the loss of a controller with dirty cached data by using ASM to mirror across storage arrays and forcing all disks to be mapped to a single controller. When ASM perceives a controller failure, it labels the associated disks as failed and only uses the data from the mirror. Therefore cache mirroring can be safely turned off in a Database Storage Grid in many situations. This can improve performance significantly.

One benefit of a cache that vendors frequently tout is that it allows the controller to prefetch. In practice, all of the database's sequential accesses use large 1 MB I/Os. In addition, in situations where prefetching would improve performance, the database performs the prefetching itself. Therefore, the value of the prefetching performed by the controller is questionable and, in heavy mixed workloads, may even hurt performance. It is most useful when large I/O operations issued by Oracle are fractured into small I/Os by the operating system or driver.

### MULTI-USER SEQUENTIAL WORKLOAD

Data warehousing, backup, and reporting applications are often limited by the throughput of sequential 1 MB read and write I/Os. The vast majority of the time, the sequential data stream is accessing the disks concurrently with other database traffic, either from the same or another database. Even during backup, RMAN is typically invoked with multiple sequential data streams to speed the backup process. Furthermore, striping spreads streams that are sequential at the application level across many disks. Therefore, consecutive I/O operations from the same stream at the disk level are widely spread apart in time. This gives a lot of opportunity for another stream or another user to move the disk head between sequential accesses by the same stream. Consequently, at the disk level, application level sequential data streams are usually seen as random 1 MB I/Os and any prefetching support in the controller will not necessarily be triggered. Therefore, a more accurate representation of a storage array's ability to handle sequential data streams can be obtained from a random 1 MB I/O workload. We call this a multi-user sequential workload.

ATA and Fibre disks have comparable multi-user sequential performance: around 30 MBps. A 2Gb Fibre Channel can sustain about 180 MBps, or the multi-user sequential throughput of 6 disks. Therefore, most storage arrays are channel limited rather than disk limited. For optimal performance for multi-user sequential applications, the Database Storage Grid should be configured with as many active network connections as possible.

In practice, the multi-user sequential performance is also heavily dependent on the host server and the storage array implementation. Good multi-user sequential performance requires that large I/Os be issued to the disk. The database issues 1 MB I/Os, but the host operating system, the host device driver, or the storage array may break it into multiple smaller I/Os. I/O fracturing has a considerable impact on performance, sometimes leading to a 90% performance degradation. Even on mid-range Fibre-based storage arrays, we have observed total throughput as low as 30 MBps because of fracturing.

On the host side, a default Linux configuration breaks a large I/O into multiple 32KB I/Os. A tuned Linux kernel

will ask the driver on the HBA the maximum I/O size it can support and only fracture the I/O if necessary. The driver itself may unnecessarily fracture the I/O by claiming it can only handle small I/Os. This is often the case for vendors who offer both low-end and high-end HBAs, but provide a single Linux driver that supports the smallest I/O size across all of its HBAs.

Therefore, for optimal performance, the storage system should be tuned to avoid I/O fracturing, or be capable of handling at least 128KB I/Os. In general, most administrators pay a lot of attention to file system disk fragmentation. In a database environment, I/O fracturing is at least as important.

Even after the host is tuned to emit 128K or larger I/Os, the throughput of a storage array is extremely vendor dependent. Our evaluations of many ATA and Fibre based arrays have shown that in most cases, while the storage array benefits from more network connections for multi-user sequential workloads, it cannot saturate its network connections.

While the sequential I/O performance is comparable on ATA and Fibre disks, the cost of the throughput is much lower on ATA-based storage. Also, when using a grid of single shelf ATA arrays, more controllers are added as more disks are added. In configurations in which there is one head and a set of chained disk trays, the head and the chained connections can become a bottleneck to throughput. Throughput of multi-user sequential workloads is particularly crucial in Data Warehouses.

## **ORION**

The performance numbers in the previous sections were obtained with a tool called Orion: Oracle I/O Calibration Tool. It measures Oracle performance without having to install Oracle or create a database. It issues I/Os against raw disks using the Oracle database's I/O libraries. It can simulate the I/O patterns of random I/O, multi-user sequential, and other common Oracle workloads. In addition, it simulates the striping performed by ASM.

While evaluating the performance of low-cost storage arrays from multiple major storage vendors, we observed that their performance is very dependent on the host server's operating system and HBA as well as the storage array implementation. The storage array settings, such as enabling / disabling write cache mirroring, cache line size, prefetching parameters, and disk write size, also made a noticeable impact.

Therefore, it is extremely important to evaluate a storage array based on its anticipated workload and tune it for optimal performance. Orion is an easy-to-use tool that outputs a broad characterization of a storage array's capabilities.

## **USES OF LOW-COST STORAGE**

### **FLASH RECOVERY AREA**

An ideal initial usage for low-cost storage is Oracle's Flash Recovery Area. The Flash Recovery Area specifies the on-line disk storage that is used by Oracle to store all recovery-related files, including full backups, incremental backups, and backups of various database logs. The Flash Recovery Area can be configured to use a low-cost storage system while the main database area remains on a storage system that supports very high I/O rates, as depicted in Figure 3. The performance of backups and restores is considerably improved by using disk rather than tape. In addition, by using RMAN's nightly incremental backup feature where changed blocks are tracked by the database and applied to the previous backup, backups can be made even faster. Because low-cost storage is cheap and the Flash Recovery Area is predominately accessed with sequential I/O streams, it is an ideal use of low-cost storage.

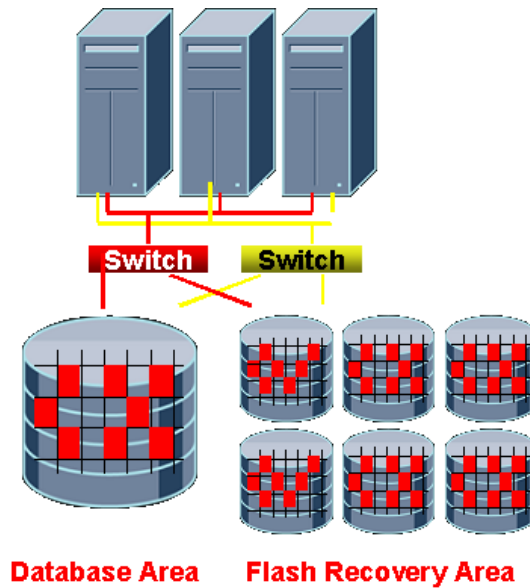


Figure 3 Flash Recovery Area on Low-Cost Storage

### SUPPORTING DATABASES

In an enterprise environment, a production OLTP database is typically surrounded by many supporting databases: data warehouses and reporting databases, development and testing databases, and standby databases. See Figure 4. They consume the bulk of the storage and often have lower requirements for random I/O performance. They, as well as lightly loaded OLTP databases, are ideal candidates for low-cost storage.

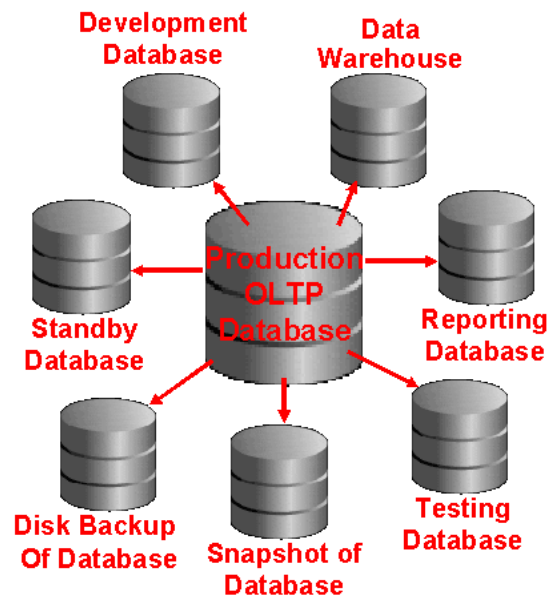


Figure 4 Types of Enterprise Databases

### ILM

A popular theme for storage is Information Lifecycle Management (ILM). For most applications, as the data ages, activity on the data declines and the total volume of the data grows. In other words, most applications typically have small amounts of active data and large amounts of less active data. During the data's lifetime, data can be partitioned

amongst multiple tiers of storage; active data can be stored on high-performance storage, while less active or historical data can be stored on cheaper, lower-performance storage.

Oracle can partition the data amongst multiple tiers of storage using table partitioning where each storage tier is assigned the table partitions corresponding to a specific range of dates. As shown in Figure 5, recent data can be assigned to a storage tier implemented on high performance storage while older data can be assigned to a storage tier implemented by low-cost storage. Oracle is an ideal platform for ILM because it can partition the data in a way that is completely transparent to the application while taking advantage of low-cost storage arrays.

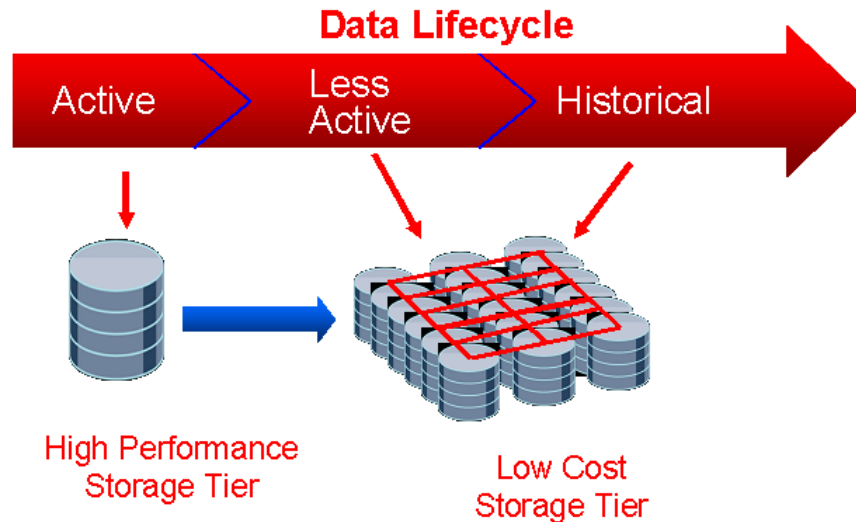


Figure 5 Information Lifecycle Management on Low-Cost Storage

## EXAMPLES OF PRODUCTION DEPLOYMENTS

### **ORACLE IT**

Low-cost storage has been successfully deployed within Oracle for the Oracle Collaboration Suite application for email, voicemail, and calendar. In the original configuration, a traditional Fibre-based array was used for both the database and Flash Recovery areas for a deployment that supported 1000 users. A new configuration was required to support an additional 3000 users. Oracle maintained the data on the Fibre-based array, but implemented the Flash Recovery Area on a grid of Apple Xserve RAID arrays connected using QLogic SAN switches.

The per mega-byte cost of this low-cost storage grid is about three times lower than that of the Fibre-based array that was originally considered. The time required for backing up the database onto the Flash Recovery Area has remained the same and the low-cost storage grid has been stable and easy to administer. This low-cost storage implementation has been so successful, both in terms of manageability and performance, that Oracle is rapidly expanding its plans for deploying low-cost storage across multiple divisions and types of applications.

### **AMAZON.COM**

Amazon.com is an example of a leading corporation that has adopted low-cost storage. It has implemented a 23 TB data warehouse on a 16 node Linux RAC cluster connected to 64 Hewlett-Packard MSA1000 storage arrays by 8 SAN switches. It can deliver over 2 Gigabytes per second of table scan throughput and is substantially less expensive than their previous storage configuration. The Amazon.com data warehouse is listed as one of the ten largest warehouses in the world in the Winter Survey. It demonstrates that low-cost storage can be used successfully in very demanding environments.

## RESILIENT LOW-COST STORAGE INITIATIVE

To help customers successfully deploy low-cost storage, Oracle has launched the Resilient Low-Cost Storage Initiative. This initiative is analogous to the existing Oracle Storage Compatibility Program (OSCP) but with a focus on low-cost

storage. Its goal is to ensure that low-cost storage can be easily managed, is very reliable, and performs well.

This initiative partners Oracle with storage vendors to validate that their low-cost storage arrays meet the minimum price, functionality, and performance requirements for use in a Database Storage Grid. Vendors use Orion to characterize their storage array performance. After verifying that all requirements can be satisfied, the storage array is certified for this initiative and the vendor together with Oracle write a Best Practice white paper to instruct Oracle customers how to optimally configure the storage array for a Database Storage Grid environment.

### **RESILIENT LOW-COST STORAGE WEB SITE**

The web site for the Resilient Low-Cost Storage Initiative is

<http://www.oracle.com/technology/deploy/availability/htdocs/lowcoststorage.html>. It contains valuable information, including:

- How to configure Linux and the host HBA driver to avoid I/O fragmentation.
- General “best practice” white paper on implementing a low-cost storage grid.
- List of certified low-cost storage arrays.
- Vendor “best practice” white papers on configuring and tuning the storage array.

### **CONCLUSION**

Low-cost storage arrays are an important building block for creating a flexible and cost-effective storage solution for Oracle databases. They have proven to be an ideal candidate for data warehouses, low volume databases, and implementing a Flash Recovery Area. With the storage configuration information and best practice white papers available on the Low-Cost Storage web site, low-cost storage can provide excellent performance and availability at an outstanding price.

### **ACKNOWLEDGEMENTS**

Our analysis would not be possible without help from various storage vendors. In particular, we'd like to thank Alex Grossman, Bill Lloyd, Chuck McClelland, and Matt Sturges from Apple; Richard Vanderbilt and Mark Register from Engenio; Baila Ndiaye and Kevin Lernihan from Hewlett-Packard; Bob Ng, Kathy Sharp, and Stu Champion from EMC; Vu Pham, Kevin Deierling, and Thad Omura from Mellanox; and Scott Kirby, Ryan Klein, and Lynn Lehan from QLogic for being so generous with both their time and insights.

Many thanks to everyone at Oracle who worked very hard to setup, tune, and evaluate the storage arrays, particularly Andrew Babb, Rahim Mau, Sally Piao, Lawrence To, and Doug Utzig. Thanks to Joel Becker, Wim Coekaerts, and Margaret Susairaj for helping us improve performance on Linux. Orion is the creation of Sridhar Subramaniam, Jonathan Giloni, and Margaret Susairaj; thanks for creating this indispensable tool. And many thanks to Paul Tsien for launching the Resilient Low-Cost Storage Initiative.