Another (Off) White Paper on High Availability Storage Options and Their Impact on Performance on the HP e3000

By Walter McCullough

Version Date January 20, 2004

This is an update to a white paper I wrote back in 1997 when we were receiving a number of questions regarding the introduction of storage arrays and their use on MPE systems. It seems we continue to have some confusion regarding this subject despite the passage of time and improvement in the storage technology.

My goal for this (short) paper is to lay out a somewhat high level overview of what makes MPE different from other operating systems and importantly why MPE goes to disk to satisfying an I/O request. Then I would like to describe the available HA storage solutions for your HP e3000 and just a little behind the curtain view on how that new technology works with MPE.

You can view the older paper at:
http://jazz.external.hp.com/mpeha/papers/off_white_paper.html

Architecture

The current (and last) incarnation of MPE and its lowest (machine dependent) layer was specifically designed for the PA-RISC architecture. This thin layer allowed the MPE lab to create an operating system that had very little shielding from the hardware layer. While the HP-UX approach was to create a (thicker) layer which allowed for greater hardware independence, MPE's approach allowed operations to move more expeditiously through the computer, thus giving it the ability to do more (and generate more I/O).

The disadvantages (for the HP e3000) of this thinner machine dependent hardware layer or shielding is that MPE is limited to the PA-RISC architecture while HP-UX's thicker shielding allowed it much more freedom and flexibility to move to newer faster computer architectures.

MPE I/O Behavior

First of all MPE/iX is different from HP-UX and Unix (like) systems on how it relates to file access and how it utilizes the combination of disk storage and server memory. While Unix (like) systems are usually referred to as using a paging system to access files and data to and from a swap area, the MPE operating system uses the file's disk location as the file swap area and only needs to allocate process/user state information/data which is "swapped" out to areas of the system volume set. This technique greatly reduces the number of times that the operating system needs to go to disk to satisfy an application’s I/O requirements.

To further reduce the need to go to disk, MPE employs a cache as part of its design. This design uses available main memory to keep active parts of the file accessible by an approach that is similar to direct memory access. As active applications require access to their data those requests are satisfied through MPE's memory cache and are handled by the MPE/iX memory manager facility.

The next technique employed by MPE/iX to increase performance is that of reducing the need to post data to disk. MPE's journaling file system is called the Transaction Manager (XM for short). This facility copies just those bytes that change in a file, because of a file record update, and copies them to XM's facility. *(This XM protection is afforded to critical MPE/iX structures and some file types like KSAM and Image data bases.)* This reduces the need to post files/data to disk to ensure data durability and this posting action can significantly impact the overall system and application performance.

The last couple of techniques I want to mention is the way MPE stripes data files across multiple disk members within a volume set and how it sends multiple I/O request for any given Ldev (volume) at the same time. This helps reduce the impact of disk seek and latency times, part of the normal performance impact when dealing with disk drives. MPE bundles a number of I/O requests (8 per Ldev) for most types disks and arrays and sends them asynchronously. This lets the operating system return to other tasks while those operations complete.

How is performance perceived

The number one question I usually get is "how fast will this HP e3000 server run with that storage array?". The answer is, "It depends"! Without any performance data or an understanding of the current (or future) application's I/O characteristics, that question really can't be answered except very conservatively.

Using a car story analogy; A customer goes to a car dealer looking for a car that has a 400 horse power engine and "price isn't an option" says the customer. The dealer sells the customer a real nice sports car but is dismayed when two days later the car is returned dirty, smoking and badly wrecked. The customer wants his money back because, it seems, the car was unable to navigate off-road, down hill and with a large boat attached (which explains the dents on the rear and roof of the car). <Enough of cars now>

Without understanding what the needs are and understanding how the technology works together, we are all destined (in our small way) to repeat this story.

Understanding the Technology

The benchmark for storage technology is set by the performance of JBOD (Just a Bunch Of Disks). Remember too, that modern storage arrays are built on top of JBOD but use a memory (cache) and a small operating system to add features that ultimately allow for the added protection of data and in some cases improve ease of management and performance.

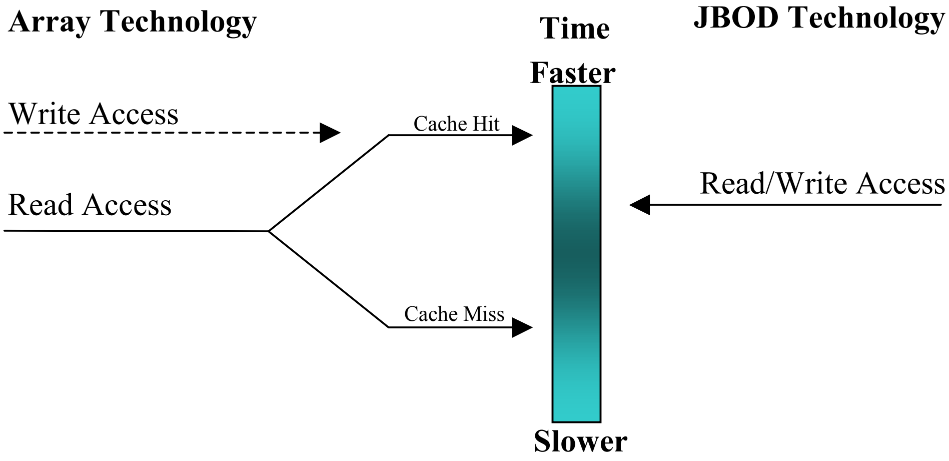
The added value that arrays provide over that of JBOD is that of data protection. It does this by laying the data across a number of disk drives along with data recovery information that can be used to reconstruct the data in the event that a drive mechanism fails. The vender may have created a way for the reconstruction to happen on the fly or as a background activity. The main point to remember is that whatever technology is in place, the goal is to protect the data.

Storage Performance Features

One of the biggest features that all the venders tout as their answer to performance improvements is the use of large amounts of memory in the array used as a cache. Some even give the impression that this cache will improve performance over JBOD and that arrays are faster because of the cache. Not always so.

The right side of the picture below shows an arrow that indicates the (relative) time it takes to do either a read or write to disk. The time is based on a combination of events that culminate in I/O response time, such as bus transfer speed, disk seek, settle and latency times. The value of this response time is usually in milliseconds (which I will not go into because I'll have to change it every time a new disk hits the market). Just use the placement of the JBOD Read/Write as a relative reference point to compare that of the storage array's Read/Write response times.

When we factor in the use of a storage array cache we see that when a Read is issued to the array and the information is already in the cache (Cache Hit), it can respond back with the data faster than a Cache Miss, which needs to wait until the data is found on the disk (seek, settle and latency) and then transferred to the Cache. As you see in the picture, a Cache Hit is faster than a JBOD Read/Write and a Cache Miss is slower than a JBOD Read/Write Access but much slower than the array Cache Hit.



Now this same picture will look a little different on the left (array) side if we look what happens to a Write Access. In this case, depending on available array cache, the Write Access can complete faster than the right side (JBOD) because the array has the ability to store the Write data in the array's cache, then inform the server that the Write has completed and unbeknownst to the server, transfer the data from the array cache to disk later. This only works if there is room in the array cache. If the array is getting more data than it can transfer to disk then I/Os will queue up and wait for room in the cache. This then slows the I/O completion down just like a cache miss.

Early enterprise array technology (1995-2000) had response times for cache hits only marginally faster than JBOD but cache misses were many many times slower than JBOD. With this scenario the only way to achieve decent performance numbers was that MPE's cache was large enough to reduce the need to do I/O to the array or that the array either had a better algorithm to anticipate MPE's needs (highly unlikely) or a large enough cache to allow MPE's I/O needs to be satisfied with a great proportion of cache hits.

MPE Performance Environment

Lets spend a little time painting the picture of an environment where MPE was best designed to give optimum performance. We will break this part up into a number of categories: memory, connection technology and Ldev (spindle count/size). The order of the categories also indicates its impact on performance, where the fastest access is memory and the slowest is disk. We will then go through the list of available array products for the HP e3000.

Memory

The greater the amount of memory for the HP e3000 the more efficient it will run (to a point). MPE uses available (server) memory as a cache to store user process information and also keeps a virtual copy of the file. If there is enough memory to cache the user process information and data then MPE does not have to wait for disk access when making changes to the file until either the user makes an explicit call to post-and-wait or lack of memory causes MPE to push that data from memory to make room for other processes that need to execute. The amount of memory needed depends on an understanding of the application's needs, the number of users and the amount of available disk space. This information can be supplied by the application's vender or with the help of an MPE performance specialist/consultant.

Connection Technology

The HP e3000 server comes in two flavors, an older NIO based machine, such as the 9x7, 9x8 and 9x9 and a newer PCI based machine such as the A-Class and N-Class machines. The NIO based machines are limited to HVD-SCSI (High Voltage Differential) but can use the A5814 Router to connect to newer HP fibre channel storage arrays.

Without playing too much with numbers, an NIO based machine can usually pump out about 12Mbytes per second of data through the backplane and HVD-SCSI HBA (Host Based Adapter) card while the theoretical limit of this SCSI specification is 20Mbytes. This sounds like a definite bottleneck compared to fibre channel speeds but remember, performance is based on all the components (memory, connection technology and disk) working at best case and disks are still a mechanical device limited to mechanical speeds.

Another connection technology is the HVD-SCSI to Fibre Channel Fabric Router. This device connects to an MPE (or HP-UX) HVD-SCSI HBA and converts SCSI to fibre channel (1Gbit) for connection to the newer HP fibre channel portfolio of storage arrays. The router solution has been thoroughly tested and has not been shown to increase any performance overhead by its presence as a connection solution.

The last connection technology for the HP e3000 is that of native fibre channel. This technology is only available on PCI based servers, usually referred to as the A-Class and N-Class, running MPE/iX version 7.5.

This fibre channel technology is rated at 2Gbits per second (theoretical limit). This translates down to about an effective rate of 200Mbytes per second but let me caution you right now. This speed limit or bandwidth is no guarantee that the disks can keep up with this limit. Using a car story analogy (again), while you can go 65mph on the freeway you still have to pull into your driveway and wait for the garage door to open before you can enter.

Ldev Count & Spindle Size

The term Ldev in this paper referrers to an MPE configured disk or volume.
The following is a list of features that MPE/iX provides which will help clarify the relationship between the number of Ldevs and their size to the impact on performance:

Extent Striping
Given a single large file, MPE/iX was designed to break up that file into extents (chunks) and then spread each of those extents across the different disks within its volume set (volume group) which we will refer to as "extent striping". This striping was shown to improve file access the HP e3000 by spreading the I/O load across multiple disks much the same way as the newer array technology does by spreading its I/O by the use of RAID-0 features.

Asynchronous I/O
MPE/iX is designed to initiate up to 8 individual I/O requests per Ldev (disk or volume).

XM Facility
XM supports multiple instances of itself for each volume set. This means that XM single threading is greatly reduced when the HP e3000 is configured to have a number of user volume sets.

Given the feature set listed above let's look at the advantages of using 8 separate 9Gbyte Ldevs instead of using a single 72GByte drive.

Extent Striping
With a number of Ldevs MPE will utilize its extent striping feature to improve performance. This allows MPE to spread the access across multiple I/O paths and drives.

Asynchronous I/O
MPE/iX is designed to initiate up to 8 individual I/O requests per Ldev (disk or volume). One big drive (72GByte) will be limited to only 8 asynchronous I/O requests at any given time while that same disk capacity (72Gbytes) divided 8 ways allows MPE/iX to issue a total of 64 asynchronous I/O request for that given capacity.

XM Facility
The best way to take advantage of XM's support of multiple instances of itself for each volume set is by dividing the 8 Ldevs into two or more volume sets. This will, depending on the application's I/O characteristics, also reduce single threading on XM resources (like the XM recovery log file).

High Availability Storage

Lets begin our description of storage by limiting our storage products to only those that are available, purchasable and supported by HP and that work on the HP e3000. I will start off with describing entry level storage solutions and go through the top of the line enterprise storage arrays. A good single point of information of arrays and HA features available for the HP e3000 can be found at http://jazz.external.hp.com/mpeha/ and in the left navigation window click on the ha & storage matrix button.

MPE Mirrored Disk/iX

Though mirrored disk is not an array it is the lowest level entry into high availability data protection. This technology is based on (SCSI) JBOD and relies on having a single piece of data (file) written to two different disks that are on two separate paths. This method protects the data from any single point of failure along the I/O path. The greatest advantage of this technology is its small cost to implement.

The downside of this technology is that it can disable itself under high I/O loads which leave the user unprotected. These false failures can also lead to long recovery times which also leave the user unprotected. For this and other reasons the MPE/iX R&D Lab does not recommend this solution for environments where there is a great deal of I/O activity or where the disk capacity exceeds 50Gbytes of protected storage.

VA7xxx Storage Arrays

These are HP's entry and mid-range fibre channel high availability storage arrays. The VA7100 (now discontinued) is a dual controller (active/passive) 1Gbit fibre channel array that is only slightly faster than the 12H AutoRAID but has much more features and is easier to manage.

All VA7xxx storage arrays must be ordered with dual controllers for MPE/iX and MPE/iX requires that a VA manage software package called the CommandView SDM, to configure and monitor the VA, be on a separate PC workstation or HP-UX server attached to the VA storage array.

For older NIO based HP e3000s the A5814A (option #003) Fabric Router can be used to connect the fibre channel arrays to the older SCSI based NIO HP e3000s. *(See the ha & storage matrix for support information.)*

The VA7110 and VA7410 are HP's newer entry and mid-range virtual arrays. Both these arrays support 2Gbit redundant controllers. The VA7410 (mid-range array) is better suited for heterogeneous connections but still don't perform as well as some (Ultra-SCSI JBOD) Mirrored Disk/iX solutions. What you do get is added hardware reliability, added data protection and ease of management because most array component failures (disk or media problems) are handled transparently by the array. Adding MPE's High Availability Failover/iX increases availability to both the VA7110 and VA7410 by protecting the I/O path components between the HP e3000 and the storage array.

Purchasing array controllers with as much memory cache as possible is important along with making sure there is enough available server memory for MPE/iX so that it does not have to go to disk will help improve the performance picture.

There have been a number of VA7xxx configuration change recommendations issued by HP that help to resolve HP-UX problems that if implemented on arrays attached to the HP e3000 have caused major performance problems. One such change describes reducing the Queue Depth of the VA from the default value of 750 and to set it to a very small number. This change solves an I/O saturation problem when HP-UX issues too many I/O requests for the VA to handle. MPE handles its own requests and will only issue 8 I/O requests for each MPE/iX configured Ldev. The "Queue Full" message is an unexpected error and at best MPE/iX will stop all I/Os on that bus (I/O pathway), abort all those pending I/O requests, reset and clear the bus and then restart all those pending I/Os each time it encounters this error message or worse do a System Abort if this reset and I/O restart happens during critical operating system I/O.

Business Copy and Continuous Access are not supported and there are no plans to release these features for use on the HP e3000.

Enterprise Virtual Array (EVA)

Not supported for the HP e3000.

Early indications are that it is much faster than the VA Family of storage and has many of the same features as the VA Family but is a higher storage capacity and connectivity solution.

But is not supported on the HP e3000.

HP SureStore Disk Array XP256

This product was born out of the partnership between HP and Hitachi Storage and was the replacement for the EMC Symmetrix line (see older white paper) and provided much of the same features but added an increased array cache, faster disks and provided a host based program to manage/control Business Copy XP and Continuous Access XP high availability features. Though this product used the Hitachi 7700 Storage hardware frame, HP provides value add with its HP proprietary firmware base. This means that the XP256 and Hitachi version are different and the Hitachi version will not work with the HP e3000.

Both the XP256 and (older) EMC Symmetrix provided an enterprise storage and connectivity solution for a heterogeneous server environment with autonomous control within each defined domain or operating system environment.

Both storage products were originally offered with a number of HVD-SCSI ports/connections and later the XP256 started supporting fibre channel for HP-UX. Only HVD-SCSI was/is supported for the HP e3000.

The XP256's much larger cache and faster drives played an important role in improving overall performance issues caused by Read/Write Cache Misses for the HP e3000.

For a list of supported high availability features for this product please see the ha & storage matrix at http://jazz.external.hp.com/mpeha

The XP256 is no longer available from HP but may be still available in the used product market.

HP SureStore Disk Array XP512/XP48

The XP512 and smaller capacity XP48 were HP's entry into 1Gbit per second fibre channel connectivity. The XP512/XP48 again supported multi-host heterogeneous environments with up to 16 FC ports.

The earlier versions of MPE/iX released during that year (2000) only supported HVD-SCSI so HP OEM'ed a product from Vicom Systems that (basically) converted HVD-SCSI to 1Gb FC for a direct connect to an XP512/XP48 (no FC switches or FC hubs). This product is still supported in this environment and was referred to as the A5814A SCSI Fibre Channel Router.

The XP512/XP48 is no longer available from HP but may be still available in the used product market.

HP StorageWorks Disk Array XP1024/XP128

The XP1024 (and smaller XP128) are HP's latest entry into 2Gbit per second fibre channel connectivity. The XP1024/XP128 again supports multi-host heterogeneous environments and along with HP's latest (last) released version of MPE/iX 7.5 on PCI based hardware support native fibre channel (no router) to the portfolio of HP FC switches and FC hardware.

For those customers still using older NIO (HVD-SCSI only) based hardware can upgrade to the latest version of MPE/iX and use the latest version A5814A-003 SCSI-Fibre Channel Fabric Router. This newer hardware supports the newer VA family and XP family of storage along with the portfolio of FC switches. ***(The A5814A is not an upgradeable product to the A5814A-003.)***

The XP1024 (and XP128) support an even larger cache along with previous features like Continuous Access XP and Business Copy XP. MPE/iX version 7.0 and 7.5 support an enhanced version of MPE's Failover/iX (I/O path failover) and also the Cluster/iX solution for even greater availability.

As the storage technology has improved over time in both performance and connectivity, customers have increased the number of servers connected and this has increased the demands of that storage. This increase in demand has sometimes impacted performance on individual servers like the HP e3000.

The increase in performance of the disks and their ability to transfer data to and from the array cache and controllers has helped reduce the impact of cache misses. One of the largest impacts to the overall performance of the array is the effective reduction of the amount of array cache available for each of the servers. As the number of servers (and workload) increased, the amount of array cache available to (just) one server decreased (like to the HP e3000). Changes in the workload of other the servers also has been known to drastically change the performance of jobs and applications running on the HP e3000 (and other servers) that are sensitive to disk I/O traffic delays.

Another performance issue observed which impacts performance on the array is MPE/iX XM checkpoint operations. The checkpoints have a lesser impact on overall performance when a large amount of array cache is available to handle the operation. In the vast majority of instances this is not the case and these checkpoint operations do tend to saturate the array cache which can impact both the performance of the HP e3000 and in some cases other servers connected to the array.

Depending on the application and system I/O activity, the frequency of the checkpoints can become (somewhat) predictable. MPE/iX can be re-configured to reduce the frequency of these events. When the XM log file size is increased the frequency of checkpoints decreases but the interesting observation is that the duration of each of those checkpoints do not significantly increase.

"Rules of Thumb"
for Improving MPE/iX Disk Performance

What I would like to do now is list some guide lines to help make the migration to an array a better experience. These are not hard rules and some flexibility is needed as you incorporate these "Rules of Thumb" into your environment. The most important thing is to help set realistic expectations as storage technology changes and improves:

1. MPE/iX prefers many small Ldevs over a few large Ldevs. Configure your array luns in the preferred 4Gbytes to 18Gbytes range. Also create (more) Ldevs for the system volume set. This allows MPE/iX to "spread" the file extents and system transient space (the swap area) across a number of Ldevs (MPE's extent striping implementation). This will have an improved performance over a configuration which uses a single large Ldev.

2. Restore the application's data to User Volumes. Create as many User Volume Sets as makes sense. This reduces the XM bottleneck to that of the master volume of the volume set, adds more pathways to the data and adds fault containment. This also reduces the amount of data to reload in case of catastrophic disk failure.

3. Gather data on the user's application I/O characteristics. There are several tools available like GLANCE/XL that can help collect this data.

4. Do not configure more that 16 Ldevs per fibre channel pathway or bus and make sure the number of Ldevs per pathway or bus is supported in that environment. *(HVD-SCSI should be limited to 8 while SE-SCSI should be no more than 4)* Consolidating a large number of physical devices to a fewer number of larger capacity devices might create a situation where the array is the bottleneck while too many Ldevs per pathway or bus can cause the connection technology (FC) to be the bottleneck. (Your mileage will vary depending on driving habits)

5. Purchase a high availability disk drive technology for its feature set and its ability to protect data. Performance of these products is dependent on a clear understanding of how it is used, I/O characteristics of the application and the way it is configured.
