HP StorageWorks XP Disk Array design considerations for Microsoft Exchange 2003 — white paper

Abstract

The goal of this paper is to present an overview of the benefits that HP StorageWorks XP disk array technology provides for designing the storage component for a Microsoft® Exchange environment. Because messaging is often considered a mission-critical application, an XP storage infrastructure enables customers to deploy Microsoft Exchange 2003 in a highly available way. Often, this is the result of a server consolidation project as companies migrate from Exchange 5.5.

At the onset, this paper briefly reviews the XP disk array family and the reason these arrays are often found in mission-critical environments. Then it discusses the XP array operating with Exchange and shows Exchange requirements for large volumes hosting single-file databases.

Later, possible XP array configurations and XP emulation types are discussed, and storage configuration ground rules are explained to optimally deploy Exchange in an XP array environment. The XP array is a high-end enterprise storage back-end component that is unlikely to be used solely to host Exchange. Chances are that mission-critical applications and databases (such as SAP/R3, Oracle®, and so on) are already in place and might also share the array resources. For best performance results, the customer must attempt to understand and analyze the I/O workload patterns existing on the XP array and then plan the target environment with the inclusion of Exchange.

In summary, this paper shows how a high-end XP disk array provides customers with the best storage platform for Microsoft Exchange, as they move to increase consolidation of servers and storage.

XP disk array background

The XP disk arrays represent the high-end or enterprise class of the HP StorageWorks line of disk arrays, offering maximum performance and the highest levels of availability, together with unsurpassed scalability. Through a unique set of not only OEM, but also engineering agreements with Hitachi Ltd. in Japan, the XP disk array is a full-featured storage array with its own firmware and, of course, HP management tools, together with global storage solutions and services.

The XP disk array is often seen in mission-critical environments, where data protection must not be compromised and availability must be as high as 99.999%. Traditionally, Microsoft Exchange deployments use mid-range to high-end storage arrays.

Figure 1. XP12000 disk array

The enterprise class of XP arrays is available in various models, including the HP StorageWorks Disk Array XP128 (128 disks), the HP StorageWorks Disk Array XP1024 (1024 disks), and the new HP StorageWorks Disk Array XP12000 (1152 disks)—and just like any back-end array, has progressed through many technology evolutions. The XP family was first introduced with the HP Surestore Disk

Array XP256 in 1999. The XP128 disk array, XP1024 disk array, and XP12000 disk array vary in capacity (GB), peak transaction, and peak data rates (I/O per second and MB/s) but have similar firmware functionality, including supported hosts and a full complement of highly functional array-based software, which can simplify and reduce the costs of data management.

Basics—Disks and array groups

Disks are provisioned in an XP array in groups of four Fibre Channel-Arbitrated Loop (FC-AL) dual-ported disks. The size of the disk is fairly standard (36 GB, 72 GB, and 146 GB) as is the rotation speed (10,000 rpm to up to 15,000 rpm for the 36-GB and 72-GB models). These disk groups are called parity groups or array groups, depending on the way they are later formatted and configured in the array. When formatted, an array group is used by means of a logical device (LDEV). There is a maximum of 8,192 LDEVs in the XP12000 disk array, XP1024 disk array, and XP128 array.

Two primary factors must be considered when configuring an array group:

- Level of redundancy—RAID 1 or RAID 5. RAID 1 is pure mirroring; RAID 5 is distributed parity. The possible disk configurations are 2D+2D (RAID 1 using four drives) or 3D+1P (RAID 5 using four drives and distributed parity). The parity is confined to the four drives. The cost of redundancy is therefore 100% for RAID 1 and 25% for RAID 5.

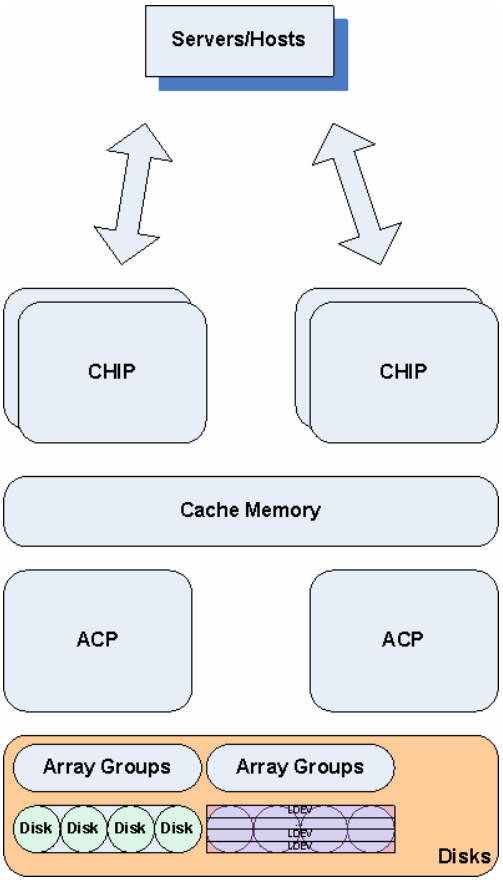
  For the XP12000 disk array, XP1024 disk array, and XP128 disk array, you could also consider a 7D+1P RAID 5 configuration, offering better throughput (eight disks instead of four) by increasing the stripe length and better disk utilization (overhead of 12.5% instead of 25%) by using the equivalent of seven disks for data and one disk for parity.

- Emulation type—This factor determines the number and size of the LDEVs presented by the array group. For now, and until Exchange specifics are discussed, consider the LDEV a slice of an array group.

Simplified structure

The XP disk array is a cache-centric array—data transfers back and forth between processors and the hosts or disks, passing through a memory cache, which is implemented in an optimized and redundant manner, as represented in Figure 2.

Figure 2. Simplified XP structure



Channel Host Interface Processors (CHIPs) control data movement between the host servers and cache memory. CHIPs come in pairs for redundancy. There are different types, depending on the host interconnect mode chosen (such as Fibre Channel, ESCON, and FICON). CHIPs are the interface to the host bus adapter (HBA), with intermediate fabric components, depending on the topology and protocol chosen.

The Array Controller Processors (ACPs) perform all data movement between the disks and the cache memory. Just like CHIPs, they also come in pairs for redundancy and load balancing purposes.

The cache memory is used temporarily to store data from the host until it is written to disk (write-back cache). It is also used to stage data requested by the host from the disk (read or read-ahead cache). The write cache area is both mirrored and backed up by a fully redundant battery. With a minimum of 2 GB, the cache size can grow up to 128 GB.

The cache memory is an integral part of the system, and all transfers flow through it—there is no provision for write-through I/O when the array is operating normally. Is this a bad thing? Not at all. The cache-centric architecture delivers outstanding performance in a variety of environments, including Microsoft Exchange. Data transferred in the cache is purged on a least recently used (LRU) basis.

In addition to the cache memory, there is an additional memory structure called shared memory, which is used for storing disk control information and cache directory entries.

Communication across these components (cache memory, ACP, and CHIP) is implemented by means of a crossbar switch with an interconnect bandwidth of 10 GB/s and above (depending on the model).

XP array operation with Microsoft Exchange

After a great deal of analysis, it is known that Exchange generates a burst I/O pattern to database and transaction log disks. Also known are its requirements for large volumes (basic or dynamic disks seen by Microsoft Windows®) for hosting single-file databases, such as the Store databases.

In some applications, the cache-centric approach of the XP disk arrays is a major performance enhancer. However, in Microsoft Exchange applications, be careful to design storage to cater to transactional applications, assuming peak load and cache avoidance I/O access patterns. This random access I/O pattern means there must be sufficient disk drives in the configuration to deliver the I/O directly from disk, assuming little or no cache-hit benefit.

Areas of consideration for Microsoft Exchange

Microsoft Exchange uses a dual-commit transaction model for its databases by committing transactions in a series of log files, while also modifying database pages (the actual result of transactions, such as a message delivery) in cache memory (RAM on the server). The modified database pages in cache are known as dirty pages, and the Store "cleans" the pages by flushing them to the database as part of the checkpoint process. Checkpointing occurs as a background process independent of user activity when enough changes have occurred in memory (20 MB worth of updated pages by default).

Exchange generates I/O through other operations such as the messages that flow through the Simple Mail Transfer Protocol (SMTP) Transport Engine or the X.400 Message Transfer Agent (MTA). Messages are often acknowledged during transmission by a peer messaging server (be it another Microsoft Exchange server or some other host) and are stored locally on an NT file system (NTFS)-based queue, known as MTADATA for the X.400 MTA and MAILROOT for the SMTP MTA, predominantly used in Exchange 200x environments.

Requirements for Microsoft Exchange

Exchange typically uses random read and writes operations of varying sizes to access its databases. The database pages are 4 KB in size, and it is common to have read I/O of 4 KB in size, whereas write operations vary from 4 to 64 KB, depending whether the pages to be written are contiguous. The underlying page structure of the Exchange databases does not adhere to the notions of index and data areas, and it is common to find that the I/O transactions flowing to the main database are purely random across the entire database seek range (that is, they could be anywhere in a 20-GB NTFS file). A storage infrastructure designed for Exchange should therefore be able to service small and parallel random read and write operations with a relatively low response time (less than 20 ms for reads and 15 ms for writes).

Exchange transaction logging is serialized in a series of sequential write operations, one at a time per transaction log stream. A storage volume designed to support transaction logs should therefore be able to service sequential I/Os as fast as possible (6 ms for non-cached sequential writes and less than 1 ms for cached writes). Some consultants often propose to host transaction log files on dedicated drives. In high-end storage area network (SAN) environments, this approach is not feasible or realistic, so be prepared to argue the case during design discussions—it is always a best practice to have a dedicated Windows drive (represented by a letter or a mount point) for the transaction log files of each storage group, but the Windows drive might actually be hosted on shared physical disks on the XP back-end. This configuration can be built out of one or more LDEVs or can be part of one or more array group.

The performance requirements for the database do not depend on the size of the database, but on the user quantity, mailbox quota, and user activity. Typical figures for medium user loads are 0.5 to 0.8 I/O per second per connected user for the database and 0.1 to 0.2 I/O per second for the transaction log area. Heavy user loads have been seen to exceed 1.6 I/O per second per connected user for Exchange 2003. Many customers quickly become focused solely on the size of the spindles or the total size of the database. The number of spindles that host the database should be the primary concern. With average drives now at 72 GB and higher, disk space is normally not an issue; too few spindles to handle the I/O demands **is** a problem. Most Exchange performance issues can be directly linked to poorly designed disk subsystems.

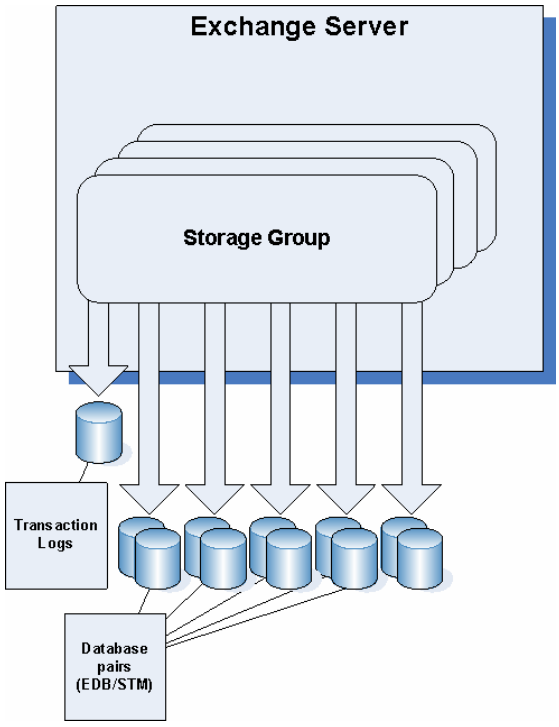
Presenting the volumes to Exchange

For each database, the Store uses a pair of files: an engineering database (EDB) file (known as the Property Store), which contains Messaging Application Programming Interface (MAPI) properties, mailbox and folder tables, and a Support Tools Manager (STM) file (known as the Streaming Store), which is used for streaming network socket data buffers directly onto disk by means of the Exchange Installable File System (IFS).

If you consider a database of 40 GB in size, you must create a volume that is at least 40 GB and generally has additional space for database growth, maintenance space, and deleted item restore. A 75% fill ratio is a usually good guideline.

For each Exchange server, you can have up to four storage groups and, for each storage group, five databases. There is a single transaction log file set for each storage group, as shown in Figure 3.

Figure 3. Exchange and message stores



As a result, you might decide to present many small volumes to an Exchange server (one for each database pair and transaction log area), a pair per storage group, or a pair for the entire server. The actual breakdown depends on the requirements for resilience, but it is a common best practice to use two volumes per storage group—one for the transaction log files and one for all the database pairs that belong to the storage group.

The next decision point involves the number of storage groups to use on the Exchange server. For scale-up environments (4,000 mailboxes and above per server), it is common to use at least three storage groups with four databases per storage group, with a goal of keeping database size to under 100 GB for recovery purposes. While Exchange 2003 SP1 introduces single-bit error checking to fix minor corruptions, a single corrupted page in a database still can cause the recovery of the entire database. Because of Virtual Address space constraints with Exchange 200x, a best practice is to create storage groups on an as-needed basis.

Connecting Exchange servers to the XP array

The following gold rule should apply: attempt to distribute, as much as possible, your host connections to the XP disk array CHIP using redundant fabric paths, each connecting to a separate CHIP configured in pairs and configured for both load balancing and fault tolerance. The I/O throughput capabilities of a CHIP pair can be estimated using the HP XP Performance Estimator Tool.

Your dependencies and constraints for the number of CHIPs to use will typically be based more on the fabric topology and SAN configuration constraints.

Creating volumes in XP arrays

As mentioned previously, you must format array groups to obtain the LDEVs desired. The first decision you must make is the RAID level—RAID 1 for mirroring or RAID 5 for distributed parity. The performance difference between RAID 1 and RAID 5 is not as large in the XP array as in other arrays (entry, medium, or high-end). Sequential writes (used for Exchange transaction logging) might actually perform better on RAID 5 because all parity calculation is done in cache and actual disk I/O is minimized when writing an entire stripe. This technique is used in other array architectures and is one advantage of the write gathering capabilities of the controller. In the case of a remote storage replication environment using HP StorageWorks Continuous Access (CA), the RAID level does not directly impact CA performance, given that I/Os are transmitted between arrays caches (and are not a hard dependency on disk transfers).

All things considered, the redundancy techniques have a cost that must considered when sizing cache and processors (ACPs). Using RAID 5 for the array groups is common in a well-balanced XP array configuration for Exchange.

Emulation types

The next step is to determine the emulation type for the parity group. The emulation type determines the resulting available capacity of the parity group and the resulting volumes available for further presentation to the host servers. When dealing with Open systems, such as UNIX® or Windows, the emulation types are known as OPEN-<code>, where the <code> determines the resulting size of the volumes. You might have only one emulation type per parity or array group. Known emulation types are OPEN-E, OPEN-L, OPEN-V, and so on. Check the XP array documentation or your nearest XP array specialist for a description of the various emulation types.

For the case of XP12000 disk array, XP1024 disk array, XP128 disk array, and Exchange, you might consider using OPEN-V emulation, which enables you to determine the size of the resulting volumes after you have applied the emulation type V. OPEN-V volumes are created either by specifying the number of volumes, the target volume size, or both. The maximum size is 62 GB (XP array formatted capacity). Other emulation types, such as OPEN-E (approximately 14 GB) and OPEN-9 (approximately 7.3 GB), can also be considered in place of OPEN-V. Because of their fixed sizes, they are more straightforward to handle and might work best in your scenario (such as the number of array groups to spread workload across).

After defining the size of the LDEV by the emulation type, uniquely identify the LDEV using a control unit (CU) and LDEV number. The resulting CU:LDEV combination uniquely identifies a single virtual XP array disk or volume. When managing the XP array, refer to the volumes using this CU:LDEV number (for example, 01:2a). The volume can either be presented to a host server or further combined to create larger logical volumes in the XP array.

Figure 4. Three main parts of an array/parity group

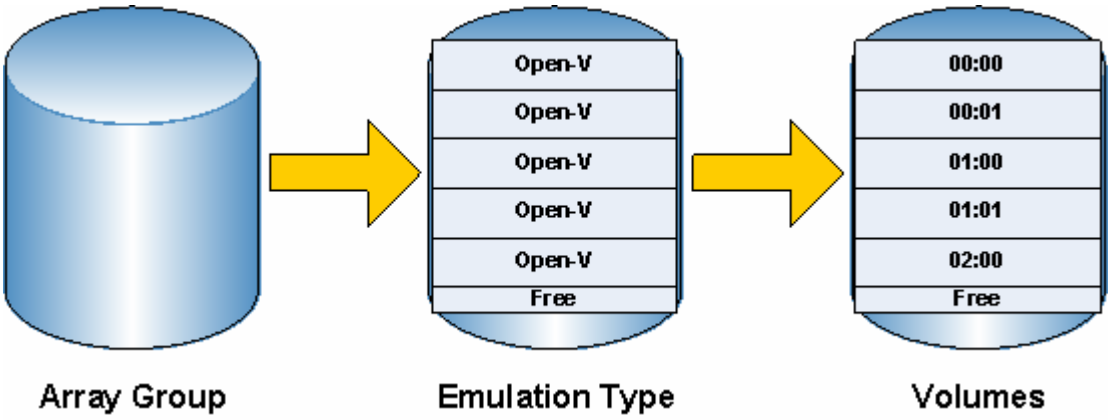
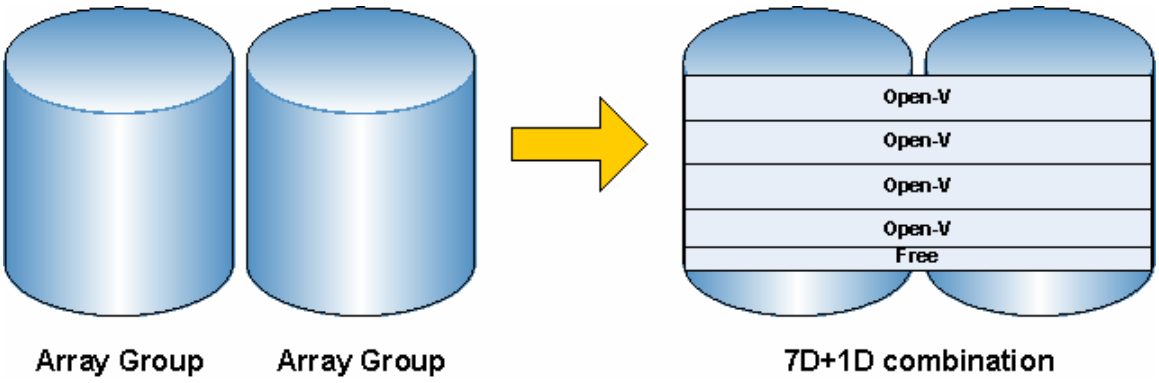


Figure 4 shows the three main parts of an array group, which eventually form volumes (LDEVs). For Exchange, first determine which data area the LDEV will eventually host. For databases, aim at using as many spindles (disks) as possible. If you can format a single array group with OPEN-V emulation, you obtain an LDEV that represents space (for example, 30 GB) served by only four disks. Because the I/O requirements for Exchange do not depend on the database size, even the first 4 GB assigned to a group of databases hosting 1,000 users will demand approximately an 800 to 1,000 I/O per second request rate.

With the XP12000 disk array, XP1024 disk array, and XP128 disk array, you can format your volumes (LDEVs) in a parity group that comprises two array groups instead of one. This mode (7D+1P) is preferable in I/O intensive workloads of Exchange (Figure 5).

Figure 5. Combining array groups in a 7D+1P parity group



The two array groups, paired in this way, must be connected to two separate ACPs. HP has successfully used this mode in Exchange deployments, and is it recommended, even though it uses a RAID 5 distributed parity redundancy scheme.

Creating LUNs for Microsoft Exchange using LUSE

After defining your LDEV, you must define logical units (LUNs) that will be associated or mapped to one or more CHIP ports, allowing for alternate path definition. Normally, the LDEVs are a fixed size, as discussed before. In the case of Exchange for which a 1,000-mailbox storage group might require 150 GB (for example), you have two options:

- Present several LUNs to the server and use host-based striping (for example, dynamic disk striping using Windows Logical Disk Manager). Although favored for performance purposes, this is not necessarily the best approach (see later in this section).
- Concatenate several LUNs inside the XP array, making a LUSE volume that functions as a large LDEV.

Logical Unit Size Expansion (LUSE) is defined as an XP array feature for creating expanded volumes that are larger than the defined emulation types (for example, OPEN-3, OPEN-9, OPEN-V, and so on). To build a LUSE volume, use LDEV issued from the same emulation type. Because OPEN-V is variable in size (between any two array groups), you might assemble a LUSE made of LDEVs of variable size, as long as the same emulation type (and size) is used in the LUSE. To use as many disks (array or parity groups) as possible, aim at using a relatively small (approximately 10 GB) LDEV size in the OPEN-V emulation (or use OPEN-E or OPEN-9 emulation). You can assemble up to 36 LDEVs in a LUSE; based on your capacity requirements, you might want to use different emulation types (for example, OPEN-E limits the maximum size of a LUSE to 36 × approximately14 GB = approximately 504 GB).

Exchange does not differentiate between a LUSE and a host-based striped volume. However, from an administration viewpoint, it is better to simplify the configuration on the Windows server and use detailed volume configuration in the XP array environment. This configuration has the advantage of possibly restructuring volumes or optimizing them with little or no impact on the Windows configuration.

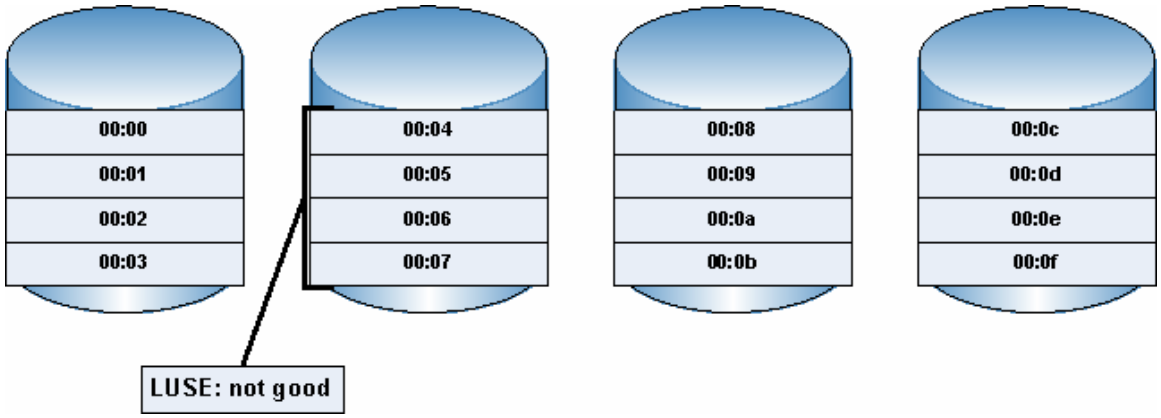
In Microsoft cluster environments, you cannot use dynamic disks anyway, so your only choice is to either make use of LUSE or use VERITAS Volume Manager to enable host-base striping in clustered configurations. Always be sure that the hardware and software combination is fully supported by Microsoft, HP, and the third-party vendors involved (here, VERITAS). There have been instances of mutual incompatibility between the HBA firmware and driver versions, multi-path software, and host stripping.

To avoid additional complexity at the Windows server level (such as the utilization of a volume manager on top of the existing storage components), you can use LUSE concatenation to make the large LUNs required by Exchange databases. However, you must be careful that these LUNs are sufficiently spread across enough array groups, so:

- Maximize the number of disks used for any given database volume.
- Avoid conflicting access to the same array group (creating a hotspot).

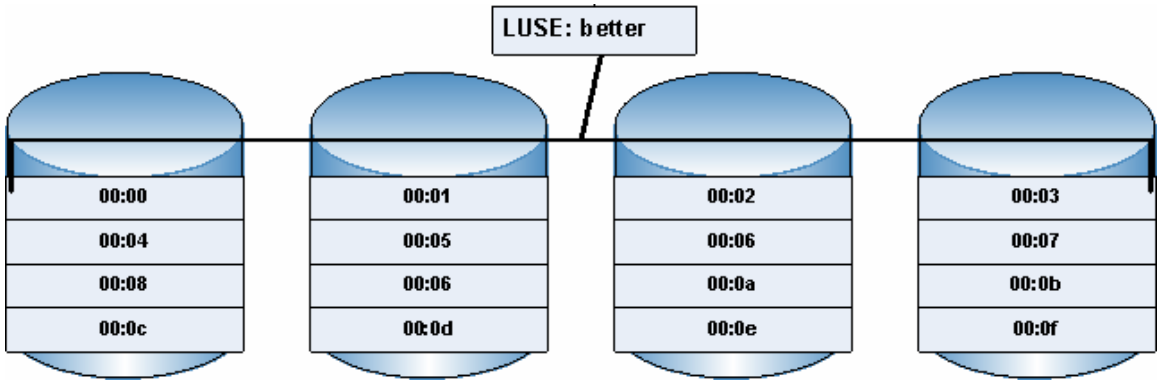
Because the allocation of LDEVs to a LUSE volume is sequential (based on the LDEV identifier), you must be careful to number them correctly during the initial formatting of the LDEVs to spread a LUSE volume across as many array groups as possible. For example, Figure 6 shows a LUSE concatenation using LDEVs of the same parity group. While the capacity objectives will be met, this volume will only use the disks defined for the LDEV (eight preferable, four otherwise).

Figure 6. Defining a LUSE using LDEVs from the same array group



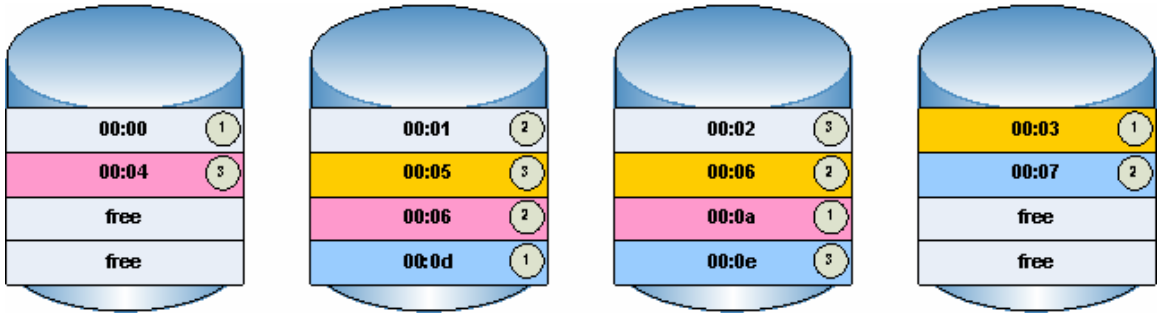
In Figure 7, the LDEV numbering allows the LUSE to be spread across the four array groups shown.

Figure 7. Dispersed LUSE



This approach is the most popular for defining the LUNs used by Exchange databases. The preceding example is interesting in so far as you should also be sure that the first LDEV of the first LUSE is not on the same array group as the first LDEV of the second LUSE, and so on. This approach enables you to spread the I/O load across LDEVs, as shown in Figure 8.

Figure 8. Avoiding hotspots by spreading LUSE start LDEV across array groups



In the case of transaction log files, there is little space required (typically 10 to 20 GB), which can easily be provided by a single LDEV. Because the workload is sequential, there is no need to provide many disks; cache can help to reduce the latency of an I/O, provided there is no queuing on the target disk. In fact, consider hosting the transaction log LDEVs on fairly idle array groups and let the XP array mitigate concurrency with proper caching and write-gathering functions. In any case, this configuration is part of the XP array cache management, and the array should not function at the detriment of Exchange. To ensure that this is not the case, do not hesitate to calibrate, characterize, and demonstrate that the I/O workload expected from the XP array can indeed be obtained. You can do this using various I/O stress tools such as Microsoft Jetstress or Iometer.

Optimal configuration?

There is not a single optimal configuration for a sophisticated storage solution like the XP disk arrays. You might find yourself with a wealth of choices and options—sometimes so many, that things become more complex!

The ground rules to deploy Exchange with the XP arrays are:

- Use as many spindles (array groups) as possible. If the XP array allows, use a 7D+1P array grouping.
- Do not attempt to isolate data areas. In an XP array configuration, isolating databases on their own spindles means also confining them to a small set of array groups, which decreases the request rate (I/O per second) capabilities of the volume. You might want, however, to avoid the first LDEV of a LUSE LUN being on the same array group as the first LDEV of another LUSE LUN. Best results are achieved by mixing I/O intensive volumes with static file stores and spreading them on as many spindles as possible.
- Spot relatively idle array groups for transaction logging. Microsoft has recommended many times that transaction logging should be on dedicated drives because the best way to achieve performance is to dedicate drives to the sequential stream of data. Microsoft recommends sequential writes to the logs should be as fast as possible. Dedicating a drive is one way of achieving this. Write cache is a better solution and might actually turn out to be appropriate for XP disk arrays.
- Do not rely on cache (but when tuning the XP array for I/O performance, this mistake rarely happens). Cache is beneficial only if the back-end disks are fast enough for destaging the data in cache. A good rule of thumb for Microsoft Exchange I/O throughput sizing is to forget about the cache initially (the term used in XP array environments is to size for "cache avoidance").
- Consider additional space for database maintenance and the use of the Recovery Storage Group.

- Do not try to isolate quorum disks (as seen in Windows clustering) on their own array group. Dedicate a LUN, but quorum disks have been deployed on shared array groups in production environment supporting more than 10,000 Exchange mailboxes in a satisfactory manner.
- Perform a full characterization of the end solution using tools such as Iometer or Jetstress.
- Consider using HP StorageWorks Business Copy for backup and recovery purposes using Exchange 2003 support for Windows Server 2003 Volume Shadow Copy Service (VSS) and Volume Disk Service (VDS).

Finally, there will be cases where you must add Exchange databases to an existing XP array environment for which data placement and array group formatting might not be the most appropriate for Exchange. For instance, RAID 5 might already have been chosen as a redundancy technique, and you might have to use a specific emulation type, such as OPEN-L (approximately 36-GB LDEV). Attempt to establish a two-way communication between the Exchange storage design and the XP array support specialist, and be ready to adapt your Exchange design.

The XP array is a high-end storage back-end component that it is unlikely to be used solely to host Exchange. Chances are that mission-critical applications and databases (SAP/R3, Oracle, and so on) are already in place and sharing the array resources. For best performance results, do not just add Exchange database volumes to the XP array—attempt to understand and analyze the I/O workload present on the XP and then plan the target environment with the inclusion of Exchange.

Summary

The XP disk array comes with configuration specifics that are typically different to other ranges of HP storage components. Be sure to always involve an XP array configuration specialist before deciding on the XP disk array and the Exchange designs.

The XP array is certainly a sophisticated and complex technology, which is typically found in mission-critical environments. To date, the vast majority of Exchange deployments have used lower-range storage technology. As customers move to increased consolidation of both servers and storage, it is likely that XP disk arrays become a popular platform for products such as Exchange, which depend heavily on storage. Be prepared to spend time with XP storage architects and administrators to understand existing deployments and to educate Microsoft and other external consultants on the XP array technology. Be ready to prove your design with the appropriate tools and follow-up performance monitoring and analysis.

5982-7883EN, Rev. 2 09/2004