Summary of Best Practices for Storage Area Networks

IBM

This document can be found on the web, www.ibm.com/support/techdocs

Version: 1.2

Date: January 26, 2012

IBM SAN Central

Jim Blue

jeblue@us.ibm.com

Notices:

This white paper reflects the IBM SAN Central support team concerns from customer feedback about the lack of a single source of best practices related to storage area networks (SANs) and storage systems. It has been produced and reviewed by members of numerous IBM support teams and is presented “As-Is”.

The use of this information or the implementation of these techniques is a customer responsibility and depends on the customer’s ability to evaluate and integrate them into the customer’s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

IBM may not officially support techniques mentioned in this document. For questions regarding officially supported techniques, please refer to the product documentation, announcement letters or contact the IBM Support Line at 1-800-IBM-SERV.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ^{TM}), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®
DS4000®
DS6000^{TM}
DS8000®
Enterprise Storage System®
FlashCopy®
NetView®
Redbooks®
Redpapers ^{TM}
SAN Volume Controller
System Storage DS®
Tivoli®
TotalStorage®
XIV®

The following terms are trademarks of other companies:

Microsoft, Windows and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Table of Contents

A Summary of SAN Best Practices for Storage Area Networks

1 Introduction

Whether a SAN environment is simple or complex, many corporate SAN administrators and their management, want to have a collection of general "rules of thumb" on best practices in a single document. In general, a vast amount of best practices information and “rules of thumb” currently exists, but these informational nuggets are scattered across numerous documents, web sites and other sources. In most cases, these best practices snippets are found within product-specific documentation and are not necessarily presented from a solution viewpoint. Based on the historical experience of various SAN Central team members, and others, a common question is “Where can I find all of this information in one place?”   This white paper is an attempt to bring a number of these basic rules and their benefits towards effective SAN management including change management techniques into one document.

This white paper will continue to be a work in progress as the storage industry evolves with new products, enhanced product functions, changes with standards as well as new pervasive issues are discovered and fixed. This white paper version does not go into great detail from a server perspective nor does it include best practices with tape storage and/or certain generic applications (such as data mirroring). This white paper is also presented from an open systems point of view involving Fibre Channel protocol (FCP). Later revisions will include IP storage based coverage as iSCSI (Internet SCSI) and FCoE (Fibre Channel over Ethernet) continue to gain traction and assume a bigger role within the data Center.

An extensive listing of references is provided to source publications used during the writing of this paper. Otherwise, guidelines are based on information from SMEs plus a few recommendations are based on customer situations which have been resolved with assistance from the SAN Central and other IBM support teams. A conservative approach with these best practice suggestions has been assumed which will lead to a more proactive approach to SAN operations. These guidelines are not applicable to all SAN environments nor are they intended to be a set of absolutes. The reader is invited to use, or ignore, the following recommendations and suggestions where applicable to their particular SAN environment.

2 Zoning

Zoning is the way Storage Area Networks (SANs) keep devices isolated from each other; this has implications for security, fabric stability, and resource management. Without zoning, it would be very common for a problem on one host to be able to affect every host in the fabric; without zoning the problems this can cause are difficult or impossible to troubleshoot.

A number of zoning implementation methods are listed below, starting with the worse case and progressively improving the granularity of zones.

- One-big-zone, also known as no zoning (worse case)
- Zone by operating system
- Zone by HBA vendor
- Zone by application
- Zone by cluster groups
- Zone by initiator port (best scenario)

The worst case scenario of one big zone (effectively, no zoning) means that all devices can communicate with all other devices. This approach may be workable and stable for very small (1-5 hosts and a single storage system) SAN environments, but it is strongly advised to never use this method.

Creating zones based on operating systems, HBA vendor and or by application improves the granularity, or scope, of individual zones. This reduction in scope is due to the operational functionality of various components. Some operating systems will try to access the resources of any device that it can, or has ever been able to, access. When HBA adapters register with the nameserver, they normally specify whether they are an initiator or a target. However, some ports will register as both (target and initiator) and thus cause potential confusion when a host system requests information about what target devices within the SAN environment are accessible.

The suggested implementation method is to create zones for individual initiator ports (typically a single host port) with one or more target ports. Although fibre channel standards and switch vendors allow zones to be created using device WWNN (world-wide node names), this practice should be avoided. When a WWNN is used in place of a WWPN for a device, switches interpret the WWNN to designate all associated ports for the device. Using the WWNN in a zone can cause multipathing issues where there are too number of paths between a server and storage.

A possible refinement is to keep destinations isolated from each other in “single-initiator, single-target” zoning. This is sometimes used in environments where it is common for a single host to access multiple destination devices. While this provides some additional measure of preventive isolation, it must be balanced with the additional administrative overhead involved with the increased number of zones.

No matter which of the above zoning implementation methods is utilized in a SAN environment, there can be exceptions to the rule. One such example involves clustered servers which handle some portion of intra-node handshaking across the SAN fabric.

The IBM Subsystem Device Driver (SDD) can support up to 16 paths per virtual disk with DS8000 devices. With SAN Volume Controller (SVC) disks, SDD supports up to eight paths per virtual disk. However, optimal SDD performance is obtained with just four paths per virtual disk. Similarly, other third party and native operating system multipathing utilities can support various numbers of paths per volume, but most of these utilities also recommend only four paths per volume.

Aliases can greatly ease routine zone management tasks. Aliases associate a human-readable name with the long hexadecimal world-wide port name (WWPN) and are most useful for sets of ports which are used in more than one zone definition. Aliases are typically most used with storage system ports utilized as target(s). For servers, the name of the zone itself is usually sufficient, so server aliases do not provide similar benefits as do storage system aliases for an equal amount of effort.

Aliases will allow the incorporation of path sets, or a limited number of paths being assigned to a single alias. For example, multiple paths to a given SVC IOGroup and/or DS8000 storage system can be readily defined with a single alias in a zone definition. With the creation of just a few such path set aliases, it will be easier to manually rotate host connections between the different ports on a storage system and thus achieve good workload balance among the storage resources while minimizing the work effort by the SAN and storage administrators.

The key point is that zoning is an important method to balance the workload across multiple ports for a given edge device, whether it is a disk storage system with dozens of ports or a server with four ports. Balancing the workload across a storage system with multiple ports will help prevent a single storage port from being the source or cause a performance bottleneck due to high latency or outfight congestion.

One overall theme for SAN best practice guidelines is consistency, and this concept is equally applicable to zone element naming conventions. A zone name should indicate which devices are enabled to communicate by the specific zone along with enough info to determine which fabric the zone can be found. At a minimum, the zone name should provide information about the following items:

- Location
- Server
- Server port
- Storage resource
- Fabric

For example, the zone name “PilRd_SQL12_ed74_DS8K1_1132_A” denotes the data center location (Pilgrim Road), the server (SQL12), which port on the server (ed74 - which are the last 2 bytes of the server port’s WWPN), the storage system (DS8K1), the storage port (1132 - DS8000 rack 1, IO bay 1, slot 3, port 2) and which fabric (A) within the SAN environment.

Never mix tape and disk traffic on the same server Host Bus Adapter (HBA) port with one or more zone definitions. This issue is a source of confusion for many administrators due to a number of reasons. Most HBA vendors state that they support mixing tape and disk traffic on the same port. Based on the collective experience of the IBM SAN Central team, customer environments with tape and disk traffic on the same HBA port will eventually experience problems.

Fibre channel is nothing more than a transport mechanism, and the bulk of the traffic is SCSI. So consider this question: would you mix tape and disk devices on an older style parallel SCSI controller bus? Most SCSI logical interfaces associated with an HBA port will use optimized settings for either tape or disk. The unit of time for disk IO operations is measured in milliseconds compared to seconds or even minutes for tape. Thus, it should be obvious that the expected timing parameters used for one type of traffic is radically different for the other traffic type.

Keep test and production environments separated by zoning. This suggested best practice statement is due to the high risk of disruptions that typically occur in test and development environments compared to the desired goal of stable operations for the production side.

Switch vendors offer management tools and/or commands which can assist a SAN administrator with planning and changing a zone configuration. When such tools are available, the SAN administrator should make use of them as a validity check if for no other reason. Switches will attempt to keep the impact of zone changes as local as possible to just the edge devices that may be impacted by the changes.

However, if there are enough changes and/or the change scope is sufficiently large, the fabric may respond by notifying all connected devices of a new zoning configuration. In these cases, there is likely to be some minor fluctuations in traffic levels while the individual edge devices respond to the state change notifications. As a result of this potential impact, it is strongly suggested that changes to the zone configuration be introduced during periods of low overall SAN traffic. In general, plan on approximately 2 to 3 seconds per active device port in the fabric before full stability after a significant zoning change has been made.

Another general SAN best practice guideline is routine housekeeping. If a zoning element (zoneset, zone, zone member, or alias) is removed from the active configuration, delete the unused element when the zone configuration is changed. This simple action will reduce potential confusion on the part of other SAN administrators as well as any technical support personnel assisting with troubleshooting.

3 Fabric and Switch

A key maintenance operation to proactively monitor a fabric involves of clearing switch and/or port statistics on a regular basis. Clearing statistics provides a simple baseline of zero values for error counters. After clearing the error counters, the SAN administrator will have a much easier time looking for non-zero values when performing routine checks of switch operations. One important question arises with this practice which is “how often should these statistics be cleared?”

In regards to fabric, there are two primary information sources that can be cleared:

- **Port Statistics Counters** (link reset, loss of sync, loss of signal, detection of faulty frames, dropped frames)

- **Event Logs** (Fabric reconfiguration, fabric rebuild, device leaving fabric, hardware failure)

Since **'Event Logs'** can be configured to show (or not) different types of informational, warning and error messages, the need to routinely clear event logs will vary. The more different types of messages are enabled to appear in the event log, the more likely the event log should be cleared whenever port statistics are cleared.

When port statistics are cleared, certain routine activities need to be considered. One such example is routine maintenance activities which may cause error counters to increment which can be problem indicators but are the result of maintenance actions. Thus, link resets, loss of signal, loss of synchronization, CRC and other errors should be expected to some degree whenever the following routine actions occur:

- Server or Storage leaving the fabric due to reboot, Firmware upgrade, manual invoke of High Availability failover

- Manual invoke switch reboot, Switch Firmware upgrade

- Manual Fabric port reset

- Fabric links removed / reseated

The approach of routinely clearing of port statistics counters will insure that the statistics represent recent time periods and not include historical events which have already been accounted for. The time span between clearing port statistics counters is dependent on the fabric administrator's judgment and comfort level. The time period can range from every couple of days to once a week to once a month. Based on the historical experience of SAN Central, it is highly recommended the time period is not longer than once a month.

Routine clearing port statistics should not be allowed whenever some degree of instability in the fabric has been detected since these counters can now provide important information about unplanned events. Once these data points have been collected as part of the routine problem data collection routines, then it is safe to clear the counters.

Switch names should be unique and follow some naming convention which will identify the fabric in which the switch is a member. With modern switches now capable of being configured into multiple virtual switches and each virtual instance has a different Domain ID, the inclusion of a Domain ID in the switch name is not suggested. Switch Domain IDs within redundant fabrics should be unique between the fabrics if at all possible. This practice allows administrators and product support specialists to review data from edge devices (servers or storage systems) and be able to quickly determine which fabric and switch a specific device port is a member.

Switches in a common fabric should have similar code and/or firmware levels whenever possible. Having a fabric comprised of ten switches of the same model and type running nine different code levels is a recipe for problems. Vendors perform extensive testing of new code versions prior to public release, but most test plans do NOT include scenarios involving different code levels unless there are known situations and/or configurations. One such example involves newer hardware being tested with a new code level where the fabric also contains one or more switches with older hardware which has a “ceiling cap” of some older code version. In these interoperable test scenarios, the release notes for the higher code level will specific information about the older hardware platforms and code level(s) were used during testing.

Switch configuration parameters within the same fabric should be consistent. One example is the time-out parameters such as R_A_TOV. For some of these parameters, the switch will not merge into the fabric unless they are the same values and the resultant error messages will provide information to be able to quickly resolve the parameter mismatch. However, some switch parameters can be different from one switch to another within the same fabric and connectivity and/or performance issues may not be as easy to troubleshoot.

When ports are not in use on the switch, please keep the port cap inserted in the port. Port caps are provided to keep both dust and light contamination within a port to a minimum. If dust particles or other small debris items are able to get inside the physical connector while not in use, later attempts to activate and use the port may be problematic due to a poor connection.

Ports not in active use should be administratively disabled. The reasoning for this recommendation is that disabled ports will not generate and/or log spurious, phantom errors which can be misleading. When a switch port is activated, allow the attached device to negotiate to its desired port speed and then hard configure the switch port for that port speed. This action will reduce the time for the remote device port to synchronize in the future. At the time a switch port is activated, configure a meaningful descriptor for the switch port. Examples of port descriptors include:

- ISL to switch Edge-1-port_13
- SVC1-node 1-port 10

In SAN environments with redundant fabrics and switches from the same vendor are deployed, use the same topology in the redundant fabrics. Using the same topology will be easier for administrators to keep the workload balanced between fabrics. Some exceptions are to be expected, such as single data paths to certain tape drives. Overall, the number of paths between a given server and associated disk storage resource should be balanced between redundant fabrics. Further, one of the redundant fabrics should be capable of handling the total traffic workload for the SAN environment. If the SAN environment is composed of more than two redundant fabrics, the complete failure of one fabric should not overwhelm the other fabrics. Consult Section 6 on High Availability topics for more details on fail-over scenarios.

Locality is the term used to describe the aspect of a SAN design where devices with high traffic levels between each other are connected and zoned across the same switch. One example is the nodes of an SVC cluster on the same switch as the back-end disk systems utilized by the SVC cluster. Thus, high degree of locality means less traffic over ISLs between multiple switches and thus less risk for latency and/or fabric congestion. The implication with locality is that the SAN administrator must have a clear understanding of all traffic patterns to determine which SAN-attached devices should be collocated on the same switch. In certain situations, the situation is very clear, such as a TSM server and its associated tape drives; but not always. Either way, a high degree of locality within a fabric is preferred.

As newer devices are incorporated into a SAN environment, there is a tendency for the newer devices to utilize newer hardware which are capable of higher port speeds than existing fabric devices. As these newer devices are added to the existing SAN environment, the attach point of these devices needs to be considered. Simply stated, devices capable of higher port speeds which will be heavily utilized by existing SAN devices should be located closer to a fabric's center. The reasoning of this approach is a result of the fabric center (the core in a core-edge design) has sufficient connections and bandwidth radiating outwards to the edge for the additional workload.

It is a common practice for storage systems to share a given port among multiple HBA ports on multiple servers to increase the level of connectivity to the storage system. The server-to-storage port ratio is called the fan-out ratio. As the size of fabric grows, the number of switches in a given fabric is very likely to correspondingly increase to the point where servers are connected to one fabric switch and their shared storage port connects to a different switch. Thus, the server-to-storage traffic must traverse at least one inter-switch link (ISL) between the two switches. Since this configuration is repeated for multiple storage ports, there can be more and more servers communicating across a single common ISL. The ratio of total server bandwidth to ISL bandwidth is known as the ISL oversubscription.

For example, there are eight servers connecting to a common storage port traversing an ISL. Each server has a 2 Gbps port connection while the ISL is running at 4 Gbps. The calculated ISL oversubscription would be:

(8 servers * 2 Gbps/server) / 4 Gbps (ISL) = 16 / 4 = 4 for a 4:1 ISL oversubscription ratio.

Switch vendors typically offer guidelines for ISL oversubscription ratios between 7:1 to 10:1. In cases where the traffic levels and patterns are not well understood, this range is a good best practices rule of thumb. However, SAN administrators can make use of higher oversubscription ratios with a good understanding of the traffic pattern utilizing a particular ISL. Ratios significantly greater than 10:1 can be incorporated into the SAN environment.

One such example is a ratio which on paper appears to be 20:1; but the servers sharing the common ISL are evenly divided between servers running heavy database operations during normal working hours, data warehousing applications during the early evening and finally back-up operations late evening. As long as these three workloads have no, or minimal, overlap during the same time of day; the higher oversubscription ratio is not a true representation of the projected ISL load factor. Thus, the range of 7:1 to 10:1 is a good generic best practice statement in cases of unknown workloads, yet not a hard and fast rule which can’t be exceeded when the traffic pattern is known.

When expanding a fabric by adding a new switch, or replacing an existing switch, make sure the new switch does not have any zoning configuration. During the fabric merge operations, the new switch will learn the fabric’s existing zoning configuration from its switch neighbors. Attempting to introduce one or a few new zones to the current zoneset can be problematic including successful merger, switch segmentation, replacing the established zoneset with the newer (and typically much smaller) zoneset or other equally disruptive results.

4 Disk Storage

4.1 General topics

One theme which is repeated throughout best practice statements centers on one word: consistency. And consistency with disk storage systems is very applicable across many areas of operation. One area which is often overlooked is consistency of code version among multiple storage systems of the same type. The primary reasoning for this suggestion is the code level on a storage system is increasing playing a role in the overall interoperability within the SAN environment. Multipathing applications, HBA BIOS firmware, device driver levels, certain operating system and application fixes and/or patches as well as other devices (such as SVC) are dependent on a valid matrix of code versions which includes the storage system. In situations where the server may have resources from multiple storage systems, trying to reduce the size and complexity of the interoperable code matrix is an important consideration for SAN and server administrators.

Consistency should also be considered with the workload balance on a given storage system. The individual access ports of a given storage system should have equal workloads. If a detailed understanding of the workload is not known, then port fan-out ratios are the determining factor for balanced traffic per port. If workload characteristics are known, then fan-out ratios are not as useful as the total IO profile across multiple storage ports.

Maintaining good workload balance across storage system controllers and RAID arrays is just as important as consistent workloads on the storage system's ports. Overloading a controller is possible, so the administrator needs to keep focus on a controller's load factor when allocating storage to some device in the SAN. Similarly, the workload on RAID arrays needs to be balanced for optimal performance of the storage system. Some general guidance is available when calculating the abilities of a RAID array and the values are based on the number of data drives utilized in the RAID array without consideration for any parity drives. Depending on the rotational speed of the disk drive, the following are good rules of thumb:

• 7200 RPM:	75 - 100 IOps
• 10000 RPM:	120 - 150 IOps
• 15000 RPM:	165 - 200 IOps
• SSD:	1500 IOps

The above values are for 3.5 inch hard disk drives (HDD), and can be increased by approximately 15% when using the smaller 2.5 inch HDD. Thus, an eight 15K RPM, 3.5" drive RAID 5 array (7 + P) should be capable of between 1155 to 1400 IOps.

Another generic factor for disk-based storage systems involves separation of different types of workload over storage system ports. In particular, storage ports used for the following purposes should be dedicated for each type and not be shared:

- Data mirroring and replication to another storage system
- SVC cluster
- Direct host access

These three workloads have subtle differences between each other, yet can also stress a given storage system port severely.

Another generic best practice statement involves "housekeeping". Whenever an allocated volume, or LUN, is no longer needed by any server, release the allocated storage back to the free space pool. The reasoning for this suggestion is to help the storage administrator more easily and quickly determine the availability of storage resources on a storage system for other needs. Similarly, delete the host definition information on the storage system whenever a server has been decommissioned, or sunset. Many storage systems are limited in the total number of host definitions within their internal configuration. Removing host definitions will also reduce the risk of production data corruption in cases where an old production server is repurposed for testing and/or development use.

4.2 DS8000

A DS8000 array is a RAID 5 or RAID 10 array made up of 8 disk drive modules (DDM). A DS8000 array is created from one array site. DS8000 RAID 5 arrays will be composed of array width of 6+P+S or 7+P. A rank is created for each array. There is a one-to-one relationship between arrays and ranks. SAN Volume Controller, as part of the open system server can access and use ranks created in Fixed Block (FB) format. FB format divides ranks into 1 GB Extents.

In the DS8000 architecture, extent pools are used to manage one or several Ranks. An extent pool is visible to both of the two servers in the DS8000, but is directly managed by only one of the storage controllers. You must define a minimum of two extent pools with a pool to be managed by each storage controller to fully exploit the DS8000's resources. There is a minimum of one rank per extent pool.

Although it is possible to configure many ranks in an extent pool, for SAN Volume Controller performance optimization, best practice guidelines recommend a configuration of one extent pool per rank per array. This configuration allows for directing storage allocations to a known location within the DS8000. Furthermore, this configuration enhances the ability to manage and monitor the resultant logical disk performance when required.

The DS8000 provides a mechanism to create multiple volumes from a single extent pool. This is useful when the storage subsystem is directly presenting storage to the hosts. In a SAN Volume Controller environment, however, where possible, there should be a one-to-one mapping between extent pools and Volumes. Ensuring that the extent pools are configured in this way will make the subsequent load calculations and the managed disk and managed disk group configuration tasks a lot easier.

Mixing array sizes within a managed disk group in general is not of concern if the differences are not too significant. Testing has shown no measurable performance differences between selecting all 6+p and all 7+p arrays verses mixing 6+p and 7+p arrays and in fact mixing array sizes can actually help balance workload since it places more data on the ranks that have the extra performance capability provided by the 8th disk. There is one small exposure here in the case where an insufficient number of the larger arrays are available to handle access to the higher capacity. In order to avoid this ensure that the smaller capacity arrays do not represent more than 50% of the total number of arrays within the managed disk group.

In summary, to make the most of the performance available from the DS8000 storage subsystems and avoid potential I/O problems:

- When using virtualization, ensure that the storage devices are configured to provide some type of redundancy against hard disk failures (RAID algorithm).
- Create a one-to-one relationship between extent pool and rank.
- Avoid splitting an extent pool into multiple volumes at the DS8000 layer. Where possible, create a single volume on the entire capacity of the extent pool.

- Ensure that you have an equal number of extent pools and volumes, equally spread on the Disk Adapters and the two servers of the DS8000 storage subsystem.

- Ensure that managed disk groups contain managed disks with similar characteristics and approximately the same capacity. Consider the following factors:

  - ➢ The number of DDMs in the array site (for example 6 + P + S or 7 + P) and the physical disk type (for example, 10K/15K rpm).

  - ➢ The underlying RAID type that the DS8000 storage subsystem is using to implement the managed disk.

- Same disks capacity provides efficient use of the SVC striping.
- Do not mix managed disks of greatly differing performance in the same managed disk group. The overall group performance will be limited by the slowest managed disk in the group. Some disk controllers may be able to sustain much higher I/O bandwidths than others, do not mix managed disks from low-end subsystems with managed disks from high-end subsystems.

Best practice guidelines dictate that a storage port should be segregated such that only one of the following is connected to a given port:

- Single SVC cluster

- Hosts

- Mirroring connection to another DS8300

HA oversubscription

Each of the two IO port pairs per storage system host adapter (0/1 and 2/3) should be split between the redundant fabrics to minimize the impact to storage traffic as a result of a fabric-wide issue or failure of an individual host adapter in the storage system.

4.3 XIV

The main goal for the host connectivity is to create a balance of the resources in the XIV Storage System. Balance is achieved by distributing the physical connections across the Interface Modules. A host usually manages multiple physical connections to the storage device for redundancy purposes by a SAN connected switch. It is ideal to distribute these connections across each of the Interface Modules. This way the host utilizes the full resources of each module that is connected to and can obtain maximum performance. It is important to note that it is not necessary for each host instance to connect to each Interface Module. However, when the host has more than one connection to an XIV storage system, it is beneficial to have the connections spread across multiple interface modules.

Although XIV storage systems supports up to twelve logical connections per host server, the multipathing utility on the host server may not be able to fully utilize this number of connections. Utilizing more than 12 Fibre Channel ports for host connectivity will not necessarily provide more bandwidth. Best practice is to utilize enough ports to support multipathing, without overburdening the host with too many paths to manage. There are various key points when configuring the host for optimal performance.

Because the XIV Storage System is distributing the data across all the disks an additional layer of volume management at the host, such as Logical Volume Manager (LVM), might hinder performance for workloads. Multiple levels of striping can create an imbalance across a specific resource. Therefore, it is best to disable host striping of data for XIV Storage System volumes and allow the XIV Storage System to manage the data.

Based on the host workload, the maximum transfer size that the host generates to the disk to obtain the peak performance may need to be adjusted. For applications with large transfer sizes, if a smaller maximum host transfer size is selected, the transfers are broken up, causing multiple round-trips between the host and the XIV Storage System. By making the host transfer size as large as or larger than the application transfer size, fewer round-trips occur, and the system experiences improved performance. If the transfer is smaller than the maximum host transfer size, the host only transfers the amount of data that it has to send.

Due to the distributed data features of the XIV Storage System, high performance is achieved by parallelism. Specifically, the system maintains a high level of performance as the number of parallel transactions occurs to the volumes. Ideally, the host workload can be tailored to use multiple threads or spread the work across multiple volumes.

The XIV Storage architecture was designed to perform under real-world customer production workloads, with lots of I/O requests at the same time. Queue depth is an important host bus adapter (HBA) setting because it essentially controls how much data is allowed to be “in flight” onto the SAN from the HBA. A queue depth of 1 requires that each I/O request be completed before another is started. A queue depth greater than one indicates that multiple host I/O requests might be waiting for responses from the storage system. So, the higher the host HBA queue depth, the more parallel I/O goes to the XIV Storage System.

The XIV Storage architecture eliminates the legacy storage concept of a large central cache. Instead, each component in the XIV grid has its own dedicated cache. The XIV algorithms that stage data between disk and cache work most efficiently when multiple I/O requests are coming in parallel - this is where the queue depth host parameter becomes an important factor in maximizing XIV Storage I/O performance.

A good practice is starting with a queue depth of 64 per HBA, to ensure exploitation of the XIV’s parallel architecture. Nevertheless the initial queue depth value might need to be adjusted over time. While higher queue depth in general yields better performance with XIV one must consider the limitations per port on the XIV side. Each HBA port on the XIV Interface Module is designed and set to sustain up to 1400 concurrent I/Os (except for port 3 when port 4 is defined as initiator, in which case port 3 is set to sustain up to 1000 concurrent I/Os). With a queue depth of 64 per host port as suggested, one XIV port is limited to 21 concurrent host ports given that each host will fill up the entire 64 depth queue for each request.

Generally there is no need to create a large number of small LUNs except when the application needs to use multiple LUNs to allocate or create multiple threads to handle the I/O. However, more LUNs might be needed to utilize queues on the host HBA side and the host Operating System side. However, if the application is sophisticated enough to define multiple threads independent of the number of LUNs, or the number of LUNs has no effect on application threads, there is no compelling reason to have multiple LUNs.

4.4 DS3000, DS4000 and DS5000 Storage Systems

With DS3000, DS4000 or DS5000 arrays, the number of physical drives to put into an array always presents a compromise. On one hand striping across a larger number of drives can improve performance for transaction based workload and on the other it can have a negative effect on sequential workload. A common mistake made when selecting array width is the tendency to focus only on the capability of a single array to perform various workloads, however also at play in this decision is the aggregate throughput requirements of the entire storage server. Since only one controller of the DS3/4/5000 will be actively accessing a given array a large number of physical disks in an array can create a workload imbalance between the controllers.

When selecting an array width, an additional consideration is its effect on rebuild time and availability. A larger number of disks in an array will increase the rebuild time for disk failures which can have a negative effect on performance. Additionally more disks in an array increases the probability of having a second drive fail within the same array prior to rebuild completion of an initial drive failure which is an inherent exposure to the RAID5 architecture. If RAID6 architecture is implemented, then the array can tolerate a second drive failure.

The storage system will automatically create a logical drive for each host attached (logical drive id 31). This drive is used for in-band management, so if the DS Storage System will not be managed from that host, this logical drive can be deleted. This access drive does count towards the maximum of accessible volumes per host maximums, so this action will allow for one more logical drive to use per host. If a Linux or AIX based server is connected to a DS Storage System, the mapping of this access logical drive should be deleted.

When storage resources are utilized by a SVC cluster, best practice guidelines suggest RAID arrays using either four or eight data drives plus parity drive(s) depending on the RAID level.

Whenever DS3/4/5k storage systems are configured for data replication, the ports connecting the source and target storage systems should be dedicated for data mirroring only. No other traffic types, such as SVC or host connections, should be shared on the mirror ports.

With direct attached hosts, considerations are often made to align device data partitions to physical drive boundaries within the storage controller. For the SVC this turns out to be less critical based on the caching it provides, and less variation in its I/O profile used to access back-end disks.

For the SVC the maximum destage size is 32k and thus it is not possible to achieve full stride writes for random workload. For the SVC the only opportunity for full stride writes occurs with large sequential workloads, and in that case the larger the segment size, the better. Larger segment sizes can have an adverse effect on random I/O however. It turns out that the SVC and controller cache does a good job of hiding the RAID5 write penalty for random I/O and therefore larger segment sizes can be accommodated.

The main consideration for selecting segment size is to ensure that a single host I/O will fit within a single segment to prevent accessing multiple physical drives. Testing has shown that the best compromise for handling all workloads is to use a segment size of 256k.

For the earlier models of DS4000 using the 2 Gbps FC adapters the 4k block size performed better for random I/O and 16k performs better for sequential I/O. However, since most workloads contain a mix of random and sequential, the default values have proven to be the best choice. For the higher performing DS4000 storage systems, such as the DS4700 and DS4800, the 4k cache block size advantage for random I/O is less obvious. Since most customer workloads involve at least some sequential workload the best overall choice for these models is the 16k block size.

4.5 SVC

The arrangement of RAID arrays into mDisks and mDisk groups is one of the most troublesome parts of SVC Installation Architecture, and unfortunately the proper method to create RAID arrays and arrange them into mDisk groups varies widely by vendor, applications, necessary capacity, and the layout of the particular disk array in use.

Some “rules of thumb” for mapping back-end arrays to managed disks (MDisk) and MDisk Groups:

a) Any given MDisk Group should only contain disks of a single size and speed.

b) All MDisks within an MDisk Group should be of the same capacity and RAID level, and in keeping with rule 1, made up of the same kind of disk.

c) *Never* spread MDisk Groups across multiple back-end disk units, such as two different IBM 2107 units.

d) If an array has multiple segment sizes to choose from, choose the largest if a particular MDisk Group will be used for multiple kinds of applications. (This rule may not apply to some disk arrays, which do not allow the selection of disk segment size.) The reason for this is that any medium-sized I/O requests received on the MDisk would then be striped across many platters, robbing them of needed IOps per Second capacity. The throughput loss through the disk reading more information than requested is minimal, due to the read speed of modern disk platters.

e) Deciding how large to make an MDisk Group is a delicate balancing act between reliability, flexibility, and performance “smoothing”. Larger MDisk Groups are more flexible and can smooth out I/O load, reducing “hot spots”. However, striping an MDisk group among a large number of disks makes it statistically more likely that a given application will be affected in the case of an individual RAID array failure. At the other end of the spectrum, if each MDisk Group were made up of just a single back-end RAID array, you lose many of the flexibility advantages the SVC is designed to provide, and your disk array will likely suffer from hot spots caused by high-traffic servers.

f) It is important to balance the MDisks among all of the RAID controllers in a chassis. If a disk array has two controllers (like an IBM DS4000 series), you would want the MDisks balanced among those controllers. In a DS8x00 series, you would try to avoid loading up all the mDisks in an MDisk Group on a single Device Adapter pair.

g) MDisk Groups should contain a number of MDisks equal to a multiple of the available paths into the disk array. This allows the SVC to provide better load balancing among the paths.

h) Transaction performance is optimal when each array consists of a single MDisk. If streamed I/O was dominant, two MDisks per array might be preferred.

Due to the nature of Fibre Channel, it is **extremely** important to avoid Inter-Switch Link (ISL) congestion. While Fibre Channel (and the SVC) can, under most circumstances, handle a host or storage array becoming overloaded, the mechanisms in Fibre Channel for dealing with congestion in the fabric itself are not very effective. The problems caused by fabric congestion can range anywhere from dramatically slow response time all the way to storage access loss. These issues are common with all high-bandwidth SAN devices and are inherent in Fibre Channel; they are not unique to the SVC.

When an Ethernet network becomes congested, the Ethernet switches simply discard frames for which there is no room. When a Fibre Channel network becomes congested, the FC switches will instead stop accepting additional frames until the congestion clears, in addition to occasionally dropping frames. This congestion quickly moves "upstream" and clogs the end devices (such as the SVC) from communicating anywhere, not just the congested links. (This is referred to in the industry as head-of-line blocking.) This could have the result that your SVC will be unable to communicate to your disk arrays or mirror write cache because you have a single congested link leading to an edge switch.

All ports in an SVC cluster should be connected to the same dual-switch fabric as all storage devices the SVC is expected to access. Conversely, storage traffic and inter-node traffic should **never** transit an ISL, except during migration and some SVC Stretched Cluster scenarios. Due to the nature of Fibre Channel, it is **extremely** important to avoid Inter Switch Link (ISL) congestion. While Fibre Channel (and the SVC) can, under most circumstances handle a host or storage array becoming overloaded, the mechanisms in Fibre Channel for dealing with congestion in the fabric itself are not very effective.

The problems caused by fabric congestion can range anywhere from dramatically slow response time all the way to storage access loss. These issues are common with all high-bandwidth SAN devices and are inherent in Fibre Channel; they are not unique to the SVC. High-bandwidth-utilization servers, (such as tape backup servers) should also be on the same switch as the SVC. Putting them on a separate switch can cause unexpected SAN congestion problems. Putting a high-bandwidth server on an edge switch is a waste of an ISL.

When an Ethernet network becomes congested, the Ethernet switches simply discard frames for which there is no room. When a Fibre Channel network becomes congested, the FC switches will instead stop accepting additional frames until the congestion clears, in addition to occasionally dropping frames. This congestion quickly moves "upstream" and clogs the end devices (such as the SVC) from communicating anywhere, not just the congested links. (This is referred to in the industry as *head-of-line blocking.*) This could have the result that your SVC will be

unable to communicate to your disk arrays or mirror write cache because you have a single congested link leading to an edge switch.

If at all possible, plan and design for the maximum size configuration the SVC cluster could eventually reach. The design of the SAN can change radically for larger numbers of hosts. Modifying the SAN later to accommodate a larger-than-expected number of hosts will either produce a poorly-designed SAN or be very difficult, expensive, and disruptive. This does not mean you need to purchase all of the SAN hardware initially, just that you need to layout the SAN while keeping the maximum size in mind. Always deploy at least one “extra” ISL per switch. Not doing so may lead to consequences from complete path loss (this is bad), to fabric congestion (this is even worse).

With the exception of a Stretched Cluster SVC solution, n***ever*** split the two nodes in an I/O group between different switches. However, in a dual-fabric configuration, half the nodes ports must remain on the same switch with the other half of the ports on another switch.

Unless two clusters participate in a mirroring relationship, all zoning should be configured so that the two clusters do not share a zone. If a single host requires access to two different clusters, create two zones, each to a separate cluster. The storage zones should also be separate, even if the two clusters share a disk controller.

If a FlashCopy mapping is created, where the source VDisk is a target VDisk of an active Metro Mirror relationship, then this will add additional latency to that existing Metro Mirror relationship (and possibly affect the hosts which are using the source VDisk of that Metro Mirror relationship as a result). The reason for the additional latency, is that the FlashCopy prepare disables the cache on the source VDisk (which is the target VDisk of the Metro Mirror relationship), and thus all write I/Os from the Metro Mirror relationship need to commit to the storage controller, before the complete is returned to the host.

When connecting an SVC to a DS8000, there are several standard configurations that should be used:

• 4-adapter/8-port configuration:	32 array max	
• 4-adapter/16-port configuration:	48 array max	(marginal configuration)
• 8-adapter/16-port configuration:	>48 array max	(or multiple SVC clusters)

Within a particular adapter, the ports should be selected based on the port group (0/1 and 2/3). The 4-port adapter in the DS8300 are oversubscribed on port group boundaries

- 4-adapter/8-port configuration, 4 port-0's and 4 port 2's
- 8-adapter/16 ports, 8 port-0's and 8 port-2's

If two SVC clusters share a common DS8000 storage system, they should use separate paths per SVC cluster per DS8000 host adapter. (E.g. SVC A uses ports 0 and 2, while SVC B uses ports 1 and 3.) More than two clusters should not share the same DS8300.

Additional information on best practice considerations for IBM DS8000 used in conjunction with SVC can be found in the "DS8000 with SVC Best Practices" TechDoc ID PRS4465.

5 Cable Plant

With most fiber cabling used in the data center today, information transfer occurs in two directions simultaneously. This method uses 2 optical fibers contained in a single fiber optic cable and physically connects to ports at each end which houses the transmitter and receiver in a single assembly. The fiber element within an optical cable usually consists of a glass core and a cladding. The glass core provides the light path, the cladding surrounds the core, and the optical properties of the core and cladding junction cause the light to remain within the core. Although the core and the cladding diameters, expressed in micrometers (µm), are often used to describe an optical cable, they actually indicate the physical size of the fiber element.

Mishandling a fiber optic cable will lead to microscopic internal faults within the glass core. In some cases this can cause a complete failure of a cable. Often, faults result in intermittent problems that require a specific cable orientation. This type of fault can be difficult to isolate There are a number of precautions that should be taken when handling fiber optic cables.

- Make sure the cable cutouts in the floor tiles have the appropriate protective edging.
- Route the cables away from any sharp edges or projections that could cut the outer jacket.

- Do not route the cables near unprotected steam or refrigeration lines.
- Do not coil or bend the cable to less than a 96.0-mm (3.78 in.) diameter. If any strain is induced on the cable, the minimum diameter will need to be increased.
- Do not pull cables into position; place them.
- Do not grasp the cable with pliers.
- Do not attach a pull rope or wire to the connectors.
- Always clean the connectors before attaching them.
- Do not remove the protective plugs or protective covers until you are ready to clean the connectors and attach the cables to a device.
- Always leave the protective plugs and protective covers on unused ports and cable connectors.
- Connect the cable carefully to prevent damage to the connector housing or the fiber optic ferrules.
- Before inserting the connector, make sure the connector and receptacle keying are aligned.
- Do not mix different cable sizes (primarily 62.5 and 50 micron) cables in the same port-to-port link.
- Use cables of the correct length so that coiling excess cable is not necessary.
- Use cable management guides for vertical and horizontal support as well as to prevent bend radius violations.
- Avoid using any cable with damaged connectors.
- Avoid using any cable which maintains kinks and/or loops.

A common issue with cable management is cables overlapping multiple pieces of equipment. This simple design flaw has two major affects on infrastructure. A single component may be hindered or prevented from being replaced without added interruption. A typical environment implements a SAN design that has both resilient and redundant components. In order to benefit from this architecture, it is critical that a component can be replaced without interruption to the SAN. Due to overlapping cables, adjacent devices may need to be disconnected to remove or replace a component. This can turn a simple procedure that would only slight reduce the integrity into a massive disruption.

The second major impact on the SAN infrastructure occurs when overlapping or bundles of cables impede the exhaust of a device. This condition needs careful monitoring to ensure cable integrity is not reduced from exhaust heat or the device has poor air circulation which results in the device overheating.

When installing fiber cabling, use labels on each end of the cable which provides sufficient information to identify the appropriate connection ports. Use a logical naming scheme to uniquely identify every cable along with its source and destination points. Good labeling reduces potential confusion as well as easing the task of finding a specific cable in a bundle. Cable labels should always be included as a key step for routine change management documentation and standard operating procedures.

If zip ties are used with cables, extreme care must be utilized. If pulled too tightly, the zip tie will damage the cable's core and/or outer cladding. Velcro tie wraps are strongly suggested. Do not bundle a large number of cables together. Instead, bundle a few cables together and then combine these smaller bundles into larger groups. This practice provides for proper support of cables yet allows for easy isolation and/or replacement of a bad cable with minimal impact. Bundle ties should be located every 2 feet, or less, of cable run.

Depending on the maker of the fiber cabling, the proposed minimum diameter for bending a fiber optic cable can vary significantly. Some vendors provide a simple measurement, while other vendors state that bend diameters should not be less than some number of cable thicknesses. In the absence of a ruler in the data center, a simple guideline is to use the width of four fingers of one's hand. If the diameter of a cable bend is less than that measure, then the bend radius is likely too tight.

Good cable management involves planning within every rack, between racks, at patch panels, in the data center and beyond. Good cable management implies a structured and consistent approach when planning and then implementing cabling. This key concept applies to both optical and wired media since both are likely to be co-located in the same space. Good cable management will address concerns for the current needs while having the flexibility to meet future needs due to technology changes, increased port counts, increase port density or some custom requirement.

A good source of information about planning cable infrastructures can be found in the Telecommunications Industry Association standard TIA-942 titled "Telecommunications Infrastructure Standards for Data Centers". Beside cabling infrastructure, this standard also covers topics such as site space and layout, tiered reliability and environmental considerations.

6 High Availability

In general, the size of the SAN as measured in the number of physical switches will normally determine the design that is most effective to employ. A single switch operates as a stand-alone with no connections to other switches. Two switches are cascaded, or simply connected to each other with a sufficient number of ISLs to meet oversubscription ratios. As the switch count increases, mesh or partial mesh designs can be considered until six or seven total switches. At that point, a core-edge design should be considered.

In a core-edge design, the core can consist of either one or two core switches or directors with cross connections to all of the edge switches. The core switch acts as the focal point for the SAN with connections to the high-throughput devices (typically storage and high-end servers) while the edge switches provides connectivity for the rest of the initiators. Traffic that must traverse across an ISL (inter-switch link) should adhere to oversubscription ratios.

An often overlooked item with the layout of port connections within a fabric is the grouping or clumping of connections to systems with a high port count. If a director is used with multiple line cards, do not place all of the connections to a storage system on the same line module. If a given line module should fail, then accessibility to the storage system in the fabric has been reduced and not completely removed. The same concept should be applied to the deployment of ISLs between switches as well as servers with four or more HBA ports capable of high traffic levels. Modern switches will automatically attempt some degree of traffic balancing across multiple ISLs or trunks (or port-channels). From a resiliency point of view, having two, or more, smaller trunk groups across multiple line modules is better than having a single big trunk using just one line module.

The peak loads should always be taken into consideration and not just the average loads. For instance, while a database server may only use 20 MBps during regular production workloads, it may perform a backup at significantly higher data rates. Congestion to one switch in a large fabric can cause performance issues throughout the entire fabric, including traffic between hosts and their associated storage resources, even if they are not directly attached to the congested switch.

The reasons for this are inherent to Fibre Channel flow control mechanisms, which are simply not designed to handle fabric congestion). This means that any estimates for required bandwidth prior to implementation should have a safety factor built in. On top of the safety factor for traffic expansion, implement a spare ISL or ISL trunk. The fabric still needs to be able to avoid congestion if an ISL were to go down due to issues such as switch line card failure.

As stated earlier, a SAN design can exceed the best practice guidelines for ISL oversubscription ratios of 7:1 with a clear understanding of the traffic patterns. Whenever the “standard” 7:1 oversubscription ration is exceeded, then it highly recommended that fabric bandwidth threshold alerts be implemented. Any time an ISLs exceeds 70%, fabric changes should be planned and implemented to spread out the load further.

Consideration must be given to the bandwidth consequences of a complete fabric outage. While this is a fairly rare event, insufficient bandwidth could turn a single-SAN outage into a total access loss event. _ Take the bandwidth of the links into account. It is very common to have ISLs run faster than host ports, which obviously reduces the number of required ISLs.

7 Servers

Clusters are another means of achieving highly available SAN environments. Server clusters shift the focus from the hardware of a single server over to the application which must be cluster aware. One interesting implementation being more involves the use of server clusters in which the individual cluster nodes are not located in the same data center. While this approach addresses disaster recovery and business continuance concerns in a cost effective manner, it increases complexity in terms of management. In addition, the use of split clusters is somewhat distance limited and may require a significant work effort with tuning to account for the distance between the cluster nodes.

In general, host resources such as memory and processing time are used up by each disk volume that is mapped to the host. For each extra path, additional memory may be used and some portion of additional processing time is also required. The user may control this effect by using fewer larger volumes rather than lots of small volumes. However it may require tuning of queue depths and I/O buffers to support this efficiently. If a host does not have tunable parameters such as Windows, then it does not benefit from large volume sizes. Conversely, AIX will greatly benefit from larger volumes with a smaller number of volumes and paths presented to it.

For systems in high availability SAN environments, individual hosts will have multiple paths to their respective storage volumes. Without multipathing utilities, many operating systems would treat each path as a completely separate and unique resource even though the end storage resource was the same in all cases. Multi-pathing utilities/functionality are typically inserted into the IO path between the physical HBA's device driver and the operating system's kernel, or logical volume manager, to "hide" the details of multiple paths and only present one logical instance of each storage volume. Another purpose of these utility applications is to silently handle a path failure while normal IO operations continue.

However, multi-pathing can also experience the situation of "too much of a good thing" by having too many paths between a host and a storage resource. As the number of paths increases, the load on the CPU by the multi-pathing application can increase significantly. One misunderstood fact is that doubling the number of paths between a host and a given storage volume does not mean the total usable bandwidth has also doubled. This observation is due to the computing needs of multipathing algorithm as it makes use of the additional paths.

One condition which has been observed too many times within the SAN Central team is a lack of coordination between server and SAN/storage administrators concerning HBA maintenance. The historical experience has been a matter where the server administrator considers HBA firmware updates as a function of the SAN/storage administrator since the HBA is part of the SAN. Conversely, the SAN/storage administrator considers HBA maintenance activities to be in the purview of the server administrator since the HBA is located within the server. The key point and best practice guideline is that the administrators need to arrive at a mutually agreeable means where one, or both, are responsible for routine HBA maintenance actions.

8 Change Management

"Change Control" is one of the most discussed concepts in the management of Information Technology. There are entire international standards dedicated to defining it, and very expensive and complicated software created to implement it. However, in practice, change control often ends up being implemented in ways that do little to actually make the environment operate better.

Change control processes should be designed to make the execution of change go as smoothly and quickly as possible, not merely slow down change (through endless checks, verifications, and meetings) in an attempt to reduce risk. The delay imposed by inefficient change controls lead to rush jobs that actually increase net risk.

This is not to say that anybody should be able to implement any desired change at any time, but rather that the purpose of change control should be to help the appropriate administrators (with some level of review) make the right implementation decisions to begin with as opposed to imposing roadblocks that are ultimately treated as an inconvenience instead of a control.

It seems counter-intuitive that eliminating double and triple checking can actually *reduce* risk. However, for a given amount of manpower, the development and maintenance of standard operating procedures for common processes, major review of truly major changes, along with coherent planning for growth, is a far better use of time than attempting to review every change with a fine toothed comb. Based on the collective experience of SAN Central, some highly dynamic environment's use of double and triple checks have become double and triple rubber-stamping, because there is simply insufficient time to *truly* inspect every last "i" for dotting and "t" for crossing.

Will unplanned outages due to changes still occur? Yes; SAN administrators are human, and mistakes will happen. However change management should allow for the evaluation of risk versus benefit for the company with reasonable review efforts to prevent such mistakes. If administrators are given the correct tools to enable them to consistently make error-free changes, everybody wins.

The author of this white paper is not a management consultant. It is the role of the business to adequately evaluate how many layers of approval should be necessary for various types of changes within the SAN environment as well as provide the means to determine the appropriate administrator and/or team responsibility for change reviews. Another key concept of change management centers is effective communication to all personnel that may be impacted by the planned change, whether other administrators or end-user, of the upcoming change so they can evaluate the potential impact within their respective area of responsibility.

A change ticket, or record, should contain sufficient technical detail that any administrator skilled in the deployed products with some degree of familiarity with the SAN environment should be able to sit down at the appropriate console(s) and make the changes given the Standard Operating Procedures. These tickets should be available to the entire SAN administration staff.

In addition, a completed record should contain a complete list detailing how the changes were made. This can be as simple as copying a checklist out of the Standard Operating Procedures, checking off each step as it is completed, and pasting the completed list into the record. The total information on implementation should allow the change to be backed out later if needed.

For every common change, there should be a documented standard operating procedure (SOP) implementing the change. This procedure should contain sufficient detail that any administrator familiar with the systems and devices involved could make the change, and end up with a functionally equivalent configuration. These procedures enable the rapid training of new personnel, reduce errors in configuration changes, increase change consistency, and make documentation simpler and more accurate. This procedure can be integrated in with the checklist templates that will appear in the change ticket.

The SOP should never incorporate a detailed “cookbook” method. Optimal SOPs do not have to be extremely detailed step-by-step instructions written so any computer literate person can follow them. Instead, they should concentrate on the things unique to the SAN environment. For instance, the SOP for storage allocation (either for a new or existing server) need not include instructions on which buttons in the GUI to push or step-by-step listing of specific commands. Rather, the SOP should include information on how to decide on which storage system to be used and additional information to fulfill the request. The primary assumption of the SOP is the administrator is expected to know the basic steps for allocation (create a volume, define a host definition, map the volume to the host.

If the standard operating procedures are well-developed, and change requests are well-documented, an effective change management system will eliminate the need for multiple approval levels for common requests and thereby allow for better time utilization of administrators for unique requests.

To bring it all together, the following is an example of a SOP and change record. This example is for the addition of a new server to the switch configuration and allocating the new server with some SVC-based storage. It assumes that a previous request (following its own procedures) for physically cabling the server has already been completed.

Author’s Note: This example is provided with the sort of detail that is required. It has not been given a “test drive”, and is probably missing vital information for your business purposes. Actually developing a complete set of change procedures and documentation templates is beyond the scope of this white paper. Certainly, the example “procedure” for choosing an MDisk Group is inadequate for most complex SAN environments.

The easiest way to develop a Standard Operating Procedure is to integrate it with the ensuing checklist to implement the operation. The procedural notes that assist the Storage Administrator in developing the ticket are in *italics.* Those notes would obviously not be part of the ticket template, just the SOP document.

The fields that will change for each ticket are delimited by __double underscores__.

Sample Change Record

******* BEGIN RECORD*******

Request: __ABC456__
Purpose: Add new server __XYZ123__ to the SAN and allocate __200GB__ of space.
Date of Implementation: __04/01/2007__
Implementing Storage Administrator: __ SAN admin’s name __
Server Administrator: __ Server admin’s name __
Impact: None. This is a non-disruptive change.
Risk: Low.
Time estimate: __30 minutes__
Backout Plan: Reverse changes

Implementation Checklist:

1. ___ Verify (via phone or e-mail) that the administrator has installed all code levels listed on the website http://www.thecompany.com/storage_code.html
2. ___ Verify that the cabling change request, __CAB927__ has been completed.
3. ___ For each HBA in the server, update the switch configuration spreadsheet with the new server.
   To decide which ioGroup to Use: These should roughly be evenly distributed. Consult the storage inventory, which contains a running total of the number of servers assigned to each I/O group.
   To select which Node Ports to Use: If the last digit of the first WWPN for the host is odd, use ports 1 and 4; if even, 2 and 3.

HBA A:

Switch: __48000_1__	Port: __39__	WWPN: __00:11:22:33:44:55:66:77__	Port Name:__XYZ123_A__	Slot/Port: __5__	__Cluster 2, Group 1, Ports 1__

HBA B:

Switch: __48000_2__	Port: __39__	WWPN: __00:11:22:33:44:55:66:88__	Port Name:__XYZ123_B__	Slot/Port: __6__	__Cluster 2, Group 1, Ports 2__

4. ___ If this is the first change of the evening, collect a supportShow from the telnet console of both core switches, compress them and attach them to this ticket with the filenames of <ticket_number>_<switch name>_old.zip. If this is not the first change, record the ticket number that does contain the information here: __ __

5. ___ Add new zones to the zoning configuration using the standard naming convention and the information above.

6. ___ If this is the last ticket of the evening, collect supportShows from both core switches again and attach them with the filenames of <ticket_number>_<switch name>_new.zip. If it is not the last ticket, record the number of the ticket that contains this information here: __ __

7. ___ If this is an AIX host, have the server admin run cfgmgr to log the HBA into the SAN.

8. If this is the first change of the evening, log onto the SVC Console and:
   ___ Obtain a config dump and attach it to this ticket under the filename <ticket_number>_<cluster_name>_old.zip If this is not the first change of the evening, record the ticket that contains this information here: __ __
9. ___ Add the new host definition to the SVC using the information above and setting the host type to __the default__ Do **not** type in the WWPN. If it does not appear in the drop-down list, cancel the operation and retry. If it still does not appear, check zoning and perform other troubleshooting as necessary.
10. ___ Create new vDisk(s) with the following parameters:
   To decide on the mDisk Group: For current requests (as of 1/1/08) use DS8300_300_1, assuming that it has sufficient free space. If it does not have sufficient free space, contact John prior to submitting this change ticket and request an update to these procedures.
   Use Striped (instead of Sequential) vDisks for all requests, unless otherwise noted in the request.

Name: __XYZ123_1__	Size: __200GB__	ioGroup: __2__	mDisk Group: __DS8300_300_1__	Mode: __Striped__

11.___ Map the new vDisk to the Host

12.___ If this is the last ticket of the evening, obtain a config dump and attach it to this ticket under <ticket_number>_<cluster_name>_new.zip If this is not the last ticket, record the ticket where this information may be found here: __ __

13.___ Map the new volume(s) to the Host

14.___ Update the Host information spreadsheet using the above information, and the following supplemental data:

Ticket: __ABC456__	Project: __Foo__

15. ___ Also update the entry in the inventory database for the remaining free space in the extent pool and mDisk Group.

16. ___ Call the Server Administrator in the ticket header and request storage discovery. Ask them to obtain a path count to the new disk(s). If it is not 4, perform necessary troubleshooting as to why there are an incorrect number of paths.

17. ___ Request that the storage admin confirm R/W connectivity to the paths.

18. Make notes on anything unusual in the implementation here: ____

***** END RECORD *****

Note how the variables necessary to implement the change are not separated out from the checklist. This makes it more likely that the checklist will be used as the change is implemented, instead of just checked off in one fell swoop at the end of the change.

Also note how that was not very detailed (sometimes referred to as a cookbook approach). It does not say “click here and select option X”. It contains a basic list of the steps necessary and provides exactly the information required to execute the change; no more, no less.

The change record assumes that the administrator is familiar with the products, and knows where to locate the necessary documentation files that will require updating. If the change went badly and did not work, the information in this ticket and in the files attached to it would be invaluable information in determining what went wrong. If the administrator originally assigned to the change was sick on the big day, it would be trivial for another administrator to carry out the implementation.

There are varying practices in the industry as to when to schedule changes to an environment which range from once per quarter to an as-needed, almost daily, basis.

Key factors to consider are how dynamic is the business' SAN environment and the typical degree of urgency for changes.

The business management and SAN administrators will need to determine a schedule for routine changes (such as new storage allocations or re-zoning operations) deemed appropriate and prevents the deployment of massive numbers of changes being implemented in a single maintenance window.

For example: schedule all changes to take place on Saturday morning along with a requirement that all new requests must be submitted a full week prior to the window where they will be implemented. Such timing gives the appropriate administrator(s) sufficient time to develop and review the implementation plan for the change. If appropriate detailed checklists and plans are developed, reviewed and approved, then the change will be implemented.

One final change management related topic to be discussed involves proper documentation of the change and device configurations which were modified. In short, the configuration documentation for a solution should theoretically be sufficient to re-create a functionally equivalent copy of the solution, entirely from scratch. Change Management records should be detailed enough to "roll-back" the current configuration to any given earlier version. Moreover, the documentation should be detailed enough that any administrator skilled in the products used should be able to obtain a similar configuration result as somebody intimately familiar with the solution. Lastly, this information should be in a form easily read by the administrators and other technical personnel involved in the solution.

Why is it a best practice to keep these records? Some might argue that the configuration information for a storage solution is largely "self-documenting." After all, most storage devices have ways of obtaining a configuration dump. (Indeed, maintaining copies of these dumps is part of Best Practices for change tickets.) However, a report from the storage device on what the configuration *is* does not leave an Administrator with any information as to what it ***should be***. A mismatch between the records and what the device reports can be a useful starting point for troubleshooting (or better yet, preventing) errors.

Also, a configuration dump may not store the information in a way that makes it useful to the administrator. For instance, in a SAN environment with multiple storage systems, it would be very tedious to find which storage system has resources allocated for a given host. In addition, the configuration interface may completely lack business-related information that is needed for the configuration. The configuration interfaces will not keep track of which project a given storage allocation was intended for.

Lastly, there may be a great many things that need to be kept track of that a SAN device will never record, such as the name of the project/application that owns a server; contact names, in-service date, etc.

NOTE: For obvious reasons, none of this documentation should be stored on a SAN-attached server. (It is surprising how often this mistake is made.)

Code upgrades

By and large, the most stable SAN environments are ones that carry out major code upgrades every 12 months, and generally do not use a major code release (for instance, a 7.x -> 8.x upgrade) sooner than two months after it is released, unless there is a compelling reason to install the more recent version. One example of an exception to this guideline is a “field flash” containing a warning of a severe code bug. The purpose of regular code upgrades is to avoid code and/or firmware levels getting so far behind that vendors and IBM support are reluctant to examine issues, but not so far on the bleeding edge that potentially problematic code is introduced into the production environment.

Another significant best practice guideline concerns similar devices running with a consistent code version across multiple systems. The purpose for this particular guideline is to maintain consistency in the behavior of a device group in the solution, plus the fact that vendor testing rarely includes ***ANY*** testing and/or certification of environments where code levels are mixed. From a practical viewpoint some customer data centers may have a valid reason for mixing code versions, such as requirements for a given application, but these situations are extremely rare and should be avoided as much as possible.

Naturally, if a new level of code becomes available that has an especially desirable set of new features and/or fixes, there is no hard and fast requirement to wait for the next “cycle” to perform a code upgrade. Also, the importance of examining the readme file and/or release notes included with almost all software and firmware packages cannot be overstated. It is the author’s experience that this crucial step is often overlooked by administrators around the world. The release notes often disclose important dependencies of code on another device that can lead to catastrophic results if not followed.

The following is a list of general best practice recommendations for upgrades:

1) An environmental dependencies checklist should be created and used. As an example; if upgrading an HBA driver on a server, what are the implications for other server utilities and hardware (such as multipathing driver, HBA firmware, etc.) and will support be maintained attaching to another device. These types of questions must be answered prior to upgrading code on any of the devices.

2) IBM recommends that a proactive code maintenance schedule be put into place for all devices. While being sensitive to the fact that there is a business to be run the chances of running into “known issues” increases substantially when running significantly older code versions. This condition becomes more likely if the entire environment is not addressed in a systematic manner.

3) IBM Support recommends maintenance of the disk subsystem twice per year. Reviews for newer releases should be performed more frequently. Please note that many of the code updates may not be "bug" fixes, but still include improvements in error correction and diagnostics.

4) Before *any* “non-disruptive” or “concurrent” upgrade, it is *imperative* to verify that hosts have all paths to external disk storage available. If, for whatever reason, a particular host has lost a significant number of its storage connections, that host is likely to have major issues during an upgrade.

5) Perform regular inspections of the error logs from critical devices in the SAN environment. Open a trouble ticket with the appropriate support group on any suspicious pattern of events that cannot be unexplained.

6) When performing updates to switches within a high availability environment, wait a minimum of 1 to 2 days before making similar code upgrades in redundant fabrics.

To subscribe to information about regular updates to IBM products that you specify, IBM offers a free service called MyNotifications. This program was called “MySupport” in the past. Additional information about the program, how to register and then use this program can be viewed at:
ftp://ftp.software.ibm.com/systems/support/tools/mynotifications/overview.pdf

Once a profile of the topics and products of interest has been created, you will receive regular emails with links to any new or modified information which has been released since the previous update. These update emails are typically sent once a week, but the more topics and items of interest may cause the system to send more than one email per week. Each update item will have a very brief description as well as a link to browse directly to the information.

9 Monitoring

One of the most often overlooked things that need monitoring in a SAN is host path monitoring. A **very** common reason for server outages in general is missing paths that turn what should be non-disruptive maintenance (or failures) into a real problem. For example: For whatever reason, a server loses connectivity over one of its HBAs. The multi-pathing utility is designed to re-route traffic over to a different path (typically across the redundant fabric), and no symptoms are noticed on the host. If the original path meets certain criteria, the path will be flagged as dead and not usable until manual intervention by an administrator. Sometime later (it could be hours, days, or months); the fail-over path for this host fails for some reason such as a damaged fiber cable or routine maintenance. The host now experiences a full outage, which could have been prevented with routine path monitoring. Since paths may fail due to internal software errors, not all path loss will be discovered via SAN switch monitoring, or by dedicated management applications. A host-based system therefore is also a useful backup to monitoring of the SAN itself.

Server logs can be an important bellwether of storage problems, since after all it is the server that ultimately must process the data requests. There are many different kinds of storage problems that will show up only in server logs, and never in disk or fabric logs. For instance, a bad transmitter in a switch port will raise no alarm in the switch itself, as the switch is not capable of monitoring the integrity of data that it transmits; only the receiving port can perform that function.

Ideally, every error in a host error log needs to be understood, if not eliminated. Even “temp” disk errors can be important advance indicators of future problems. If an innocuous system error is repeated many times, measures should be taken to eliminate the error from occurring in the future.

When monitoring switches, ports of particular interest to monitor should be those for disk storage system and SVC connections. The setting of appropriate thresholds mandates that some degree of performance monitoring must take place over a period of time to develop meaningful thresholds that will generate true alerts and not numerous false alarms. As painful as it is to say “it depends”, there really are not any appropriate general guidelines or suggested threshold levels to determine when a connection and/or device is performing well or poorly. Since each network is unique, meaningful thresholds will need to be determined via a trial and error method.

One alternate method involves using TPC to collect performance metrics over a period of time, such as a full week, and then use the various peaks as a starting point for setting thresholds. A key proviso with this method is that SAN operations during the initial monitoring period must be normal without any notable disruptions or abnormal activities. Needless to say, performance metrics should be collected on a regular basis (such as once a month and no longer than quarterly) to better monitor changes in the SAN environment over time.

If necessary, thresholds may warrant modifications. However, a threshold should never be set much higher than 80% of a connection's bandwidth. Levels over this threshold level are an indication can either additional port-channel bandwidth are needed or that the workload on a particular device port may need to be reallocated to another port on the same device, or moving some of the workload to another device.

9.1 Performance and Metrics

The answer to some performance questions is the same two words; *it depends*. This is not much use to you when trying to solve storage performance problems, or rather perceived storage performance problems. But there are no absolutes with performance so it is truly difficult to supply a simple answer for the question: "what is a good performance number for a VDisk?"

The best rule of thumb metrics for any system are derived from current and historical data taken from specific configurations and workloads that are meeting application and end user requirements. Collect new sets of metrics after configuration changes are made to important storage resources. Keep a historical record of performance metrics.

It can be better to move users up the performance spectrum rather than down. Users will rarely complain if performance increases. So, if there is uncertainty as to the correct pool use the one with lower performance, and move them up to the higher performing pool later if required.

The key question of what needs to be monitored is also the most difficult question to answer. "It depends." There are a number of questions which have to be answered first.

- What monitoring tools are available?
- How much of the SAN environment is appropriate and/or would benefit from proactive monitoring?
- How often will monitoring be consulted and how will it be used?

Depending on the capability of available tools, the following items are a good starting point for proactive monitoring.

- Overall Data Rates and I/O Rates
- Backend I/O Rates and Data Rates
- Response Time and Backend Response Time
- Transfer Size and Backend Transfer Size
- Disk to Cache Transfer Rate
- Queue Time
- Overall Cache Hit Rates and Write Cache Delay
- Read-ahead and Dirty Write cache
- Write cache overflow, flush-through and write-through
- Port Data Rates and I/O Rates
- CPU Utilization
- Data Rates, I/O Rates, Response Time, Queue Time for
  - ➢ Port to Host,
  - ➢ Port to Disk
- Mirror Rates (while getting synchronized, data change rate, metro or global)
- Peak Read and Write Rates

A key point to understand is that while some or all of the above listed metrics may be routinely collected by a management application, the administrator must fully comprehend the meaning and how to interpret a given metric. If this skill is missing, then the value of performance data gathering is significantly reduced.

9.2 Tools

Management is one of the key issues behind the concept of infrastructure simplification. The ability to manage heterogeneous systems at different levels as though they were a fully-integrated infrastructure offering the system administrator a unified view of the entire SAN environment is a goal that many vendors and developers have been striving to achieve. SAN management tools can range from huge packages capable of monitoring and managing one or more fabrics to a collection of vendor-supplied management applications for specific products.

Regardless of which approach is deployed within a SAN environment, the tool(s) should also have the functionality which aids the administrator with diagnosis and troubleshooting. This paper does not attempt to advocate one approach or management package over another. However, certain key points of consideration will be discussed so the administrator has a better understanding when evaluating various tools.

As stated, some products will have management tools which are supplied by the product vendor. Many of these product-specific tools are free, while some are not. Using vendor-supplied tools may be cost effective, but the administrator should remember that a dedicated management tool for a limited product set is just that … limited. The management tool is very unlikely to be integrated with other management tools as well as being able to monitor or manage other SAN components from other vendors.

Another decision point is how the management tool(s) interfaces with SAN devices. Interface mechanisms include:

- Command line interface (CLI) via a Telnet or secure TCP/IP connection
- Web-based graphical user interface (GUI)
- Simple network management protocol (SNMP)
- Dedicated management server
- SNIA’s SMI open standard

Finally, the increasingly frequent adaptation of virtualization at various points within the SAN environment is only increasing the number of pain points for administrators as well as requiring broader range of functionality and features for the management applications. Being able to virtualize servers, storage (disk or tape) and fabrics plus much more will be a pain point for years to come.

10 Solution-centric

There are a number of best practice items that deserve some amount of coverage which do not readily lend themselves to one of the previous topic headings. Thus, this section is called solution-centric and not miscellaneous since these items can have an impact of overall SAN operations.

10.1 Security

Security has always been a major concern for networked systems administrators and users. Even for specialized networked infrastructures, such as SANs, special care has to be taken so that information does not get corrupted, either accidentally or deliberately, or fall into the wrong hands. And, we also need to make sure that at a fabric level the correct security is in place, for example, to make sure that a user does not inadvertently change the configuration incorrectly.

The SAN and its resources may be shared by many users and many departments. The SAN may be shared by different operating systems that have differing ideas as to who owns what storage. To protect the privacy and safeguard the storage, SAN vendors came up with a segmentation tool, zoning, to overcome this. The fabric itself would enforce the separation of data so that only those users intended to have access could communicate with the data they were supposed to.

Zoning, however, does not provide security by itself. For example, if data is being transmitted over a link it would be possible to “sniff” the link with an analyzer and steal the data. This is a vulnerability that becomes even more evident when the data itself has to travel outside of the data center, and over long distances. This will often involve transmission over networks that are owned by different carriers.

One approach to securing storage devices from hosts wishing to take over already assigned resources is logical unit number (LUN) masking. Every storage device offers its resources to the hosts by means of LUNs. For example, each partition in the storage server has its own LUN. If the host (server) wants to access the storage, it needs to request access to the LUN in the storage device. The purpose of LUN masking is to control access to the LUNs. The storage device itself accepts or rejects access requests from different hosts. The user defines which hosts can access which LUN by means of the storage device control program. Whenever the host accesses a particular LUN, the storage device will check its access list for that LUN, and it will allow or disallow access to the LUN.

Server-level access control is called persistent binding. Persistent binding uses configuration information stored on the server, and is implemented through the server's HBA driver. The process binds a server device name to a specific Fibre Channel storage volume or logical unit number (LUN), through a specific HBA and storage port WWN. Or, put in more technical terms, it is a host-centric way to direct an operating system to assign certain SCSI target IDs and LUNs. When zoning, LUN masking and persistent binding features are used in combination, the result is a more secure SAN.

SANs and their ability to make data highly available, need to be tempered by well thought out, and more importantly implementing, security policies that manage how devices interact within the SAN. It is essential that the SAN environment implements a number of safeguards to ensure data integrity, and to prevent unwanted access from unauthorized systems and users.

It is a well-known fact that "a chain is only as strong as its weakest link" and when talking about computer security, the same concept applies: there is no point in locking all the doors and then leaving a window open. A secure, networked infrastructure must protect information at many levels or layers, and have no single point of failure.

As true as it is in any IT environment, it is also true in a SAN environment that access to information, and to the configuration or management tools, must be restricted to only those people that are need to have access, and authorized to make changes. Any configuration or management software is typically protected with several levels of security, usually starting with a user ID and password that must be assigned appropriately to personnel based on their skill level and responsibility.

Whether at rest or in-flight, data security comprises of both data *confidentiality* and *integrity.* This is a security and integrity requirement aiming to guarantee that data from one application or system does not become overlaid, corrupted, or otherwise destroyed, whether intentionally or by accident, by other applications or systems. This may involve some form of authorization, and/or the ability to fence off one system's data from another system.

This has to be balanced with the requirement for the expansion of SANs to enterprise-wide environments, with a particular emphasis on multi-platform connectivity. The last thing that we want to do with security is to create SAN islands, as that would destroy the essence of the SAN. True cross-platform data sharing solutions, as opposed to data partitioning solutions, are also a requirement. Security and access control also need to be improved to guarantee data integrity.

Encryption is the translation of data into a secret code and is the most effective way to achieve data security. To read an encrypted file you must have access to a secret key, or password or passphrase that enables you to decrypt it. Unencrypted data is called plain text; encrypted data is referred to as cipher text. There are two main types of encryption: symmetric encryption (uses a single common key) and asymmetric encryption (also called private-public-key encryption).

However, there is still an issue when talking about public-key crypto-systems: when you initially receive someone's public key for the first time, how do you know that this individual is really who he or she claims to be? If “spoofing” someone's identity is so easy, how do you knowingly exchange public keys? The answer is to use a *digital certificate*. A digital certificate is a digital document issue by a trusted institution that vouches for the identity and key ownership of an individual—it guarantees authenticity and integrity.

On the LAN side of a SAN environment, IP security can be problematic. To address this concern, a number of protocols have been developed. First, the Simple Network Management Protocol (SNMP) was extended for security functions to SNMPv3. The SNMPv3 specifications were approved by the Internet Engineering Steering Group (IESG) as a full Internet standard in March 2002.

IP security (IPSec) uses cryptographic techniques obtaining management data that can flow through an encrypted tunnel. Encryption makes sure that only the intended recipient can make use of it (RFC 2401). IPSec is widely used to implement Virtual Private Networks (VPN).

Other cryptographic protocols for network management are Secure Shell (SSH) and Transport Layer Security (TLS, RFC 2246). TLS was formerly known as Secure Sockets Layer (SSL). They help ensure secure remote login and other network services over insecure networks.

A common method to build trusted areas in IP networks is the use of firewalls. A firewall is an agent that screens network traffic and blocks traffic it believes to be inappropriate or dangerous. You will use a firewall to filter out addresses and protocols you do not want to pass into your LAN. A firewall will protect the switches connected to the management LAN, and allows only traffic from the management stations and certain protocols that you define.

At a high level, some of the security best practices include the following:

- Default configurations and passwords should be changed.
- Configuration changes should be checked and double checked to ensure that only the data that is supposed to be accessed can be accessed.
- Management of devices usually takes a “telnet” form—with encrypted management protocols being used.
- Remote access often relies on unsecured networks. Make sure that the network is secure and that some form or protection is in place to guarantee only those with the correct authority are allowed to connect.
- Make sure that the operating systems that are connected are as secure as they ought to be, and if the operating systems are connected to an internal and external LAN, that this cannot be exploited. Access may be gotten by exploiting loose configurations.
- Assign the correct roles to administrators.
- Ensure the devices are in physically secure locations.

Make sure the passwords are changed if the administrator leaves. Also ensure they are changed on a regular basis. Finally, the SAN security strategy in its entirety must be periodically addressed as the SAN infrastructure develops, and as new technologies emerge and are introduced into the environment.

10.2 Consistent clock settings

All devices within a SAN environment which support network time protocol should have this feature enabled and refer to the same time server. The end result is that time-stamped data and logs from different devices will be consistent (a second or so at most) which will significant help administrators and support personnel cross reference data from multiple devices. Having consistent clock settings will ultimately save time during outages as well as assist during routine maintenance activities.

10.3 Interoperability

A SAN is a simple thing, a path from a server to a common storage resource. So, where do the complexity and interoperability concerns originate? The first key point to remember is that interoperability is based on the lowest common denominator for features, functionality and available services.

Limited budgets and too few skilled people have pushed many organizations into looking for short term solutions. When a new application or project appears, the easy, inexpensive option is to simply introduce another server, whether it is a physical or virtualized server is an implementation item. Now add storage to the sprawl. Every server has two or four fiber Host Bus Adapters (HBAs) and a share of the consolidated storage.

To make things more difficult, most SAN environments have servers from multiple vendors, with decisions made on cost, suitability to a specific application, or merely some administrator's personal preference. Different vendor's servers were tested on very specific SAN configurations. Every server vendor has their own interoperability matrix or list of SAN configurations that the vendor has tested, and as a little understood fine point, the only configuration(s) which a particular vendor supports. Further complicating matters, the configuration lists account for specifics such as:

- Operating system level
- Specific HBAs and the drivers for these HBAs
- Multipathing software
- Types and models of SAN switches
- Code level for these switches
- Storage subsystems and the microcode level that was tested

Because the SAN and storage are common to all servers, therefore, the interoperability matrix for every server then has to be checked out and compared before any change is made to the storage network, or even to individual servers.

A reasonably complex SAN could have servers from Sun™, IBM, HP, and Dell and storage subsystems from the same or other vendors. There is no guarantee that SUN, Dell, and IBM support the same level of SAN or disk microcode. Changes, upgrades, or new implementations can be delayed for months, while waiting for the different vendors to support a common code stream or driver. A best practices rule for interoperability in SANs is that it is the manufacturer of the storage device that defines interoperability. They do the testing that produces a matrix of supported configurations, from the storage device to different types of server.

There is an additional complication with certain types of applications, such as backup solutions. In these cases, the application vendor will also produce a matrix of tested, supported hardware.

10.4 Separation of Production and Test Segments

A test environment is useful in many instances. A key factor for this testbed is that it should contain at least one of the production environment's critical devices. It is impossible for any hardware vendor to test every combination of hardware and software, much less extend test scenarios to include application software. It can be very helpful to use the testbed to check new code releases against applications known by the staff to be particularly troublesome. The testbed can be as small as a single server with a pair of HBAs mapped to some old, slow, SAN storage and a cast-off switch. Such a setup would be adequate to perform some internal tests on a new failover driver.

Conversely, the test environment can be as elaborate as a miniature recreation of the whole SAN system, including a dedicated storage system mapped from each brand and/or model of installed storage controller, similar, albeit smaller, versions of the switches used in the production environment, and servers "beefy" enough to run application simulations. Such a setup can be used for application testing and development, as well as general testing. Many shops give their internal application development teams such setups, and use it as a "sandbox" for planned changes to their production environments.

Why is such a test environment necessary? There are many obvious answers to this question. One such example is multi-path software. Multipath software is pivotal to ensuring servers maintain redundant paths to storage. Without properly functioning multipathing, the dual physical SAN infrastructure is somewhat less useful. IBM designs and performs test plans to ensure that during a given path failure, no errors are reported by the operating system. However, the failover code stream may take a few moments to verify that a path is truly down by attempting some number of IO retries before failing the IO stream to a different path. For this reason, it is not at all uncommon for some applications to suffer from IO timeouts before any OS timer "fires" and issues appropriate error messages. Certainly some application testing takes place, but it is simply impossible for IBM to test all scenarios based on hardware and application family.

Appendix A - References and additional materials

The following listing has been used in the development of this paper. These publications and materials were not the sole reference source used for this paper. They are not presented in any particular order. Some of these references may have multiple revisions, and the author has strived to utilize the most recent version. Additional information and material used to create this paper is based on IBM trouble tickets (PMRs) handled by SAN Central and various IBM PFE support teams.

IBM Redbooks. SG24-5470. Introduction to Storage Area Networks.

IBM Redbooks. SG24-5250. IBM System Storage Solutions Handbook.

IBM Redbooks. SG24-6116. Implementing an IBM/Brocade SAN.

IBM Redbooks. SG24-7545. Implementing an IBM/Cisco SAN

IBM Redbooks. SG24-6384. IBM TotalStorage: SAN Product, Design, and Optimization Guide

IBM Redbooks. SG24-7114. Introduction to Storage Infrastructure Simplification

IBM Redbooks. SG24-6419. Designing and Optimizing an IBM Storage Area Network

IBM Redbooks. SG24-7543. IBM/Cisco Multiprotocol Routing: An Introduction and Implementation

IBM Redbooks. SG24-6363. IBM Midrange System Storage Implementation and Best Practices Guide

IBM Redbooks. SG24-7659. IBM XIV Storage System: Architecture, Implementation, and Usage

IBM Redbooks. SG24-7904. IBM XIV Storage System: Architecture, Implementation, and Usage

IBM Redbooks. SG24-8786. IBM System Storage DS8700 Architecture and Implementation

IBM Redbooks. SG24-7146. IBM TotalStorage DS8000 Series: Performance Monitoring and Tuning

IBM Redbooks. SG24-7521. SAN Volume Controller: Best Practices and Performance Guidelines

Brocade Data Center Best Practices Guide. GA-BP-300-00. Fabric Resiliency Best Practices

Brocade Data Center Best Practices Guide. GA-BP-329-00. SAN Design and Best Practices

Brocade Data Center Fabric Best Practices Guide. GA-BP-036-02. Cabling the Data Center

Brocade Publications. 53-1000672-03. Performance User Manual

Brocade Publications. 53-0000350-04. LAN Guidelines For Brocade SilkWorm Switches.

Brocade Publications. 53-0001573-01. Building and Scaling Brocade SAN Fabrics.

Brocade Publications. GA-RG-250-00. SAN Security: A Best Practices Guide.

Brocade White Paper Series. Secure SAN Zoning Best Practices.

Cisco Publications. OL-12518-01. Data Center High Availability Clusters Design Guide

Cisco Publications. OL-14856-01. Cisco MDS 9000 Family Cookbook for Cisco MDS SAN-OS Release 3.1

Cisco Publication. OL-23396-01. Cisco MDS 9000 Family NX-OS Fabric Configuration Guide

Cisco Systems White Papers. C11-330664-00. Intelligent Traffic Services with the Cisco MDS 9000 Family Modules

Cisco Systems White Papers. ETMG 203154—LSK. Scalable Fabric Design — Oversubscription And Density Best Practices.

EMC Engineering White Paper. EMC CLARiiON Best Practices for Fibre Channel Storage.

HP Publications. AA-RW86D-TE. HP StorageWorks: SAN Design Reference Guide

(This page is intentionally blank)