

# **Virtualization Storage Deep Dive for the Controls Engineer (with a slight VMWare slant)**

**Andrew Robinson**

**Avid Solutions**

## **Before we begin.... a few thank you's**

First I'd like to thank Brent Humphreys of Stone Technologies for his technical review. I met Brent a number of years ago at a Wonderware conference after seeing his talk about virtualizing what was then Industrial Application Server. After chatting with him for just a few minutes I quickly figured out we had a pretty similar outlook on many things. I was also very impressed by how willing he was to share his knowledge, even if I worked for a competitor. We've stayed in touch over the years and he was the first person I thought of when I needed a technical reviewer for this work.

The second person I'd like to thank is Shannon Ford. Shannon joined our organization in the last couple years and she has done an amazing job with all things marketing and social media. She is also a fantastic copy editor. Let's just put it this way. If I had submitted my draft to a college professor and they bled on it as much as Shannon did the professor would have quit 1/3<sup>rd</sup> of the way through and just given me an F. Thankfully Shannon powered through my engineerish and helped me pull together something respectable.

## Introduction

The typical process control engineer has characteristically been quite slow to adopt new technologies from the classic IT space. For example, how many engineers are still running critical systems on Windows XP and Server 2003? How many of you have stopped to think how long Windows 7 and Windows Server 2008 have been out and proven in use? Are there some inevitable bumps in the road as you transition? The answer is obviously yes. However, one must weigh the pain and impact of these road bumps versus the inevitable issues involved in running on unsupported platforms. Now that most people have finally starting cutting over to Windows 7 and Server 2008, virtualization is the next frontier that the pioneers appear to be approaching. However, what I find most interesting is how far ahead most vendors are in front of their customers. Many of the major vendors are now fully supporting virtualized environments long before customers are actually considering moving in that direction. Below is a summary of some of the major vendor stances listed as of May 2012

Product	Virtualization Support (VMWare)
Invensys System Platform	Yes
Rockwell Software	Yes
Emerson DeltaV	Yes
Honeywell Experion	Yes
ABB 800XA	Yes

Table 1. VMWare Support by Vendor

In summary, for most customers ready to move to a virtual infrastructure, you are likely to find the support needed from your vendor.

## Critical Choices

When creating your ideal virtual environment, there are a handful of critical choices to make including the core software, compute, and shared storage. For the core software, the major players are VMWare VSphere, Microsoft HyperV, and Citrix Xenserver. While the majority of my personal experience is with VMWare, the choice of hypervisor is incidental with respect to storage considerations at a high level. The hypervisor aggregates CPU, memory, and network connectivity, allowing these resources to be shared between multiple virtual machines on a single physical host. Due to the high-availability options existing in modern hypervisors, the ability to add a new node and move workloads with no disruption makes the choice of an actual physical hosting platform less critical. If additional memory or new NICs are needed, you can easily transfer workloads, shutdown the server, add physical resources, and power back up; all with no interruption to your environment. The final critical choice is storage. This is typically the component that people understand the least. The primary reason for lack of understanding is the fact that outside of the traditional large data centers, shared storage has rarely been seen or used. If a common file storage environment is needed, a user typically installs extra hard drives in a server and shares a drive or a folder. The next question asked may be "Why shared storage?" Whereas most hypervisors can utilize the local storage on a server to run virtual machines, the user typically gives up almost all high availability and portability functions if the machine data is stored on a local machine as opposed to on some type of shared storage. While there are some software solutions that can expose

local storage to make it look like shared storage, this work-around is more of a niche product as opposed to mainstream shared storage.

A late breaking piece of information that changes this scenario slightly is the introduction of “shared nothing” migration. The current releases of most hypervisors require virtual machine data to be stored on shared network storage. However, with the upcoming releases of Microsoft Hyper-V 2012 and VSphere 5.1 both vendors will introduce technologies that allow migration of machines between servers without having access to common shared storage. This solution does not allow for more advanced high-availability scenarios but it is a major step in the right direction.

## Connections

Once you have decided to virtualize your environment by accepting the fact that you need shared storage, you have also made the decision that you need high availability features. The first major choice involves connectivity to the device. Commonly, shared storage is more accurately referred to as network storage. Do not assume that network means ethernet. The first type of network shared storage used in years past was a technology called Fibre Channel. Fibre Channel involves a proprietary protocol (as opposed to TCP/IP) transported over a fibre optic cable. Like traditional ethernet, Fibre Channel requires a specialized adapter in the host along with switches to aggregate connectivity. The second major technology choice, ethernet, is much more familiar to the everyday controls engineer. Using ethernet involves standard network interface cards located in the server, standard ethernet switches, and copper cabling for up to 1GB/s speeds. For many years, there were three primary reasons for choosing Fibre Channel. First, Fibre Channel was typically always faster than any competing ethernet protocol. The most common top end speed of Fibre Channel today is 8 GB/s, with 16 GB/s becoming more common. Second, because Fibre Channel utilizes a proprietary protocol that does not contain the overhead of TCP/IP, the communications latency is extremely low, typically < 1 ms. Finally, in the not so distant past, almost all storage arrays of decent quality implemented Fibre Channel as their interface of choice. While these are all compelling reasons to choose Fibre Channel, modern implementations of ethernet protocols are eliminating these advantages. First, the speed advantage has effectively been nullified. Ethernet at 10GB/s has been in common use since 2010. In addition, 40GBe and 100Gbe will be available in the near future. Specialized ethernet switches can help you obtain ultra-low latency connections within your ethernet fabric. Finally, as more and more vendors consider the rise of ethernet protocols, you can typically connect to all but the most high-end network storage devices with ethernet protocols.

Another modern optimization is the use of specialized ethernet cards with TOE (TCP Offload Engines). From the Wikipedia article on TOE:

*TCP offload engine or TOE is a technology used in network interface cards (NIC) to offload processing of the entire TCP/IP stack to the network controller. It is primarily used with high-speed network interfaces, such as gigabit Ethernet and 10 Gigabit Ethernet, where processing overhead of the network stack becomes significant.*

In previous computing environments, the CPU was typically so oversized that the additional overhead of processing the TCP/IP stack was of little to no concern. However, with our attempts to push utilization to the 60-70% level, a 3-5% reduction in overhead can be significant.

Today, it is atypical to see a new installation utilize fibre channel. Instead, most new small and mid-sized environments are choosing ethernet protocols. The primary rationale is that you already have most of the expertise needed to implement and maintain an ethernet network. Also, of special note is the need to separate your storage and compute networks, a concept discussed in further detail later

## Protocols

After you have chosen a connection method, the next step is to select the protocol. If you have chosen Fibre Channel as your method of connectivity, then you have no choice with respect to protocol. However, if you have chosen ethernet as your transport mechanism then you have essentially two choices, NFS and ISCSI. In an effort to be complete, Fibre Channel over Ethernet is a third choice but that would just confuse the conversation at this point. Another popular term used when discussing NFS vs. ISCSI is File vs. Block. Understanding this difference will help you understand the difference in the protocols.

Block Storage Options (SAN)	File Based Storage Options (NAS)
iSCSI	NFS
Fibre Channel	CIFS
AoE (ATA over Ethernet)	

Table 2. Block Storage Options

The following discussions of SAN vs. NAS are from Wikipedia Articles:

### **SAN**

*A storage area network (SAN) is a dedicated network that provides access to consolidated, block level data storage. SANs are primarily used to make storage devices, such as disk arrays, tape libraries, and optical jukeboxes, accessible to servers so that the devices appear like locally attached devices to the operating system. A SAN typically has its own network of storage devices that are generally not accessible through the local area network by other devices. The cost and complexity of SANs dropped in the early 2000s to levels allowing wider adoption across both enterprise and small to medium sized business environments.*

*A SAN does not provide file abstraction, only block-level operations. However, file systems built on top of SANs do provide file-level access, and are known as SAN filesystems or shared disk file systems.*

### **NAS**

*Network-attached storage (NAS), in contrast to SAN, uses file-based protocols such as NFS or SMB/CIFS where it is clear that the storage is remote, and computers request a portion of an abstract file rather than a disk block.*

NFS stands for Network File System. It was originally created by Sun Microsystems as a way to allow multiple clients to access files on a central network storage device. When the hypervisor accesses data on an NFS share, it accesses the files directly because the protocol itself provides the file system, hence the term File protocol. iSCSI, also known as internet SCSI, essentially takes standard SCSI disk commands and instead of executing them over a local SCSI connection, the commands are encapsulated in TCP/IP packets and transmitted over ethernet. These low level commands do not know how to directly interact with files. Instead, they interact with arbitrary blocks of data on disk. The system that is reading and writing data implements a file system on top of the iSCSI share, or LUN (logical unit number), to be able to read and write data for files. In the case of vSphere, this file system is called VMFS. Hyper-V also implements a file system on top of iSCSI LUN's. This allows the Hypervisor to implement a file system (like VMFS) that is optimized for the IO needs of the hypervisor.

Once put into operation, both protocols are capable of perfectly acceptable performance in a small to medium-sized environment. The primary differences come with respect to setup and scalability. For NFS, there are essentially three steps involved in setting up a datastore. First, create an NFS export on your storage device. Second, from vSphere create a special network port over which you will connect to your NFS datastore. Finally, connect vSphere to the NFS export and create a new datastore. Once connected, the user can immediately begin to store virtual machines on the newly created datastore. For iSCSI, there are a few additional steps. First, create an LUN on your storage device. Second, from vSphere create a special network connection for iSCSI data. Next, add your storage device as an iSCSI target. After adding the device, perform a rescan of iSCSI targets. At this point, the user should see the LUN created. Select the LUN and create a new datastore. Once the datastore is created, format the datastore with VMFS. Only after the datastore is formatted can you begin to store data on the datastore. Although it seems as if setting up an iSCSI datastore is more complex, remember this is typically a one-time activity that only takes a few extra minutes than configuring an NFS datastore.

One final consideration is the optimizations present in the VMFS file system utilized by VMWare when implementing block storage. While NFS was designed from the ground up as a multi-user protocol, it was never designed with the intent of handling really large virtual machine disk files. VMFS, on the other hand, has always been and will continue to be designed to handle extremely large files with simultaneous multi-host access.

Below are tables summarizing high level Pros and Cons for NFS and SAN protocols.

<b>Pros</b>	<b>Cons</b>
Tends to be less expensive	Higher CPU Overhead
Simpler configuration	
Simpler management	

Table 3. NFS Pros and Cons

Pros	Cons
Higher performance	Tends to be more expensive
Ability to offload protocol to hardware components.	More complex configuration
Allows for hypervisor to utilize specialized file system	

Table 4. SAN Pros and Cons

Another major difference of any substance is the scalability of the protocols. While you can connect multiple network cables to an NFS controller, only one of them can be used at a time between a computer host and the storage device. This is an inherent limitation in the protocol. On higher end storage arrays, you can purchase specialized software that overcomes this limitation, but those high end units are not the focus of this study. The primary purpose of using multiple connections to an NFS array is for redundancy, not throughput. ISCSI, on the other hand, can use as many network connections as you configure between the computer host and the network device. Once again, this is a feature of the protocol that allows for this functionality. The real question is determining if this difference actually matters. Based on my own personal experiences and a large amount of study on the subject by other experts, a single host rarely saturates a typical 1 GB/s link between the host and storage unit. Only in extreme corner cases when the user is running a specialized test, do you actually see saturation of the 1 GB/s link.

Other considerations for VSphere users are a couple issues that plagued older, pre 5.0 versions of the product. First, the maximum VMFS datastore size was 2 TB – 512 bytes. While 2 TB is quite a large datastore, once a user begins configuring multiple database servers or making multiple copies of these machines for development purposes, size limitations occur. As a result, some users have created multiple, smaller datastores. This, in turn, can lead to greater maintenance and overall cost of ownership with respect to OPEX (operational expenditures, i.e. day to day cost). With the latest release, 5.0 and later, this limit is now 64TB. I do not foresee any of my colleagues approaching this limit anytime soon. Second, with larger datastores and more specifically numerous virtual machines on a datastore, one may experience an issue called SCSI locking. Simply explained, any time VSphere reads or writes data, it would momentarily lock access to the entire datastore while performing the modification. Even though this would happen extremely fast, if you had hundreds of virtual machines on a datastore, there was a risk that the storage system performance would be negatively impacted due to these locks stacking up and restricting access to the underlying data. With VSphere 5.0, this method has been dramatically reworked, and SCSI locking is no longer an issue on even the largest datastores. Most experts in the field recommend ISCSI users to create datastores as large as possible, only splitting them for administrative or security reasons, not performance or size.

Considering most typical environments that a controls engineer could work with, both NFS and ISCSI should perform equally well. While I personally prefer ISCSI, a quality NFS array can perform just as well under all but the most extreme conditions. The one, subtle, reason that can push one towards ISCSI is

VMware's tendency to publish features for block protocols (iSCSI and FC) before NFS. Usually, the NFS features lag by approximately one release.

## **Features**

Once connectivity methods and protocols have been determined, the user should create a system with key features when you are seeking for a storage array.

## **Controllers**

All disk systems have at least one disk controller through which read/write requests are passed. Storage arrays are no different. The primary difference between storage arrays and what you see in standard desktops and servers is the quality and quantity of controllers. In a storage array, the controllers are designed for extremely high throughput. Remember that these units are designed with the consideration of handling the disk I/O for numerous (10+) computer hosts simultaneously. For this reason, these specialized controllers usually contain more RAM and a faster chipset than what is found in a typical RAID controller in a standalone server. Also, consider the quantity of controllers. If this array is installed in a production environment, anything less than redundant controllers is unacceptable. There are two reasons for this. First, if a controller should fail, the backup controller should resume processing activities without any system interruption. Depending on the vendor's chosen architecture, Active-Active vs. Active-Passive, you may experience a slowdown in performance. An Active-Active array is always using both controllers to process workload. It stands to reason that if a controller is lost, capacity will be cut in half. In an Active-Passive arrangement, only one controller at a time is handling 100% of the workload. In the event of a failure, no performance difference should be experienced due to only utilizing a single controller. Given the performance requirements of a typical environment, this may not be much of a distinguishing factor. In addition, having multiple controllers is only useful if both units are hot swappable in the event of a failure. Another nice feature, although not necessarily required, is the ability to upgrade firmware at any time by failing back and forth between controllers.

Another feature to pay special attention to is the NVRAM or battery backed CACHE on the controllers. If the array is in the middle of writing a block of data and the power suddenly disappears, there is a chance the data on the array could be corrupted due to an incomplete write or modification of data. In this event, the data is stored in NVRAM or the battery backed CACHE until power is restored. Depending on numerous different factors, this downtime can be as long as a week or two before potential corruption becomes an issue. This is a feature that quality arrays will implement as a matter of standard configuration. If this is offered as an option, it is a red flag that this array may not be a unit that is suitable for your demanding environment.

Downtime is bad, but corrupted data is unforgivable in a manufacturing environment.

Finally, take a look at the number of network ports on each controller. At a minimum, there should be two network ports for data and one network port for a maintenance interface. Having a separate port for maintenance allows routing to your production network for configuration and maintenance while

leaving the actual storage data on separate ports on a separate network. See the diagram (Figure 1) on proper segmentation.

### **Expansion**

When considering an array, one should not only consider the amount of storage in the base unit, but also whether or not the unit can be expanded with additional “shelves” or “enclosures.” Typically, these units take the form of a 2U device with nothing but hard drives and interconnect. These enclosures are connected via an external SAS cable from the controllers in the first enclosure. On each of these shelves, there are usually IN and OUT connections, allowing more shelves to be added in a daisy chain fashion. If implemented properly, these new enclosures can be added without any disruption to the running array. It is not atypical for a modest array to support as many as 48 to 96 hard drives on a single set of controllers.

### **Online Maintenance**

Just the same as modern DCS systems are engineered to run for years without downtime, the same can be said for quality arrays. Downtime can typically have two sources. One source of downtime is component malfunction. This is mitigated via component level redundancy and careful design of piece parts to maximize MTBF. The second, avoidable source of downtime is system upgrades and modifications. A quality array will allow for the creation, maintenance, and resizing of disk volumes with no downtime on the system or the volume. Taken to an extreme, one unit I have purchased actually allows you to relocate hard drives extemporaneously with no downtime. Also, as mentioned before, a quality array will allow for firmware updates with no downtime.

### **Software Features**

The list of software features available in the modern array is astounding. Some of the major features include compression, deduplication, snapshots, and replication. Compression is a simple concept that most can easily understand. Compression works in a similar manner to creating zip files on your computer and unzipping the file when you need to access the files. Deduplication is a little more complex. It is a process that involves the system looking at each block of data being stored and determining if an exact copy already exists. These blocks may be 1 MB, 128 KB, or some variable value. If a copy exists, then the system simply stores a pointer to the existing block instead of storing the data a second time. Typical deduplication ratios in a relatively homogenous environment (i.e. lots of Windows installs) are approximately 5x to 10x. Therefore, what previously took 5 TB to store now only takes 1 TB. Snapshots are essentially a backup method that takes place on the array itself instead of using some agent inside the machine. While these are very useful and efficient, some users may find they are slightly more difficult to work with as opposed to a typical virtual machine backup software package such as Veeam or PHDVirtual. Finally, some software provides the ability to perform near real time replication. Provided you have the budget, this is a spectacular method for ensuring business continuity in the event of a physical disaster taking out your primary array. However, one must be careful of how much trust placed in this method. Imagine if you corrupt a file or entire database, and that activity gets

replicated. This will create two corrupted files or databases. Some replication schemes do allow a rollback to a specified point in time to deal with this particular kind of situation.

## Performance

The final and easily most important item to consider when purchasing an array is performance. Array performance takes two major forms; IOPS and throughput. IOPS is measured in total read/write operations per second. Throughput is typically measured in MB/second. While both are important measures, the primary limiting factor in most environments is IOPS. Whenever data is written to or read from the disk this is considered an IO operation. With an understanding of the basics of IOPS, it is important to understand what controls IOPS in a storage array. There are three major factors under the user's control that influence IOPS. The first is disk speed. The faster the speed of the underlying disk, the more IO operations a particular disk can support. Technically seek and rotational latency are factors as well, but for simplicity sake, we will focus on disk speed. Second, the total number of drives, commonly referred to as spindles, in a volume (aggregated set of disks with a particular capacity) can influence IOPS. The more spindles in a volume, the more IOPS it can support. Using multiple slower disks can sometimes provide better performance than fewer fast disks. Finally, the RAID configuration of the volume can have a substantial effect on the IOPS performance. The easiest way to see the effect of each is to calculate the average IOPS for a particular disk arrangement while adjusting different parameters to see the effect.

Speed (RPM)	Raid Level	Number of Disks	%Reads/%Writes (8 KB chunks)	Per Disk IOPS	Total IOPS
<i>Adjusting Raid Level</i>					
7200	0	6	25%/75%	76	442
7200	1	6	25%/75%	76	252
7200	5/50	6 (5+1)	25%/75%	76	136
7200	6/60	6 (4+2)	25%/75%	76	93
<i>Adjusting Disk Speed</i>					
7200	5/50	6 (5+1)	25%/75%	76	136
10000	5/50	6 (5+1)	25%/75%	83	149
15000	5/50	6 (5+1)	25%/75%	90	162
<i>Adjusting Spindle Count</i>					
7200	5/50	6 (5+1)	25%/75%	76	136
7200	5/50	7 (6+1)	25%/75%	76	159
7200	5/50	8 (7+1)	25%/75%	76	181
7200	5/50	9 (8+1)	25%/75%	76	204
<i>Adjusting Read/Write Balance(Not realistically under your control)</i>					
7200	5/50	9 (8+1)	0%/100%	76	165
7200	5/50	9 (8+1)	25%/75%	76	204
7200	5/50	9 (8+1)	25%/75%	76	265
7200	5/50	9 (8+1)	75%/25%	76	379
7200	5/50	9 (8+1)	100%/0%	76	662

Table 5. Disk Performance

*\*\* Thanks to <http://www.wmarow.com/strcalc/strcalc.html> for providing the calculation engine for these values.*

In my opinion, the least understood, yet most critical finding in this table is the influence of RAID configuration on the performance of the system. The reasons are beyond the scope of this study, but consider when a write to a RAID volume occurs, the system must not only split the write across multiple disks, but also calculate parity bits to be written to the balance of disks in the volume. The more parity bits required to be written, the more severe the hit on write performance becomes. Also of interest is the dramatic range, almost 4x, as the R/W mix goes from all writes to all reads. Refer back to the previous discussion on parity calculation for why writes are so much more taxing than reads.

While the Read/Write Balance is not specifically in the users control, you can use this information to your advantage when working with your array provider to determine sizing requirements. For instance, if you measure disk performance on a typical Application Object Server on a Wonderware System Platform environment, it is almost 100% writes. Historians will typically have a high percentage of writes, but also consider how many clients may be running trends at the same time. Instead of guessing what this mix and the performance might look like, it is always best to measure actual data. Thankfully, included in all Microsoft operating systems is a tool called Perfmon. The details of running Perfmon are beyond the scope of this study. When you do run Perfmon you should collect all statistics on all volumes, as well as all physical disks. I typically run these metrics at one minute intervals for 24 hours. This should account for daily activities and daily backups.

From a real world case study, here are results that can be used for proper sizing considerations. Across multiple Application Object Servers, we found an average of approximately 130 write operations per second, essentially 100% writes. This type of activity is almost exclusively the result of the application engines writing checkpoint files and historical store/forward data to protect against engine failures. On a side note, the system originally distributed its entire load across three machines instead of five. When the system only had three machines, the check pointing was slowed to once every five seconds because the machines simply could not keep up. At first, it was speculated to be a shortage of RAM and CPU. However, studying disk statistics revealed the bottleneck was the disk subsystem. A pair of 2.5" 10K RPM drives in a RAID 1 configuration was installed in each machine. According to the online calculator, this configuration was capable of supporting 140 IOPS. A quick check of the math yields

$$(130 \text{ IOPS} * 5 \text{ new machines}) / (3 \text{ old machines}) = 217 \text{ IOPS/old machine}$$

Reviewing this data makes it obvious why the systems had trouble with check pointing. So, the question is how to solve this issue without purchasing new machines. According to the calculations, the user could install two new hard drives and increase the capacity to 280 IOPS in a RAID 1 configuration. If we wanted equivalent performance in a RAID 5 configuration, that would require an exorbitant total of eight drives. On a contrasting side note, terminal servers in this environment averaged around 20 IOPS, 95% writes.

If there is one takeaway from this study of performance, it should be that performance is your primary nemesis when specifying a new array for your environment. The uneducated specifier will typically look at an array and see a large amount of capacity for a relatively cheap price and think “what a great deal.” With the cost of 2TB and 3TB drives continuing to plummet, this tendency will only increase in the future. What this reaction does not take into consideration is that 20 TB of usable space is great, but if you cannot push the IO through to the disks, all of that capacity is essentially useless. It is for this exact reason that you will typically see greater quality servers with disk capacity choices like 146, 300, and 600 GB. The manufacturers realize the user will typically run out of IOPS before GB’s, so the user is forced to purchase more disks/spindles than otherwise needed for a typical desktop.

**Performance Addendum (Solid State Disks)**

When shopping for arrays, you will encounter those using solid state disks. In the past, these devices have typically been specified to fight vibration and shock issues in industrial PC’s. While this is a side benefit, it is a minor factor in the inclusion of SSD’s in today’s array. The primary reason for inclusion of SSD’s in newer arrays is performance. Take the example of our sluggish disk above that could generate about 75-100 IOPS. Compare this to a \$200 SSD drive found on Amazon.Com in May 2012. Below are the key stats from the website.

Intel Solid-State Drive 330 Series 180GB SATA 6 Gbps 2.5-inch SSD - SSDSC2CT180A3K5
180 GB Raw Capacity
SATA 6Gb/s (Compatible with SATA 3Gb/s)
Sequential Reads Up To 500MB/s
Sequential Writes Up To 450MB/s
Random 4 KB Reads Up To 42,000 IOPS
Random 4 KB Writes Up To 52,000 IOPS
Form Factor: 2.5 inch
Minimum Useful Life/Endurance Rating: The SSD will have a minimum of three years of useful life under typical client workloads with up to 20 GB of host writes per day.

Table 6. SSD Performance Summary

Although, this is not an apples to apples comparison, it is helpful to compare the Read and Write speeds. Simply put, even the consumer grade SSD’s are orders of magnitude faster than classic “spinning rust” (as the SSD vendors like to refer to it). Just as critical, however, is the last line discussing the useful life. Users are accustomed to hard drives lasting at least five years, and it is not unusual for a drive to last even longer. However, SSD’s have a shorter useful life and typically fail suddenly and catastrophically. Without discussing the technical details too in depth, SSD’s write data by turning bits on and off electrically as opposed to magnetically, as is the case with traditional hard drives. As data is written and re-written to a particular bit of an SSD, the controllers ability to distinguish between ON and OFF becomes impaired, and at some point, that bit is no longer useful. If enough of these bits fail, the drive itself eventually fails. Consumer grade SSD’s, utilizing what is known as MLC or multi-level cell technology are especially susceptible to this phenomenon. Enterprise grade SSD’s typically utilize SLC or single-level cell technology and typically last longer. The only disadvantage is cost; SLC drives can typically be 10x the cost of an MLC drive on a \$/GB basis.

SSD's are great for performance; however, lasting only three years seems like a risky proposition for critical data. Array vendors are taking several different approaches to the use of SSD's in their environments. Many of the more traditional vendors are utilizing SSD's as a Tier of storage. In this scenario, the array watches the blocks of data that are most active in terms of read/write activity. As these blocks "heat up," they are automatically moved from the slowest 7200 RPM drives up to the faster 15K drives, and finally to a layer of SSD's with the most performance. The technique, from an economics standpoint, is this "hot" data is typically less than 5% of your overall stored data. This means your SSD tier may only be 10% of your total storage capacity, allowing for the specification of much more expensive devices without making the overall unit unaffordable. A second approach used by other vendors is using SSD's as a conduit to slower disks. Essentially, all writes go to the SSD's first. Once the CPU cycles are available, this data is transferred to slower, cheaper disks in the array. Through the use of inline compression and deduplication, the number of writes to the SSD's is dramatically reduced, effectively extending their useful life. A third and much more bold approach is to pack the entire array with consumer grade SSD's but use sophisticated software to perform inline compression, deduplication, and other advanced techniques to reduce the number of writes required. Only time will tell which of these approaches wins out in the mainstream marketplace. The one thing that is certain is that SSD's or some future derivative are a disruptive technology that will dramatically change the way we address performance issues with arrays. It is my opinion that SSD's will essentially eliminate the IOPS bottleneck, leaving us with some new and different hindrance to address.

## **Networking**

Network storage infers one obvious aspect of the device; you must communicate via a type of network. If your chosen technology is Fibre Channel, then you have no choice but to separate the network from what I label your "computer network". "Computer network" refers to the network used for your computers to communicate with your clients' computers. The second network in a properly designed system is a "storage network." This is a physically isolated network managing all storage traffic communication. For many of the same reasons needed to isolate the PLC network from general client-server traffic, the storage network should be a dedicated fabric of cables and switches. Secondly, when designing your network, the switches should be redundant. Losing your storage backend will typically allow machines to run for about 20 or 30 seconds in a frozen state until they fail. A typical layout will look something like this.

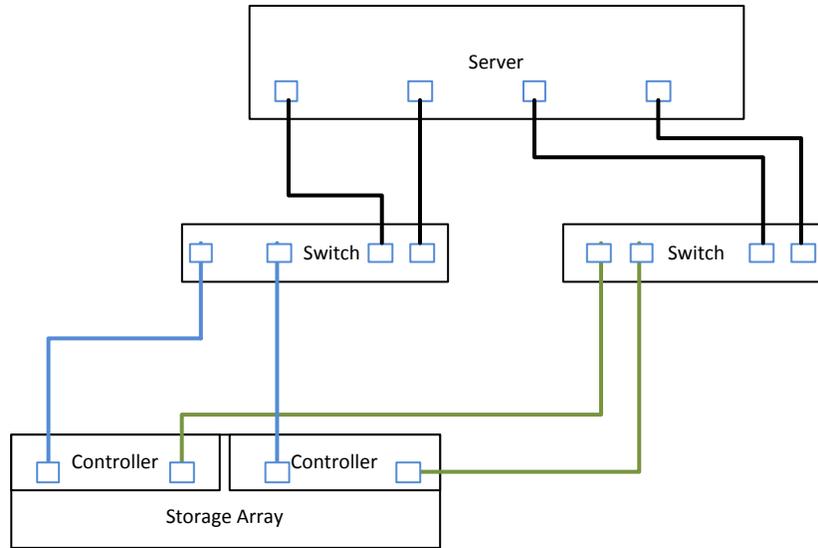


Figure 1. Typical Storage Network

A few items should be noted in this simple design. First, note that a single controller connects to multiple switches. This ensures that the system can continue operation in the event of a controller and switch failing simultaneously. Second, consider the number of network interface cards on the server itself. For a minimum setup, the server requires at least four network connections. Once again, this allows for maximum robustness in the event of a NIC and switch failing simultaneously. One final detail to consider that is not necessarily conveyed in this diagram includes what is called a multi-port NIC. The connections from a multi-port NIC should serve a diverse set of duties. For instance, on a quad port NIC, one port should be used for a connection to switch 1, another port for a connection to the “computer network”, and an additional port for a connection to your “management network.” Therefore, if one of the common components such as the PCIe interface fails, you do not lose all of the redundancy.

### Practical Shopping Considerations (Vendors, Dollars, and Sense)

This guidance will be useful, but turning it into actionable information is the next step. Learned by many years of experience, I do not recommend incorporating an array into your system simply with cost as a primary driver. A project to be executed with a limited budget should include the traditional multi-server, non-virtualized environment. It is critical to wait for the needed funds to create the ideal system. If the correct equipment is not used from the beginning, especially storage components a massive failure or terrible performance will happen at some point. Either way, an individual is going to be responsible for a poorly implemented project. While I will not express a preference for one vendor over another, I have provided a list of vendors, in no particular order, that produce quality products that should perform well in your environment.

EMC, NetApp, Fujitsu, Hitachi, IBM, HP, Dell Equallogic

In the corollary to the instrumentation engineer’s quote of “Nobody ever got fired for buying Rosemount,” you will not be risking any reputation if you purchase products from one of these vendors.

To be fair, there are a lot of other great vendors out there along with some really exciting startups. However, when it is time to make such a critical decision, you can feel comfortable with any of these providers.

Second, the dollars and sense of the matter must be evaluated. My best guideline developed from years of experience is that storage should be at least 50% of the cost of your hardware budget for a significant virtualization project. For example, three Dell R710's with 64 GB of RAM and dual Quad Core processors is about \$18-20K. Four switches (two for virtual machine traffic and two for storage traffic) should cost about \$5K. This results in the base price for a storage array costing approximately \$20-25K. With a similar budget, a user should be able to easily acquire a storage array to fit their needs. Depending on the vendor, some will include only base functionality in a starting price, and then allow you to select features such as snapshots, replication, thin provisioning, deduplication, etc. in an a la carte fashion. Beware, this can sometimes double or triple the starting price of your unit.

Finally, pay close attention to warranty costs. The higher end units will typically include three to five years of base warranty. After the warranty expires, maintenance costs can become extremely expensive. This can be a driver in the refresh cycle for a typical IT organization. The hardware may be old but functioning well. However, the economics of maintaining the warranty for the five year old hardware sometimes makes it more affordable to simply purchase new hardware. The dynamics of capitalization and depreciation can play a big factor, and it is highly recommended to work closely with your budget managers on this detail.

## **Conclusions**

Specifying and acquiring a network storage device that is adequate for a robust virtualization infrastructure in a manufacturing facility is neither easy nor cheap. While there are plenty of vendors and independent providers who are willing to help with the specification process, I always find it necessary to be armed with a solid foundation along with a keen understanding of what should be important to me and my application.

In the end, consider basic economics regarding cost. A three physical host system can easily support 24-30 servers. An average physical server should cost approximately \$5K if properly specified. The acquisition cost for these servers, ignoring the additional cost of networking, would be approximately \$120K. Contrast that with a \$50K acquisition cost for a virtualized three host system with storage. When viewed through this lens, a virtualized system with high quality storage is much less expensive. Although I would technically never advise a manufacturing customer to virtualize based on CAPEX (capital expenditures) alone, the economics are significant once you surpass 8-10 servers.