



Creating a Scalable Architecture for Accommodating Long-Term Geometric Data Growth in SANs

By Marc Farley, Building Storage, Inc.

September 2006

Abstract

This paper looks at some of the assumptions made about SAN storage that hinder the development of solutions for data growth and examines some of the technologies that have been used to manage it. The second section of the paper examines the various virtualization approaches that exist and compares their relative strengths and weaknesses. The final section is an in-depth look at one particular virtualization implementation, from SAN vendor EqualLogic, Inc., that has several important advantages over all other existing virtualization methods.

The Problem of Geometric Data Growth

There are many factors contributing to the geometric growth of stored data, including the continued introduction of data-rich software features and the ease with which users collect and exchange information over the Internet. The impact of new government regulations and the growing threat of information-borne legal costs have compounded data growth by significantly increasing the amount of redundant data copies organizations are storing. The imminent question from IT professionals is becoming more pressing every year: “What should we do to handle the problems of data growth, and how do we avoid an information management crisis?”

The alarming realization about data-growth issues is that there is no crystal-clear picture of how to solve the problem. My personal belief is that storage consolidation and virtualization are the keys to building scalable storage solutions. Consolidating storage resources in storage area networks (SANs) makes it easier to share and monitor them. Unfortunately, the number of different virtualization approaches makes it difficult to determine which one to use for a given environment. I have written about storage virtualization a great deal in the past and worked on presenting solutions from several different vendors. By writing this paper, I hope to clarify many of the questions people may have about virtualization and how best to deploy it.

This paper looks at some of the assumptions made about SAN storage that hinder the development of solutions for data growth and examines some of the technologies that have been used to manage it. The second section of the paper examines the various virtualization approaches that exist and compares their relative strengths and weaknesses. The final section is an in-depth look at one particular virtualization implementation, from SAN vendor EqualLogic, Inc., that I believe has several important advantages over all other existing virtualization methods.

Assumptions about SAN Storage That Hinder Solutions for Geometric Data Growth

The challenges of any sufficiently difficult problem, such as long-term geometric data growth, are great enough that false assumptions about the technologies involved will lead to disappointing results. Effective problem solving begins with questioning the assumptions that might preclude a more complete analysis of the problem's root causes. In the sections that follow, we examine some of the assumptions about SAN storage that have made it difficult to make progress on managing rapid, compound data growth.

Assumption Number 1: SANs are “SCSI on Steroids”

SANs replace the traditional physical SCSI bus with a network where SCSI commands and data are transported using an application layer protocol running over an underlying network, such as Ethernet or Fibre Channel. This allows SANs to function more or less the same as SCSI buses. This similarity between SAN and SCSI bus operations has led many to describe SANs as “SCSI on steroids.” This simplified view of SANs tends to overlook the profound benefits they enable, such as making multi-pathing much easier and more affordable, which means many more systems are now protected by high-availability storage connectivity.

The SCSI-on-steroids view of SANs subtly implies that SANs have the same roles as SCSI buses, which is to conduct storage data transfers between host systems and storage. People often assume that SANs do not extend beyond the network ports (HBA) into the end nodes. Although many storage

subsystems use SAN technology internally to connect devices, such as Fibre Channel loops, these device connections are managed by the subsystem controllers, which provide a distinct boundary between the SAN infrastructure and the internal workings of the subsystem.

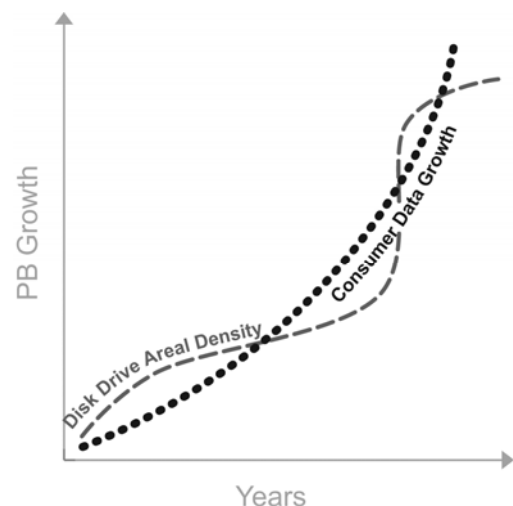
Using SCSI as an application protocol tends to hide the capabilities of the network running beneath it – including the potential to exceed the capabilities of the SCSI protocol. It's worth pointing out that the underlying SAN network supports many different processing models in addition to the limited master/slave processing model that is used for SAN traffic. Distributed and cooperative processing can certainly be done in SANs, even though traditional SAN subsystems do not support it. The problem is that once certain architectural assumptions have been made in product designs, they often are carried forward as unspoken requirements. Despite the inertia among entrenched subsystem providers, architectural progress will be made. For instance, just as SAN technology expanded to include Ethernet and IP networks, it will also necessarily grow and incorporate distributed and cooperative processing.

If you look at SANs from the perspective of network processing, they appear to be artificially constrained by the assumption that storage subsystems are single, large, discrete, isolated islands of storage. While they can be accessed by multiple host systems, they are not designed to function as part of a clustered or distributed storage system. It's true that certain applications, such as remote data copying *are* accomplished by communications between different subsystems, but this should not be confused with having the primary storage function based on a distributed or cooperative processing model.

Assumption Number 2: Storage Capacity is Determined by Component Disk Drive Capacity

As discussed above, one of the outcomes of the “SCSI on steroids” view of SANs is the isolation of storage resources within traditional storage subsystems. The storage capacities of traditional storage subsystems are capped by their physical designs – particularly the number of controllers and disk drives that fit inside the subsystem and any possible expansion cabinets. This has given rise to the assumption that storage capacity is primarily determined by the capacity of their component disk drives.

I don't know how many times I've heard storage professionals remark on how some new, larger capacity drive will make such a big difference. Invariably, the expected relief that comes from using larger drives doesn't last very long. In fact, many have already discovered that disk capacity increases are not keeping up with their data growth rates. For example, if disk drive areal densities double every 18 months while year-to-year data growth is 50 percent per year, it takes less than three years for the growth in data to overcome the capacity improvements in disk drives. Clearly, a strategy for managing data growth that depends on disk drive product cycles is not a very good strategy.



The problem of capacity scaling in a subsystem is more complicated than it appears at first. It is practically impossible to design a single, infinitely scalable storage subsystem due to the need to specify certain performance and availability minimums. If performance and availability weren't important, it might be possible to design an enormously scalable storage subsystem; but, of course, performance and availability are almost always defining variables.

One of the most common ways customers protect themselves against geometric data growth is to buy storage products that are not fully populated with disk drives so they can later increase capacity by adding additional disk drives. Unfortunately, after the maximum number of disk drives has been installed, customers have to start planning their next move, which is a “fork lift” physical upgrade to replace their existing subsystem with a new one. The disruption caused by a physical upgrade is typically onerous, even if many consider it to be business as usual.

Traditional SAN subsystem architectures are showing a pronounced scalability weakness in the face of geometric data growth and need to be replaced more frequently than desired at a significant cost to customers. There certainly are better ways to provide capacity scaling than through the arcane practice of guesswork, sandbagging, and praying.

Assumption Number 3: You Need Professional Services to Manage a SAN

SAN technology has followed a well-worn evolutionary path. Early customers depended heavily on vendor-provided professional services to make the technology work, but over time customers have become much more self-reliant. Today, the ability to manage SANs effectively depends on the combination of applications, requirements, products, and skills involved. Nonetheless, many customers still believe that costly professional services are a necessity for managing SAN storage.

One of the common misconceptions about SANs is rooted in the management shortcomings of traditional SAN storage subsystems. For example, various methods for *fencing* storage resources, such as zoning, LUN masking, and virtual networking, were developed in response to requirements that emerged after customers started building their SAN infrastructures. Adjusting to these types of fundamental technology changes was challenging for customers and elevated the risk of owning and operating a SAN, which, in turn, helped justify the expense of vendor-supplied professional services.

A great deal has changed in the short history of SAN technology. The technology has matured, and concepts that were once foreign are now broadly understood. For example, provisioning storage was once a great mystery, but now it can be done much more quickly and with automated tools that have effectively eliminated the risks of do-it-yourself storage management. More important, new technologies such as virtualization, iSCSI, and disk-based backup were invented to address management requirements that were not well understood when SANs first came on the scene. These inventions have made a huge difference in the effort needed to manage SANs, making it possible for customers to take the reigns.

When customers have greater control over their own storage resources and facilities, it is easier for them to understand how to deal with big



picture problems such as geometric data growth. Relying on vendor professional services for data-growth answers is likely to result in more traditional approaches, such as the construction of monolithic islands of storage that need to be replaced every five years or so – with the able assistance of the vendor's professional services organization, of course. To use a familiar analogy, the smartest foxes don't simply raid the henhouse, they establish permanent residence there and become fast friends with the chickens.

Assumption Number 4: Faster Networks Are Needed for Higher Capacity Levels to Maintain Performance

While capacity scaling is the primary problem facing storage administrators today, another closely related problem is performance scaling. As more data is stored, it becomes increasingly difficult to ensure that application and end user performance levels can be maintained.

The SAN industry has spent a great deal of energy and money solving this problem by increasing the speed of Fibre Channel networking technologies. There is no doubt that increasing the speed of network transmissions in traditional SAN architectures can increase overall SAN performance levels, but that does not mean that increasing network bandwidth is the only way to solve the problem. One of the most effective ways to increase throughput in a system or a network is to use parallelism where the workload of the SAN is spread over a higher number of network paths.

Unfortunately, monolithic SAN storage designs do not add network paths when the capacity of a subsystem is increased. By restricting the number of paths to stored data, the throughput of a traditional SAN depends on network transmission rates. As capacity is added to these systems, it is possible that network bottlenecks can occur where they were not previously a problem.

By comparison, a distributed SAN architecture that increases the number of network paths as capacity is increased gives IT organizations more options for successfully dealing with the performance “side effect” of geometric data growth.

Virtualization and Other Technologies for Managing Geometric Data Growth

One of the most powerful techniques for dealing with geometric data growth is storage virtualization. The concept of storage virtualization is fairly simple: storage address spaces are subdivided or combined to form smaller or larger address spaces, and these virtual address spaces are then made available for use by host systems in the SAN.

RAID is the most common form of storage virtualization. RAID level 1 (mirroring) duplicates data across two storage address spaces and presents these duplicate address spaces as a single virtual address space to host systems. RAID levels 3, 4, and 5 are formed by spreading data across three or more same-sized storage address spaces and writing calculated parity information on one of them. Again, the multiple address spaces in these RAID arrays are presented as a single address space to host systems.

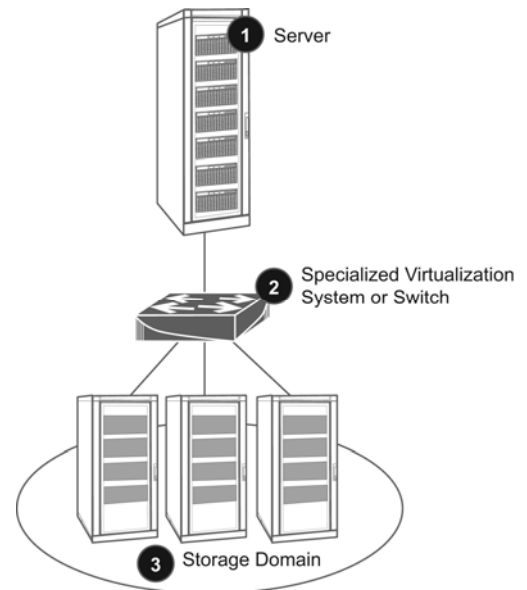
The ability to combine address spaces using virtualization techniques to create larger address spaces has intrinsic value for managing geometric data growth. From today's perspective, virtualization is the most effective technology for adapting storage technology to new high-growth environments.

To date, virtualization architectures have been distinguished by the location of the virtualization function in the I/O path. The three most obvious locations are:

- 1) In host system software – usually referred to as volume management
- 2) In the network, either in specialized virtualization systems or in switches
- 3) Within the storage domain, as a function provided by storage subsystem controllers

Volume Management in Host Systems

Volume management is the implementation of storage virtualization software running in a host system. The volume manager performs storage address space manipulations, providing a virtual view of storage resources to the system. Volume managers typically have code executing in kernel space, which may involve undocumented programmatic interfaces and has significant implications for product design, testing, and support.



Volume managers have a one-to-many ratio with the storage resources they access. A system with volume management software can potentially access the complete set of address spaces on all devices and subsystems connected in a SAN. The disadvantage with this arrangement is that it does not provide centralized control of storage resources for host systems, making overall SAN storage management more tedious and time consuming. Designs for clustered and distributed volume managers attempt to address this weakness.

To summarize, volume management software is a well-known technology for managing geometric data growth for individual servers; however, it is not thought of as a tool for managing the geometric data-growth problems of the entire business.

Out-of-Band Network Virtualization

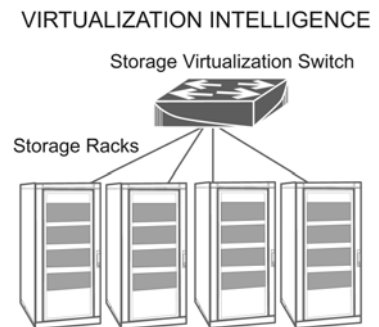
An alternative approach to host volume managers is *out-of-band* virtualization, which uses an independent control system located in the SAN that coordinates storage resource usage among multiple host systems, each running their own virtualization *agent* software. The main difference between out-of-band virtualization and volume management software is that the former is designed to accomplish resource allocation and sharing in the SAN using a many-to-many ratio between systems and storage. For instance, when new storage capacity is added to the SAN, its access is controlled by the virtualization control system(s), which then make it available to the various host systems in the environment.

The primary drawback with out-of-band virtualization is the overall complexity of the solution, including the development and maintenance of host system agents. The architecture is simple enough, but the details of maintaining agent code across mixed platforms can be devilish. Customers do not have to keep up with OS changes – their virtualization vendors do – but there could be times when the customers must wait to roll out a new operating system upgrade in order to make sure they do not “break” the functionality of their out-of-band virtualization system.

Virtualization in the Network

Storage virtualization can also be implemented in the network by running an application in a network system, which may be implemented as a virtualization appliance, or in a network switch. Such products perform virtualization functions in the I/O path between host systems and the storage domain.

Multiple servers can access a network-based virtualization system, which in turn can access multiple storage resources and address spaces in the SAN. Similar to out-of-band virtualization, network-based virtualization (sometimes called *in-band virtualization*) has a many-to-many ratio between host systems and storage resources. Network-based virtualization provides a mechanism for effectively managing geometric data growth by allowing new storage resources to be transparently integrated into the SAN environment.



The primary shortcoming of network-based virtualization is the complexity involved in conducting storage data transfers between host computers, through the virtualization system, and to multiple storage subsystems in the SAN. Specifically, differences in time-out and error-recovery processes involving multiple independent network sessions pose unpredictable problems for the virtualization system. In the simplest cases there are two sessions: the first is between the host system and the network-based virtualization system, and the second is between the virtualization system and the storage subsystems with which it is working. Of course, in actual implementations, the complexity is much greater due to the increased numbers of host systems and storage subsystems involved.

For instance, assume the network virtualization system is distributing I/Os across two or more storage subsystems. How should it behave when one of the subsystems does not respond to a query or command? It is possible that the unresponsive subsystem is temporarily busy handling work from another system or that a network switch may have dropped frames. In this case, it would make sense to wait a reasonable amount of time and try the operation again. The problem is that time-out values for storage processes are not standardized across storage subsystems, which means the virtualization probably needs to take a conservative (slow) approach to error handling. Subsequent long delays in I/O processes can occur, having a direct negative impact on application performance. In the final analysis, network-based virtualization is easy to imagine but very tricky to implement, which is why compatibility testing is so important for these types of products.

Another potential problem area for in-band network-based virtualization is performance. The process of terminating and retransmitting storage processes necessarily takes time and involves limited resources, such as queue memory in the virtualization system. As the amount of storage connected through a virtualization system increases, the more likely it is that the virtualization system will become an I/O bottleneck.

Virtualization in the Storage Domain

The other location for virtualization is in the storage domain. In a traditional storage subsystem, virtualization functions are performed in subsystem controllers and are limited in scope to the storage devices within the subsystem and any attached expansion cabinets. For the most part, this type of virtualization represents the current situation where SANs are assumed to be “SCSI on steroids” along with the lackluster characteristics of subsystems based on the limited capabilities of disk drives.

The combination of virtualization with distributed processing models in the storage domain is one of the brightest areas for the future of storage virtualization. The concept is to have multiple subsystems operating in a cooperative fashion and sharing their aggregate resources. It is not necessary for these subsystems to use clustering technology, thereby avoiding all the overhead that clusters require, as long as they provide timely I/O services to each other when they are needed. An architecture with these characteristics would be able to keep pace with geometric data growth without the disk drive dependencies of traditional monolithic storage subsystems.

As SAN technology matures, methods for manipulating storage address spaces will become more sophisticated. One method, developed by EqualLogic, is discussed in the final section of this paper.

Alternative Technologies for Managing Data Growth

This section briefly looks at other technologies that have been used or could be used to manage geometric data growth.

Hierarchical Storage Management (HSM)

The first technology developed to address the problems of data growth was hierarchical storage management, or HSM. HSM was initially introduced by IBM for mainframe customers and provided a way to relocate data from primary storage to secondary storage where it can be accessed again, if needed.

HSM is one of the most complex technologies in storage, involving operating system and file system modifications in addition to policy-based data management and metadata extensions. When a storage volume exceeds a certain threshold for filled capacity, the HSM system starts copying data files from primary storage to secondary (or tertiary) storage. It then creates a special file called a stub that replaces the copied data file and acts as a pointer to the data file's new location on secondary storage. When a user or program attempts to access the stub file, the HSM system intercepts the request and copies the data file back to its original location, overwriting the stub file.

In general, HSM really only works when the operating system, file system, backup, and HSM system vendor are the same. Open systems HSM products have not been very successful due to the difficulties in implementing kernel-level modifications consistently across different OS platforms. Integrating HSM with backup systems is also a challenge, requiring special considerations for backup I/O processes. Most important, customers have legitimate concerns about the intricate system of stub files and metadata – and what happens if it is damaged or lost. There is some risk that the HSM capacity management system could end up losing data.

Information Lifecycle Management (ILM)

Unlike HSM, information lifecycle management, or ILM, is a concept encompassing several different storage technologies that are intended to manage data resources from their inception until the time that they are no longer needed. Unlike HSM, which is primarily a capacity-management technology, ILM incorporates tiered storage, backup and recovery, data archiving, and regulatory compliance within its scope.

ILM is not intended to manage data growth as much as it is intended to manage digital records, including the creation of redundant data copies for various historical purposes. In fact, ILM probably does more to exacerbate the problems of data growth than it does to solve them. Nonetheless,

considering the nature of ILM processes, it is certainly possible that ILM applications could emerge that would be used to manage geometric data growth. One such possibility would have ILM software taking advantage of tiered storage, which is discussed in the next section.

Tiered Storage

Tiered storage is a concept that encompasses different classes of storage resources under the control of a single management system, so that data having different purposes and storage requirements can be located on the most appropriate storage resources. In many respects tiered storage is an attempt to get the benefits of HSM without the complexity and risk. The economic motivation behind tiered storage is that data that does not require fast performance can be placed on lower-cost, slower storage. In other words, customers can place high-priority data on high-performance storage and lower priority data on lower performing storage.

Tiered storage does not reduce the amount of stored data; instead, it is intended to reduce the amount of stored data on high-performance, expensive storage subsystems. Essentially, tiered storage provides a way to manage data growth for the most important on-line applications. That means the management of data growth on secondary storage is still a major problem.

EqualLogic's Dynamic Paging Architecture for Managing Geometric Data Growth

Given that existing storage products and technologies have not provided solutions for geometric data growth, it follows that a new approach is needed to solve the problem. One of the things to look for is an architecture that does not conform to the assumptions about isolated storage resources discussed previously in this paper. An excellent example of an architecture that avoids traditional storage shortcomings is found in the storage products from EqualLogic, Inc. But before discussing EqualLogic's storage architecture, first we will examine a different technology, virtual memory page swapping.

Storage Paging

Virtual memory technology was invented to give systems the ability to use disk space as a temporary location for application data that is not currently being processed. *Page swapping* is a virtual memory technique that identifies specific memory address ranges called *pages*, with the idea that some of these pages can be temporarily stored on disk in order to make room for other additional application programs and data in memory. Page swapping can be described as an anti-caching mechanism, where data that isn't being used by programs is put on disk where it can be retrieved later.

In the parlance of networking, page swapping allows systems to oversubscribe the number of applications they can support. Page swapping is independent of any addressing mechanisms that applications use, and the work of paging is completely transparent to the applications running in the system. The concept of the page as a regularly sized quantity of data is extremely important, because it means all swapping operations can be done with optimum efficiency – there is no need to calculate and communicate the size of the data or its boundaries because they are always the same.

EqualLogic's architects built a system based around the concept of a *storage page* (the obvious pun would be to say they took a page out of the virtual memory book). They reasoned that disk storage could be segmented into regularly sized address ranges, similar to the way memory is segmented in page swapping implementations. The main difference is that the storage page is not intended to be

swapped between system memory and disk; instead, the storage page can be located and relocated within any number of storage systems working together in a SAN.

A hypothetical example of storage paging and page relocation follows. Think of a computer system that has been operating with a single disk drive that is approximately 80 percent full of data. Let's assume the system has storage paging and that it has already defined and created pages for this drive. To improve system I/O performance and get some peace of mind, the administrator puts an additional disk drive in the system, adds its capacity to the existing volume, and expands the file system. Now, instead of redistributing the file system in an attempt to balance the workload, the storage paging software starts relocating pages from the first disk over to the second as a background task and reducing the percentage of used space on the first disk to acceptable levels.

Throughout this process, the file system continues to operate as it normally does, allocating storage in the virtual address space formed by the two drives while the storage paging system monitors reads and writes, adjusting the workload between the two drives in order to balance I/O activities. It is important to realize that that file systems have no way to balance the I/O workload among multiple devices because they have no way of knowing for certain what their address space boundaries are. On the other hand, a storage paging system *does* have that information and can ensure that storage resources are being exercised to their optimum potential.

But this paper is about managing geometric data growth, and to fully appreciate the ability of the EqualLogic architecture to manage geometric data growth, it is necessary to analyze the potential for storage paging in a SAN context. EqualLogic designed its products to be network storage *systems* that are deployed in SANs, where they are expected to work cooperatively with other EqualLogic systems, sharing storage resources and relocating storage pages among those shared resources as demands dictate. This distributed resource approach sets this architecture apart from traditional “SCSI-on-steroids” architectures where storage resources are isolated from each other over the network.

The Invisible Hand of Pool-Based Allocation

In stark contrast to traditional SAN storage, storage resources in an EqualLogic environment are not limited by the maximum disk capacity of the system. EqualLogic systems allocate storage as *pools* that can be merged and combined with other pools on other EqualLogic systems to form virtual storage address spaces. Pools are a virtualization structure that is used to organize storage resources so they can be more easily identified and manipulated. It may help readers to think of EqualLogic storage pools as macro-storage structures that are protected by RAID and ready to function independently or as a member of a team of storage pools.

As a tool for managing geometric data growth, pools give storage administrators a non-disruptive way to increase the storage capacity of existing storage address spaces. Increasing an existing storage address space involves adding available storage pools in the SAN to the storage address space that needs capacity relief. The growth scenario in the EqualLogic architecture involves adding a new storage system to the SAN, creating its storage pools through the administrator interface and adding some of those pools to other, existing pools in the SAN.

This is much simpler than the onerous upgrade processes needed to expand capacity with traditional SAN storage subsystems. To put a point on it, there is never a need to perform disruptive and risky fork-lift upgrades to storage subsystems because incremental capacity is a relatively short and easy network hop away. In the EqualLogic architecture, incremental capacity is not metered by the

capacities of individual disk drives but is generated at a system level as a storage pool. A worthwhile and time-saving by-product of adding storage capacity at the system level in this manner is that it is already protected by RAID – unlike individual disk drives in monolithic subsystem designs where adding drives to existing RAID groups can be a painstaking task – and may require expensive vendor professional services.

The virtual address space in the EqualLogic architecture includes system coordinates, such as network identifiers, similar to the way network virtualization systems do. This means a storage address space that is exported to a host system can be composed of the total aggregate capacity of several EqualLogic systems to meet the requirements of certain applications. However, in most cases, customers have more urgent needs for managing numerous, shifting capacity pressure points. The distributed, cooperative design of EqualLogic's architecture coupled with the ability to share resources and relocate pages online within those shared resources gives their customers a leg up on dealing with the symptoms of geometric data growth.

Automated I/O Balancing and Paging

While network-based virtualization is used effectively by many, there are a few comparisons worth noting in relation to EqualLogic's architecture. First and foremost, the EqualLogic architecture provides automatic workload balancing as part of its paging algorithms. These systems work together in a SAN, monitor their conditions, and periodically shift pages between systems in order to avoid disruptive capacity crises. In other words, an EqualLogic SAN provides online capacity management that significantly reduces the burden of monitoring storage capacity associated with traditional SAN storage subsystems. This capability is supported architecturally through the definition of storage pages, which makes inter-system transfers efficient and reliable. Because all EqualLogic storage systems in a SAN are well-known entities to each other, they have predictable behaviors for time outs and errors that might occur during inter-system transfers.

EqualLogic systems are designed to discover each other. When new systems are added to a SAN, the systems recognize the addition of the newcomer, and the storage administrator is given the opportunity to automatically add (provision) the new system's storage pools to existing pools. The ease with which this is done should not be underestimated, nor should the value of EqualLogic's automated provisioning, which eliminates the potential for human error during installation and provisioning. The end result is that these products can be and often are installed, and managed, by customer employees, circumnavigating the need for costly professional services.

As new storage systems are added to an EqualLogic SAN, the number of network paths to storage also increases. Combined with the automated I/O balancing across pools in different systems, the aggregate throughput of the SAN is increased without having to go through the protracted performance analysis work of traditional monolithic SANs. In other words, the time spent tuning the SAN performance is far less; in many cases, the automated performance benefits are completely transparent to the administrators because they never notice performance degradation.

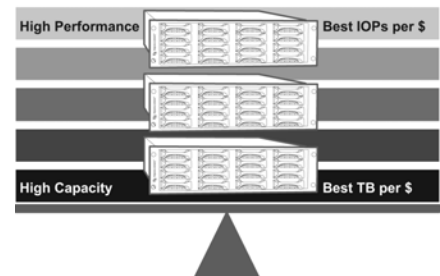
Pooling and Tiered Storage

A pooling architecture lends itself well to the concept of tiered storage. If you consider the possibility of a storage system that supports different types of disk drive technologies, it is easy to imagine how pools could be established based on the type of disk drive used.

While this might seem obvious to readers familiar with discussions of tiered storage, the subtle point to make about an integrated pooling and storage paging architecture is that storage systems can be

added to a growing SAN with the idea of changing the mix of resources applied to the various tiers being used. If you need more first-tier storage, simply add tier-one storage arrays employing high-performance disk into the existing pool(s). More important, these systems will automate data distribution to balance capacity usage. As IT professionals add sophisticated tools for classifying and storing data, the desire to change the mix of storage in response to performance and economic drivers will increase. The need to apply automated tools for managing storage operations will become a baseline requirement.

SINGLE VIEW BALANCED SYSTEM



Summary

Compound geometric data growth is one of the most urgent problems facing IT organizations today. Unfortunately, traditional storage technologies, products, and methods are showing definite shortcomings in managing the situation.

One of the problems we face as an industry and as a market is the set of historical assumptions about how storage can and should work. Traditional SAN storage products are based more or less on the processing models of older direct-attached storage. If we are going to overcome the problem of geometric data growth we need to shed our “SCSI-on-steroids” view of storage and find ways to unlock the resources that are isolated within the subsystems around which we build our storage infrastructures. In addition, it is important to overcome the arcane assumption that storage capacity is somehow related to the capacity of disk drives. While disk drives certainly have an impact on storage capacity, it's almost laughable that a sophisticated infrastructure component, such as a subsystem or the storage volumes it exports, should be constrained by the number of disk drives that can fit in a cabinet. Another assumption that confounds many is that professional services are required to own and run a SAN. As long as we believe that key infrastructure technology is beyond our grasp to control, we will not be able to come to grips with our geometric data-growth problems.

Several technology concepts, such as HSM, ILM, and tiered storage have been conceived to attempt to deal with geometric data growth, but none of them have shown much ability to enable most organizations to deal with their data-growth problems. There is still a possibility that these technologies will someday become more effective; but unless they start showing more real potential than promises, customers should try to find more direct answers to their data-growth problems.

One place to look for answers is in newer processing models and architectures for data storage that fundamentally change how storage works. An excellent example of a new storage architecture is that designed by EqualLogic for its SAN solutions. In brief, replacing the traditional master/slave SCSI block processing model with a combination of distributed/cooperative processes and an underlying paging structure and mechanism holds a great deal of promise for fundamentally tilting the odds in the favor of IT professionals.

There are a number of reasons why geometric data growth probably cannot be arrested, but that does not mean it cannot be dealt with successfully. Cooperative storage paging is a significant breakthrough that shows very real potential for getting a firm grip on geometric data growth.

###