Distributed Storage for a Scalable Grid Infrastructure

Network Appliance | Rajesh Godbole and Chuck McManis | February 2005 | TR 3385

TECHNICAL REPORT

Network Appliance, a pioneer and industry leader in data storage technology, helps organizations understand and meet complex technical challenges with advanced storage solutions and global data management strategies.

Table of Contents

1) Executive Overview

The primary business driver for grid computing is the need to apply massive computing power to the manipulation of massive amounts of data—for example, for rendering animated motion pictures, analyzing geologic data, and running large-scale business applications. As network technology has advanced and computer servers have become more powerful, large monolithic systems have given way to distributed computing grids composed of many discrete processors and storage devices interacting in a cohesive way to form a unified system.

Grid computing solutions have already proven their ability to reduce costs and boost capacity at many organizations. However, deploying and maintaining these grids require careful system design, not simply ad hoc network expansion. For example, simply adding more storage capacity won't necessarily improve the performance of the overall system. You must begin with the right kind of storage infrastructure, and you need mature management tools for detecting and dealing with continually shifting application priorities and workloads.

Network Appliance (NetApp) offers a variety of industry-leading solutions for configuring and managing dynamic storage grids, which yield significant improvements in storage utilization. With these solutions in place, organizations can efficiently deploy, allocate, and expand storage resources, with little or no effect on users. NetApp solutions provide capabilities for identifying and alleviating storage "hot spots," for managing data sets across file servers, for adding storage capacity on-the-fly, for moving selected data from one place to another, and for making changes to the storage infrastructure without impacting online applications.

This paper describes these unique storage solutions and management utilities, with attention to how different types of organizations are using them to build scalable, reliable, and imminently serviceable grid-based computing architectures. The paper begins with a historical overview of the technological advancements that precipitated today's computing grids. It then discusses common challenges storage administrators face when deploying storage grids. The heart of the paper revolves around three particular NetApp solutions—SpinServer®, Distributed Network File Services (DNFS), and NetApp SharedStorage™—which are setting the standard for innovation in today's rapidly evolving storage industry. The paper concludes with scenarios and case studies that demonstrate how these NetApp solutions are being used today to solve real-world problems.

This paper will help IT managers answer the following key questions:

- What exactly is a storage grid, and how does it differ from other types of distributed storage systems?
- What are the criteria for evaluating whether or not my company needs a grid, and, if so, what is the optimal grid architecture?
- What are the benefits and costs of a storage grid versus traditional storage implementations?
- How are other companies using grids, and what can I learn from their experiences?
- What are current and future NetApp grid offerings?

2) A Very Brief History of Trends in Computing Models

Several computing generations ago, at the dawn of the minicomputer era in the late 1970s, a computer's central processing unit (CPU) was made up of several boards. Each board handled one or several distinct sets of tasks, and the full set of tasks composed the operations of the CPU. Deploying the tasks across multiple boards allowed subsets of tasks to be performed in parallel, yielding processing speeds greatly exceeding those that would have been possible if all tasks had to be executed sequentially.

As new and faster chips were developed, however, the distance between the boards became a bottleneck in the effort to achieve more powerful architectures. If the components could not communicate with one another, across the boards, quickly enough to keep up with the increased processing power of the chips on the boards, the components on one board might finish a set of tasks and then have to wait idly until the results of another set of tasks could be received from the components on another board. This communication latency could pose a problem in terms of taking full advantage of the promise of parallel processing. This problem was solved through advances in miniaturization and the emergence of the "computer on a chip," in which the components of the central processor were placed not on separate boards, but in extremely close proximity to one another.

This scenario illustrates an obvious principle: in planning improvements in information-processing architectures, you can't consider any one element in isolation. Great improvements in processing speed will not yield great improvements in overall system "power" if the processing components can't communicate quickly enough with one another and with other components of the system.

Balance of Power

An ongoing goal in computer design is to increase the scope and complexity of the kinds of work for which a computing system can be used. Toward this end, improvements in processing speed and in intercomponent communication cannot, by themselves, deliver the desired results. There must be corresponding improvements in:

Memory capacity, so that more data and more instructions can be "in the computer" at any one time

Long-term storage, so that all data that might be required in the performance of a particular application can be readily accessed, intermediate results of processing can be kept readily available, and "final" results can be retained as long as desired and made available for display, offloading, or any other desired purpose

Communication speed between internal memory, processors, and long-term storage, so that processing power is not wasted waiting for data to be processed, and bottlenecks are not created when two or more processors must write data simultaneously to long-term storage

Access speed and ease of use for client computers (users) as they interact with the rest of the system

Networks and Distributed Processing

An interim stage in the evolution of computer systems was the development of networks, over which multiple computers could exchange and share data. Each client computer in the network could run its own applications locally, while accessing remotely stored data when necessary. When networking first became popular, users who wanted to share files had to log in across the net to a central machine on which the shared files were located. The storage capacity of these central machines was quickly exhausted, and demand grew for a convenient way to share files from several machines at once. Various solutions, most notably Network File System (NFS), emerged in response to this demand.

Today, the architecture of both processing systems and storage systems is changing dramatically. The phrase "parallel processing," which used to refer to components on several different boards interacting to work as a single processor, now refers to multiple processors working together on a single application. This can mean two or more processors residing together in a single computer (for example, an enterprise-class server may have dozens of processors), or, increasingly, it can mean hundreds or thousands of *computers* working together in a single unified architecture that represents a new kind of "supercomputer." This is a new model, a *distributed* model, that is very different from (and much less expensive than) earlier monolithic models of supercomputing.

In 1994, under the auspices of the NASA Grand Challenges program, the NASA Goddard Space Flight Center decided to take 16 off-the-shelf computers based on the Intel® 80486 DX4 processor, install open-source software on each node (Linux), and link them into a single computing resource using Ethernet. The result, called the Beowulf Project (NASA1), was a high-performance computing resource that was significantly less expensive than the alternatives from Cray and other supercomputer vendors.

Since that time, three key factors have contributed to the phenomenal growth of such clusters: aggressive knowledge sharing in the open-source community, an incredibly steep price/performance ramp in commodity computers, and an insatiable demand for densely packed servers in the Web services industry. These types of systems have become so successful that dozens appear in the list of 500 fastest machines in the world.

Grid Design Isn't Just Network Expansion

Unlike "classic" supercomputers, this new model of computing is emphatically not monolithic. Yet it is still vitally important to recognize that deploying a distributed system is an exercise in system design and not simply network expansion. Just as designing a single computer requires an optimum balance of a number of different factors, so also the design and deployment of a distributed architecture require (on a much larger scale) an optimum balance of processing power, memory capacity, accessibility and capacity of long-term storage, and speed of communication between all these elements—as well as between the distributed system as a whole and the client computers that interface with it.

Maximizing Utilization of Present and Future Resources

As architectures grow in complexity, scope, and the way in which they are distributed, the challenge of *administering* these systems can become increasingly daunting. If the promise of grid computing is to be fully realized, it must be possible to configure resources to meet present needs, while at the same time ensuring the ability to scale for the future—*without having to precisely predict future needs in advance.* It must be possible to maximize utilization of resources in the near term *and* the long term, without having to interrupt service or reconfigure the architecture when it comes time to add capacity.

Underscoring the point that deploying a grid is a matter of system design and not just network expansion, a key challenge in grid computing is the need to provide instant access, for large numbers of processors, to huge amounts of data. In attempting to increase the processing speed of data-intensive applications, the benefit of adding more and more processors "hits a wall" when the storage infrastructure is unable to keep up with the data access demands of the processors. The solution to that problem is not necessarily, or not only, the addition of more storage capacity; the solution lies in deploying the right kind of storage *infrastructure* as well as the optimum capacity. The infrastructure must provide for current needs while also being easily and transparently expandable and reconfigurable to support future changes.

GX Technology (GXT), a seismic processing services company based in Houston, Texas, recently deployed an infrastructure of this type. Because GXT must complete processing on time to meet customer commitments, and because seismic processing is extremely compute- and data-intensive, GXT began migrating its time-processing service from its legacy Sun™ enterprise servers to a grid infrastructure consisting of clusters of Linux® server blades and network storage devices.

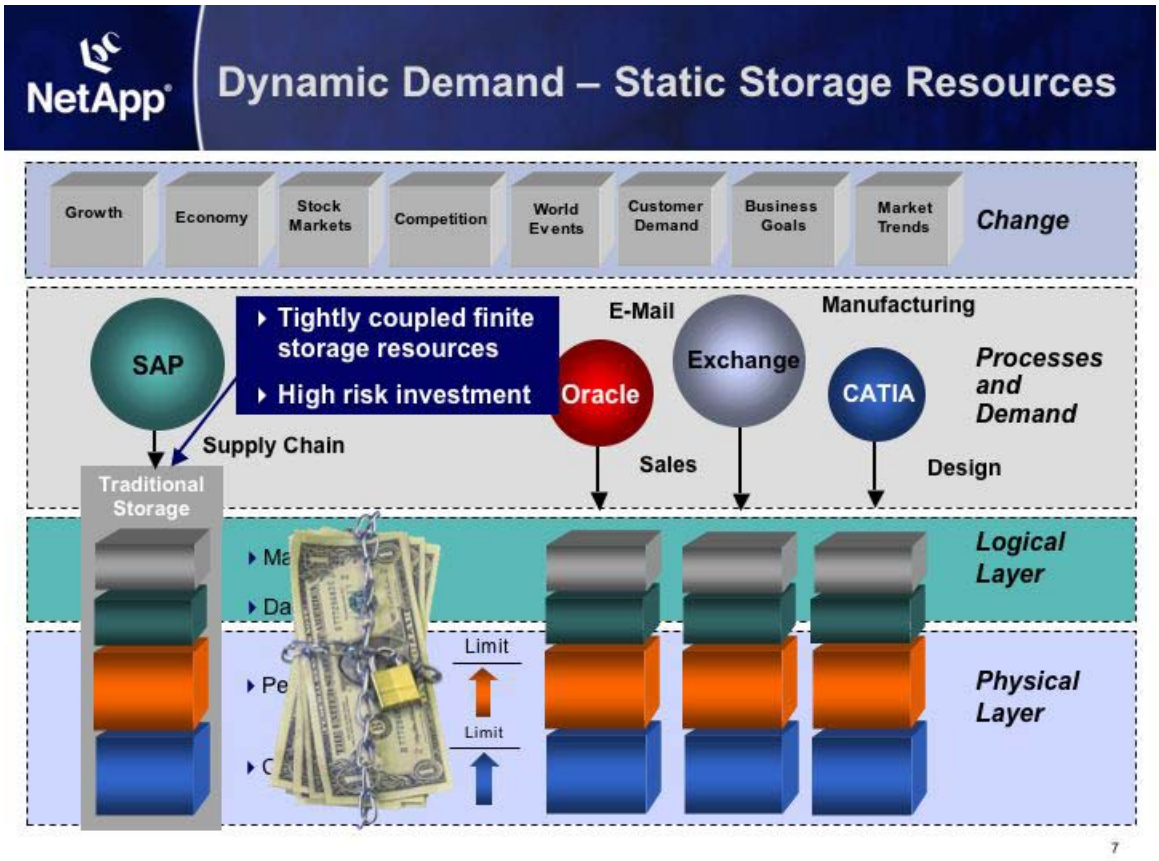
A Network Appliance™ SpinServer storage system was selected to provide all storage for the application because of its superior reliability and performance. SpinServer, discussed in more detail later in this paper, is a flexible, high-performance grid storage system that can scale to encompass as many as 512 cluster nodes and a total of 11PB (11,000TB) of storage in a single installation and under a single global namespace. This unique feature of the SpinServer architecture allows users to view all storage on any of GXT's 16 nodes as if the entire storage grid were a single large file system.

As a result of the SpinServer deployment, GXT has been able to expand production and run more jobs in parallel, with greater performance than was possible under its former monolithic architecture. Because administration of the SpinServer solution is so straightforward, a single individual manages the 250TB SpinServer installation and performs other duties as well. GXT estimates that the SpinServer deployment has resulted in at least a tenfold gain in overall price/performance compared to its previous storage environment.

Note: You can access the complete GXT case study at www.netapp.com/tech_library/3346.html.

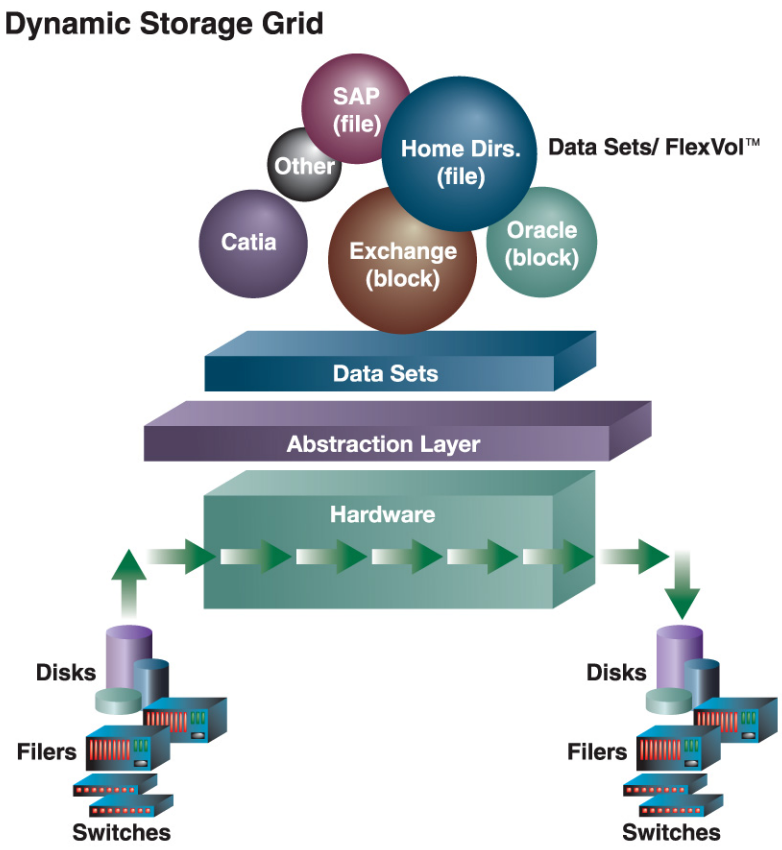
3) Toward Grid Computing

Many of today's mainstream applications are too tightly coupled to the physical IT environments with which they interact. Organizations design applications and provision them to meet anticipated peaks, but because they cannot share the underlying resources among other applications, average utilization is highly inefficient. Perhaps too much focus has been applied to cost/performance ratios and not enough to simplified integration. As a consequence, despite all the progress that has been made in application development, database management, server performance, storage access, and capacity, today's applications end up creating virtual "silos" of nonsharable, underutilized resources. It is largely because of these silos that most companies can only utilize about 20% of their server capacity and 40% of their storage resources.



Flexible volumes are a groundbreaking new technology that helps solve this problem. These volumes are *logical* data containers that can be sized, managed, and moved independently from the underlying physical storage. A flexible volume, or FlexVol™ volume, is simply a "pool" of storage that can be sized based on how much data you want to store in it, rather than on the physical size dictated by the disk drives. A FlexVol volume can be reduced or enlarged on-the-fly, without any downtime. FlexVol volumes have all the spindles in the storage pool available to them at all times, and I/O-bound applications can run much more quickly than equivalent-sized traditional volumes.

FlexVol volumes provide these new benefits while preserving the familiar semantics of volumes and the current set of volume-specific data management and space allocation capabilities. A good deal of information about flexible volumes is available on the NetApp Web site. (Go to www.netapp.com/tech_library/ and enter the keyword *FlexVol* in the search field.)



Just What *Is* a Grid?

Grid Computing and Storage Grids

A “grid” model is emerging in the area of storage as well as in the area of processing. Strictly speaking, “grid computing” refers to the use of many processors linked in a common infrastructure and operating in parallel, and “grid storage” refers to storage systems configured as a cluster or grid. But because these two technologies are so closely related in the context of a total system solution, this paper uses the phrase “grid computing” to mean the practice of deploying integrated processing and storage architectures. Terms such as “storage grid” and “storage cluster” will be used to refer specifically to storage infrastructures.

Compute Farms and Clusters

When speaking precisely about systems composed of multiple computing nodes, those systems with a dedicated interconnect carrying internode traffic are called *clusters,* whereas groups of nodes that share a common management infrastructure but not an internal communication infrastructure are more properly called *farms.*

The distinction is not always noted, and the phrase “compute farm” is often used to refer to any integrated collection of computing nodes. As used in this paper, “compute farm” is used in this fairly general way, meaning an integrated collection of computing nodes, as distinct from the storage systems that may be associated with it.

Grid Computing and Utility Computing

The phrase “grid computing” is also used in some circles for talking about the concept of a ubiquitous, extremely widely distributed (even global) infrastructure of computer resources that users tap into in order to run their applications, in much the same way as any customer can tap into electricity from the electrical power grid. The focus of this white paper, however, is not on “utility computing,” but on enterprise grid computing.

Technical Strategies

These strategies characterize current trends in grid computing:

Distribution of processing nodes: breaking the “big server” model, especially the deployment of hundreds or thousands of processing nodes in a blade format

Distribution of storage, including data replication, minimizing of the amount of data that must be replicated

Emerging state of the art for distributed storage access: global namespace, in which distributed storage appears to each node as if it were “local”

State of the art for storage manageability: ability to add capacity, perform maintenance, and reallocate storage, all without interrupting processing

Scaling Up and Scaling Out

For any given enterprise, the move toward grid computing must begin with careful analysis of the computer resources currently deployed by the enterprise, so a roadmap can be created for evolving from the current configuration to a flexible future configuration with as little disruption as possible. Tips for this kind of transition analysis are presented later in this paper. The objective, once again, is system design, as opposed to simply adding on.

A distinction is often made between “scaling up” and “scaling out.” Scaling up means adding more capacity, such as internal memory, to an existing unit of equipment such as a server. Scaling out means moving toward a distributed instead of a monolithic architecture.

Business Drivers

Grid computing is motivated by the need to apply massive computing power to the manipulation of massive amounts of data. Innovative organizations are developing grid-based replacements for a broad scope of traditional SMP solutions, bringing the benefits of grid computing to a widening range of enterprises. The reason is clear: the grid computing model provides a strategy for further utilizing and extending the ROI of existing hardware and database resources.

Specific business drivers include the following:

Price/performance

- Maximizing existing resources by taking advantage of spare cycles. Companies spend as much as $50 billion annually on servers, and yet utilization is as low as 30% of available capacity.
- In traditional IT configurations, computing resources are dedicated to specific organizations and are not available elsewhere even when they are idle. Grid computing provides a mechanism to share resources across organizational boundaries.
- Grid-based systems open the way to “utility computing,” in which computer resources are made available on demand on an ad hoc basis, as opposed to requiring the constant creation or tailoring of application-specific architectures.
- While grid computing is revolutionary in many ways, it is also attainable in an evolutionary way: it is possible to maximize the utilization of existing resources while preparing to integrate them into the new model.
- Grid computing architectures are built by clustering low-cost servers rather than centralizing massive monolithic servers. This is made possible, in part, through the use of moderately priced storage appliances that have their own software and perform operations and tasks that formerly consumed some of the processing power of general-purpose servers. These architectures are much more cost-effective than traditional IT infrastructures.

Cost sharing

- Multiple groups can contribute resources to a project.
- A body of standards already exists for grid computing and continues to evolve, enabling a broad set of diverse systems and architectures.

Improved resource management

- Traditional enterprise IT architectures are very inefficient in terms of the level of maintenance they require. Many businesses devote up to 90% of their IT budgets just to keeping their systems running.
- With grid computing, enterprises can transparently make changes in their computer-based business processes, without having to change their IT infrastructures. A good deal of scaling and assignment of resources can be done automatically by the infrastructure itself, allowing IT managers to focus on larger issues than the day-to-day maintenance and reconfiguration of architectures.
- A grid infrastructure can effectively manage applications with dramatically fluctuating capacity requirements, without the need for constant manual intervention and reconfiguration.
- Grids can start small and scale out as demand increases, without a need to redesign the architecture and without interrupting the applications that are already running.
- One administrator can manage hundreds of servers and petabytes of storage.

New class of capabilities

- Grid computing offers the potential to solve some very large problems that inherently lend themselves to parallel processing.
- There are enhanced capabilities for research and simulation and for making new discoveries.

4) Desirable Characteristics in a Grid

Whether you're designing a grid system from the ground up, expanding an existing grid, or embarking on the move from a central server–based architecture to a grid system, there are several characteristics that should be attended to in all planning and design. These are:

Scalability

Availability

Manageability

Serviceability

These characteristics all affect one another, and in an optimal solution they are all present.

Scalability

Scalability applies in two important ways:

It must be possible to scale physical resources. For example, it should be straightforward to add servers for performance, add disks for capacity, etc.

Management of the system must scale as the physical system grows. For example, global namespace makes the storage environment look like a single large system from the point of view of the storage client. More importantly, having a unified system view and single system image allows the storage administrator to manage the distributed system as if it were one large system.

Availability

Beyond the obvious requirement that the entire integrated system be reliable, it should also be possible to make on-the-fly adjustments and additions to the running system, without affecting clients and without interrupting operations. For example, it should be possible to add new storage nodes to the system and rebalance storage across the expanded capacity, with no effect on clients. Similarly, it should not be necessary to reprogram clients when storage is redeployed.

Manageability

The system's storage management interface should present a unified system view and single system image to the storage administrator, so the storage administrator can manage the distributed system as one large system.

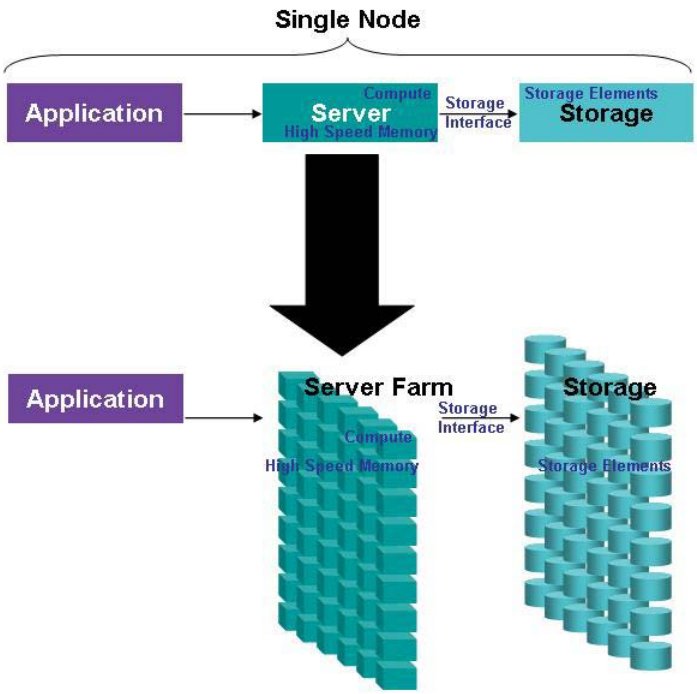
Serviceability

If a node in the system must be removed or taken down for maintenance, it should be possible to do so without affecting clients and without interrupting operations.

5) Solution Overview: What to Scale, When to Scale

Ready availability of inexpensive servers has reduced the barrier to building compute farms. Furthermore, the use of open-source systems such as Linux and FreeBSD means that more often than not the basic operating system is part of the "standard" distributions of this software. The result is that anyone can put together a basic compute farm and achieve some scalability advantages with very little effort. However, it is also true that many of the resulting problems—which were solved by supercomputer vendors in the 1980s—are cropping up as new challenges in the 21st century.

Traditionally, a single node in a traditional client-server architecture consists of a compute element, high-speed memory, a storage interface, and a storage element. Because of the proliferation of compute servers, the compute and memory elements are now spread across a server farm. This scaling out on the compute side can potentially cause scaling problems on the storage side. The following figure illustrates the importance of system design in these
instances.



Why system design is critical.

The key to understanding these problems and their solution lies in understanding the system itself. In simple terms:

> *The deployment of a grid should be approached in terms of system design and not merely as a basic exercise of "adding on" to a computing network.*

There are four primary factors that affect the performance of any computing system:

Maximum processing rate. For any computing node, there is a maximum rate at which the node can process programs.

Working set maximum. Computing nodes have a fixed amount of high-speed memory, and this defines the size of the largest program that a node can execute without degradation of performance.

Average latency. This factor is a measure of the performance of a computing system's storage resources. It refers to the time between when the data is requested and when the data is available.

Maximum I/O rate. This rate may vary, but for any given storage element there is a maximum number of I/O operations that can be processed per second. Once this limit is exceeded, storage latency begins to increase because of channel congestion.

For a technical discussion of these four factors, please see the appendix at the end of this paper. The important thing to note at this point in our discussion is that these four factors should be carefully analyzed when deciding how best to improve the execution speed of a particular application. The best solution is not always simply adding more processing power or upgrading to a bigger server with more CPUs. In fact, if the performance bottleneck in a given architecture is a result of data latency, overall system performance might even be degraded, instead of improved, by adding more processors, since the result might be an increase in the I/O request workload on the already stressed storage system. The challenge comes with achieving optimal resource utilization; this leads to maximum throughput. When every compute node and storage node is operating at its optimum efficiency, the system is balanced.

6) Challenges in Designing and Maintaining a Grid

It would probably be relatively straightforward to design a grid system that optimizes scalability or availability or serviceability. Providing for all three presents challenges like the ones listed below. NetApp solutions address all of these challenges.

The implementation of increasingly complex applications that require larger amounts of data, or an increase in the number of applications running and accessing data, results in a need to scale performance and capacity well beyond that of a single file server.

> Linux compute farms often include more than a thousand compute nodes. A single file server, no matter how fast, cannot keep up with the increasing demand of a compute farm.
>
> With NetApp solutions you can add new storage nodes as needed, without disrupting processing. This means you can plan the optimum storage configuration for your present needs and then add capacity when it's needed; you don't have to anticipate your precise future needs in advance. This allows you to maximize resource utilization at every stage of growth.

It is difficult to manage data sets and storage spread across many file servers.

> Since the demand for storage capacity eventually cannot be supported by a single file server, organizations deploy multiple file servers. This creates management challenges.
>
> NetApp provides unified management tools that enable administrators to manage hundreds of storage nodes as a single storage appliance.

The priority of jobs and data is constantly changing.

> A Linux compute farm typically uses a centralized resource for multiple projects (or jobs) simultaneously. In a semiconductor company, for example, multiple chip designs are usually occurring in parallel. At any given time, the priority of some jobs will be higher than that of others. The storage system must provide a way to prioritize data access for the highest priority jobs.
>
> NetApp solutions make it easy for administrators to reallocate access priorities or, if necessary, to reallocate storage to enhance access to certain data—with no impact on the applications and processes that are running.

Hot spots are created in the file server farm as a result of changing application priorities or workloads or from any other circumstances that result in a spike in the demand for a particular set of data.

> NetApp solutions provide capabilities for recognizing and clearing up storage hot spots (or avoiding them in the first place), for rebalancing storage deployment, for adding storage capacity on-the-fly, and for moving selected data from one place to another. All of these operations are carried out with no system downtime and with no impact on running applications.

It is difficult to plan downtime to make changes to the storage infrastructure without impacting applications.

With a traditional file server architecture, if the data stored behind a single file server becomes very hot, downtime is required to bring on a new file server to provide additional file-serving capacity. The downtime—which can be substantial—occurs when data is moved from its original file server to the new file server. To compound the problem, in a traditional architecture, all the Linux clients that use the data must be reprogrammed to find (mount) the data in its new location. With hundreds or even thousands of Linux clients in a compute farm, this can be an extremely complex task. It is especially problematical in some environments, where changes like this are necessary on a daily basis.

With NetApp solutions, no downtime is required for making infrastructure changes. And from the point of view of client computers, the movement of data from one storage location to another is fully transparent: the data is still available through the same mountpoint, and no reprogramming is necessary.

There must be fast recovery from user-created data corruption.

Mistakes and accidents are by nature unpredictable, and organizations cannot afford unscheduled downtime to make recoveries. NetApp solutions incorporate software that provides for fast, efficient protection of data and for transparent restoration of data without any downtime. In many cases, users can even restore their own files without the assistance of the IT department, leading to faster problem resolution, lower TCO, and greater ROI.

7) Network Appliance Solutions

Previous sections of this paper have discussed the importance of analyzing the complete system when designing and architecting a grid solution. Network Appliance offers a range of storage solutions, best practices, and services to support the deployment and operation of grid-based computing architectures. The right solution depends on various factors such as the number and type of nodes in the grid, the network design, the amount of data, the pattern of access, and the bandwidth/latency necessary to meet the business requirements.

A few highlights of NetApp solutions for grid computing environments include:

NetApp FAS systems. These fabric-attached storage (FAS) systems integrate easily into complex enterprise environments. They provide shared access to data while simultaneously supporting Fibre Channel SAN, IP SAN (iSCSI), and NAS. These high-performance systems have proven their ability to continuously serve data at higher than 99.99% availability and can scale from 50GB to hundreds of terabytes.

NetApp SharedStorage. This solution enables multiple FAS systems to share storage resources through a fabric. The fabric simplifies and accelerates storage provisioning across FAS systems while improving resiliency and increasing flexibility.

DNFS. DNFS allows flexible and automatic replication of read-heavy data from multiple FAS systems, thus eliminating data hot spots in read-heavy environments.

Additionally, the recent acquisition of Spinnaker Networks will allow NetApp to deliver a new generation of highly scalable solutions for grid computing environments with features such as global namespace, transparent data migration between FAS servers for load balancing, and industry-leading clustering technology that can be used to scale storage grids to petabytes capacity and hundreds of gigabytes per second of bandwidth. Please refer to the following paper for more information: www.netapp.com/tech_library/3304.html.

All three solutions are designed for high-performance computing requirements within industries such as energy, entertainment, software development, semiconductor manufacturing, and biotech. These types of organizations depend on NetApp solutions to significantly increase performance and availability while reducing the administrative requirements of running grid computing environments.

Additional information about those products is available at www.netapp.com/products/filer/index.html, and best practices for deploying the FAS line of products with Linux compute nodes can be found at www.netapp.com/tech_library/ftp/3183.pdf.

Note: The summary descriptions provided here are meant to introduce the solutions in the context of the discussion of grid computing. Detailed papers and presentations are available on the NetApp Web site. See especially the NetApp online library at www.netapp.com/tech_library.

SpinServer

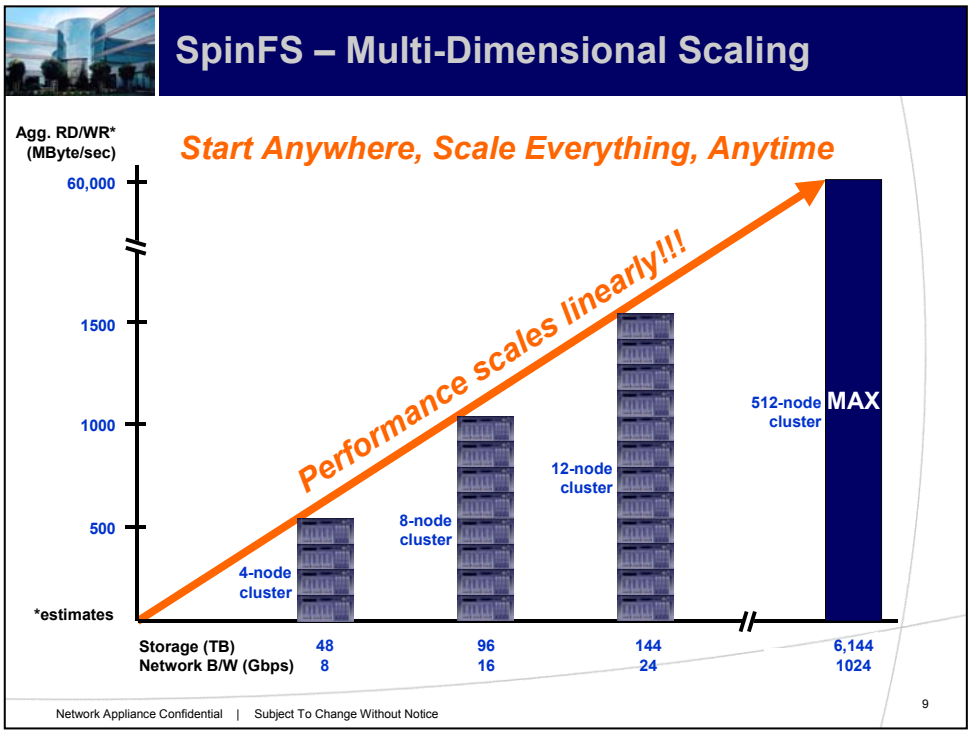
Network Appliance SpinServer is a flexible, high-performance grid storage system that can scale to encompass a cluster of as many as 512 storage servers and a total of 11PB (11,000TB) of data in a single installation. This scalability enables SpinServer to provide a tremendous amount of aggregate bandwidth to support the most data-intensive computing environments. SpinServer is delivered as a full solution with software, storage server platforms, and storage.

Massive Distributed Storage under a Single Namespace

SpinServer makes the storage capacity of the entire distributed cluster visible to clients under a single namespace. This unique feature of the SpinServer architecture allows network clients to access all of the storage resources with a single mount. A network client sees a single large file system and can access any part of the global namespace without having to mount the specific SpinServer node that stores the desired data.

Unlimited Scalability

SpinServer software provides for multidimensional scaling, meaning that the aggregate bandwidth available from the single global namespace scales linearly with the number of systems in the cluster. This makes SpinServer ideal for applications where high bandwidth is critical. As SpinServer nodes are added to a cluster, all the physical resources (CPUs, cache memory, network I/O bandwidth, and disk I/O bandwidth) are automatically kept in balance.



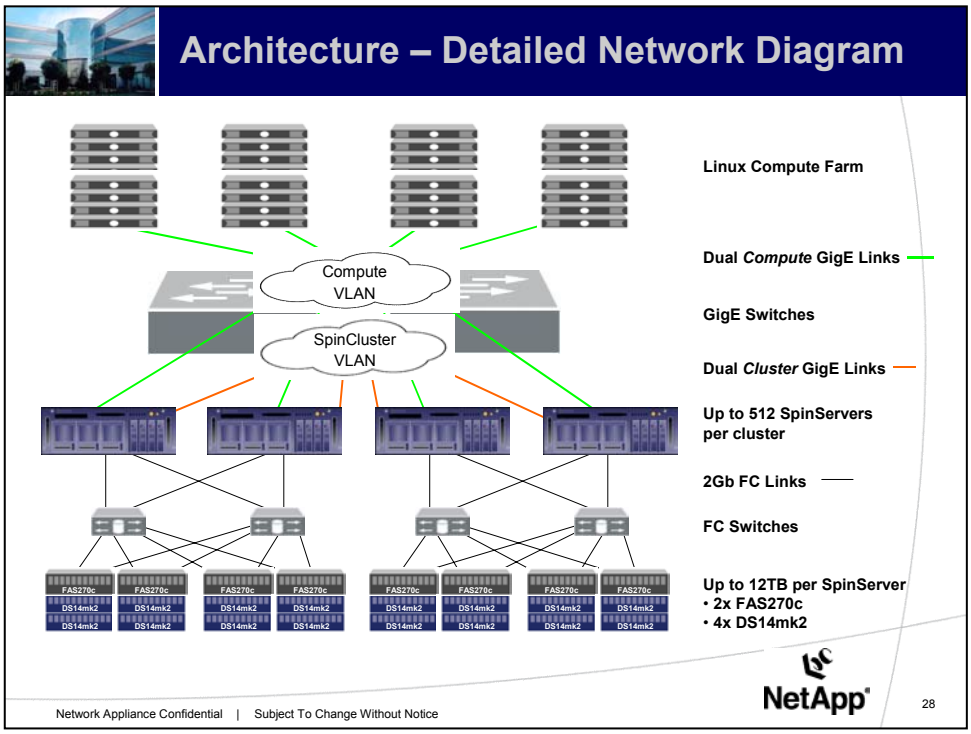
Advanced load-balancing capabilities and superior ease of use make SpinServer ideal for high-performance computing (HPC) installations such as grid environments with thousands of Linux compute servers. Virtual file systems (VFSs) stored on individual nodes can be transparently migrated between servers to balance workload and optimize performance without disrupting ongoing work.

High Availability and Bandwidth

SpinServer has a two-stage architecture that includes a dedicated interconnect (Gigabit Ethernet) for network processes and another dedicated interconnect (2Gb Fibre Channel) for disk processes.

When an NFS request comes to a SpinServer node, if the requested data is local to the SpinServer node that receives the request, this SpinServer node serves the file. If the data is not local, this SpinServer node routes the

request over the switched network to the SpinServer node where the data is stored, and that SpinServer node serves the requested data to the client. The routing of the request is completely transparent to the client. Each SpinServer node is only one hop away from any other SpinServer node in the cluster, across a low-latency cluster network.



Ease of Manageability through a Unified System View

SpinFS® enables the storage administrator to manage the entire SpinServer cluster as a single appliance. The administrator can think of each SpinServer node as a building block in a single modular appliance.

Adding or Reassigning Capacity without Disruption or Downtime

In any computing environment, an application's need for storage, and the priority in which applications must be served their data, can often change. Similarly, "hot spots" can develop in a storage system when there is a sharp increase in demand for access to a particular file or set of files. The unique SpinServer SpinMove® feature enables the storage administrator to rebalance the storage environment and to move any desired set of data to any desired SpinServer node without interrupting processing. It is not necessary to change any code or scripts in any of the client computers; their logical view of the data is unaffected by the move.

Without disrupting the operation of the Linux compute farm, you can add storage, add SpinServer nodes (for performance), modify the configuration (for example, move an IP address from one SpinServer node to another for load balancing), and migrate data (for example, remove a "hot spot" from one SpinServer node by moving data to a less loaded or newly added SpinServer node).

Quick and Transparent Recovery from Data Corruption or Hardware Problems

The SpinServer solution includes software for replicating, backing up, and restoring data. SpinShot™, SpinRestore™, and SpinMirror™ constitute a suite of data protection software with a field-proven track record unmatched by solutions from any competitor.

SpinHA® (high-availability) software enables administrators to upgrade hardware or software without downtime. Similarly, in case of a hardware problem, the administrator can cause the data from one SpinServer node to be accessed automatically and transparently by another, so that the unit with the problem can be repaired or replaced. Client applications are not interrupted, and their access to data is unaffected during the move.

A Scalable Storage Solution for Present and Future Needs

SpinServer is a scalable storage solution that can meet the needs of the most demanding Linux compute farm. Its applicability is not limited *to* compute farms. The SpinServer solution can also be used in more traditional configurations such as server-centric architectures. This makes SpinServer an especially strong solution for managing the transition from present-day configurations to grid computing.

Distributed Network File Service

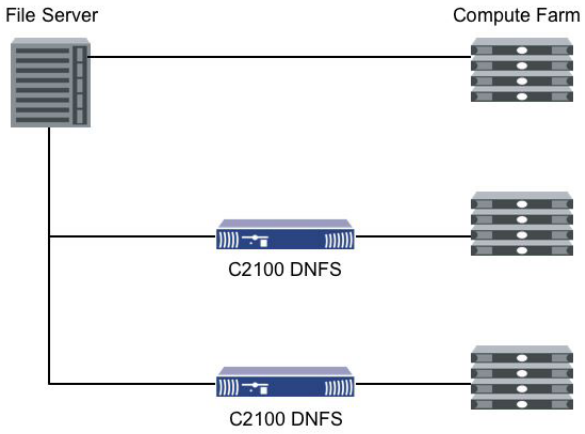
DNFS provides rapid access to shared UNIX® files for enhanced collaboration in distributed environments. This enables organizations to reduce management and infrastructure costs by automatically replicating, storing, and serving data that is requested over NFS.

Problems That DNFS Solves

In attempts to speed up the processing performed by the compute farm, computational capacity is typically increased by adding new blades or servers to the farm. The performance benefits of this kind of expansion eventually “hit a wall” when the file server becomes the overall system’s performance bottleneck. IT managers often respond by deploying additional file servers. This approach adds complexity to the storage infrastructure, and the tasks involved in keeping data synchronized across the file servers impair the performance of the system as a whole. In such a situation, DNFS appliances can be used to increase overall performance while minimizing administrative complexity.

DNFS appliances yield a significant performance increase in environments where workloads involve a high proportion of read operations, such as animation and special effects, oil and gas seismic studies, biotech research, and semiconductor design. For environments with a high proportion of write operations or read-once operations, other solutions may be more effective than DNFS.

Organizations that currently have a data center networked with remote offices can use DNFS to make the transition to grid computing when desired, because DNFS supports both environments.



DNFS appliances are deployed in conjunction with the file servers in a compute farm. This architecture increases the throughput to the clients to overcome bottlenecks without additional management complexity.

DNFS Manageability

DNFS appliances integrate seamlessly into an existing NFS infrastructure. Clients that previously mounted their data volumes exported by file servers will instead mount them from a DNFS appliance. With support for NIS domains, automounter, and NLM lock management, the use of DNFS is fully transparent to NFS clients.

For deployments using only a few DNFS appliances, administrators can use the user-friendly Web and command-line interfaces for configuration management. For larger deployments, NetApp DataFabric® Manager can simplify the monitoring and configuration of deployments with tens to hundreds of appliances. Information about DataFabric Manager is available at www.netapp.com/products/software/datafabric.html.

NetApp SharedStorage

NetApp SharedStorage provides a shared storage infrastructure to enable workload migration. NetApp SharedStorage shares many of the same capabilities of SpinServer. In the next 18 months, the two solutions will converge into a single solution that includes the advantages of both.

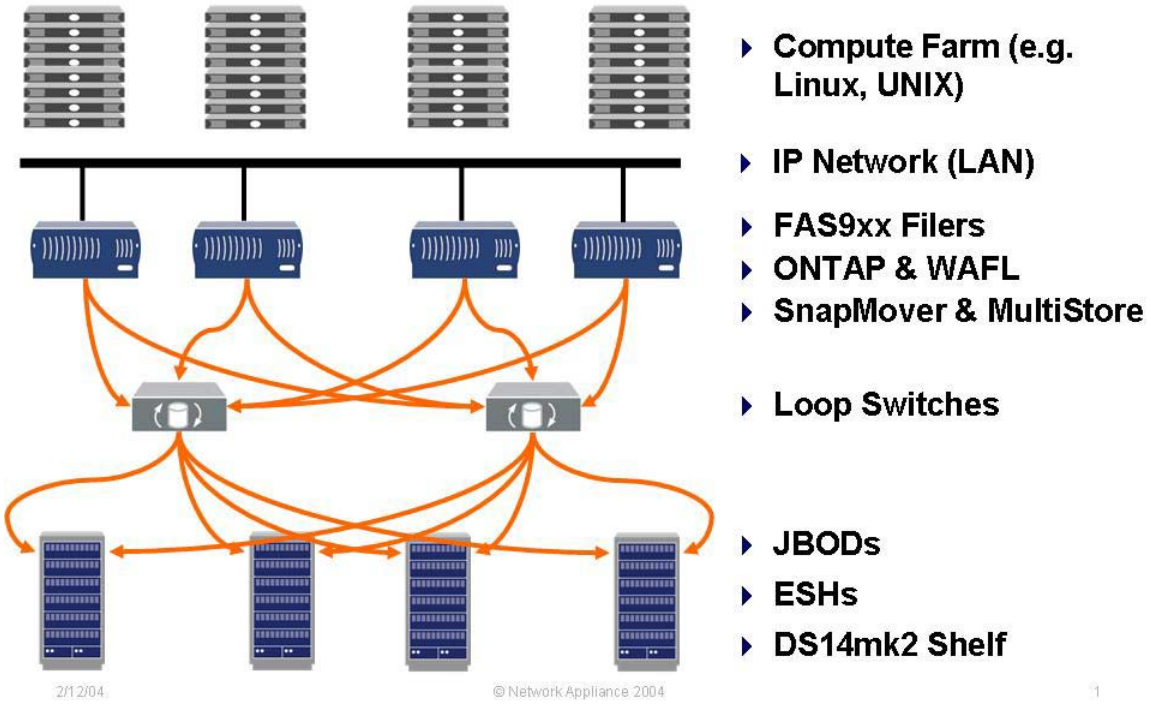
NetApp SharedStorage is an attractive solution for organizations that have NFS compute farms, already have NetApp storage appliances with Data ONTAP™, and have storage infrastructures that are not yet so complex that global namespace is a near-term requirement. (SpinServer offers global namespace today; NetApp SharedStorage will offer it in mid-2006.) As the convergence of SpinServer and NetApp SharedStorage proceeds, customers who deploy either solution will be able to migrate seamlessly to the converged solution.

NetApp SharedStorage optimizes performance via workload migration: when the workload gets out of balance, and storage hot spots are created, the file-serving workload can be moved from a hot filer to a less busy filer without requiring any downtime.

NetApp is working with Platform Computing to provide an integrated solution for customers who run, or want to run, a compute grid using platform LSF.

If hot spots or potential hot spots are detected, LSF suspends jobs that would increase hot spot intensity. This avoids oversubscription of filer heads.

LSF can trigger SnapMover® to move hot storage resources to devices that are less busy. LSF can notify the administrator, and the administrator can initiate the desired move. If hot spots continue consistently in the same pattern, LSF can trigger SnapMover to automatically move the pertinent resources.



NetApp SharedStorage architecture.

MultiStore® Features in the NetApp SharedStorage Solution

MultiStore is an optional software solution that enables secure, multiprotocol storage consolidation across enterprises. It provides secure partitioning of network and storage resources and enables multidomain and multiserver consolidation on a single storage appliance. In addition, it reduces management costs by introducing a tiered management model.

MultiStore Features

Advanced virtualization functionality

Logical partitioning of a single filer's network storage resources

Multiprotocol storage consolidation

Multiple customer storage consolidation

MultiStore Benefits

Effortless and secure storage consolidation

Simplified management for consolidated environments

Secure multidomain storage consolidation

Tiered management model to reduce administrative complexity

Storage Grid Loop Switch Features

Multiinitiator connectivity to shared storage pool

Dual active paths to each disk allow greater resiliency and performance

Best practice cabling configuration

Storage Grid Loop Switch Benefits

Sharing flexibility: physical connectivity means any filer has the potential to access any disk in the shared back end

Grid-based scalability: scalable cabling design enabled

Allows customers to efficiently add filers or storage as their CPU and capacity needs grow over time

Scalable performance: loop switches implement cut-through routing for higher throughput

Platform LSF Adapter for NetApp Storage

If you add the platform LSF adapter to the NetApp storage solution, the LSF plug-in scheduler monitors the NetApp storage resources to detect data hot spots. The monitoring is configurable to detect hot spots based on filer CPU utilization, capacity utilization, and the number of jobs scheduled per mount. If hot spots or potential hot spots are detected, LSF suspends jobs that would increase hot spot intensity. This avoids oversubscription of filer heads.

LSF can trigger SnapMover to move hot storage resources to storage that is less busy. LSF can notify the administrator, and the administrator can initiate the desired move. If hot spots continue consistently in the same pattern, LSF can trigger SnapMover automatically to perform a move.

8) Tips for Managing Grids: Scenarios and Case Studies

This section presents several scenarios of creating or scaling a computing grid. For each scenario, one or more challenges are described along with solutions for overcoming them. The discussions are not meant to be exhaustive in terms of possible scenarios and solutions, but they should illustrate how NetApp solutions can take any computing architecture to the next level of performance. NetApp technical personnel would be happy to discuss your particular scenario with you and help determine the best ways of dealing with whatever performance challenges you are experiencing.

Managing the Initial Deployment of a Grid

A Transitional Step: Moving from Direct-Attached Storage to NAS/SAN

Scenario: Your current configuration is based on one or more centralized computing servers with direct-attached storage, and you are considering the possibility of scaling out to a grid configuration. One way of starting the reconfiguration is to deploy a NetApp storage solution to offload storage-related responsibilities from your server. You can do this and take advantage of significant increases in system throughput and performance, while still retaining your existing servers. Later, if your computing power needs outgrow the computing capacity of the server, you can deploy a compute farm instead of scaling up your server horsepower. Your NetApp storage solution will continue to serve you through that transition, protecting your investment and helping to maximize the utilization of your computing and storage resources at every step along the way.

NVIDIA is an example of a major organization that successfully transitioned from server-attached storage to a NetApp storage solution. In addition to solving storage-capacity challenges, this solution resulted in a boost in the efficiency and performance of NVIDIA's servers, which no longer had to use CPU cycles to manage storage. "Ensuring our IT systems can keep up with the company's continuous, rapid growth is the biggest challenge we face," says Ed Yee, NVIDIA's director of IT Operations. "We realized that using local storage on our big Sun enterprise servers takes away horsepower they need to process massive engineering jobs. We fixed that by deploying the NetApp systems, which offload NFS file operations and allow the servers to use their CPU cycles for processing jobs."

NetApp Snapshot™ functionality also simplifies administration for Yee's staff. A unique function of the NetApp Data ONTAP operating system, Snapshot technology creates read-only versions of a file system. Administrators can perform online backups using the Snapshot feature with minimal disk space, and end users can recover lost or deleted files online, without assistance or recovery from tape. "We create Snapshot copies according to the default schedule, which is every four hours," Yee says, "and make the Snapshot directories visible to the engineers. When they need to retrieve an earlier version of a file, they do it themselves, with no intervention on our part. That saves time for everyone."

The full NVIDIA case study is available at www.netapp.com/case_studies/nvidia-ns.html.

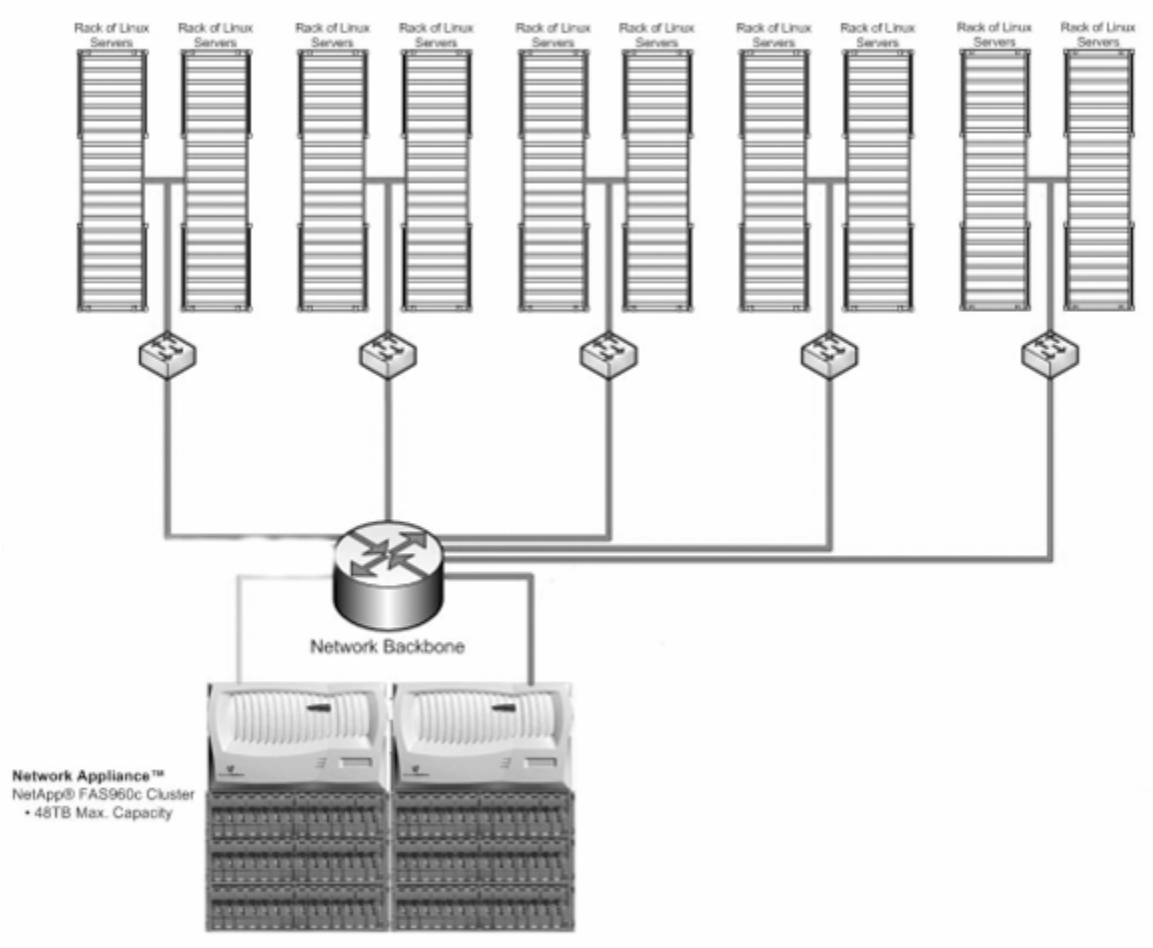
Scaling an Existing Grid

Deploying a Storage Grid when Storage Hot Spots Strain a Filer's Performance Capacity

For small farms of Linux systems, a single filer is normally adequate to handle the storage for the farm. But in a scenario such as the one described here, storage hot spots can develop, creating a negative impact on overall system performance.

The following diagram shows a typical Linux farm implementation. Racks (or small groups) of Linux servers are combined into discrete network segments that have an uplink to a backbone. The filers are also connected to this backbone, typically with multiple Gigabit Ethernet network cards to provide adequate bandwidth.

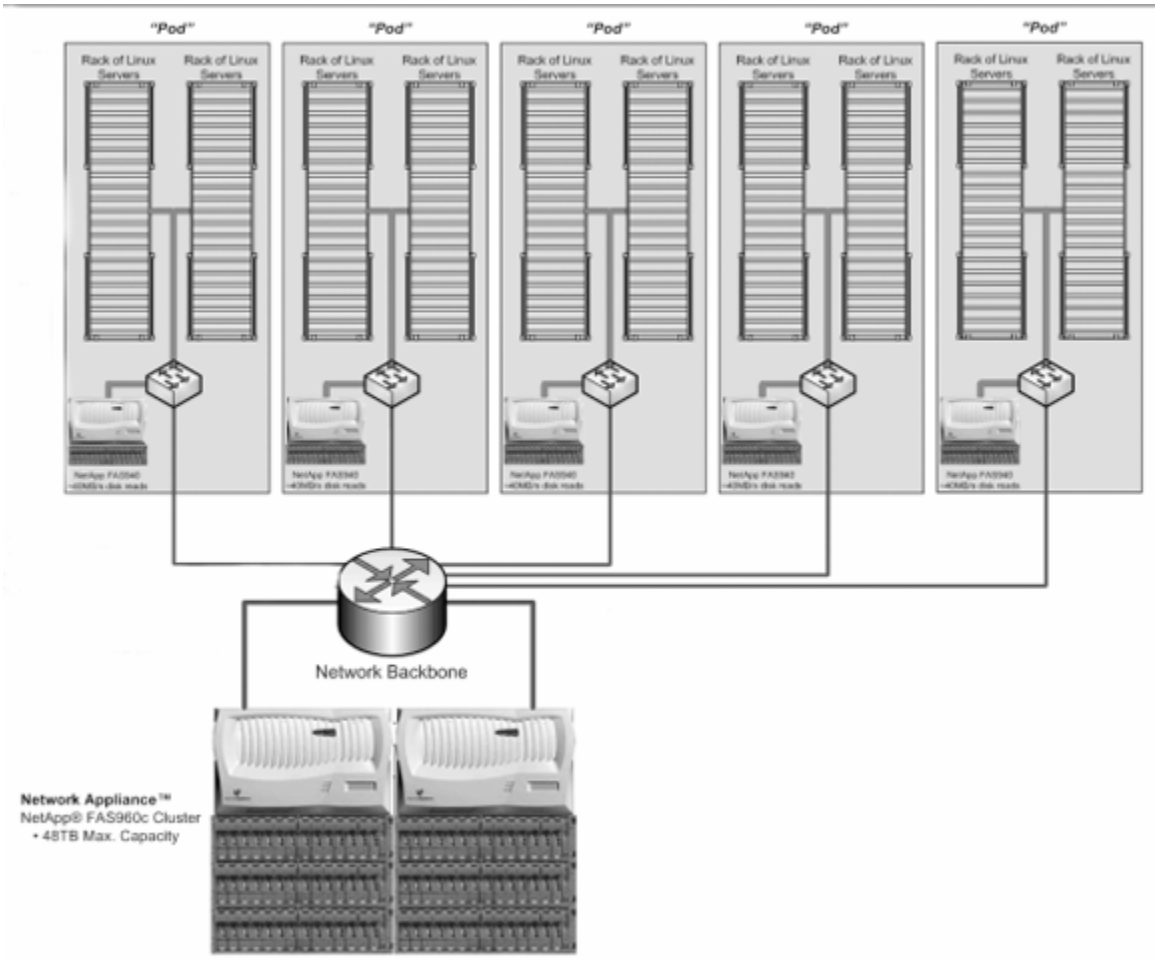
This architecture has an obvious limitation: the entire farm is limited to the throughput of a single filer. Although filers normally do their jobs well, there is a point of saturation when hundreds or even thousands of clients are requesting the same data. The filer will encounter a CPU and/or disk limitation, so that when more clients are added to the farm, system throughput does not increase and may even decrease.



Linux farm networked to a pair of filers.

Scenario: Your current computing configuration is a Linux compute farm with a large number of computing nodes. Storage for the system is handled by a clustered pair of filers such as the NetApp FAS960, connected to the compute farm over a network backbone. A storage hot spot is occurring as a result of a very large number of read requests being issued by a very large number of clients—all trying to access the same relatively small region of the same particular file. The filer where the file is stored is being pushed to its performance limit, and the result is that storage access is becoming a bottleneck for the performance of the computing grid as a whole.

This hot spot–bound scenario can be addressed by deploying a NetApp solution. A possible architecture is shown in the following figure.



Linux farm arranged in “pods” with NetApp storage solution.

In the revised architecture, each “pod” still contains a certain number of Linux hosts and a dedicated switch. Now each pod also contains one FAS940 filer and SnapMirror® software. The FAS940 provides a high-speed local copy of the most requested input files for each pod so that the origin filer can be offloaded from a large proportion of the read requests of the farm. These requests are now served by a pod’s local filer, without exchanging traffic with the origin filer.

By adding to each pod an FAS940 with SnapMirror in heavy read environments such as this one, the farm has much more read bandwidth available to the application in aggregate. Even when such an architecture contains only a small number of pods, the available bandwidth scales far beyond the capabilities of a single filer. Also, the possible number of clients in the farm becomes much greater, since each pod includes its own copy of the heavily accessed data.

Evolving a Hybrid Architecture into a Grid

GX Technology (GXT) is a seismic processing services company based in Houston, Texas. For some time, GXT has been deploying clusters of Linux blade servers to provide flexible, economical compute capabilities for several key business processes. When the Sun servers used for seismic processing services began to reach their performance limits in the face of increased processing loads, it was natural for the organization to begin moving these activities to a grid-like configuration with Linux servers and consolidated network storage.

Seismic processing is extremely compute- and data-intensive, and GXT must complete its processing activities quickly to meet customer commitments. The company was faced with these challenges:

- Expand processing capability
- Improve efficiency

Consolidate direct-attached storage from 10 Sun servers

After a careful evaluation, the organization selected NetApp SpinServer to meet its storage needs. SpinServer was evaluated against competitors and found to offer not only superior reliability and performance, but also a variety of other features that made it the ideal choice:

Cost-effective grid storage

Single global namespace

Ability to dynamically balance the load across storage systems for maximum performance without user disruption

Deployment Results

The deployment increased production capacity and streamlined management tasks, enabling redeployment of one IT professional. This resulted in lower storage/maintenance costs and a tenfold price/performance improvement.

Migrating from the Sun Servers to the Grid

Migrating several hundred terabytes of storage from Sun servers to SpinServer was a significant undertaking for the company, made even more complicated by limited physical space in its busy data center—plus the need to keep all projects moving forward.

To complete the migration, the organization purchased 25TB of additional storage capacity. All existing storage arrays were “recycled” as part of the SpinServer configuration, a utilization of existing resources that yielded significant cost savings and investment protection.

Global Namespace in the SpinServer Cluster

For its seismic processing deployment, GXT has a single SpinServer cluster, which currently consists of 16 cluster nodes, each with about 15TB of storage, for a total cluster capacity of about 250TB. The storage manager allocates storage at the outset of each project. Each VFS is configured to be about 1TB in size. VFSs are arranged within the global namespace to meet the needs of each project. A low-priority project might draw all its storage from one or two cluster nodes, while a high-priority project might utilize VFSs on all cluster nodes to ensure maximum throughput. If a project unexpectedly requires more storage, it can be flexibly allocated from free space on any of the 16 servers and made available in the correct location in the global namespace.

Workload Balancing

Because of the workload-balancing features of the SpinServer architecture, it is easy for the storage manager to adjust the storage configuration for a project to increase performance. Since the location of a VFS in the global namespace is completely independent of the node storing the VFS, a VFS move operation allows the contents of a VFS to be moved from one server to another at any time. The first stage of a VFS move copies the contents of the VFS over the cluster interconnect. Read and write operations occur with no disruption and are directed to the original location. These updates are propagated to the new location during the second stage. The final stage of the move operation transfers file-locking information between the source and destination servers, at which point the new VFS becomes active, and the old VFS is released. The names and locations of the files in the global namespace do not change as a result of the move, nor do any mountpoints become invalid.

Ease of Management

A typical seismic processing project lasts only a few months, so project setup and teardown occur frequently. This is another area where the SpinServer architecture is a major advantage for GXT. Both physical and logical storage configuration can be done from an administrator’s desktop, since everything is done in software. Reconfiguring for a new project takes less than 30 minutes. During daily operation, GXT simply monitors the performance of its SpinServer nodes and makes adjustments with VFS move when necessary, without downtime or disruption to ongoing work.

When more storage bandwidth is required, another SpinServer node can be added to the cluster without bringing the cluster down, and workloads can be transparently migrated to that system, which becomes part of the global namespace. If an individual system requires maintenance, its VFSs can be transferred to the other cluster members (assuming they have enough free storage space) so the system can be serviced without disrupting operations.

Integrating a Compute Farm with a Server-Based Environment

This scenario is an actual case study of a NetApp deployment carried out by PDI/DreamWorks in Northern California. The complete case study is available at www.netapp.com/case_studies/pdi.html.

Creating animated armies of ants, talking donkeys, and green ogres for feature films demands supercomputing—plus processing power and a storage solution to match. PDI/DreamWorks tried using high-performance UNIX servers for file serving, but they weren't up to the task and demanded constant, time-consuming tuning. To successfully meet the performance, reliability, and scalability demands of its thousand-CPU render farm, PDI deployed clustered pairs of filers from Network Appliance.

Switching from General-Purpose Servers to NetApp Filers

Established in 1980, PDI/DreamWorks is located in Northern California and employs nearly 350 people. It is an award-winning producer of high-end animation and visual effects for the feature film, advertising, and entertainment industries. PDI/DreamWorks' Commercial and Feature Effects Division provides character animation, visual effects, live action, and postproduction solutions for advertising agency and entertainment clients. PDI/DreamWorks' Feature Animation Division produces full-length, computer-animated feature films for DreamWorks Pictures. Following the success of its 1998 hit movie *Antz*, PDI/DreamWorks completed production on its second animated feature, *Shrek*, which was released in May 2001.

PDI has three categories of computing activities, each placing different requirements on its computing and data storage resources. The first is standard office and administrative computing, where users run Windows NT® applications on their desktops and store files in CIFS format. The second is work done by animators and artists using UNIX or Linux workstations and generating large data sets, which are the input data for scenes in an animated film. The third is the processing done on the input data by the render farm—a very large batch-queuing system consisting of 500 dual-processor workstations that render the data from the animators to produce images.

PDI tried using general-purpose UNIX servers for data storage and file serving for all these activities, but the devices required too much maintenance. "They are difficult to manage in an environment like ours, which places extreme stress on servers, the network, and storage devices," says Shoshana Abrass, head of Systems, PDI/DreamWorks. "We need to spend our time and money making our animators more efficient rather than tuning our servers."

After evaluating and benchmarking alternatives, PDI deployed Network Appliance filers. The company has already installed multiple pairs of filers comprising 7TB of storage in clustered failover configurations and has plans to purchase more. Key reasons PDI selected NetApp include very low maintenance; an integrated, highly reliable design; and superior performance and scalability.

"The integrated design is an important advantage," says Abrass. "We examined other file-serving products and found they're simply a collection of hardware and software we, ourselves, could put together. If you have a problem with such a device, you may need to work with as many as four vendors—the server vendor, disk vendor, file system vendor, and integrator—to troubleshoot it. NetApp filers aren't like that—they're an integrated, well-supported hardware and software device from a single vendor."

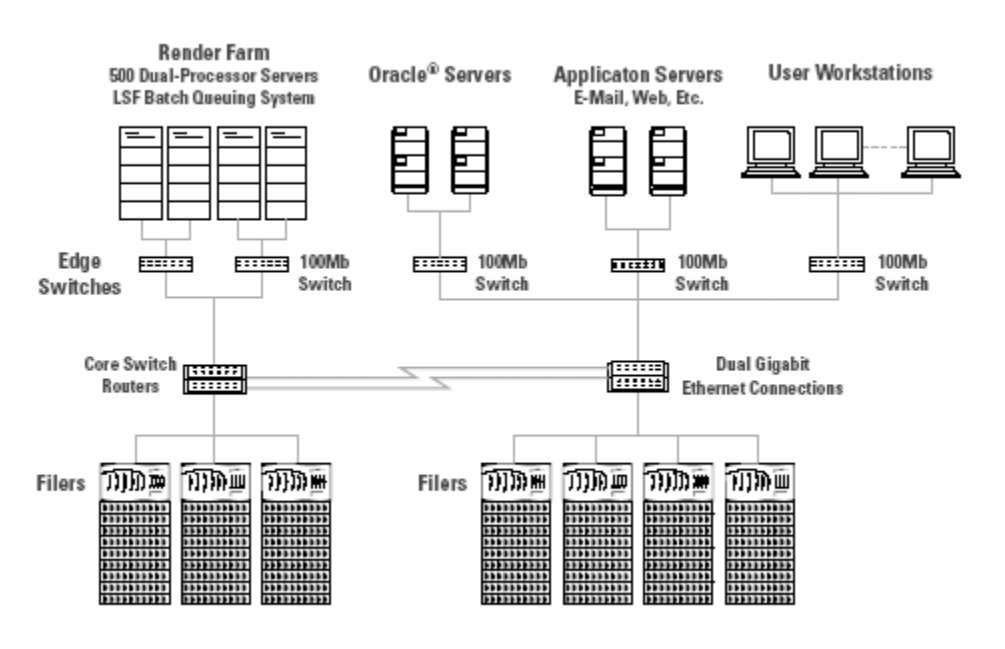
Performing under Peak Loads

At PDI, each animator creating a scene for a film works on a data set that is typically 100MB to 150MB, which the animator edits and saves and reedits throughout the workday. Hundreds of animators working together on a project continuously store and retrieve such data sets from the filers. Data sets are combined with other data defining sound, motion, and lighting, which is all turned into a 3D animated image by the render farm, which operates 24x7.Once finished, a typical frame of film represents about 12MB of data. Film is projected at the rate of 24 frames per second, so a single minute of film represents about 17GB of data, and one hour of film represents a full terabyte.

Needless to say, storing and serving all that data, and doing it quickly and reliably enough to keep a staff of hundreds of highly skilled, well-paid animators busy and productive, are major IT challenges. PDI's filers help it meet the challenges by storing and serving data for all its computing activities. Some of the filers store home directories and the binary code for custom-developed application programs. Another pair stores the data sets generated by the animators. Other filers also store these data sets as well as data that's in high demand from the render farm. All the filers are in a clustered failover configuration, which delivers the highest level of availability by implementing failover at both the hardware and software levels. Clustered failover works by "failing over" data service to the partner filer in the event that one of the filers becomes unavailable. The failover ensures that data remains accessible and customers experience minimal, if any, inconvenience.

The PDI IT architecture consists of filers connected via two Gigabit Ethernet connections to a set of core switch/routers, which are connected to several edge switch/routers, which are connected via a 100MB network to

the animators' workstations and the render farm. Like many data-intensive enterprises, PDI stores multiple terabytes on its filers and changes those terabytes several times a day as the animators submit and edit data sets and the render farm generates rendered scenes.



The integrated architecture at PDI/DreamWorks.

"The storage devices can easily become the bottleneck in our system, but the filers prevent that from happening," Abrass says. "If there's going to be a bottleneck, we want it to be the user. That is, we don't want animators sitting idle waiting for a scene to load; we want the system to respond more quickly than they're able to work. Our architecture makes that possible." Abrass says the filers respond even under peak load conditions, such as when hundreds of workstations in the render farm all write at the same time to the same filer. "In that scenario, many NFS servers bog down and lock up, requiring the administrator to manually intervene," she says. "That never happens with the filers, and it's a good thing, because we can't predict when we're going to hit such a rendering crunch."

Snapshot Enhances Department's Image

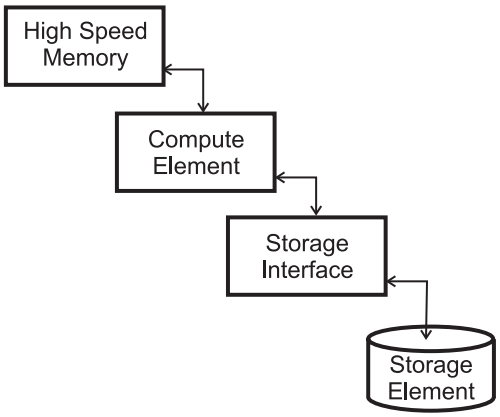
PDI uses NetApp Snapshot software several times a day to back up files and enable users to restore their accidentally changed or deleted files. A unique function of the Data ONTAP software, Snapshot stores up to 254 read-only versions of a file system. Administrators can perform online backups using the Snapshot feature with minimal disk space, and end users can recover lost or deleted files online, without assistance or recovery from tape. "As systems people, we're aware of our PR within the company," Abrass says, "and when there's an IT problem, it can hurt our image with employees and management. But a service like Snapshot is so popular, it gets us positive PR. Users often thank us for it after they've recovered files, and we didn't even have to do anything. So, in addition to its practical benefits, Snapshot can really help user relations."

According to Abrass, the superior reliability and performance of the filers play important roles in helping PDI meet a mission-critical business objective—releasing its feature films on time. "Unlike a traditional software company that has a release date that often slips a bit, we absolutely can't miss the release date for our movies," she says. "Film-printing facilities are booked months in advance, and so are theaters. If we miss a date, it costs us a lot of money and a lot of goodwill. We don't have the luxury of saying, 'The file server was down a few days last week, so we'll just deliver the film later.' Fortunately, we're never put in that position, because our downtime with the NetApp filers is functionally zero, and that helps us deliver our products on time."

9) Appendix: Analyzing the Factors That Affect Computing Performance

The fundamental unit of analysis for evaluating a computing architecture is a system node, which, for purposes of this discussion, is defined as referring to the four-element entity described below.

The required elements for a system node are shown in the following figure. These elements are a compute element (the central processing unit), high-speed memory, a storage interface, and stable storage for storing intermediate results.



The essential elements of a system node.

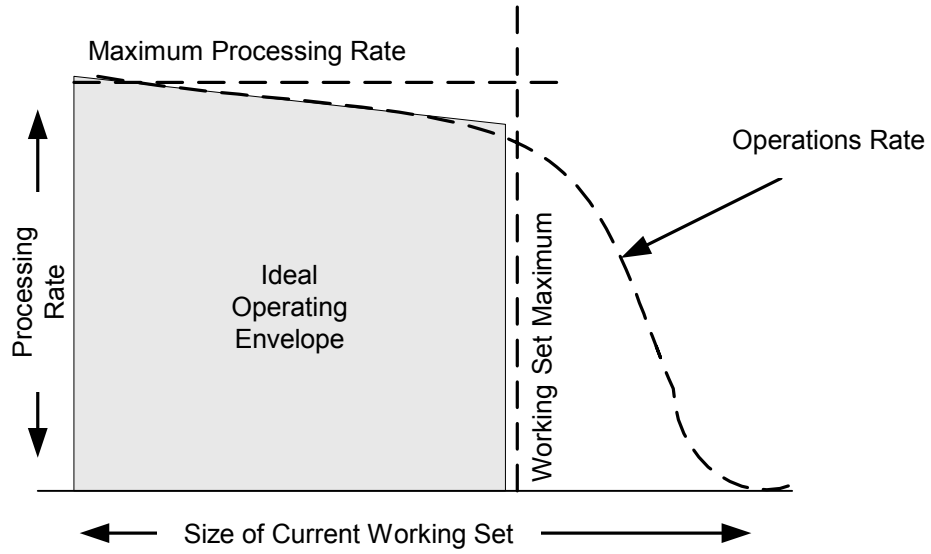
First Factor: Maximum Processing Rate

System nodes can execute a number of programs. Those system nodes that are in the process of executing are said to be active, and those that are waiting to execute are said to be idle. For purposes of this model, we assume that there is a maximum rate at which the system node can process programs. Not all nodes necessarily have the same maximum rate, but for a specific system node the maximum rate will be constant. This rate is an important factor in the set of factors we need to consider in analyzing the efficiency of a computing architecture and in determining what and when to scale in order to improve performance. We will refer to this factor as the maximum processing rate.

Second Factor: Working Set Maximum

System nodes have a fixed amount of high-speed memory, and this defines the size of the largest program that a system node can execute without degradation. The set of instructions and data in memory at a given time is called the working set. The largest program that a system node can execute without degradation will be called the working set maximum, and it is the second factor for our analysis.

The relationship between the maximum processing rate and the working set maximum is graphically depicted in the following figure. A system node typically experiences a measurable but minor impact on its operations rate as the size of the working set increases. Once the working set exceeds the working set maximum, however, the falloff in the operations rate is exponential. This rapid falloff of performance is due to the system node using a much slower secondary storage mechanism such as magnetic disks to simulate a larger memory size.



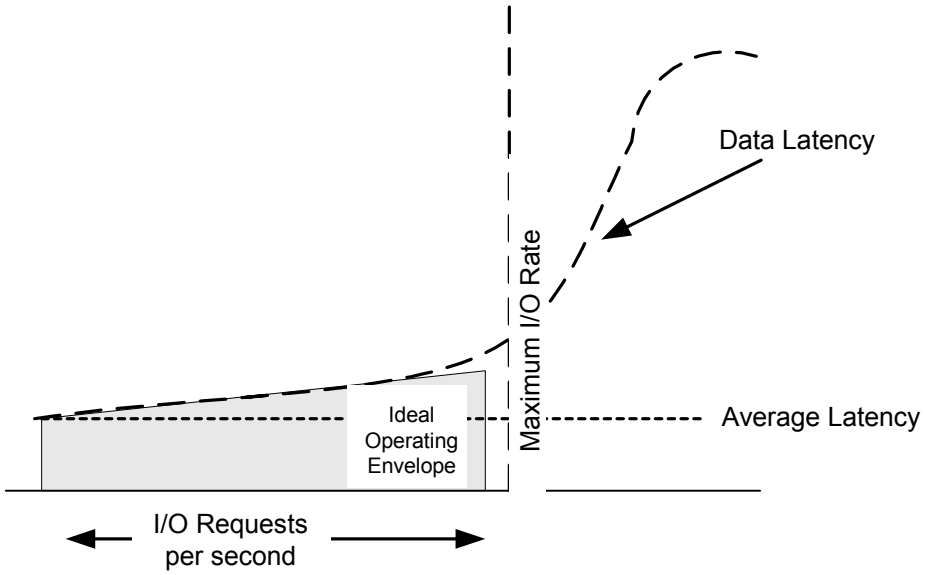
The area within the shaded portion of the illustration represents the ideal operating envelope for this system node.

Third Factor: Average Latency

An important factor associated with storage is latency, which refers to the time between when the data is requested and when the data is available. For example, with rotating magnetic media such as disks, there is an inherent latency associated with first identifying where the requested data is located and then waiting for that data to arrive under a read/write head. For any given request, the latency may vary significantly; however, for this discussion we will assume that randomness in the requests over time will yield an average value for a system node's storage element. This will be called average latency.

Fourth Factor: Maximum I/O Rate

The other important factor associated with storage is the input/output rate, or I/O rate, specified in I/O operations per second. This rate may vary from storage element to storage element, but for any given storage element there is a maximum number of I/O operations that can be processed per second. Once this limit is exceeded, storage latency begins to increase because of channel congestion. As with the compute elements, data latency begins to increase exponentially when the maximum I/O rate for the storage element is exceeded. This is illustrated in the following figure.



The area within the shaded portion of the illustration represents the ideal operating envelope for this storage element.

Jobs: Programs and Data Sets

The final concept that is needed to complete our analytical model is the one that ties storage and execution together. For this we define the notion of a job.

A job is composed of two parts: a working set and a data set. The working set consists of a set of executable instructions (a “program”) and a set of data structures. The data set portion of the job is broken up into the quantity of data and the number of I/O operations that will be executed in order to access the data set. These parameters define the total number of I/Os that will be initiated by the job during the course of its execution.

The number of instructions in a program and the number of times instructions are repeated because of looping structures, etc. constitute a fifth factor for this model of analysis. And the sixth factor is the number of I/O operations a given job will perform. The speed at which the executable instructions in the working set can be executed is related to the operations rate discussed earlier, in connection with the factors of working set maximum and maximum processing rate. The speed at which the I/O operations of the working set can be executed is related to data latency, also discussed earlier.

If it is desirable to increase the performance of the computing system, in terms of executing a particular job or set of jobs, it is possible to do a detailed analysis of the interrelationships between all the factors we have discussed. The main point is that improving the execution speed of a particular application is not merely a function of adding more processing power or upgrading to a bigger server with more CPUs. In fact, if the performance bottleneck in a given architecture is a result of data latency, overall system performance might even be degraded, instead of improved, by adding more processors, since the result might be an increase in the I/O request workload on the already stressed storage system.