



I/O Considerations for Server Blades, Backplanes, and the Datacenter

Contents

Abstract	3
Enterprise Modular Computing	3
The Vision	3
The Path to Achieving the Vision	4
Bladed Servers	7
Managing Datacenter I/O	7
Summary	10

Abstract

This paper looks at modular computing objectives and identifies the emerging I/O properties necessary to achieve those goals. It highlights the differences in I/O architectures between server blades and traditional pedestal servers. It addresses management and properties of I/O resources such as sharing and access control as datacenters transition to a unified fabric and simplify server I/O to virtualize datacenter computing.

Enterprise Modular Computing

The trend toward modular computing is driven by the need to dynamically scale datacenter resources to meet changing business needs. It merges the concepts of virtualization, automation, and modularity with management software to define new degrees of manageability.

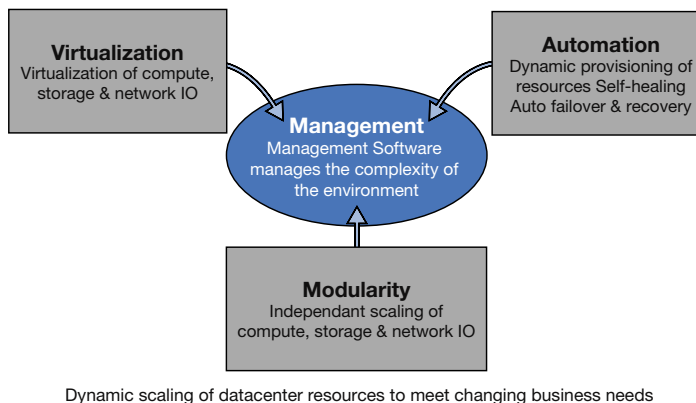


Figure 1 Modular Datacenter

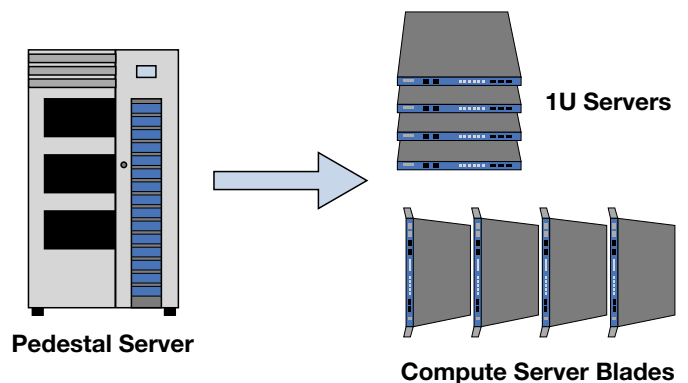
Storage virtualization is not new. File-based storage abstracts physical blocks into virtual files. For block-based storage, RAID¹ abstracts physical sets of disk drives into one or more virtual volumes. Automation is not a new concept either, however the degree of automation is growing and the need to dynamically re-provision resources as compute loads change is driving new requirements on modularity. Modularity enables compute, storage, and network resources to scale independently. The focus of modularity is on the server. That is, while storage and networking are very modular, traditionally servers have been designed for particular applications. Web servers have different I/O properties than application servers, which differ from database servers. The goal is to separate the I/O properties from the compute complex (i.e., CPU and memory) and thus morph server platforms into server modules that are extremely versatile. Pools of server modules will replace dedicated server platforms and the notion of a server platform changes from being a machine with its application and I/O, to being a set of compute

resources, applications, and storage that are associated to provide virtual server platforms.

The Vision

Transitioning the datacenter from an assortment of server platforms to pools of compute resources means disaggregating the server platform into server modules and fabric attached I/O. This is beginning, first with the introduction of 1U servers, and now server blades.

Figure 2 Transition to Modular Computing



Ideally, the server module's I/O is its attachment to the datacenter fabric. Networking and Inter Process Communication (IPC) are inherent in the datacenter fabric technology, and other I/O (primarily storage) attaches to the fabric so that each server module can be dynamically configured for its storage.

Thus, server modules become more universal and versatile. That is, a datacenter server will no longer be categorized by its number of I/O slots, number of Host Bus Adapters (HBAs), or number of SCSI² ports, since these characteristics move outside the server module. It is the universal nature of the server module that makes it a compute resource that can be dynamically positioned within the datacenter.

Once we take that step, any application can potentially run on any server module and we can start thinking of attaching applications to server modules instead of installing applications to a specific server platform. With that capability, one can envision that a server module being used for one purpose can quickly be re-configured for a different purpose, simply by attaching a different application and its set of storage. For example, in the morning when web surfing is low and email activity is high, a server module can be configured as an additional mail server. Then later, when web surfing activity increases, the server module can be re-purposed as an additional web server.

¹Redundant Array of Independent Disks, (a.k.a. Redundant Array of Inexpensive Disks)

²SCSI - Small Computer System Interface; an interface primarily for attaching storage devices. There are a number of physical SCSI interconnect such as Parallel SCSI and Fibre Channel (FC)

This re-purposing has several distinct advantages. Since server modules are generic, the datacenter requires fewer spare modules. Spares no longer have to be cold spares; they can be hot spares, ready to go, such that if a server module fails, its application can be switched to one of the standby modules, and this can be automated. Even more inviting, is that hot spares do not need to be idle. That is, the spare can be used to further distribute workloads for one application and if a server module running another application fails, the active spare server module can be quickly re-purposed for the failed module's application. Fewer spares equate to lower total cost of operation and active spares equate to a greater value for the investment.

The Path to Achieving the Vision

Removing I/O variables from the server module is paramount to achieving this vision and requires I/O to move outside the box and into the fabric. The key to this is the datacenter fabric.

A little background will help in understanding the role of the datacenter fabric. I/O primarily falls into three categories: Networking, Inter-Process Communication (IPC) and Storage. Storage is either direct attached (DAS) or fabric attached storage (FAS). Storage falls into two categories: file-based and block based. The most common block-based FAS technologies today are Fibre Channel and Parallel SCSI. Sharing of storage is very useful for distributed applications and clustering. Distributed applications refer to multiple instances of an application running on different platforms. Clustering refers to a single instance of an application running across multiple platforms. FAS usually permits direct sharing of storage resources. Sharing of DAS is less efficient since the access must go through the server to which the storage is attached.

Today's Datacenter

A typical datacenter today, as illustrated in Figure 3, contains various special-purpose servers interconnected by an IP network (typically Ethernet).

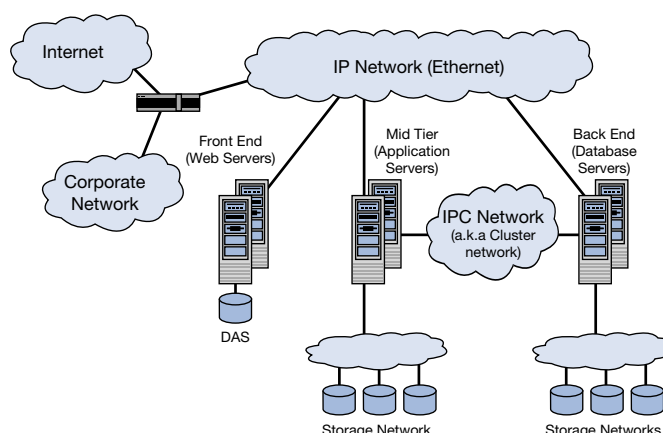


Figure 3 Today's Datacenter

The I/O for front-end servers, such as web servers, is characterized as mostly network traffic, no IPC, small amount of storage traffic, distributed applications, and no clustering. The I/O for the mid-tier (application servers) is characterized as network and storage, some IPC, distributed applications, and some clustering. The I/O for the back-end (database servers) is characterized as heavy storage, IPC, and some network traffic with extensive clustering. While LANs are capable of IPC, special IPC networks are used to achieve low latency and enhanced operations such as remote direct memory access (RDMA).

There are many flavors of I/O interconnect, and thus servers provide multiple I/O slots to house I/O cards such as Network Interface Controllers (NICs) and storage Host Bus Adapters (HBAs) as illustrated in Figure 4. Thus, a server has many thin pipes that attach to the various I/O facilities. The more connections, the greater the number of I/O slots needed.

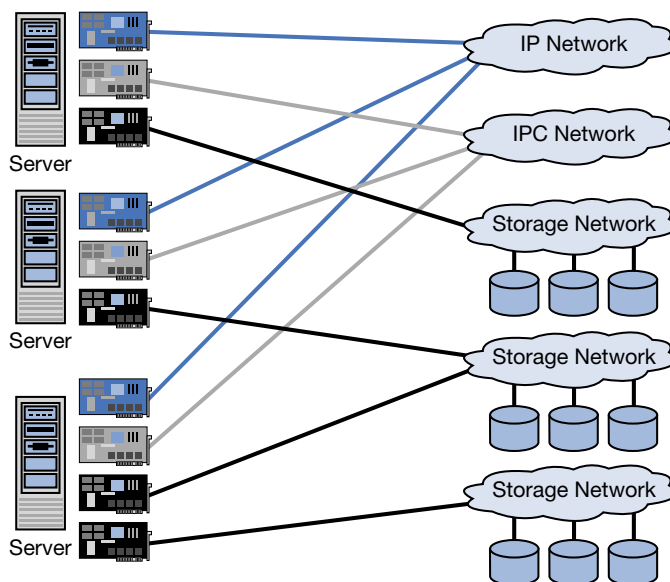


Figure 4 Today's I/O Methodology

Unified Datacenter Fabrics

The trend to unified datacenter fabric (as illustrated in Figure 5) is growing; with InfiniBand³ leading the way and RDMA enabled Ethernet on the rise. Even though Fibre Channel⁴ is a possibility, it currently lacks the features and management infrastructure to make it a serious contender for large datacenters.

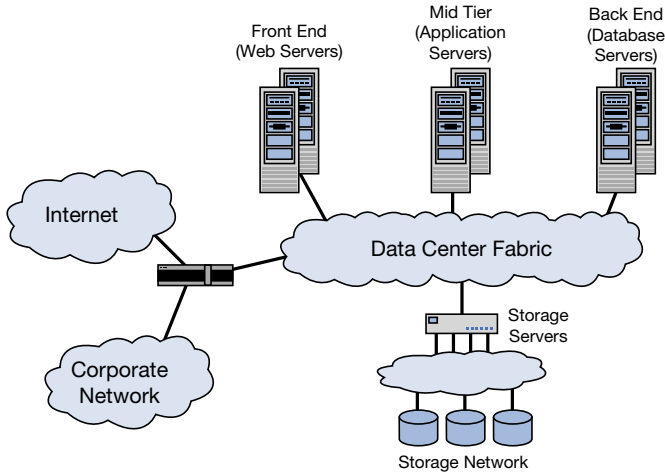


Figure 5 Unified Datacenter Fabric

For the unified datacenter, there is a single fabric, which provides networking and IPC as well as attachment to storage facilities. The unified fabric greatly simplifies server I/O design, which collapses I/O of the server to a single fabric-attach-portal (i.e., one fat pipe instead of multiple thin pipes). Thus, characteristics such as bandwidth, load sharing, and redundancy are focused on that one fat pipe (OFP) and not across a number of the thin pipes.

Figure 6 illustrates the impact of a unified datacenter fabric on the server. Instead of designing the server's I/O for the particular application, the server on a unified fabric needs to be good at just one thing. That is, I/O to the fabric, regardless of what percentage of that I/O is for networking, IPC, or storage.

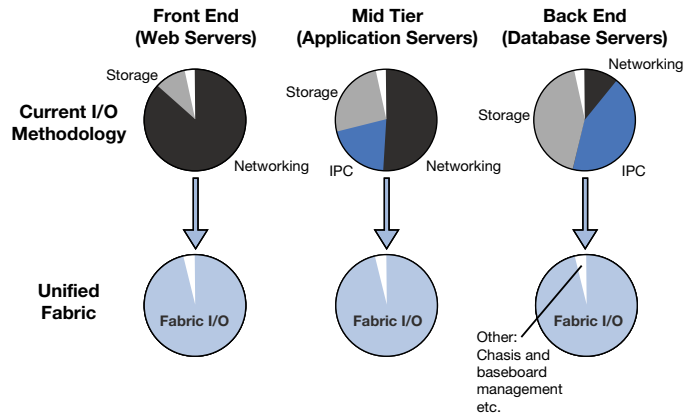


Figure 6 Distribution of I/O

Moving I/O outside the box does not mean moving the NICs and HBAs, as illustrated in Figure 7. Rather it means making I/O an integral part of the fabric as illustrated in Figure 8. This poses some unique management obstacles that are discussed later.

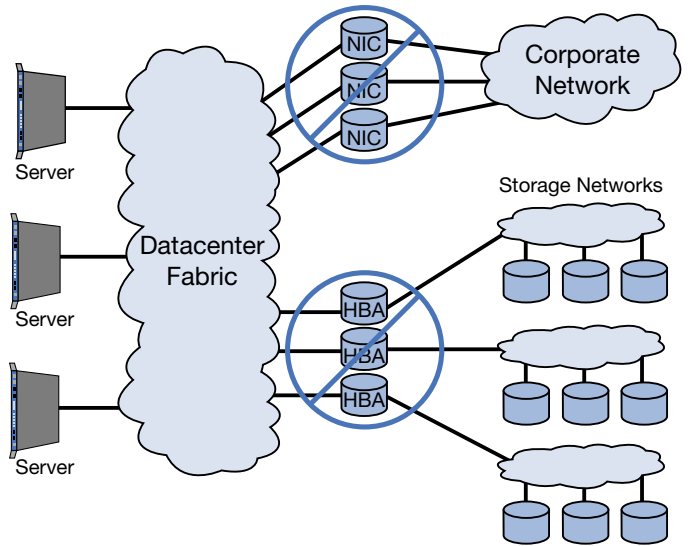
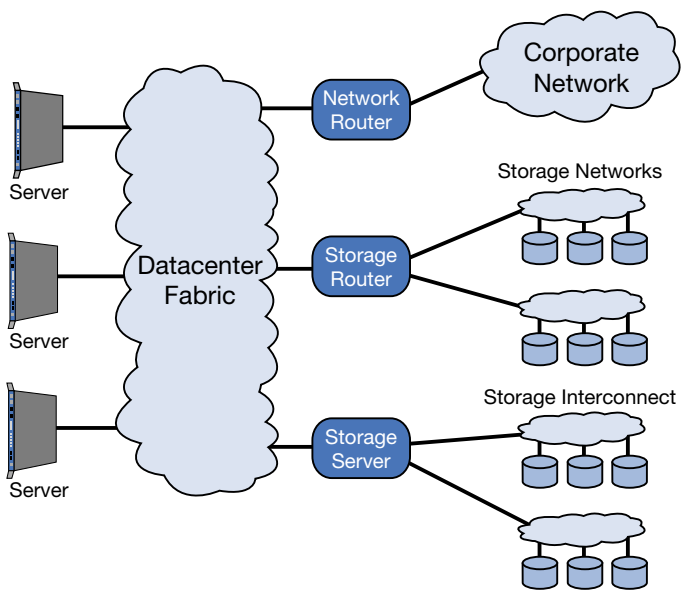


Figure 7 Incorrect I/O Attach Method

³InfiniBand is the trademark and service mark of the Infiniband Trade Association.

⁴Fibre Channel is primarily a storage network and features for networking and IPC do not compare to features for InfiniBand and Ethernet. Although FC is capable of networking and IPC, the industry is not moving in that direction.



It is changing the way we access I/O

Figure 8 I/O Service

Networking

As illustrated in Figure 8, datacenter fabrics are IP networks that can easily be routed to the Internet and the corporate network. This is standard networking where the function of the NIC⁵ is inherent in the server's fabric-attach-port. For example, for an Ethernet datacenter fabric, the server's fabric-attach-port is an Ethernet NIC and thus network packets are delivered to other servers or routed to the Internet or corporate network the same as they are today. The same holds true for an InfiniBand fabric, where the server sends and receives IP packets over InfiniBand (iPoIB) that are easily routed between the datacenter network and the Internet or corporate network.

IPC

The datacenter fabric must also provide for IPC. The primary IPC characteristic is low latency. This can be accomplished with a well-designed fabric structure and low-latency switches. Thus, datacenter network switches will mostly be a fully integrated into a full-mesh fabric that is actively managed to provide shortest paths between source and destinations. While InfiniBand was architected for low-latency, Ethernet was not. This only means that datacenters Ethernet switches will be architected and managed differently than Ethernet switches used in client networks.

Storage

Two new SCSI transport protocols are being adopted that directly relate to datacenter fabrics. One is iSCSI, a protocol for transporting SCSI commands and data over an IP network. The other is SCSI RDMA Protocol (SRP), for

transporting SCSI commands and data over RDMA enabled networks. Generally, SRP is for InfiniBand⁶ and iSCSI is for Ethernet⁷. These protocols allow a compute server to access a SCSI device (disk drive, RAID controller, etc.) that attaches to the fabric.

While disk drives attached directly to the datacenter fabric would be the simplest and most direct approach, the economics of producing Ethernet and InfiniBand disk drives are prohibitive. Thus, native datacenter fabric drives are not considered viable, especially given that desire for virtualization. Accessing legacy drives via a storage router and via a storage server, as illustrated in Figure 8, are the leading solutions. These two approaches have significant differences.

Storage Router

A storage router simply extends the datacenter fabric to include the storage network, so that each compute server can access legacy drives and storage devices on the storage network as if they were attached to the datacenter fabric. However, a storage router is not as simple as it sounds. While it is true that SCSI commands are independent of the transport, different transport protocols are used for the various types of SCSI interconnects.

The router must be capable of translating the protocol. Take, for example, a router between InfiniBand (IBA) and Fibre Channel (or Parallel SCSI). The SCSI transport protocol for InfiniBand is SRP. The SCSI transport protocol for Fibre Channel is Fibre Channel Protocol (FCP). The Information Unit (IU) for each protocol has a different format and content. The router converts the SRP request (which includes an RDMA address) to an FCP request (that contains a context-tag instead of an RDMA address) and sends it to the FC SCSI device. The device returns the context-tagged data that the router returns to the initiator, but the router must convert the FCP data into an IBA RDMA transaction, relating the context-tag to the original RDMA address. The router also converts the SCSI completion status it receives from the SCSI device as it forwards the completion status back to the initiator as an IBA Send transaction. All of this requires the router to maintain a certain amount of state, so in comparison to a network (IP) router, the storage router actually functions as a gateway. An Ethernet storage router (Ethernet to Fibre Channel) has the same considerations routing between iSCSI and FCP or parallel SCSI. Management considerations are discussed later.

Storage Server

A storage server (or storage appliance) abstracts the physical storage devices (and the bus or network that connect them to the storage appliance). This approach allows a number of features such as storage virtualization, RAID⁸, and caching. A storage appliance can be a file-based or a block-based device. File-based storage

⁵The server's Ethernet controller will most likely include a TCP offload engine (TOE).

⁶Even though InfiniBand is an IP network, and thus it could execute iSCSI, RDMA provides better benefits and thus SRP is the SCSI protocol for InfiniBand.

⁷RDMA-enabled Ethernet can support SRP. However, there is an effort underway to extend iSCSI to take advantage of RDMA enabled Ethernet networks.

⁸Redundant Array of Independent Disks, (a.k.a. Redundant Array of Inexpensive Disks).

appliances (a.k.a. network attached storage) are gaining popularity and are a natural solution to fabric-attached storage. The primary difference between a file-based and a block-based storage appliance is that the file-based appliance contains the file system that virtualizes disk blocks into files, and thus the compute server communicates with a file-based appliance using a file-based protocol, such as NFS and CIFS for Ethernet and DAFS for IBA and RDMA-enabled Ethernet. This is in contrast to the compute server using a block-based protocol, such as SCSI block commands, to communicate a block-based appliance, such as a RAID controller.

The storage server still has to execute both storage protocols (i.e., the protocol with the compute server and the protocol with the physical storage devices), but abstraction of the physical devices and storage virtualization provides significant value.

Storage virtualization means that the storage server can take a large capacity disk and make it appear as multiple smaller capacity disks. It also means that smaller disks can appear as a single large disk (i.e., RAID). In the data center, different server applications have varying storage requirements. Many of the Front-End applications (such as web servers) require only a relatively small amount of storage while database applications require an enormous amount of storage. Since the cost per Mbyte of storage decreases as the drive's storage capacity increases, it makes sense to abstract a large drive as multiple smaller drives, thus reducing storage costs. The abstraction also makes it easy to expand a compute server's storage capacity, because the storage server can simply add more blocks to a virtual drive, regardless of whether the additional storage is physically contiguous, or even on the same physical drive. This type of capability becomes more and more important with the desegregation of servers.

RAID on its own is very compelling and recognized a significant value. Virtualizing a large capacity RAID set as multiple smaller disk drives not only provides economies of scale, but each virtual drive inherits the advantages of RAID (e.g., redundancy, load sharing, manageability, etc) that a single small disk drive does not provide.

Caching also provides significant performance benefits, especially when multiple servers are accessing the same blocks of storage. Historically, caching was only found in large storage arrays serving backend-clustering applications. Caching data for distributed applications in the mid-tier is becoming more and more common. Caching not only improves disk-read performance, but also write caching also permits the storage server to schedule disk writes in a more efficient fashion.

Bladed Servers

Server blades are the epitome of modular computing. The fabric interface (Fabric I/F) replaces the server's I/O subsystem as illustrated in Figure 9. Less I/O circuitry makes room for more memory and allows the fabric I/F to be more closely coupled to the memory controller. It forces storage to the edge of the fabric turning the server blade into a very universal computing resource.

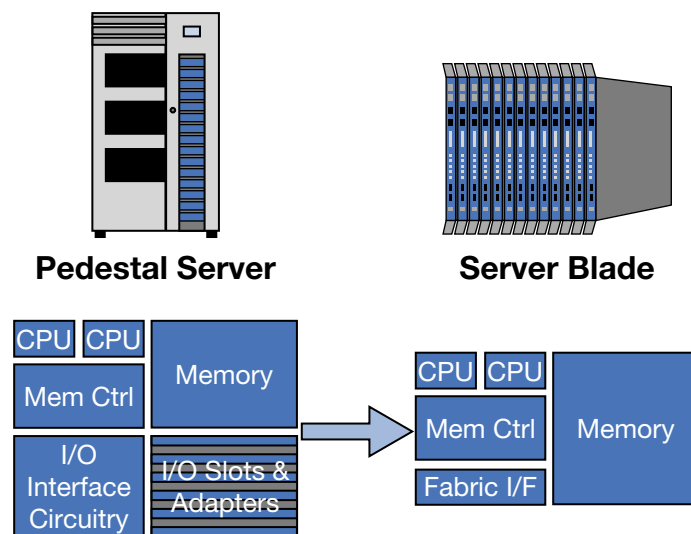


Figure 9 Transition to Server Blade

Blade products will vary by the number & type of CPUs and the amount of memory. A significant principle is that the fabric interface bandwidth, memory bandwidth, and CPU bandwidth need to be balanced. This balance is independent of the type of I/O (see Figure 6). Thus, it makes little difference if the I/O traffic is dominantly network, IPC, or storage because when the bandwidth of the fabric I/F matches the memory and CPU capability, I/O is not a bottleneck and the server operates at maximum efficiency.

Managing Datacenter I/O

Enhanced features such as a unified fabric, virtual resources, and repurposing result in significant values, such as economies of scale, automated control, and versatility. These features require enhanced management software to manage the complexity of the environment.

IT managers will invest more in management because they will get so much more out of it, and the ultimate objective is better automation. However, products will need to support management features and management needs to be architected to take advantage of those features.

Managing pools of network, storage and compute resources and turning them into virtual server platforms is a very in-depth subject that is outside the scope of this paper. Rather this paper focuses on the management features necessary to support fabric attached storage and virtual servers.

Required Management Features for the Modular Datacenter

The two prime goals of modular computing are:

- Enable virtualization of compute resources
- Increase the degree of automation

Virtualization of compute resources means breaking the paradigm that storage belongs to a server platform. The notion of server platforms is being replaced by pools of compute, storage, and network resources such that a virtual server is an association of a compute node, an application, and its storage.

One point that needs to be stressed is that storage is associated with an application. Thus, it is the application that a compute node executes that determines which storage resources it gets to access. As the realm of modular computing grows, it is equally important to recognize that the relationship between applications and server modules is very dynamic.

- ‘Server module’ refers to the hardware without any particular affiliation to an application.
- ‘Compute node’ refers to a server module associated with a particular application.
- ‘Virtual server’ refers to an application, the server module on which it is currently executing, and the storage resources.

Re-purposing of server modules requires the ability to dynamically reconfigure storage devices as to which server modules can access it, as well as configure the compute node as to where to find the I/O that its application requires. In particular, it means the ability to take resources away from a server module as well as add resources to it. A better management model is one where I/O resources are directly associated with the application and not the server module. That is, storage devices are configured with the identity of the application rather than the server module that is allowed to access the storage device.

Maintenance is another management concern. With storage devices located outside the compute node's power domain, care management needs to notify the compute node when powering down storage modules, removing storage modules, or running diagnostics that could disrupt I/O operations. Thus hot-plug and hot-swap events occur across the fabric and the facility for notifying an operating system about resource availability must be network based. The desire is that it be fully automated. For example,

an attempt to remove a module signals the configuration manager, who notifies all affected compute nodes, and once those nodes have curtailed their I/O operations with that I/O module, the configuration manager signals the I/O device, which causes module power to be removed, allowing the module to be ejected or extracted.

It is also important to understand that the datacenter is a highly shared environment and that adding or removing I/O impacts more than one compute node. This is an extremely important consideration when invoking diagnostics. Most diagnostics are disruptive to normal I/O operation. This implies that diagnostics can only be executed when the I/O device is not in use. It would be useful for I/O devices to also have graceful diagnostics (i.e., diagnostics that do not disturb normal I/O operation). Thus, a manager can routinely run graceful diagnostics and only when there is potential failure, would the manager run a diagnostic that will disrupt normal I/O operation. This implies that the manager, not the server, invokes diagnostics, and there is a means for compute nodes to inform the manager when the health of the I/O device is suspect.

Fabric Management

Fabric management is a very complex subject. The focus here is on access control because it is fabric management that configures the fabric to enable a server module to communicate with an I/O device. Examples of fabric access controls are zoning for Fibre Channel, VLAN for Ethernet, and partitioning for InfiniBand.

The fabric manager is the first to know when a node enters or leaves the fabric. Thus, there is a strong need for fabric management to coordinate with other levels of access control. Both to inform the configuration manager that device is present and active, as well as to make sure that access capabilities as consistent at all layers.

Device Configuration

Device configuration is the ability to configure an I/O device. The ability to configure an I/O device is very vendor specific and includes the ability to create or destroy I/O objects (such as virtual drives) as well as configure the object's I/O properties (such as the drive's size).

The configuration program normally executes from the manager's console, remotely accessing the I/O device outside the normal I/O channels. The easier it is for the manager to automate these functions, the more value (ease of use) the product has. The ability to standardize is highly desirable.

Since I/O devices are becoming more and more intelligent, one way to achieve this goal is for the I/O device to use an HTTP session. Thus, the I/O device can have a custom configuration program that can be executed from any console that has a browser. Such a program should be password protected to assure only trusted access.

Access Control

Access control is present at various levels.

- SCSI sharing controls (i.e., Lock/Unlock and Reserve/Release) are still very important and are used by compute nodes that already have access to I/O devices to gain atomic access.
- Before a compute node can access an I/O device, it needs to have appropriate fabric permissions assigned by the fabric manager.
- Device level access control is the enforcement of device access rights by configuring the I/O device with the identity of each client (e.g., the Initiator ID used for SCSI LOG-IN) and the permissions that the client has to access various I/O objects (e.g., LUN masking for SCSI devices). It also includes the ability to notify the configuration manager when I/O objects are created or destroyed.

Additional requirements for device level access control are:

- Enterprise class I/O devices must be able to support multiple clients.
- The dynamic nature of the modular datacenter may dictate frequent changes to access control information. Thus, device level access control needs to be standardized so it can be automated.
- The client is no longer a server platform, but rather an application that can be associated with any server module. This implies that the Client ID needs to be associated with the application and not the server module.

It was previously mentioned that there are differences between a storage router and a storage server. Those differences also apply to management. Management models need to be consistent for both I/O attachment models. The following sections discuss the issues.

Access Control Considerations

Storage Router: Talk-Through Management Model

The storage router predicates a talk-through approach where compute nodes communicates with the I/O device on the storage network (not the storage router) as illustrated in Figure 10.

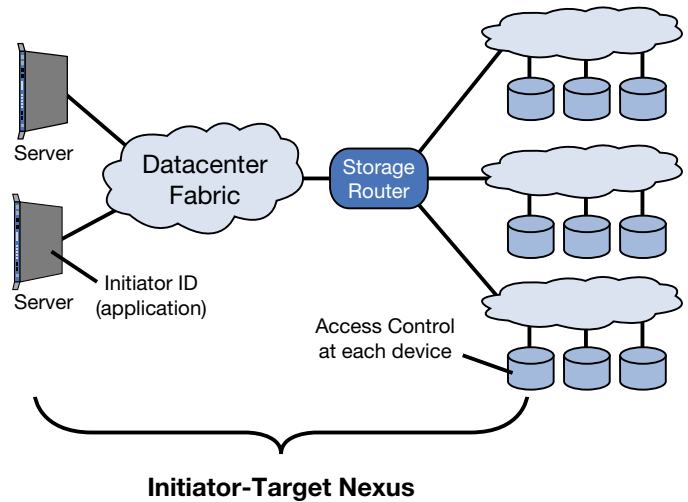


Figure 10 Storage Router Talk-Through Model

This implies that each storage device provides its own device level access control. Intelligent devices, such as RAID controllers, typically provide access control capabilities defined above, however, simple storage appliances, such as disk drives, tend not to incorporate sophisticated features, at least not to the degree that intelligent storage devices do, especially the number of initiators supported.

For SCSI, the relationship between the client and the I/O device is the Initiator-Target Nexus (I-T Nexus). Each SCSI device can have multiple I/O objects referred to as Logical Units or LUNs (Logical Unit Numbers). LUN mapping is the concept of identifying which Initiator can access which set of SCSI LUNs.

The SCSI architecture model does not place any restriction on Initiator IDs. However, certain SCSI transports such as Fibre Channel have a predefined notion that the Initiator ID is the World-Wide Port Identifier (WWID) of the Initiator's port. This is another paradigm that needs to change. First, the Fibre Channel port of the Storage Router is not of importance, so it is not that WWID that is of interest. The logical recourse would be to use the network address of the server module, however, these addresses can be dynamic, and even if they are static, it is the application that really identifies the initiator. Thus the conclusion is there needs to be a way to assign WWIDs to the application and that WWID is used for the SCSI LOG-IN.

Storage Server: Talk-To Management Model

The storage server predicates a talk-to approach where compute nodes communicates with the storage server (not the devices it represents) as illustrated in Figure 11.

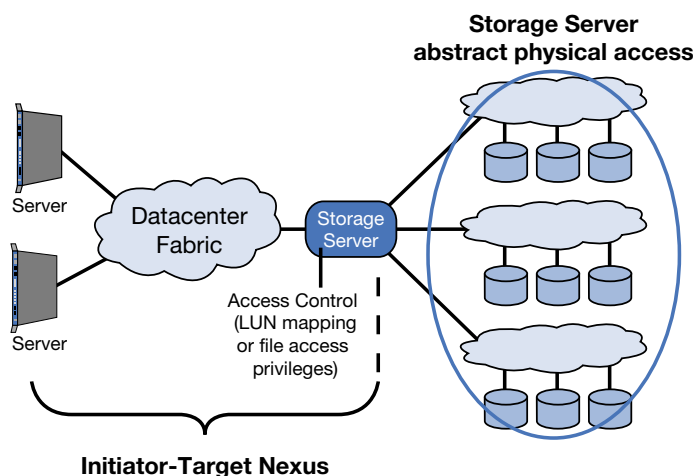


Figure 11 Storage Server: Talk-To Model

This implies that the storage router provides the device level access control, consolidating management to one point. Thus, the storage server abstracts the access to the physical devices.

The same considerations hold for identifying the Initiator as with the talk-through model. That is, the Initiator ID needs to be tied to the application and not the server module.

Server Configuration Management

Configuration management is not only responsible for configuring an I/O device with the identity of its clients, but it also supplies the client with information that identifies the I/O devices that it is permitted to access. When a server module boots, it needs a way to query the configuration manager as to its I/O resources. The answer returned by the configuration manager is very dependant on the application that the server is to execute. Thus, the client ID of a server module can be dynamic. Even though it is possible for the server to have a fixed client ID, it is undesirable for the configuration manager to reprogram the I/O devices with new client information each time the relationship between a server module and an application changes (i.e., application moves to a different server module, or the server module is re-purposed), especially considering that the number of I/O objects greatly outnumber the number of server modules. Thus, the ability for the Client ID to be associated with the application rather than the server module is highly desirable. The configuration manager also needs an efficient way to notify a server when new I/O objects get assigned to the application, to notify a server when existing I/O resources come on-line or go off-line (including off-line to run diagnostics), and to gracefully remove I/O devices from the server's control.

Summary

The trend to modular computing is driving I/O evolution. The evolution includes the transition to a unified datacenter fabric, simplifying server I/O to a single fabric attach point, and fabric attached I/O. This is done to transition the datacenter from sets of dedicated server platforms to pools of compute, storage, and network resources that can be dynamically associated to form virtual servers. A modular datacenter enables dynamic scaling of datacenter resources to meet changing business needs. The value that modular computing brings is increased degree of manageability, better automation, reduced total cost of operation, and higher return on investment.

Author Biography

Bill Futral, Server I/O Architect, Enterprise Platform Group, Intel Corporation.

For more information, visit the Intel web site at: developer.intel.com



Copyright© 2003 Intel Corporation. All rights reserved.

*Other names and brands may be claimed as the property of others.