LUN Masking in a SAN

Bill King
QLogic Corporation
bill.king@qlogic.com
1.952.932.4000

Describes what a LUN is and how LUN masking is used in a SAN

Abstract: SAN's provide the infrastructure on which sophisticated storage solutions are built. One benefit is the ability to share a single large storage device across many servers or applications. To enable this the storage is typically carved into smaller pieces which are each assigned a SCSI logical unit number or LUN. Each LUN is then assigned to one or more servers as required by the configuration. LUN masking is the ability to exclusively assign each LUN to one or more host connections.

LUN Masking is used to assign appropriately sized pieces of storage from a common storage pool to various servers. It allows for large devices to be divided into more manageable pieces, it allows reasonably sized file systems to be built across many devices to improve performance and reliability, and it is often times part of the fail-over process when a component in the storage path fails.

Overview

A storage area network (SAN) provides a dedicated network for storage functions. This is appropriate since the requirements of a SAN are quite different from a general purpose ethernet network. The primary technology to implement a SAN today is based on fibre channel technology which is well suited to the requirements of a storage area network.

Storage Consolidation

Adding a network to the storage subsystem allows many new capabilities to be implemented, one of the most fundamental of these is storage consolidation. Storage consolidation creates a single pool of storage which is managed and serviced centrally. The administrator can carve out pieces of the storage pool and grant one or more servers access to them. By having a single large pool with central management it is easier to establish and enforce policies for backup, upgrades, and allocation of storage. This results in more flexible policies and lower cost of management.

Physical and Logical Storage

There are two types of storage which can be managed, a physical storage device and a logical storage device. A physical storage device is the entire storage device, you can usually hold a physical storage device in your hands or maybe with a forklift. A physical storage device is the RAID controller and its associated disks, a disk drive or a tape drive (not the tape). A logical piece of storage is usually created to make the physical storage more manageable and is typically built from one or more pieces of a physical storage device. A logical piece of storage is not something you hold in your hands. A modern marketing term for creating and managing logical pieces of storage is storage virtualization.

A physical piece of storage is very often measured in TeraByte's, and is built from engineering specifications that specify reliability, serviceability, performance or a specific price per MegaByte. A logical piece of storage is usually measured in MegaBytes and is created to meet the requirements of a system administrator, such as planning availability, backup policies, disaster recovery or other high level storage requirements.

Host Connections

A host will have connections for fibre channel cables, these are the plugs into which a fibre channel cable is connected. These may be provided as an integral part of the server or workstation or they may be options that are added through PCI or SBUS host bus adaptors (HBA). Regardless of how they are provided, a given host will usually have more than one fibre channel connection. A single HBA may provide multiple host connections that are implemented on a single piece of hardware, but each connection is unique for configuration purposes.

Shared Storage

A logical piece of storage is usually not shared between host connections. If a logical piece of storage is shared then support for synchronization of access must be provided. For the purposes of this article sharing is not

discussed since it requires a higher level of control than a SCSI LUN. However a few examples are given below to provide context.

Active/Passive connection. Multiple connections are often allowed in clusters and other high availability configurations, however they are implemented as *active* and *passive* connections. There are two connections, but only one that is *active*. The other channel is not used and is called *passive*. The *passive* channel is used only if a failure occurs on the *active* channel.

Active/Active connections. Some software products allow two host connections to simultaneously share a single logical piece of storage. These are usually minor extensions to the standard device driver or file system and provide synchronization for multiple reads but only a single write. Variations of this are possible, but usually they are quite simple to avoid the complexities of a shared file system.

Shared file systems. There are file systems that are truly shared and they allow reads and writes from different host connections and even different hosts. These file systems provide synchronization so that corruption of data does not occur. Another example of shared data that some readers may be familiar with is the Oracle Parallel Server (OPS), in OPS a special version of the Oracle database provides synchronization of writes so that corruption does not occur. The OPS product does not require a file system to be used.

Overview of SCSI LUN's

Most storage devices use the SCSI (small computer system interface) command set to communicate. This is the same command set that was developed to control storage devices attached to a SCSI parallel bus. The SCSI command set is not tied to the SCSI parallel bus and the command set is now commonly used for all storage devices with all types of connections, including fibre channel. The command set is still referred to as the SCSI command set.

What is a LUN?

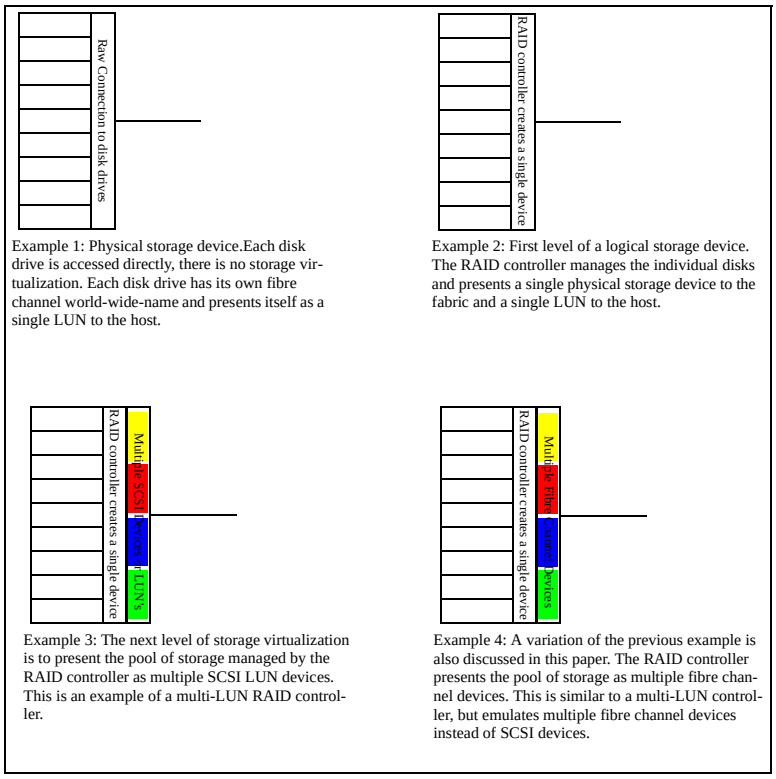
On a SCSI parallel bus (a type of inter-connect used to connect storage devices, it is different from fibre channel) it is possible to have different devices, and each one is given an address called a logical unit number (LUN). For those of you that have set jumpers or toggle switches on the

back of a SCSI disk or tape drive, this is the number that is being set. The LUN on a SCSI parallel bus is actually used to electrically address the various devices. The concept of a LUN has been adapted to fibre channel devices to allow multiple SCSI devices to appear on a single fibre channel connection.

The LUN is required to be unique among all devices that are visible to each other. In the old world of a SCSI parallel bus it was very easy to define which devices were able to see each other since they were physically attached to the same cable. In the world of SAN's this definition is much broader since it is possible, although usually not desirable, that all devices attached to a SAN can see each other.

A RAID controller that simulates multiple SCSI devices with different LUN's is said to have multi-LUN support. This allows a single RAID controller to present its storage as many smaller devices.

FIGURE 1. Progression from physical storage device to logical storage device



Example 1: Physical storage device.Each disk drive is accessed directly, there is no storage virtualization. Each disk drive has its own fibre channel world-wide-name and presents itself as a single LUN to the host.

Example 2: First level of a logical storage device. The RAID controller manages the individual disks and presents a single physical storage device to the fabric and a single LUN to the host.

Example 3: The next level of storage virtualization is to present the pool of storage managed by the RAID controller as multiple SCSI LUN devices. This is an example of a multi-LUN RAID controller.

Example 4: A variation of the previous example is also discussed in this paper. The RAID controller presents the pool of storage as multiple fibre channel devices. This is similar to a multi-LUN controller, but emulates multiple fibre channel devices instead of SCSI devices.

The Relationship of SCSI Devices and Fibre Channel Devices

It is important to distinguish between a SCSI device and a fibre channel device. A fibre channel device is a lower level device that emulates one or more SCSI devices as an abstraction (the lowest level of storage virtualization), there is not an actual SCSI device. The SCSI device is emulated by responding appropriately to the SCSI protocol.

> Example: This is somewhat like speaking different languages over a telephone connection. The low level connection (fibre channel) is the same for conversations in English, French or Japanese (SCSI command set).

How are LUN's Used?

A modern RAID controller manages from 100GigaByte to more than 5000 GigaByte of storage. The entire amount of storage is typically attached by one or more fibre connections. If the RAID controller does not have multi-LUN support then it appears as one piece of storage. If there is multi-LUN support, then the storage can be carved into many smaller pieces and allocated as appropriate.

A pair of RAID controllers (paired for reliability) will support many LUN's and divide the storage into many pieces and associate a unique LUN with each piece. Each LUN can be assigned to a given host connection as appropriate. Each LUN could be a different size, although typically a large number of identical LUN's are created and then allocated and combined to form volumes.

What is LUN Masking?

LUN Masking is the capability that allows a specific LUN to be exclusively assigned and accessed by a specific list of host connections. Usually only one host connection will access a LUN at a time. By implementing LUN Masking it is possible to reliably attach a single LUN to a single host connection. Most importantly, other host connections will not be able to access LUN's to which they are not assigned.

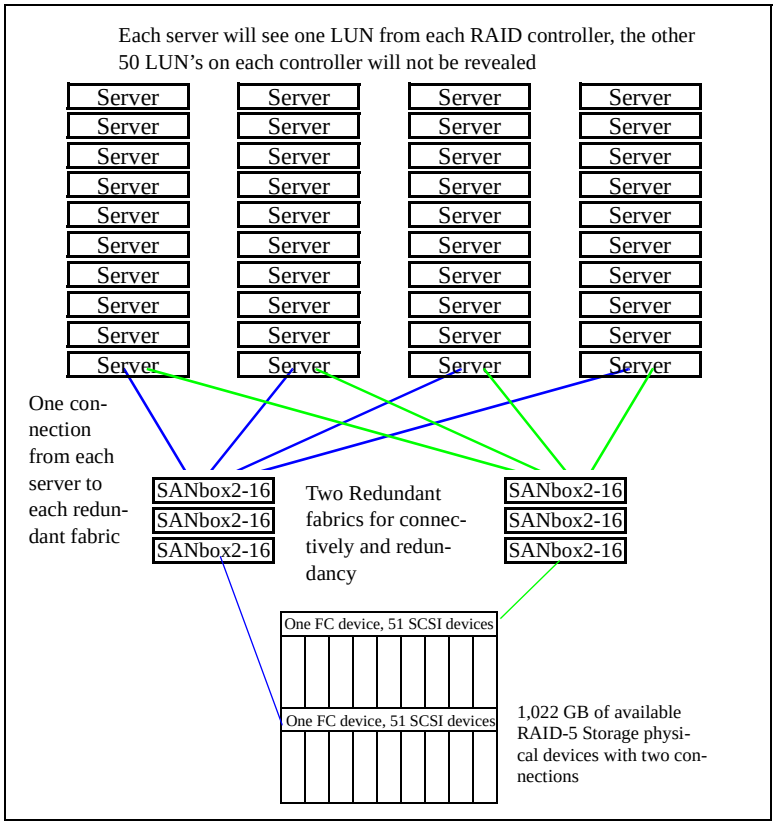
The critical point here is that the allocation of a LUN to a host connection is made by hiding devices that are not assigned. There is actually no special connection made when LUN Masking is performed. The implementation is simply to not reveal any LUN's to a host that have not been assigned.

> Example: This is like having an unlisted phone number. The number is accessible to anyone who dials it, however because it is not listed it is very difficult to find the number.

Example of Storage Virtualization/ Consolidation

This example demonstrates the most basic example of what is called storage virtualization. The specific example is of storage consolidation, that is all of the storage for many servers is consolidated into a single pool. The physical storage is built from 18 separate disk drives and two RAID controllers, but is presented to the servers as 102 highly available drives of 10 GigaByte each.

FIGURE 2. Storage Virtualization/Consolidation Example



Requirement. A group of 40 Window 2000 servers each require 20 GigaByte of highly available storage.

Solution. Propose a pair of RAID controllers with fail-over between the controllers and multi-LUN support. Assume that each RAID controller provides 511 GigaByte of available storage (9 drives of 73 GigaByte each, 7+1 RAID-5 and one hot spare). Each controller is configured to have 51 identical LUN's each, each LUN is 10 GigaByte in size. This would result in 1 GB of the available physical storage not being allo-

cated. One LUN from each RAID controller will be assigned to each Windows 2000 server providing a total of 20 GigaByte of available storage to each server. There will be 11 LUN's on each controller that are available for future expansion or other uses.

Evaluation. Obviously with no failure this solution meets the requirement. In the event of a disk failure, the RAID controller will provide access to the data and rebuild the lost drive transparently after it is replaced. In the event of an HBA or cable failure one of the servers would lose access to one of its LUN's. In the event of a controller failure all of the servers would lose access to half of their LUN's. In either case, the LUN's that could not be accessed would be failed-over to the remaining RAID controller. The server(s) would then be able to access all of the storage through the redundant path. After the HBA, cable or RAID controller is repaired the LUN(s) would fail-back to the original configuration and the HA configuration would be restored.

Assumptions . To implement a solution like this requires that all of the servers be connected to the storage, this is done by using a SAN, an example of which is shown in the figure. If the fabric is configured as shown it will be redundant. In this configuration the storage and HBA's can be either public or private, because all of the servers need to be connected to the same storage, nameserver zoning and other fabric services are not required.

Implementing LUN Masking

There are three places where LUN Masking can be implemented. The first is in the storage, the second is in the servers, and the third is either in a device through which all of the I/O passes or the SAN itself. Each of these has its benefits. In practice, LUN Masking at a customer site is implemented in multiple ways reflecting the different methods used by each vendor.

The method used by the host when it discovers storage is to query on each connection as to which LUN's are available. If only a specific subset of LUN's is return in response to that query then LUN Masking is being performed. LUN Masking is performed by hiding some of the LUN's.

Implementing LUN Masking in the Server

The easiest place to implement LUN Masking is in the server. This is because a server can be upgraded with software to provide the capability and in theory the same software can be run across all of the servers.

The way LUN Masking is implemented on the server is to have the server query for LUN's, but then ignore all but the assigned LUN's. This means that each host must be configured to ignore all but the LUN's they have been assigned. This type LUN Masking is more manageable if there are only a few large servers connected to many heterogeneous storage devices.

LUN Masking at the host level can be performed by the driver and HBA as is done by QLogic, or can be done by the OS itself. The driver/HBA method is most typically done since having a common HBA across many platforms is reasonable.

The issue with implementing LUN Masking at the server level is that the host actually sees all of the LUN's, but will ignore LUN's not assigned to it. This requires that all of the hosts be trusted and that they be administered together. LUN Masking at the host level is reliable, but does require that all hosts have common administration. LUN Masking on the host also requires that the storage vendor implement multi-LUN support in their RAID controller. Because most recently released RAID controllers with multi-LUN support, also have LUN Masking support a host based solution is not always required.

Implementing LUN Masking in the Storage

The second place to implement LUN Masking is in the storage. This means that each RAID controller is configured to allow each host to see only a subset of the actual LUN's. By implementing LUN Masking in the storage, their is no need to configure each host. In the example above the two RAID controllers would need to be configured, but the 50 Windows 2000 servers would not need to be configured. The servers would simply use the storage that they see. If LUN Masking is implemented in the storage there is no requirement to make any changes to the servers. This type of LUN Masking is especially attractive when there are many more servers than storage devices.

The issue with LUN Masking in the storage, as with all of the methods, is that it must be implemented by the storage vendor. In a typical data center with heterogeneous storage solutions each type of storage will have different capabilities.

Implementing LUN Masking in the SAN

A third place to implement LUN Masking is with some device in the SAN itself. This is very attractive because it is independent of the hosts and the storage. There are many devices which perform this function, generally under the marketing term of storage virtualization.

The benefit of a LUN Masking device is that it can work with any server and any storage device. The device is simply placed in the path of the storage device and LUN Masking is available. In addition, most of these devices can implement multi-LUN support if it is not available from the storage. This makes it possible to integrate both old and new storage from many vendors into a consolidated storage pool. The device also has its own management interface so there is a single management interface for all of the LUN management. This also creates a single point of administration.

The issue with a LUN Masking device is that it introduces yet another device into the solution with another management interface. Storage management typically requires the host, the storage, and the SAN to be carefully coordinated. While the capability of the device is very attractive, having yet another device to manage is not. In addition, for large storage configurations there must be many of these devices and they can introduce a performance bottleneck.

Implementing LUN Masking in the Fibre Channel Fabric

A variation of the third type is to implement LUN Masking in the SAN itself. If the SAN is based on fibre channel then this would be done by the fibre channel fabric. This is an attractive solution because the fabric is managing access to the storage devices and LUN Masking is a natural extension to this. In addition, the fabric is typically implemented across many switches and this method scales well as more complex configurations are required. Unfortunately, implementing LUN Masking in the fabric would cause an enormous impact to performance using existing technology.

As discussed earlier the SCSI protocol and fibre channel protocol are completely separate. A SCSI data transfer is typically many thousands of bytes long and is broken into many pieces by the fibre channel protocol, There is no access to the SCSI LUN number in each frame header. Even if the LUN information was available it is not related to the routing tables used in a fibre channel switch.

To implement LUN Masking in the switch would require that a time consuming table look-up be performed that is not currently possible within

the memory constraints on the fibre channel switch ASIC. This means that all of the data would need to be staged to a central cache before being forwarded on. This is simply not possible with today's technology without increasing the latency by a factor of 10 to 100 times.

> Example: Using the example of speaking different languages over a telephone connection this performance penalty can be easily seen. If the telephone company could offer a service where different languages were translated it would be very convenient. However, this service would cause a significant delay in the rate at which information was communicated.

The latency of the newest QLogic SANbox2 switch is approximately 400 nano-seconds. The header of the frame is already moving out of the switch before the tail of the frame has arrive. If this is increased by 10 to 100 times the latency of each I/O will become significant and it would not be possible to build high performance SAN's. Technology in the future may make this possible.

Support of Device Masking by the SAN

Another way of describing LUN Masking is as device masking. Essentially the idea is to mask the existence of a storage device from all but a desired set of host connections. Because fibre channel fabrics support zoning based on individual devices (WWN Zoning) it is possible to perform device masking in the fabric.

The fibre channel standard requires that each fibre channel device be assigned a unique 64 bit number called the World-Wide-Number or World-Wide-Name (WWN). Typically the WWN is is assigned to each physical storage device. However, just as a storage device can emulate the presence of multiple SCSI devices, it can also emulate the presence of multiple fibre channel devices. There are products which do this, but they are currently only available in very high end storage products.

Fabric addresses, WWN's and the Nameserver. All devices in a fabric are assigned a 24 bit public address by the fabric. This address is associated with only one device. In addition the fabric maintains a database of all the devices and their addresses, this database is call the fabric *nameserver.* The nameserver is used by host connections to determine which devices they are allowed to communicate with.

Fabric Services and Zoning. The nameserver provides another service called zoning. The administrator of the fabric can define zones and assign devices to each zone. When a host connection requests a list of allowed

devices it is only returned the devices that it shares in the zones. This is device masking for fibre channel devices, just as LUN Masking is device masking for SCSI devices.

If a RAID controller is designed to implement multiple fibre channel devices then it is possible to configure the fabric to implement device masking that is very similar to LUN Masking. This is attractive because all fabrics support WWN zoning and this would provide a standard solution. This solution also has no impact on performance and is a natural extension to the fabric management.

The primary negative to this solution is that it still requires the storage vendor to implement the functionality in the RAID controller, so this is not a more universal solution than LUN Masking. It also requires each RAID controller to be allocated a range of WWN's by the manufacturer so that each WWN is guaranteed to be unique, however with 18,446,744,073,709,551,615 available device numbers, we should be OK for a while.

Another reason this is not an ideal solution is that the RAID controller is where the multiple fibre channel devices are configured and it is natural to perform the device masking at that level as well. Finally, while it is attractive to have the fabric perform the device masking, it does not provide a superior solution to the traditional method of multiple LUN's and LUN Masking.

Summary

LUN Masking is used to assign appropriately sized pieces of storage from a common storage pool to various servers. It allows for large devices to be divided into more manageable pieces, it allows reasonably sized file systems to be built across many devices to improve performance and reliability, and it is often times part of the fail-over process when a component in the storage path fails.

This white paper focused on what LUN Masking is and the various ways it can be implemented. Each vendor implements LUN Masking differently and the method used by a customer typically reflects whatever is available in the equipment or software purchased. Each method has its

benefits, however they all require some level of central administration and each is implemented slightly differently.

- Implementation of LUN Masking on the server is attractive because it can be done with software and potentially this could span all server types. However host based LUN Masking requires all hosts to have a central administration to avoid conflicts.
- If LUN Masking is implemented in the storage it is often times easier to have the central administration, however it is then dependent on the support the storage vendor provides.
- Special devices to create and implement LUN Masking are attractive because they work with any products, however they require adding a new device, new management, and they can cause a performance bottleneck in large configurations.
- Finally the idea of masking fibre channel devices with the fabric was presented, but this depends on the storage vendor to implement special functionality in their products. This method also does not provide any additional functionality over a multi-LUN storage controller with LUN Masking.

While each solution has weaknesses, the preferred place to implement LUN Masking is in the storage, since it is a simple extension to the creation of the LUN's that is required in any multi-LUN controller. The second best place to implement LUN Masking is in the host bus adaptor on each host. This provides a standard solution on all hosts with the same HBA.

Over the next few years higher level management software will be available which will unify the various implementations under a single management structure. This will allow different methods to be used, but managed in a standard fashion so that more robust configurations can be built with a range of underlying technology.

