Performance Analysis of a Fibre Channel Switch supporting Node Port Identifier Virtualization

Suresh Muknahallipatna and Joseph Miles
Department of Electrical and Computer Engineering, University of Wyoming
1000 E University Ave, Laramie, WY – 82071, USA
Email: sureshm@uwyo.edu
Howard Johnson, Brocade Communications Systems, Inc,
San Jose, CA - 95110 USA

Abstract-**The server virtualization architecture encompassing sharing of storage subsystems among virtual machines using fibre channel fabrics, to improve server utilization and reduce the total cost of ownership, was pioneered by IBM through their System *z9* mainframe and its predecessors. With the advent of sharing small computer system interface storage subsystems among host servers through fibre channel based storage area networks, has cropped up new set of security and associated performance issues when the host servers are virtual machines on a single physical server. To address the security issues and reduce the total cost of ownership, IBM introduced new storage virtualization architecture known as node port identifier virtualization enabling thousands of virtual machines on a server to share storage subsystems through a few numbers of host bus adapters.**

In this paper, we introduce the node port identifier virtualization architecture and the associated fibre channel switch latency performance issue that would affect virtual machine instantiation when supporting thousands of virtual machines. We first show the architectural problem in hard zoning mechanism contributing to the large fibre channel switch latency by actual performance measurements on a switch using hardware simulators. Next, we suggest a modification to the hard zoning mechanism to reduce the fabric channel switch latency significantly and demonstrate the reduction using hardware simulators. The performance issue we have identified and addressed will allow a single fibre channel switch to support thousands of virtual machines on a server using only a few numbers of host bus adapters.

Index Terms-**Fibre Channel, NPIV, SAN, Virtualization**

I. INTRODUCTION

The need to deploy enterprise class applications rapidly to support the expanding business process has led to infrastructure growth in an unplanned way: often new servers and storage subsystems are purchased to deploy a new application. In many cases, this is led to data centers populated with underutilized servers, with heterogeneous storage subsystems. The average use of server capacity is only 10-35 %, wasting valuable resources and increasing the complexity of maintaining this sprawl of servers and storage subsystems.

The problem of heterogeneous storage subsystems in a data center has been addressed, to a certain extent, by implementing storage area networks (SAN). A SAN is a network that provides the capability to connect servers and storage subsystems to move small computer system interface (SCSI) protocol based data packet encapsulated in fibre channel protocol (FCP) based packet. A SAN allows enterprise class applications hosted on servers to access large storage repositories distributed across multiple storage subsystems satisfying availability and security requirements.

A typical SAN illustrated in Fig. 1 consists of switches, directors and physical media, known as fabric in addition to servers and storage devices. The fabric transfers' SCSI packet encapsulated in a fibre channel (FC) packet between servers and storage subsystems leading to SCSI over FCP (SCSI-FCP). The storage subsystems and the servers are connected physically to the fabric by point-to-point links between a node port (N_Port) on the host bus adapters (HBA) and a fabric port (F_Port) on a switch/director. A HBA similar to gigabit network interface cards improves the performance of server/storage subsystems by relieving the server/storage subsystems CPUs of both data storage and retrieval tasks. In a FCP SAN, the HBA is identified by a manufacturer hard coded unique 64 bit wide number known as world-wide node name (WWNN) and each N_Port on the HBA by a 64 bit wide number known as world-wide port name (WWPN).

DOI: 10.5176_2010-2283_2.1.156

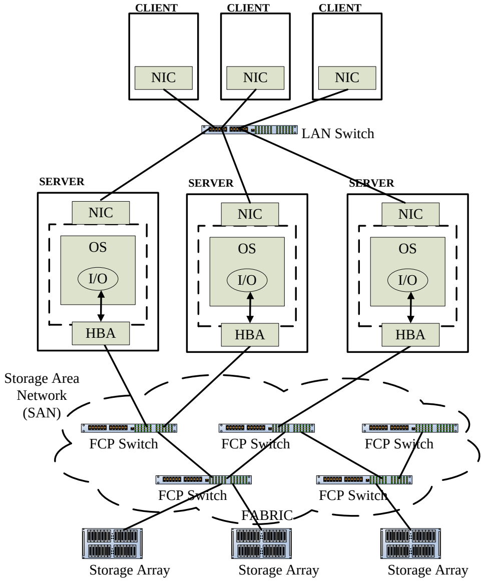

Fig. 1. Architecture of a typical Storage Area Network.

The WWPN in case of a server node is used to identify the server OS image across the entire fabric. During the node initialization process (discussed in detail in subsequent section), a N_Port transmits its WWPN and WWNN to connected switch/director in the fabric, wherein the received WWPN and WWNN are stored in the switch zone server database. The switch in turn assigns a unique 24-bit FC address identifier (N_Port ID) to the N_Port to be used in future transmissions. In Fig. 1, it can be seen that, it is possible for each server to have access to all storages in the SAN, which would lead to I/O access security issues. A SAN administrator typically limits access to storage subsystems to only desired servers through a mechanism known as soft zoning [8]. The soft zoning mechanism consists of creating a zone set of multiple zones in the zone server database on a switch. Each zone consists of nodes that are supposed to have I/O access only among them. The nodes in the zone are identified either by their WWPNs or WWNNs (single port HBA) but not by their assigned N_Port IDs.

The problems of server underutilization and administration are being addressed through an architectural approach known as server virtualization. Server virtualization architecture consists of physically or logically partitioning a physical server to allow concurrent execution of virtual machines (OS images) thereby increases the utilization of the physical server. Recently, server virtualization normally confined only to mainframes like IBM zSeries or specialized servers are now being implemented on off-the-shelf inexpensive servers increasing the popularity of server virtualization.

The VMs are spawned and managed by special software known as the hypervisor. The hypervisor spawns a number of abstract copies of the physical hardware leading to multiple VMs. A number of companies have developed hypervisors for server virtualization. The most widely used is ESX by VMWare, Intel Virtualization Technology [22], Microsoft Hyper-V, and Linux-based Xen. All the above virtualization products are based on the VM architecture proposed by IBM [20]. The primary objective for server virtualization is to reduce the total cost of ownership (TCO) and increase corporate data center utilization by consolidating multiple OS/applications onto a single physical server. In addition to the reduction in TCO and increase in utilization, improvements in system security and reliability are also achieved through server virtualization. System security is improved by confining an intrusion to an individual VM. Reliability is improved by isolating software stacks on their own VM and hence software failures in one VM do not affect other VMs.

With server virtualization becoming more prevalent in data centers the need to seamlessly combine server virtualization with SAN has become a necessity. Fig. 2 illustrates the architecture of server virtualization with a SAN.

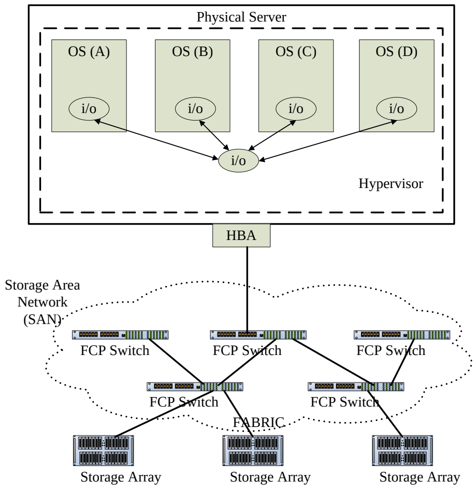

Fig. 2. Architecture of Server Virtualization with a SAN.

In Fig. 2, it can be seen that each VM is provided with an abstract copy of the I/O hardware (HBA and N_Ports) by the hypervisor. The VMs will access the storage subsystems using this abstract I/O there by sharing a single or multiple N_Ports on the actual HBA. As discussed previously, a N_Port on the HBA is identified by a single WWPN and a corresponding FC address. Therefore, then I/O traffic from multiple VMs sharing a N_Port will also share the WWPN and FC address of the N_Port. The sharing of WWPN and FC address will prevent the SAN from segregating I/O traffic between different VMs and thereby introduce new drawbacks listed below: A single storage subsystem would be visible to multiple VMs sharing the same WWPN and thereby susceptible to I/O access security breaches across VMs. The system security improvement of confining intrusions to a single VM by virtualization is negated. A single storage subsystem would be visible to multiple VMs sharing the same WWPN due to the soft zoning mechanism is based on identifying each N_Port (in turn OS) with a unique WWPN. A specific storage subsystem cannot be assigned to a particular VM among a group of VMs sharing the same WWPN thereby limiting quality of service implementations.

One of the key features of server virtualization is the ability of a VM migrate from one physical server to another due to hardware failure using a suitable migration tool. The migration tool is used initially to take an existing physical server and make a virtual hard drive image of that server with the necessary modifications to the driver stack so that the server will boot up and run as a virtual server. On the event of hardware failure, the migration tool is used to restore the virtual server on a healthy physical server using the previously created virtual hard drive image. This key feature would be severely restricted when the VM is accessing storage subsystem by sharing a N_Port. The abstract I/O stack would need to reflect the WWPN of the HBA on the health physical server.

To overcome the above drawbacks, a simple solution would be to equip each VM with a dedicated N_Port on a HBA port thereby not share WWPN and FC address. The IBM system z9 with z/VM hypervisor [21] can host more than a thousand VMs and even the Microsoft Hyper-V/VMWare ESX hypervisors can host 32 VMs on an off-the-shelf server. This high number of VMs would require a large number of HBAs to be installed on a single physical server to provide dedicated N_Ports to the VMs. This would increase the total cost of ownership (TCO) and reduce the utilization of the physical server since individual VMs may not have a workload to utilize the full capacity of a dedicated N_Port. Typically, servers provide for the installation of a few numbers of HBAs and thereby limiting the number of available N_Ports. Thus equipping the physical server with large number of HBAs is not a practical solution. Hence, there is a need for a new mechanism allowing the sharing of N_Ports among multiple VMs with an ability to identify each VM to the SAN fabric and thereby the existing SAN zoning protocols would ensure security and reliability. The need for the new mechanism led to the development of N_Port ID virtualization (NPIV) by IBM [21] and adaptation [6] considering the drawbacks of other approaches.

In this paper, we introduce the node port identifier virtualization architecture and the associated fibre channel switch latency performance issue that would affect virtual machine instantiation when supporting thousands of virtual machines. We first show the architectural problem in hard zoning mechanism contributing to the large fibre channel switch latency by actual performance measurements on a switch using hardware simulators. Next, we suggest a modification to the hard zoning mechanism to reduce the fabric channel switch latency significantly and demonstrate the reduction using hardware simulators. This paper is continuation of the short paper [1] previously presented at the local computer network conference in 2009.

The remainder of this paper is organized as follows: Section 2 presents related precursor mechanisms to NPIV to address I/O isolation issues. Section 3 provides a discussion of the physical N_Port Initialization process. In section 4 a detailed discussion of NPIV mechanism is presented. Performance results, analysis and modifications to improve the performance of the fibre channel switch are presented in section 5. Finally, section 6 concludes highlighting the latency issue, solution and future direction.

II. RELATED WORK

The NPIV mechanism is based on the approaches like FC process associators, VM identification at upper level protocol and FC hunt groups that were investigated to address virtualization and SAN interfacing issues. A brief discussion of these approaches is presented in this paper and detailed discussions can be found in the literature [2][3][15][21][23].

II-I FC Process Associators

This mechanism proposed including an optional seven-byte number known as the process associator of a VM as a part of the FC frame header. A unique process associator is chosen by the host/hypervisor for a VM at the time of creation and subsequently this unique process associator is included in every I/O frame transmitted identifying the VM to the SAN fabric. In order to implement this mechanism the existing FC and SCSI-FCP protocols had to support the use of process associator. The use of process associator is supported by FC protocol, whereas SCSI-FCP protocol does not support this feature due to lack of support in existing SCSI devices. Therefore, the use of this mechanism would eliminate use of SCSI devices in SAN, which is an important component of FC.

II-II VM Identification at Upper Level Protocol Layer

Initially, IBM developed a storage protocol known as enterprise systems connection (ESCON) protocol based on an optical serial interface between IBM mainframe computers and peripheral devices such as storage and tape drives. The ESCON protocol uses a separate host logical addresses (HLA) at the upper layer of the protocol for each VM, to identify the VM to the network. Later, this was adopted to run on FC fabrics leading to the fibre connectivity (FICON) protocol. Again the lack of HLA capability in upper layers of SCSI-FCP protocol, made this solution also not applicable. Use of this mechanism required modification to the SCSI-FCP protocol, which has been in existence for decades, leading to the necessity of modifying a large base of SCSI devices in use.

II-III FC Hunt Groups

A group of N_Ports controlled by a common entity such as a VM is known as a hunt group and the common entity is identified to the network by a hunt group identifier. The hunt group concept being a part of the FC standard [7] does not require any changes to existing SCSI devices or FC protocol, but it has its limitations. The FC protocol by means of the 8-bit hunt group identifiers can identify a maximum of 256 VMs whereas thousands of VMs needs to be identified to a single SAN. The hunt groups have to be managed by an alias server [8] in the fabric, leading to increased configuration complexity.

Mechanisms like multicasting, parallel multi addressing [19] and fabric virtualization were also investigated and deemed unsuitable for implementation due to their complexity and modifications required to the existing infrastructure. Recently, another proprietary alternative by Cisco Systems Inc.'s known as VFrame technology has been proposed, but it requires infrastructure changes as it utilizes InfiniBand [18].

All the drawbacks of the various approaches discussed in this section lead to the development of N_Port identifier virtualization mechanism to address the issues of interfacing server virtualization and SAN. Next, the N_Port initialization process describing how a physical N_Port acquires 24-bit FC address is discussed.

III. PHYSICAL N_PORT INITIALIZATION PROCESS

The manufacturer assigned unique WWPN for a N_Port on a HBA is used for purpose of identifying the N_Port by various fabric management services like the name and zone servers. To perform actual I/O data transmission using the FC protocol a N_Port would require the N_Port ID (24-bit FC address identifier) of itself and the destination which is assigned by the fabric during N_Port initialization process.

An N_Port on either server or storage subsystem when physically connected to the fabric through a F_Port on a switch will initiate N_Port initialization process to request a N_Port ID. The N_Port ID is divided into three fields namely the domain identifier (D_ID), the area identifier (A_ID) and the port identifier (P_ID) as shown in Fig. 3.

Bits	23:16	15:8	7:0
Value	D_ID	A_ID	P_ID

Fig. 3. N_Port Identifier Address Format.

The content of the individual fields is dictated by the connection location of the N_Port in the fabric. The connection location can be defined as the 2-tuple (D_ID, A_ID) where D_ID is the domain identifier of a switch and A_ID is the area identifier of a F_Port on that switch. In Fig. 4 a fabric with two switches and two HBAs in a server and storage subsystem are shown. First, the two switches will go through a fabric initialization process [11][12] to obtain unique 8-bit domain identifiers and then each switch assigns predefined unique 8-bit area identifiers to all of its F_Ports as shown in Fig. 4.

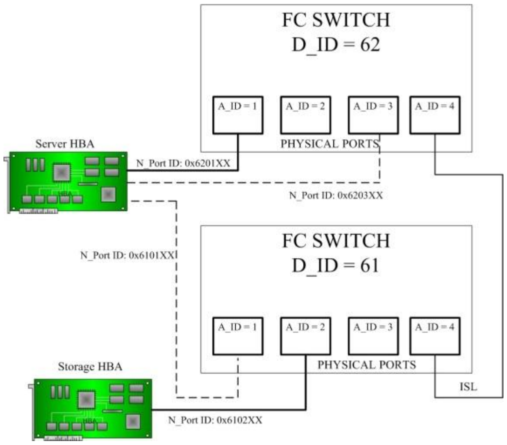

Fig. 4. Architecture of N_Port ID Allocation Process in a typical SAN.

Next, a N_Port, when plugged into a F_Port on either of the two switches will initiate N_Port initialization process shown in Fig. 5.

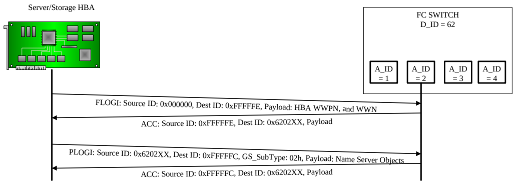
Server/Storage HBA
FC SWITCH
D_ID = 62
A_ID = 1
A_ID = 2
A_ID = 3
A_ID = 4
FLOGI: Source ID: 0x000000, Dest ID: 0xFFFFFE, Payload: HBA WWPN, and WWN
ACC: Source ID: 0xFFFFFE, Dest ID: 0x6202XX, Payload
PLOGI: Source ID: 0x6202XX, Dest ID: 0xFFFFFC, GS_SubType: 02h, Payload: Name Server Objects
ACC: Source ID: 0xFFFFFC, Dest ID: 0x6202XX, Payload

Fig. 5. Architecture of N_Port Initialization Process.

The N_Port initialization process starts with the N_Port transmitting a fabric login (FLOGI) frame (command) to the fabric login server [8] located on the connected switch (each switch in the fabric hosts an identical fabric login server and other generic services) at the well known address (WKA) (0xFFFFFE) [9] as illustrated in Fig. 5. The source N_Port ID in the transmitted FLOGI frame is set to 0x000000h the destination N_Port ID is set to 0xFFFFFEh, and the payload of the frame contains the WWPN of the N_Port and the WWNN of the HBA. The switch on receiving the FLOGI frame acknowledges with a link service accept (LS_ACC) frame, which has the source N_Port ID set to 0xFFFFFEh the destination N_Port ID set to 0xD_IDA_IDXXh (0x6201XXh as shown in Fig. 5) formed using the connection location 2-tuple and the payload containing the operational parameters [9]. The port ID field is set to a random number, and it ignored in a fabric that supports only physical N_Ports. The received WWPN, WWNN and the assigned N_Port ID are recorded in the local fabric login table by the switch. The switches in a fabric exchange their local fabric login table to build a global fabric login table. The N_Port will use the destination N_Port ID of the received LS_ACC frame as its N_Port ID in all of its subsequent transmissions. Next the N_Port registers with various fabric services by transmitting a port login (PLOGI) command to the directory server on the switch. The PLOGI frame has its source ID N_Port ID set to the assigned N_Port ID, the destination N_Port ID set to the WKA: 0xFFFFFCh of the directory server and with an appropriate payload. The N_Port registers with various fabric services so that its information can be provided to other N_Ports (server discovering storage subsystems).

Since, every N_Port executes the N_Port ID initialization process; the fabric login table will contain the WWPN, WWNN and the assigned N_Port ID of all N_Ports. A SAN administrator using fabric manager software can access the WWPNs of the N_Ports to create zone sets in the zone server database. Next, when a N_Port queries the fabric for other N_Ports, information of N_Ports belonging only to the same zone set will be provided by the zone server [8]. If the server N_Port with the N_Port ID (0x6201XXh) is unplugged and connected to another F_Port shown as dotted lines in Fig. 4, the server N_Port will be receiving a new N_Port ID. But, the zoning information would not change since zoning is implemented using the WWPN.

The work load on a switch due to N_Port initialization process is minimal since a switch at any given time has to service a maximum of 256 N_Port initialization requests. Also, a server or storage device in an enterprise data center will initiate the physical N_Port initialization process very infrequently, since the devices do not experience frequent power cycling, link failures and timeouts. Therefore, the delay and performance of a switch during a physical N_Port initialization process is not of importance. In the next section, a detailed discussion of the NPIV mechanism is presented.

IV. N_PORT IDENTIFIER VIRTUALIZATION MECHANISM

The goal of NPIV mechanism is to present a single physical N_Port as multiple virtual N_Ports to the fabric and thereby obtain unique N_Port identifiers for each virtual N_Port. The N_Port IDs of the virtual N_Ports would then be assigned by the hypervisor to each instance of a VM. The NPIV mechanism is developed by using the ignored P_ID field of the physical N_Port ID. A partial command sequence to implement NPIV mechanism was proposed in the fibre channel device attach standards [6] and later a complete command sequence has been developed in consultation with ANSI T11 organization, IBM, Brocade and other manufactures. The complete command sequence to create a virtual N_Port is shown in Fig. 6.

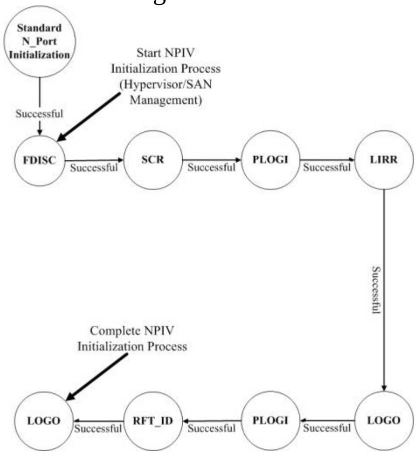
Standard N_Port Initialization
Start NPIV Initialization Process (Hypervisor/SAN Management)
Successful
FDISC
Successful
SCR
Successful
PLOGI
Successful
LIRR
Successful
Complete NPIV Initialization Process
LOGO
Successful
RFT_ID
Successful
PLOGI
Successful
LOGO

Fig. 6. NPIV Command Sequence.

The hypervisor or fabric management software initiates the NPIV mechanism on the physical N_Port which has obtained a N_Port ID through a successful physical N_Port initialization process. The NPIV initialization process starts with the physical N_Port transmitting a discovery fabric (FDISC) service frame (command) to the fabric login server. The source N_Port ID is set to 0x000000h the destination N_Port ID is set to 0xFFFFFEh (fabric login server) in the FDISC frame identical to a FLOGI frame. The payload of the frame will contain the WWNN of the HBA, a new virtual WWPN (supplied either by the hypervisor or fabric management software) and class of service parameters. The switch prior to responding to the FDISC command with a LS_ACC frame will perform the operations listed below:

- Search the name server database [8] to determine if a N_Port with an identical WWPN to that in the FDISC payload exists in the fabric. Logout existing N_Port and then allocate a new virtual N_Port ID corresponding to the WWPN in the FDISC payload.
- Register the N_Port ID, WWPN, WWNN and the class of service fields of the new virtual N_Ports with the name server database.
- Determine with which existing N_Ports the new virtual N_Port can communicate (permissions) by searching the active zone set through the soft zoning mechanism.
- Add permissions into the switch F_Port's hardware routing table through the hard zoning mechanism [8]-[11] to allow future communications.

The switch then responds with a LS_ACC frame with the destination address set to the assigned virtual N_Port ID (0xD_IDA_IDYYh) which is used in all future communications by that particular VM. Since the virtual N_Port ID and the physical N_Port ID will differ only in the P_ID field 255 virtual N_Port IDs can be assigned to a single physical N_Port.

Initially, the virtual N_Port in comparison to the physical N_Port will not have capabilities like; request state change notifications of N_Ports in same zone; request notifications when link errors occur; and notify the level of FC protocol supported by the N_Port to the fabric. To achieve capabilities similar to that of the physical N_Port additional commands have to be transmitted by the virtual N_Port after it has successfully obtained a virtual N_Port ID as shown in Fig. 6.

After the successful FDISC command, the state change registration (SCR) command is transmitted to the fabric controller (WKA: 0xFFFFFDh) to request a notification in the event of other N_Ports in the same zone undergo state changes (An example of a state change event is a storage subsystem performing login/logout). If, there is a state change event the fabric controller will send registered state change notification (RSCN) command to all other N_Ports in the same zone.

Next, the virtual N_Port requests to be notified of any link errors like link drop, loss of synchronization, etc., by transmitting the link incident record registration (LIRR) command to the management server during a session. The session is established by first transmitting a PLOGI command to the management server (WKA: 0xFFFFFAh) followed by the transmission of the LIRR command and termination of the session by transmitting the logout (LOGO) command.

The virtual N_Port next informs the fabric the FC-4 level protocol (SCSI-FCP) it supports by transmitting a register FC-4 protocol type (RFT_ID) command to the name server during a session.

By repeating the command sequence in Fig. 6 with new virtual WWPNs, the hypervisor or fabric management software can create additional virtual N_Ports and assign them to individual VMs. Since, a virtual N_Port has the same capabilities as that of a physical N_Port; the various fabric services are available flawlessly to perform management operations, like establish zones and isolate a part of the fabric to be used by particular VMs.

V. PERFORMANCE ANALYSIS OF A FIBRE CHANNEL SWITCH SUPPORTING NPIV

From the NPIV mechanism discussions in the previous section, it can be seen that each VM is now uniquely identified across the SAN fabric and then I/O security/reliability is ensured. But, at the same time a fabric switch experiencing significant increase in its workload due to NPIV mechanism should not cause any performance issues with stable fabric operation and VM instantiation process. The performance issues related with respect to VM instantiation process are the instantiation time and the number of VMs instantiation at a given time. Based on a number of studies of VM instantiation, replication and management [5][16][17] the time required to instantiate or replicate a VM without NPIV on typical server hardware configurations is in the range of 300 to 800 msecs. In data centers, requesting a large number of concurrent/successive requests for virtual N_Port IDs (from a single IBM zSeries physical server or multiple hypervisor/physical platforms) from a single FC switch during initial VM setup, reactivation (due to internal errors) and workload balancing [14] are common scenarios. Hence, a NPIV supporting switch should be capable of handling large number of concurrent NPIV requests and not significantly increase the time required to instantiate or replicate a VM. Any significant increase in this time will cause timing violations at the hypervisor [3] and an unstable fabric due to violation of the Error Detect TimeOut Value (E_D_TOV) fabric timer [10]. The

E_D_TOV is the basic error timeout used for all FC error detection with a default value of 2 seconds.

Therefore, the performance of the switch during the process of issuing virtual N_Port IDs is crucial and the performance can be ascertained by measurements of the delay introduced by the switch in servicing a NPIV request.

V-I Test setup and procedure

The test setup consists of an early Brocade Communications Systems Fibre Channel switch[1] connected to storage and fibre channel simulator (FCSim) cards as shown in Fig. 7. The FCSim cards are special purpose real time programmable embedded systems (on-board processors, memory and OS) which are programmed to generate the FCP commands in the NPIV mechanism shown in Fig. 6. Individual FCSim cards are connected together through a shared PCI bus in a special custom built chassis for synchronization. The main processor on the chassis is used to trigger the tests on each FCSim card. A test program capable of repeating the NPIV command sequence with increasing WWPNs is loaded on each FCSim card.

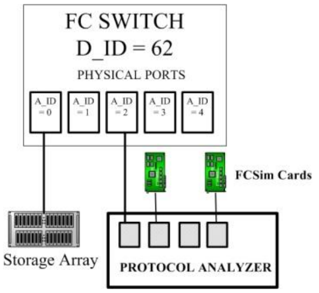

Fig. 7. Performance Analysis Test Setup.

A zone set comprising of number of zones equal to the desired number of virtual N_Port IDs that would be requested by the test program is created first on the switch. Each zone is populated with the WWPN of the storage node and a unique WWPN for which the FCSim cards will request a virtual N_Port ID through the NPIV mechanism. This will ensure that only a single RSCN command is sent only to the storage array (a typical server storage behavior) for each NPIV request, keeping the delay due to state change notification messages to a minimum. The protocol analyzer in Fig. 7 is used to measure the time elapsed between the reception of a command by the switch and its response (LS_ACC frame) known as the switch latency. The tests are repeated ten times to determine the average and standard deviation of the measured latency. The Brocade Fibre Channel switch used in this test can support a maximum of 1000 virtual N_Port IDs with a maximum of 255 virtual N_Port IDs through a single F_Port.

V-II Performance results of virtual N_Port requests on a single F_Port

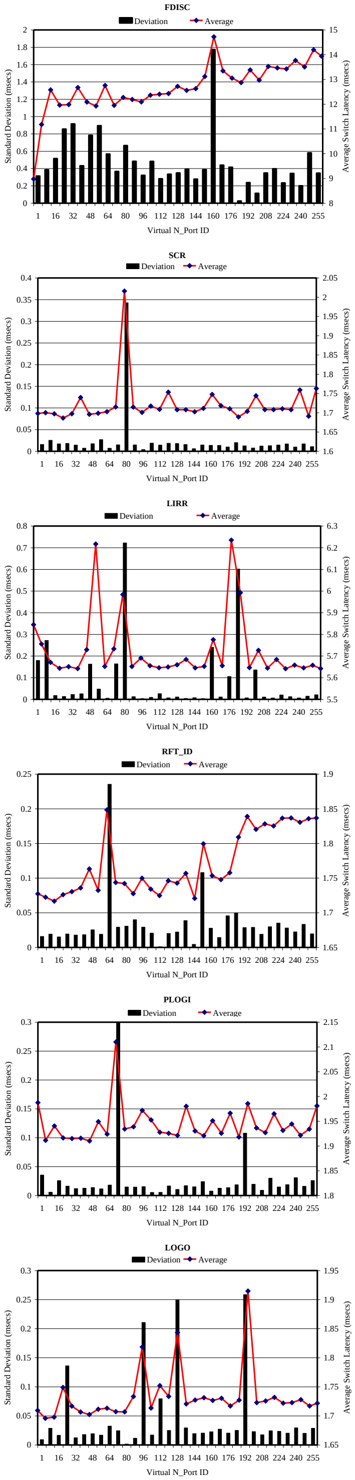

Fig. 8. ASL of Individual Commands in the NPIV Sequence.

[1] Brocade Communications Systems Sphereon 4500.

The first set of latency measurements were performed by running the test program on a single FCSim card connected to a F_Port on the switch and requesting 255 virtual N_Port IDs consecutively. The average of the switch latency (ASL) and standard deviation of individual commands of the NPIV sequence are shown in Fig. 8.

Examining the values of ASLs of the individual commands in Fig. 8, the FDISC command experiences maximum delay, which correlates with the large and increasing workload (previously discussed in section V) the switch has to perform before responding, whereas the workload for the other commands is very small and constant [11]. For the FDISC command, the difference in ASL between the 1st and 255th request is 5 msecs due to the increasing number of name server database search [8] operations the switch has to perform, whereas the ASLs for the other commands is approximately constant. Combining the individual command ASLs, it can be seen that the total latency introduced by the switch in granting the virtual N_Port ID through a single F_Port will increase the VM instantiation time only by few milliseconds and not cause any hypervisor timing violations. Also, this small value of latency will not cause any fabric timeouts. For some instances of the virtual N_Port ID request, the standard deviation is higher due to the switch performing mandatory management operations. In subsequent tests the ASL of only the FDISC command is presented since it has a very large impact on the performance in comparison to the other commands.

V-III Performance results of virtual N_Port requests on multiple F_Ports

A set of latency measurements were collected by requesting a total of 765 virtual N_Port IDs through multiple F_Ports on the same switch using three FCSim cards. The test was performed by first requesting 255 virtual N_Port IDs through the 1st F_Port and then requesting two sets of 255 virtual N_Port IDs on the 2nd and 3rd F_Ports. The ASL of the FDISC command for 255 requests through each F_Port is shown individually in Fig. 9.

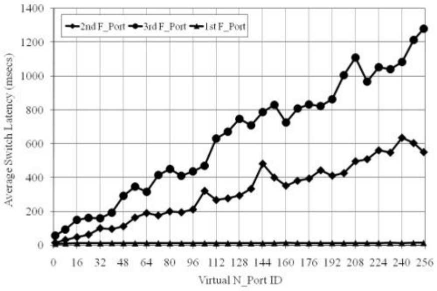

Fig. 9. ASL for FDISC Command through Multiple F_Ports.

The ASL of the FDISC command for the first 255 requests through the first F_Port is identical to the measurement shown in Fig. 8, whereas the ALS for the 255 requests through the second and third F_Ports are increasing linearly and reach values of 600 and 1300 msecs respectively. These high values of latency will cause two to three fold increases in the VM instantiation time causing hypervisor timing violations [3]. Also, as hypervisor requests additional virtual N_Port IDs due to the creation of VMs to accommodate increasing application workload, the server will keep waiting for longer periods to receive virtual N_Port IDs [21]. Even if the hypervisor is redesigned to handle this huge increase in VM instantiation time, the fabric is being pushed towards instability due to the ASL of the 255th virtual N_Port ID request through the 3rd F_Port approaching 66% of the E_D_TOV. Requests through additional F_Ports or any disturbances in the fabric during the requesting phase can cause fabric rebuild and the servers will be denied temporarily access to the storage subsystems.

To verify whether this linear increase of ASL was unique to Brocade Fibre Channel switch, measurements were repeated with switches from Cisco Systems Inc. and Qlogic Corp., and they also exhibited similar linear increase of the FDISC ASL indicating flaw(s) in the NPIV mechanism architecture described by the FC standards [6]-[9].

Since the requests were not sent to the three F_Ports at the same instance of time by synchronizing the FCSim cards, there are no switching operations between F_Ports and buffering of incoming frames, which could increase the FDISC ASL. In Fig. 9, it can be noticed that the FDISC ASL through the 3rd F_Port increases at a higher rate in comparison to that through the 2nd F_Port and for a particular virtual N_Port ID request, the ASL is dependent on the number of virtual N_Port IDs that has been granted successfully through that F_Port. Based on the above observations, the linear increase of FDISC latency has to be due to some unforeseen operations within the name server and zone server databases performed by the switch on receiving the FDISC command. As discussed in section 5, a switch on receiving the FDISC command performs the four operations namely a search in the NS database; registers with the NS database; determine communication permissions using soft zoning mechanism; and finally implement hardware routing tables using hard zoning mechanism. The first three operations mentioned above are common between the virtual N_Port ID physical N_Port initialization processes. In a related work [4], we have determined through actual measurements and simulations that the latency introduced by a switch due to the first three operations is in the order of few milliseconds. Since the hard zoning mechanism is implemented only in case of virtual N_Port IDs, next the hard zoning mechanism examined in detail to identify the flaw causing large latency.

V-IV Mechanism of hard zoning

The zoning mechanism discussed in section 3 known as the soft or name server zoning using the WWPN of N_Ports to limit access of storage subsystems only to desired servers is most commonly implemented in all switches forming the fabric. The access to the storage subsystems is limited to servers, which are also members (identified by their WWPNs) of the same zone. The advantage of soft zoning is that even when a server or storage subsystem physical connection (F_Port) to the fabric is moved or removed there is no need to change the zone information. The disadvantage with soft zoning is the security risk associated due to potential for spoofing a WWPN by a host. In case of a physical N_Port, the WWPN is never communicated to the upper levels of I/O protocol (limited to physical layer) on a server and therefore a hacker has to guess the WWPN in order to cause a security breach.

In case of virtual N_Port IDs the WWPN generated by the hypervisor is communicated to higher levels of the protocol since each VM has to use its WWPN for any I/O operation. The WWPN at the higher levels of protocol can be potentially hacked and used in spoofing. Also, due to errors in a hypervisor multiple VMs can end up with same WWPN after the virtual N_Port IDs have been assigned. To address these security issues with virtual N_Port IDs the FC standards have made hard zoning mandatory in addition to soft zoning on fabrics support NPIV mechanism.

Hard zoning involves creating zone sets using assigned N_Port IDs, which are never communicated above the physical level of the protocol and using the zone set to implement a routing table on application specific integrated circuit (ASIC) on each F_Port on the switch. The routing table on a F_Port will contain N_Port IDs between which FC frames can be exchanged. The hard zoning involves checking source and destination N_Port IDs of each received frame against the routing table entries and dropping the frame if they are not in the same zone. The routing table generation algorithm of the hard zoning mechanism is discussed below:

1. A switch receiving a N_Port ID request on a F_Port (source_port) will compare with other previously assigned N_Ports on all other F_Ports (destination_ports) on the same switch to determine if the N_Port IDs are members of the same zone.
2. If the N_Port IDs are in the same zone, the N_Port IDs are now added to the routing table of both source and destination F_Ports.
3. If a login/logout operation occurs on a F_Port the steps 1 & 2 are to be repeated.

Based on this algorithm, on a switch with 256 F_Ports supporting only physical N_Ports when a N_Port ID request is received a maximum of 255 zone comparisons are possible. In case of a switch supporting NPIV on receiving a FDISC command at a F_Port (source_port), all N_Port IDs (physical and virtual) at the source_port has to be compared with other N_Port IDs (physical and virtual) at all other F_Ports (destination ports) to update the routing table.

Applying this routing table algorithm to measurements shown in Fig. 9, it can be seen that when the FDISC commands are issued on the 1st F_Port, there are no other assigned N_Ports IDs on any other F_Ports and the switch will not perform any zone comparisons leading to a very low FDISC ASL. On receiving the first FDISC command on the 2nd F_Port, zone comparisons between the previously assigned physical N_Port ID on the 2nd F_Port and 256 N_Port IDs (255 virtual and 1 physical) on the 1st F_Port is performed. Next, the newly assigned virtual N_Port ID on the 2nd F_Port is again compared with the 256 N_Port IDs (255 virtual and 1 physical) on the 1st F_Port resulting in a total of 512 zone comparisons. On receiving the second FDISC command on the 2nd F_Port will result in a total of 768 zone comparisons and 65,536 zone comparisons for the 255th FDISC command causing a 600 msecs latency as shown in Fig. 9. In case of the 3rd F_Port, a total of 131072 zone comparisons is performed for the 255th FDISC command introducing 1300 msecs latency. The increasing number of zone comparisons with each FDISC command received is due to repeated zone comparison of previously assigned N_Port IDs, whenever a login/logout operation occurs on the F_Port in order to reflect the current device attachment of the fabric in the routing table [6].

If, FDISC commands are synchronously issued on multiple F_Ports, the FDISC ASL will increase to large value on all participating F_Ports. The FDISC ASL on both 1st and 2nd F_Ports shown in Fig. 10 is seen reaching a high value of 600 msecs when the FDISC commands are synchronously issued. Based on the current NPIV mechanism architecture and hard zoning requirements described in the FC standards it can be seen that a single switch can support only 255 virtual N_Ports without causing hypervisor timing violations and fabric instability.

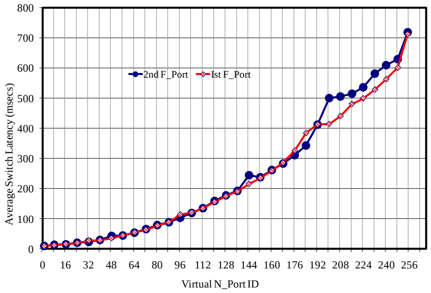

Fig. 10. ASL for Synchronized FDISC Commands through 1st and 2nd F_Ports.

To support more than 255 virtual N_Ports, a hypervisor has to send FDISC commands to multiple switches, which would increase the management complexity of the hypervisor and TCO of the fabric. Therefore, an appropriate modification to the hard zoning architecture would be a better solution to address the latency issue with NPIV.

V-V Modified hard zoning architecture

As discussed in the previous section, the large value of FDISC ASL is due to growing number of repeated zone comparisons, which is required to update the routing table. The zone comparison for an existing physical N_Port ID whenever a new virtual N_Port ID request is received is not necessary since it is possible to issue a virtual N_Port ID request only through a logged in physical N_Port. If the logout event of a particular virtual N_Port is not updated in the routing table it would not cause any malfunction since the hypervisor keeps track of the state of virtual N_Port IDs. Therefore, a switch supporting NPIV on receiving a FDISC command should perform zone comparisons each only between the new virtual N_Port ID on the source port and all N_Port IDs on all other destination ports.

Tests were again repeated with a new firmware incorporating the suggested modification to hard zoning on the same Brocade Fibre Channel switch. The ASL of the FDISC command for 255 requests through each F_Port is shown individually in Fig. 11. The FDISC ASL is approximately constant in comparison to the linear increase seen in Fig. 9. Each FDISC command through the 3rd F_Port will cause 512 zone comparison operations, whereas through the 2nd F_Port will cause only 256 zone comparison operations.

The reduction in FDISC ASL from 1300 to 50 msecs in case of the 3rd F_Port will eliminate any hypervisor timing violations and fabric instability issues. The reduction in latency due to this suggested modification enables a single switch to support large number of virtual N_Ports thereby without introducing adverse effect the performance of the hypervisor and provides the switch manufacturer's capacity to support larger a number of virtual N_Ports on a single switch.

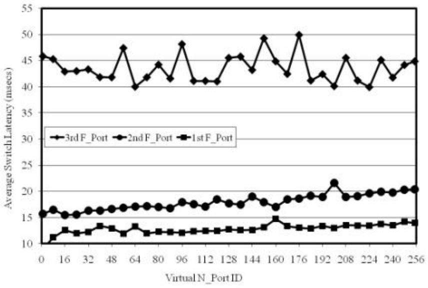

Fig. 11. ASL of FDISC Command through the 2nd and 3rd F_Port with New Firmware.

VI. CONCLUSION

In this paper, the importance of the node port identifier virtualization mechanism to seamlessly combine server virtualization with SAN has been discussed. The performance issue of large latency to FDISC commands when the FC switch allocates virtual N_Ports IDs in excess of 255 was identified. The reason behind the linear increase in FDISC ASL was identified as the repeated zone comparisons in hard zoning mechanism as mandated by the current FC standards. With a modification to the hard zoning architecture, a significant reduction in FDISC ASL was achieved. The performance improvement achieved through the suggested modification should allow seamless integration of large scale server virtualization and storage area networks satisfying security and reliability issues. Currently, all FC switches and directors manufactured by Brocade Communication Systems, implements NPIV with the modified hard zoning architecture. In the future, analysis of the performance of a FC switch handling actual data from multiple I/O channels from a virtualized server will be performed. Also, the effect of overhead by the hypervisor introduced due to NPIV on the actual I/O operation will be studied.

REFERENCES

[1] Muknahallipatna, S., Miles, J., Hamman, J., Denson, P., Lewis, R., Johnson, H., and Noe, K., Performance Analysis of a Fibre Channel Switch Supporting Node Port Identifier Virtualization: Preliminary Results, 34th IEEE Conference on Local Computer Networks (LCN), 2009, pp. 229 – 232.

[2] Adlung I, Banzhaf G, Eckert W, Kuch G, Mueller S, Raisch C. FCP for the IBM eServer zSeries Systems: Access to distributed storage, IBM Journal of Research and Development, July/Sept. 2002, Vol. 46, No. 4/5, pp. 487 – 502.

[3] Banzhaf G, Mueller S, Friedrich R, Rund C. Host-Based Access Control for zSeries FCP Channels, IBM z/Journal, August/September 2005, pp. 99 – 103.

[4] Brothers JT, Muknahallipatna S, Hamann CJ, Johnson H. Fibre Channel Switch Modeling at Fibre Channel-2 Level for Large Fabric Storage Area Networks using OMNET++: Preliminary Results, 32nd IEEE Conference on Local Computer Networks (LCN), Oct. 15-18, 2007, Dublin, Ireland, pp. 191- 199.

[5] Constandache I, Yumerefendi A, Chase J. Secure Control of Portable Images in a Virtual Computing Utility, Proceedings of the 1st ACM workshop on Virtual Machine Security, October 27, 2008, pp. 1 – 8.

[6] FC-DA-2: Fibre Channel Device Attach, International Committee for Information Technology Standardization (INCITS): NPIV Acquisition Procedure, October 2008, Rev. 1.03.

[7] FC-FS-3: Fibre Channel Framing and Signaling, International Committee for Information Technology Standardization (INCITS), April 2009, Rev. 0.81.

[8] FC-GS-6: Fibre Channel Generic Services, International Committee for Information Technology Standardization (INCITS), March 2009, Rev. 9.3.

[9] FC-LS-2: Fibre Channel Link Services, International Committee for Information Technology Standardization (INCITS), May 2009, Rev. 2.11.

[10] FC-PH-3: Fibre Channel Physical and Signaling Interface, International Committee for Information Technology Standardization (INCITS), November 1997, Rev. 9.4.

[11] FC-SW-5: Fibre Channel Switch Fabric, International Committee for Information Technology Standardization (INCITS), June 2009, Rev. 8.5.

[12] Kembel RW. Fibre channel: A Comprehensive Introduction, 1st Edition, Northwest Learning Associates Inc.; 2002.

[13] Kembel RW. Fibre Channel Switched Fabric, 1st Edition, Northwest Learning Associates Inc.; 2002.

[14] Khanna G, Beaty K, Gautam K, Kochut A. Application Performance Management in Virtualized Server Environments, The 10th Network Operations and Management Symposium , NOMS 2006, April 3- 7, 2006, pp. 373 – 381.

[15] Kipp S, Guendert S, Johnson H. Consolidation Drives Virtualization in Storage Networks, IBM z/Journal, December/January 2007 pp. 40 – 44.

[16] Krsul I, Ganguly A, Zhang J. VMPlants: Providing and Managing Virtual Machine Execution Environments for Grid Computing, Proceedings of the ACM/IEEE SC 2004 Conference, Nov. 2004, pp. 7-7.

[17] Lagar-Cavilla HA, Whitney JA, Scanell A, Patchin P, Rumble SM, Lara ED, Brudno M, Satyanarayanan M. SnowFlock: Rapid Virtual Machine Cloning for Cloud Computing, Proceedings of the fourth ACM European Conference on Computer Systems, April 01-03, 2009, pp. 1 - 12.

[18] Liu J, Huang W, Abali B, Panda DK. High Performance VMM-Bypass I/O in Virtual Machines, Proc. of USENIX ACT, 2006, pp. 29 – 42.

[19] Meng B, Khoo PBT, Chong TC. Design and Implementation of Multiple Address Parallel Transmission Architecture for Storage Area Network, Twentieth IEEE/Eleventh NASA Goddard Conference on Mass Storage Systems & Technologies, April 2003.

[20] Smith JE, Nair R. The architecture of virtual machines, IEEE Computer, May 2005, Vol. 38, Issues: 5, pp. 32 - 38.

[21] Srikrishnan J, Amann S, Banzhaf G, Brice FW, Dugan R, Frazier GR, Kuch GP, Leopold J. Sharing FCP adapters through virtualization, IBM Journal of Research and Development, January/March 2007, Vol. 51, No. 1/2, pp. 1 – 16.

[22] Uhlig R, Neiger G, Rodgers D, Santoni AL, Martins FCM, Anderson AV, Bennett SM, Kagi A, Leung FH, Smith L. Intel virtualization technology, IEEE Computer, May 2005, Vol. 38, Issue: 5, pp. 48 – 56.

[23] Wyman LW, Yudenfriend HM, Trotter JS, Oakes KJ. Multiple-logical channel subsystems: Increasing zSeries I/O scalability and connectivity, IBM Journal of Research and Development, May/July 2004, Vol. 48, No. 3/4, pp. 489 – 505.