

# N\_Port\_ID Virtualization

## Functional Specification

Version 1.0

### 1 Project Description

N\_Port\_ID Virtualization (NPIV) allows one FibreChannel (FC) Port to represent many physical ports, thus virtualizing the hardware. System administrators can use their existing management schemes such as LUN Masking and Switch Zoning to perform access control against these logical ports. NPIV is especially useful for virtual machine environments such as Xen or Solaris Logical Domains.

#### 1.1 Definition

This project has several different modules. At the lowest level, the Solaris FibreChannel device driver stack (a.k.a. Leadville) must be changed to support NPIV. Specifically, the FCA (FibreChannel Adapter) drivers, which actually interface to the hardware must be changed. The interface between the FCA driver, and the generic FibreChannel port drivers (fp/fctl) must also be changed to support NPIV.

Further up, the HBA-API will be changed to manage NPIV. HBA-API is an industry standard interface designed to report status of the HBA and attached devices. Although the HBA API will be changed to support NPIV, the standard is immature, so we will be implementing a proprietary extensions, to be replaced when the standard is mature.

We will change the fcinfo(1M) utility to support NPIV. The utility will have new administration options to create and delete NPIV ports; this utility will be called fcdm, but implemented as a hardlink to fcinfo.

Containers, Xen and Logical Domains will all be supported. For each guest operating system, we will optionally map an NPIV WWN. If a volume appears on that NPIV WWN, it will be automatically mapped to the appropriate guest operating system.

Although we are planning to support Containers and Logical Domains with NPIV, with similar concepts, this work will be put back later. We will file a new PSARC case at that time. This two phase approach is due to the missing ability to dynamically add block devices to Logical Domains, which is already on the roadmap for Logical Domains, as well as resource constraints in the NPIV implementation team.

## **1.2 Motivation, Goals, and Requirements**

Target customers are those customers deploying relatively complex SANs and use Zoning and/or LUN Masking. They may also use Solaris virtual machine technology such as Xen, Logical Domains, or Containers. Using NPIV allows customers to use existing Zoning and LUN Masking techniques.

## **1.3 Changes From the Previous Release**

This project increases functionality of existing products. This project is a backwards compatible change to existing functionality, and should be considered a minor release.

## **1.4 Program Plan Overview**

### **• 1.4.1 Development**

Development is ongoing in Beijing. Many areas of the design, including FibreChannel device driver modifications, HBA API changes, and fcadm configuration tool have been prototyped and design is complete. Other areas, such as Xen and Containers have been partially prototyped.

### **1.4.2 Quality Assurance/Testing**

- Testing will consist of functionality, both without virtualization, and with xVM.
- Additionally, we will perform basic I/O performance tests, as well as SAN reconfiguration tests. We will test the maximum number of NPIV addresses on an HBA.
- We will test interoperability with Sun's OEM'd switches which support NPIV. We will test against both of Sun's OEM'd HBAs (Qlogic and Emulex)

### **1.4.3 Documentation**

This project requires updates to man pages as listed in the interface documents. It will require updates to manuals for FibreChannel and xVM. specifically:

- Solaris Fibre Channel and Storage Multipathing Administration Guide
- Solaris Express System Administration Collection: System Administration Guide: Solaris Containers-Resource Management and Solaris Zones

### **1.4.4 Release Cycle**

We did a beta release of the driver/CLI on Opensolaris in January 2008. We plan put back Q4 FY08.

### **1.4.5 Technical Support**

Standard Solaris technical support

### **1.4.6 Training**

No special training requirements.

## **1.5 Related Projects**

NPIV is an addition to the Solaris FibreChannel stack, commonly known as Leadville. It modifies the fcinfo CLI.

### **1.5.1 Dependencies on Other Sun Projects**

Existing completed projects:

PSARC 1997/385 Fibre Channel Driver Re-architecture (Leadville)

PSARC 2004/291 Fibre Channel HBA Port Utility

### **1.5.2 Dependencies on Non-Sun Projects**

None

### **1.5.3 Sun Projects Depending on this Project**

None

### **1.5.4 Projects Rendered Obsolete by this Project**

None

### **1.5.5 Related Active Projects [Describe the relationship.]**

None

### **1.5.6 Suggested Projects to Enhance this Program**

Integrate SCSI passthrough for Xen. SCSI pass through allows SCSI devices to be presented as SCSI devices to a DOMU (not block devices, as this project supports). This is especially valuable for non disk SCSI devices such as tape drives.

Fujitsu is working on this support for Linux/Xen, and it will likely be included in Xen 3.2 release. NPIV is complimentary because one NPIV port's devices can be mapped to a DOMU.

## **1.6 Competitive Analysis**

The other major virtualization environments (Xen/Linux, Microsoft Virtualization and VMWare) are all implementing NPIV. At this point, limited support is available from the HBA vendors with special drivers, although we anticipate that support for VMWare, and likely Microsoft virtualization will be included in the operating system. We are working with

members of the Xen community on the operating system independent portion of NPIV support.

## 2 Technical Description

### 2.1 Architecture

The NPIV project encompasses many different layers. This is summary; details are included in section 4 and beyond.

- At the lowest level level is the Leadville device driver. The change to the Leadville device driver is relatively simple. The current architecture of the Leadville stack (PSARC 1997/385) encompasses several device drivers. At the lowest level is the FCA (or FibreChannel Adapter) which actually communicates with the FibreChannel Adapter hardware/firmware. It is able to send/receive encapsulated FibreChannel commands. The FCA driver communicates with the fp and fctl drivers. The fp driver currently has one instance per physical and is responsible for per port handling (for example reconfiguration). There is one fctl instance per system and it is responsible for maintaining knowledge of all ports and upper layer protocols.

The architectural change for adding NPIV is to change the current one to one relationship between the FCA instance and fp instance to a one to many relationship.

- Currently Leadville is managed using the HBA API for status reporting as well as Solaris specific project private interface. There has been work in T11, the standards body responsible for HBA-API, to include support for NPIV. However, according to the editor, it will not be completed due to lack of interest from the standards community
- Currently we have a CLI called fcinfo which is used to report status of HBAs and devices attached to the HBA. We will extend this tool to support NPIV. We will create a hardlink to the tool called fcadm, used for administering NPIV. This change will not break compatibility with fcinfo.
- For each Virtual machine environment (xVM) we will do the necessary work to bind a NPIV port to a guest operating instance. In the case of xVM, we will migrate a DOMU's WWN when the DOMU migrates.
- We will not support Logical Domains or Containers in the initial release but will support them in the future.

### 2.2 Interfaces

#### 2.2.1 Exported Interfaces

<b>Interface Name</b>	<b>Proposed Stability Classification</b>	<b>Specified in What Document?</b>	<b>Former Stability Classification or Other Comments</b>
FCA	Project private	NPIV_FCA_Interface_Doc-0.2.pdf	Base document is specified FibreChannel Case
Proprietary HBA-API extensions for NPIV	Project Private	HBA-API_ext.man	
fcadm command line	Committed	fcinfo.man	
fcadm output	Uncommitted	fcinfo.man	
fcinfo output	Uncommitted	fcinfo.man	
fcadm SCF	Project Private	scf_schema	Storage for NPIV WWN
xenstore	Volatile	Xenstore.man	Changes for existing Xen database
xm input/output	Volatile	xm.man	Xen CLI
/usr/lib/xennpivd file	Project Private		Xen reconfiguration daemon
xnpivd file	Project Private		Xen reconfiguration driver

### 2.2.2 Imported Interfaces

<b>Interface Name</b>	<b>Proposed Stability Classification</b>	<b>Specified in What Document?</b>	<b>Former Stability Classification or Other Comments</b>
T11 NPIV Specs	Standard	T11 web site	See Appendix A for exact titles

## 2.3 User Interface

See section 4 and beyond for details of user interface.

There will be several extensions to existing CLI user interfaces, and one new user interface:

- fcadm will allow creating and deleting NPIV ports. fcadm is intended to be used in non virtualized environments. Fcadm shares functionality with fcinfo and will likely be implemented as a hardlink to fcinfo.
- We will extend the fcinfo command to support NPIV.
- We will extend the Xen xm command to support NPIV. This will support creating/deleting and reporting status NPIV virtual ports associated with a Xen DOMU.

## 2.4 Compatibility and Interoperability

### 2.4.1 Standards Conformance

We are implementing N\_Port\_ID Virtualization. NPIV is a T11 standard specified in T11 FC-LS/FC-DA/FC-GS-4. See Appendix A for full titles.

### 2.4.2 Operating System and Platform Compatibility

Solaris Nevada on both X86 and SPARC platforms will be supported. Both Emulex and Qlogic HBAs will be supported. Both Emulex and Qlogic support NPIV in their 4Gb products; previous products (1Gb/S and 2Gb/S) will not support NPIV.

SCSI 3 PGR (persistent group reservations). NPIV will impact PGRs – a reservation will be based on NPIV WWN. However, this should be beneficial for users; today a guest operating system using SCSI 3 PGR will share the device with any other guest operating system sharing that HBA; with NPIV using a dedicated WWN, the device will only be dedicated to that affiliated guest operating system.

### • 2.4.3 Interoperability with Sun Projects/Products

On the SAN, NPIV will interoperate with 4Gb/S FibreChannel switches, and all target devices.

- NPIV will support xVM, Logical Domains and Containers (as well as being usable without virtualization). The initial release will only add explicit support for xVM, but NPIV ports can be used with Logical Domains and containers, although simply as

regular FibreChannel ports (there will be no explicit binding between an NPIV port and a Container or Logical Domain).

#### **2.4.4 Interoperability with External Products**

NPIV will interoperate with any Fibrechannel switch which supports NPIV – generally vendors' 4Gb/S products support NPIV. If a user attempts to use an NPIV capable HBA with a switch that is not capable, the HBA will function with one port, and no virtual ports. Legacy HBAs (2 Gb/S and 1Gb/S) will continue to operate in non NPIV mode.

- All FibreChannel targets will transparently support NPIV HBAs (NPIV functionality will be invisible to targets).

#### **2.4.5 Coexistence with Similar Functionality**

The emlxs driver for Emulex HBAs supports NPIV today from Solaris 8 to Solaris Nevada via .conf entries. When our project delivers, the support in emlxs driver for NPIV support via .conf entries will be removed, and we will transition to this implementation, migrating the .conf entries transparently.

- **2.4.6 Support for Multiple Concurrent Instances**

Only one instance of FibreChannel device drivers may be present on the machine. This includes Logical Domains, Containers and xVM.

#### **2.4.7 Compatibility with Earlier and Future Releases**

This is backwards compatible with all existing interfaces.

## **2.5 Performance and Scalability**

### **2.5.1 Performance Goals**

We expect runtime CPU requirements for adding one NPIV port to be similar to adding an additional HBA. This CPU load will be principally felt at reconfiguration time, when events occur on the SAN, and Leadville must handle reconfiguration.

### **2.5.2 Performance Measurement**

We expect no degradation compared to existing HBA today.

### **2.5.3 Scalability Limits and Potential Bottlenecks**

HBA Vendors have hardware limitations due to their HBA's physical resource limitations (on board memory). This limits the number of active NPIV ports bound to a physical HBA. This number is fairly large (~100), and varies with each model of HBA.

### **2.5.4 Static System Behavior**

SCF configuration file data will be small, approximately 100 bytes per NPIV WWN.

### **2.5.5 Dynamic System Behavior**

System behavior will be similar to non NPIV case, with the exception of reconfiguration noted above. During reconfiguration, every virtual HBA port will perform reconfiguration. This is by design.

## **2.6 Failure and Recovery**

### **2.6.1 Resource Exhaustion**

If the HBA runs out of hardware resources to support NPIV, this will be reported when the user attempts to create the port.

### **2.6.2 Software Failures**

We have no planned software failures.

### **2.6.3 Network Failures**

SAN Failures are managed as today. Cable failures between the adapter and switch are handled by disabling the link. Failures between switch and device create RSCN (State Change Notification) messages to the NPIV port. The NPIV port can then act on the message by determining the nature of change.

### **2.6.4 Data Integrity**

n.a.

### **2.6.5 State and Checkpointing**

n.a.

### **2.6.6 Fault Detection**

n.a.

### **2.6.7 Fault Recovery (or Cleanup after Failure)**

n.a.

### **2.6.8 Hardware Compatibility**

NPIV must be supported in the HBA as well as the switch. It is only supported in Fabric configuration, not loop or point to point topology. The following describes these incompatibilities, and how they can be reported at user level command/libraries:

1. HBA does not support NPIV: error can be detected from HBA capabilities.

Alternatively, creating an NPIV port will return an error.

2. Topology (i.e. Loop or point to point) does not support NPIV: NPIV port create will return error. This may also be detected by querying the topology.
3. Switch does not support NPIV: error is detected when creating a port

## **2.7 Security**

NPIV is an important part of controlling access to devices on the SAN. Storage Administrators use it to segregate devices on the SAN. However, it's not intended as a secure protocol. FC-SP (FibreChannel Security Protocol, a part of the T11 FibreChanel standard) is intended as a secure protocol. FC-SP includes authentication of nodes and encryption. FC-SP is not widely implemented in the industry and is not implemented in the Solaris FibreChannel stack.

With NPIV, it is possible to spoofing existing devices on the SAN. However, this capability is already present in non NPIV device drivers for Linux and possibly other operating systems. Other NPIV implementations will also allow spoofing NPIV addresses.

In short, although NPIV can allow the user to do spoofing, this capability is already a problem on the SAN. Although there is an industry standard addressing this problem, there has been insufficient demand to implement the solution.

## **2.8 Software Engineering and Usability**

### **2.8.1 Namespace Management**

We introduce no new packages or names.

### **2.8.2 Dependencies on non-Standard System Interfaces**

No non standard/stable interfaces are being used.

### **• 2.8.3 Year 2000 Compliance**

NPIV does not use dates or time, so is Y2K compliant.

### **2.8.4 Internationalization (I18N)**

No change in existing levels of I1ILN. fcadm will follow same conventions as fcinfo.

### **2.8.5 64-bit Issues**

Existing FibreChannel device drivers are 64 bit clean and will remain this way.

### **2.8.6 Porting to other Platforms**

Some portions of Xen code are by their nature platform independent, and needs to be portable to other operating systems (e.g. Linux). Most of this project however is specific to Solaris and would require significant effort to port.

#### **3.1.1 Package Overview**

The fcadm command will be added to SUNWfcprt package. This package currently contains fcinfo.

New Xen daemon, xennpivd will be created in new package, name TBD.  
New Xen driver (xfcb), will be in existing package for Xen drivers.

#### **2.8.7 Accessibility**

No new interfaces, just additions to existing interfaces. All interfaces are CLIs.

## **3 Release Information**

Note: Some of the packaging and installation details may not be available at design time. Describe expected solutions, and augment the description as the details are decided.

### **3.1 Product Packaging**

Bundled with Solaris.

#### **3.1.2 (Default) Installation Locations**

fcadm will be installed in /usr/sbin/  
Xen NPIV deamon will be installed in /usr/svc/method/  
Xen NPIV device driver (xfcb) will be installed in /platform/i86xpv/kernel/  
directory structure.

#### **3.1.3 Effect on External Environment**

None

## **3.2 Installation**

### **3.2.1 Installation procedure**

Installed during standard Solaris installation. Unchanged from current.

### **3.2.2 Effects on System Files**

None

### **3.2.3 Boot-Time Requirements**

fcadm will be called at boot time (from SMF) to enable NPIV ports. Enabling NPIV ports may slow boot time, again similarly to adding additional physical HBAs.

### **3.2.4 Licensing**

Standard Solaris/OpenSolaris.

### **3.2.5 Upgrade**

FCA interface is versioned; because Leadville components are bundled and this error is not expected to occur, error recovery will be to fail loading Leadville. Incompatibility results in failure to load drivers leaving FC devices unusable.

HBA API interface is versioned, so that necessary level will be checked for proprietary extensions. Incompatibility results in loss of functionality.

### **3.2.6 Software Removal**

Normal package tools

## **3.3 System Administration**

System administration is based on several different tools. Typically the user will select one tool to use for administration depending on his or her environment. For non virtualized environments, the user will use the fcdm command to create NPIV ports. For xVM, the xm command is used. For zones, the zonecfg command will be used.

- The administrator will use fcinfo to report status, regardless of the virtualization environment.
- For this initial release of software, there are two methods of administering the WWN name space:
  - users may manually administer name space. Documentation will guide the user in creating a correctly formatted WWN.
  - users may allow the library to select a WWN. The WWN algorithm is similar to that used by Xen for MAC addresses, by using an OUI and a random number. Due to the format of the WWN, we have 40 bits of random values, instead of 24 random bits used

for Xen MAC addresses. Statistically, if we have a relatively large SAN, assume 100 WWN per HBA port, and 1000 HBA ports on a SAN, there is a 1% chance of collision.

The penalty for collision is different in different cases

- If there is an existing device logged in with that WWN, the new device will timeout at login, and the new device will not be allowed on the SAN. The user will receive an error message.
- If there is an existing device with resources on the SAN, that is not logged in, but there are zones or LUN masking allocated to that device, that device will be accessible.
- Future HBAs will have the ability to create WWNs based upon the HBA's permanent WWN, and importantly store in non volatile memory which generated WWN are in use. It is important to be able to store in use WWN so that WWNs are not reused.
- There may also be solutions based upon a centralized management station, such as xVM OPS Center.

## **4 FibreChannel Driver NPIV Architecture**

### **• 4.1 Description**

The major architectural change is to change the relationship between the FCA (Fibre Channel Adapter, one instance per FC port) and the fp (supports a generic fibrechannel port) driver. Currently the relationship between the FCA and fp driver instances is 1:1, this project changes the relationship to 1:many.

Additionally, the interface between the FCA and fp driver changed to support NPIV. The FCA Driver must return information about support for NPIV and also return specific errors related to NPIV. The fp driver can also create and delete NPIV ports through this interfaces

The public user interface to NPIV is based on extensions to the HBA-API covered in the next section. The HBA API communicates to the driver through new ioctl calls. These are project private interfaces to the driver.

### **4.2 Interfaces**

#### **4.2.1 User-visible**

No public interfaces.

#### **4.2.2 Internal (optional for ARC review)**

Updated FCA interfaces are included in file NPIV\_FCA\_Interface\_Doc-0.2.pdf

New ioctl interfaces to support HBA API are project private.

### **4.3 Operation**

The high level operation of the FCA interface remains unchanged. The port's permanent WWN (the one that the adapter currently uses, found on the HBA's NV memory), will continue to be used automatically. The permanent WWN will never be disabled.

NPIV ports are created and deleted by private ioctl calls into the fp driver. This causes the fp driver to call the fca driver's `fca_bind_port` function to enable the port in the FCA driver. In response, the FCA driver then creates a new fp instance.

Neither the Emulex and Qlogic HBAs not have enough on board hardware resources to support fcip (IP over FC) on NPIV ports. We will not attach the fcip driver for NPIV ports. Current non NPIV ports will continue to attach fcip, so there is no problem with upgrade.

## **• 5 HBA-API Architecture**

### **5.1 Description**

HBA API library is extended to support NPIV. The current implementation of HBA-API is specified in PSARC 2002/644 FibreChannel HBA API V2. HBA-API is an industry standard interface.

### **5.2 Interfaces**

#### **5.2.1 User-visible**

See section 2.2

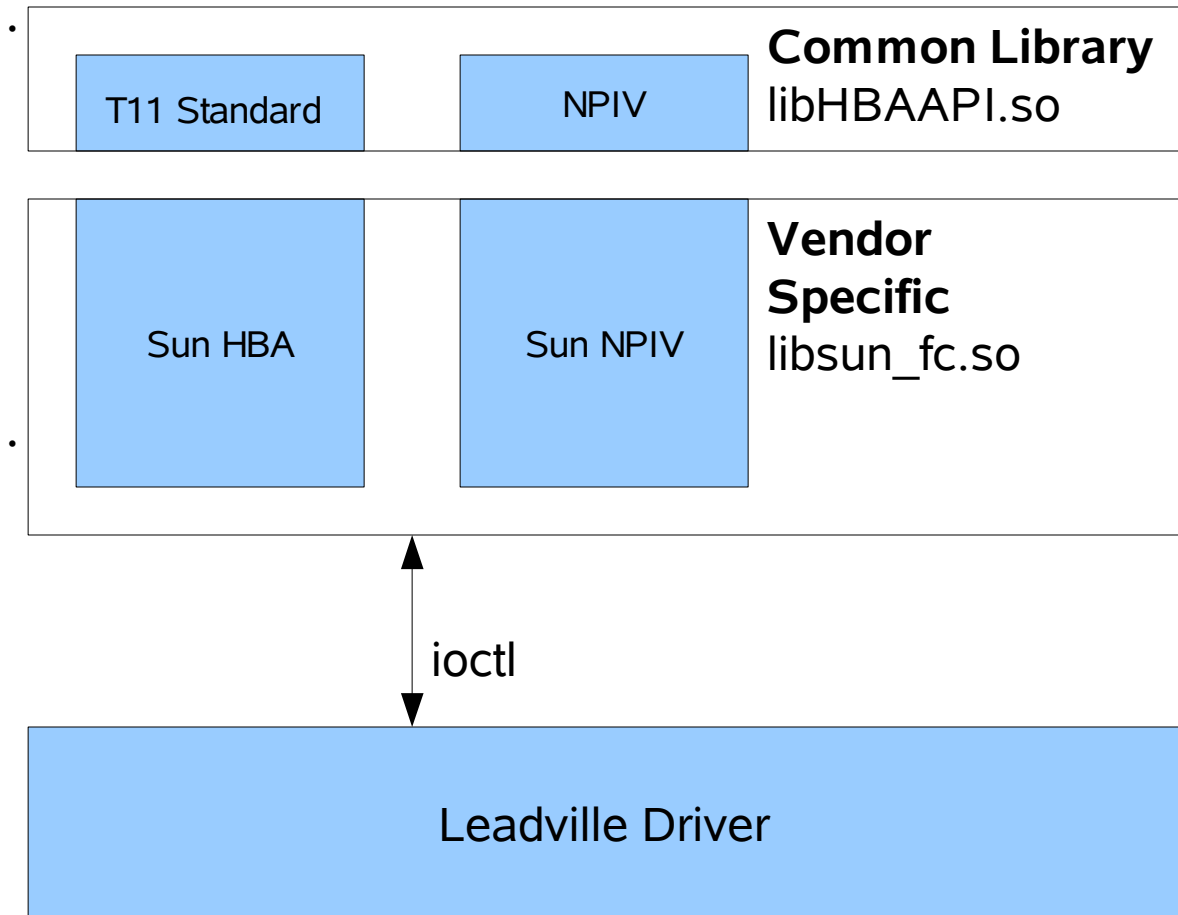
#### **5.2.2 Internal (optional for ARC review)**

HBA\_API\_ext.man

### **5.3 Operation**

HBA-API interface is a SNIA/T11 industry standard specification that allows accessing information about FibreChannel HBAs from user level libraries. Although there has been work in T11 to implement NPIV in SM-HBA-2 (a follow on specification from HBA-API), the editor has indicated that this effort is stalled due to lack of interest from the community.

In our implementation, we followed the HBA-API model of having a vendor independent layer, and vendor dependent layer. We added functionality to existing to the existing libraries: libHBAAPI and libsun\_fc.so respectively.



## 6 fcadm and fcinfo CLI

### 6.1 Description

fcadm is a new CLI command used to administer NPIV in a standalone environment. fcadm is based upon fcinfo and will contain a superset of fcinfo's functionality. fcadm will retain existing status commands, and fcadm will allow modifying FibreChannel's state, including adding and deleting NPIV ports.

Additionally, fcinfo will be extended to accommodate NPIV. fcinfo will be able to list relationship between NPIV ports and physical adapter. It will also list error states, for example when a switch is not able to support HBA API.

### 6.2 Interfaces

#### 6.2.1 User-visible

See proposed fcadm man page, and additions for fcinfo.

For SCF data format, see attached document in case materials.

### **6.2.2 Internal (optional for ARC review)**

fcadm will use the HBA-API extensions described in section 5.

Data will be stored using the SCF scheme shown in interface table.

## **6.3 Operation**

Users will create and delete NPIV ports using fcdm. fcinfo will be used to return status of NPIV ports.

At boot time, fcdm will read SCF data and enable NPIV ports.

fcadm will store configuration data in SCF.

# **7 Xen Architecture**

Xen modifications consist of several parts:

- Bind an NPIV port to a DOMU
- When a device comes online that is associated with a DOMU's NPIV port, automatically map that device to the DOMU
- When a DOMU migrates between two physical servers, first disable the NPIV port on the original server, and enable the NPIV port on the destination server at appropriate points in the migration process.

These topics are covered in more details in the following sections.

## **7.1 Description**

We modify the Xen's standard CLI, xm, to accept NPIV WWN and associate with a DOMU and store to Xenstore. This is the sole part of the change that is platform independent, and is suitable to contribute to the Xen community.

Management will be handled by a new daemon, xennpivd. It will be started at DOMU boot time from smf(5), but will execute in DOM0. The main function is to wait for changes on the SAN using the HBA API interface. It also is responsible for creating and deleting NPIV ports. The HBA API interface provides for callbacks when the SAN changes. If the daemon determines that a FibreChannel target has been added or deleted, it determines if it is from an NPIV port that is mapped to a DOMU. If it is from an NPIV port, it will perform the appropriate Xen commands to add or delete the device to the SAN. More details are in section 7.3

We introduce a small device driver, xfc. Due to Xen architecture, we need a device driver to register for Xenstore change callbacks. Its sole function is to callback to xennpivd when Xenstore is updated.

Finally, we will support migration of DOMUs across physical servers, by disabling an NPIV port on the origin server, and enabling the NPIV port on the destination server.

## 7.2 Interfaces

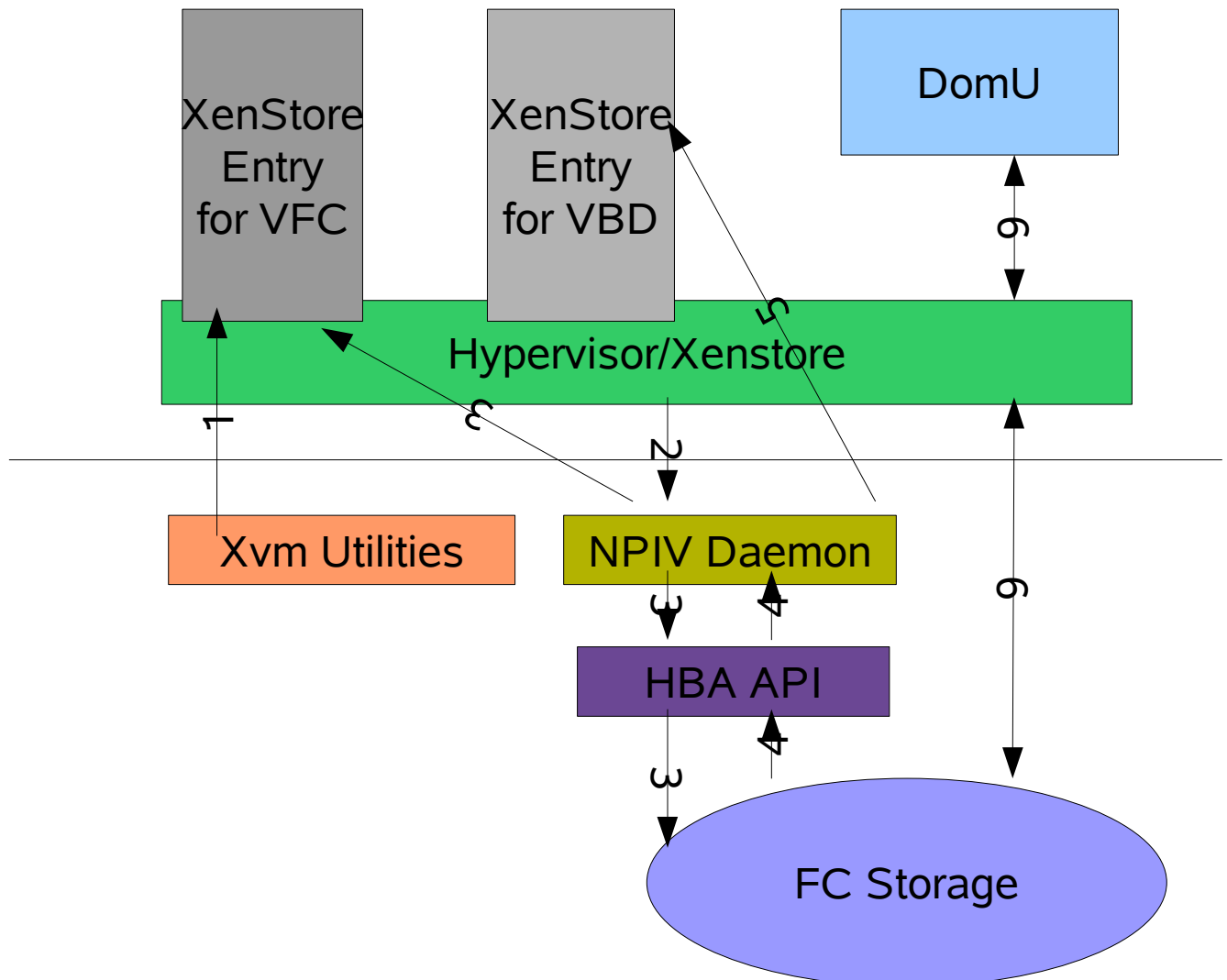
### 7.2.1 User-visible

xm will be modified. See section 2.1  
/usr/lib/xennpivd daemon file location.

### 7.2.2 Internal (optional for ARC review)

## 7.3 Operation

The following diagram shows the several modules involved in Xen. Numbers refer to the creation steps, below.



## Creating an NPIV Port

1. xm(1M) lets the xend(1M) create new entries in Xenstore under /local/domain/0/backend/vfc/\$DOMUID/.
2. xfc, the backend driver of virtual FC port, is triggered by the new entry, then it calls the NPIV daemon (npivxend) through a door during the attach routine.
3. NPIV daemon calls HBAAPI to create the virtual port. If the creation of virtual FC port succeeds, xfc continues the attach routine.
4. NPIV daemon also registers devices event via HBAAPI.
5. Devices discovered from the new FC port trigger events which are captured by the NPIV daemon. NPIV daemon calls xvm utilities to attach new devices to DomU.
6. The DomU is now able to access the storage via the virtual port.

## Deleting an NPIV Port

1. Xm marks the hotplug status of a virtual FC port as error.
2. The detach routine of xfc is triggered by the new hotplug status.
3. Xfc calls the NPIV daemon through a door to delete the virtual FC port.
4. The daemon calls HBAAPI to remove the virtual FC port from Domain-0.
5. Xfc continues its detach routine.

## Migration

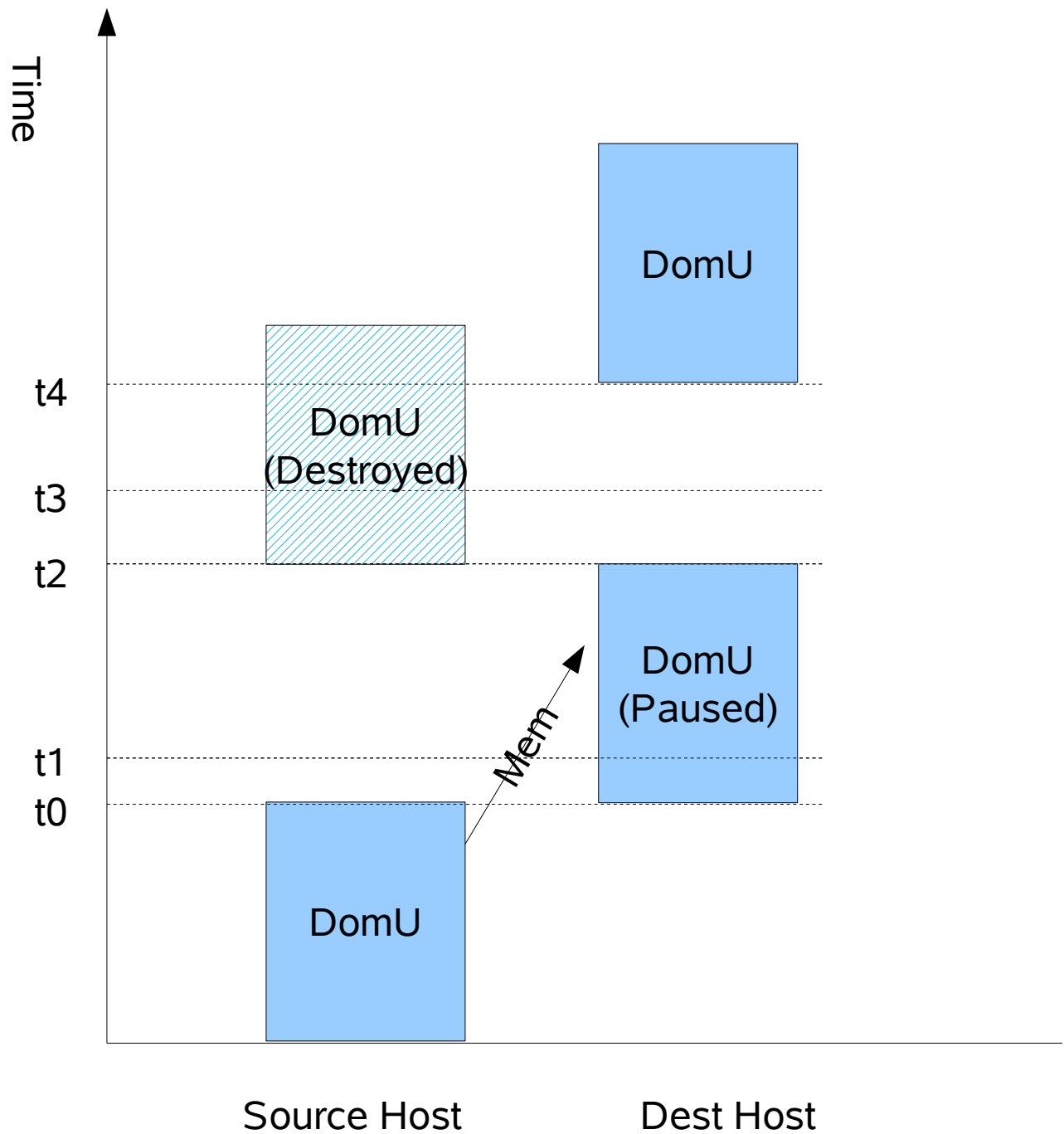
The live migration of a DomU exposes a window where a backend device is connected both on source and destination hosts. The window occurs during creating devices on the destination and the pausing of DomU on the source. During the window all memory pages are transferred from the source to the destination and the DomU keeps running on the source.

Migrating a virtual FC port must be different from normal devices in that a given WWN may be active

on the Fabric once. Therefore block devices associated with the virtual FC port is also different with other block devices attached to a DomU.

A 'delayed binding' mechanism is introduced to overcome this issue. The creation of virtual FC port and associated block devices is delayed until the DomU is going to be turned on on the destination host. Possible conflicts of virtual WWN is avoided by the delayed binding as the WWN should be already unregistered from the Fabric on the source host. A step-by-step description of the live migration of FC port is as below. Steps that are specific to NPIV are marked with a \*.

1. The description of a DomU transferred to the destination server from the source.
2. \*Save NPIV FC ports and associated block devices into a temporary list.
3. The destination host creates the DomU (paused) and recreates normal devices.
4. The destination host receives all memory pages of that DomU.
5. \*The destination host restores NPIV FC ports and associated block devices from the temporary list. The procedure is the same as operated in single physical server.
6. The destination host waits all backend devices to be connected.
7. The destination host starts the DomU



t0 : Check point is created on source host. Destination host receives the info of domain which is being migrated and creates a paused domain according to the info.

t0 - t1 : Restoration of normal devices (except virtual FC port and associated block devices).

t1 - t2 : Transferring memory pages.

t2: Memory transfer is completed, DomU is destroyed on source host along with the virtual port.

t2 - t3: The destination host creates virtual ports and associated block devices.

t3 - t4: Waiting for all devices to be connected, including devices from t0-t1 and t2-t3.

t4: DomU unpaused, is starting to work.

---

## **Appendix A: Standards Supported**

---

### **References**

T11 FC -LS FibreChannel Link Services, T11/05-345v1

- T11 FC-DA, FibreChannel Device Attach T11/04-202vA
- T11 FC-GS-4 FibreChannel Generic Services (T11/04-031v2)

### **R.1 Related Projects**

1997/385 Fibre Channel Driver Re-architecture (Leadville)

2006/260 Solaris on Xen

2002/174 Virtualization and Namespace Isolation in Solaris (Zones)

2002/644 FibreChannel HBA API V2

### **R.2 Background Information for this Project or its Product**

See OpenSolaris project page:

<http://opensolaris.org/os/project/npiv/>

### **R.3 Interface Specifications**

None

## **R.4 Project Details**

None