



Virtual Storage Ports

James Smart
Emulex Corporation

Virtual Ports - FC

■ Virtual Port :

A unique FC wwpn, which is associated with a particular fabric, and has a single N_Port_ID (aka Address or S_ID) assigned to it. Each port will have its own view of attached storage.

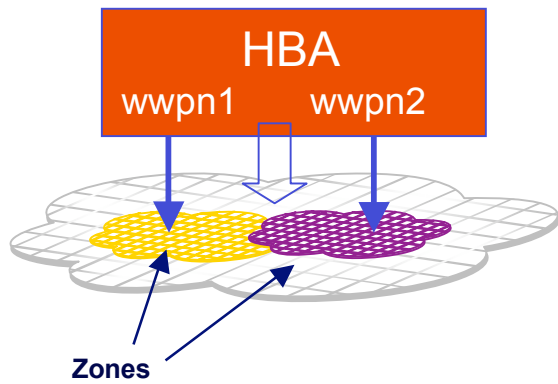


■ Standard Physical Port:

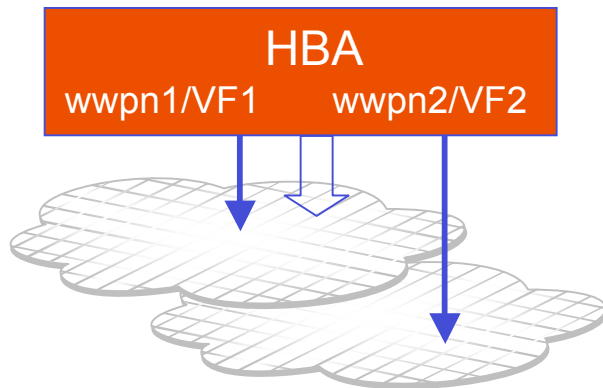
- Single WWPN per physical link
- Assigned a Single Address

■ N_Port ID Virtualization (NPIV):

- Only if Pt2Pt w/ F_Port (no loop support)
- WWPN for physical link, FLOGI gets Address1
- WWPN for each NPIV, FDISC gets Address2.

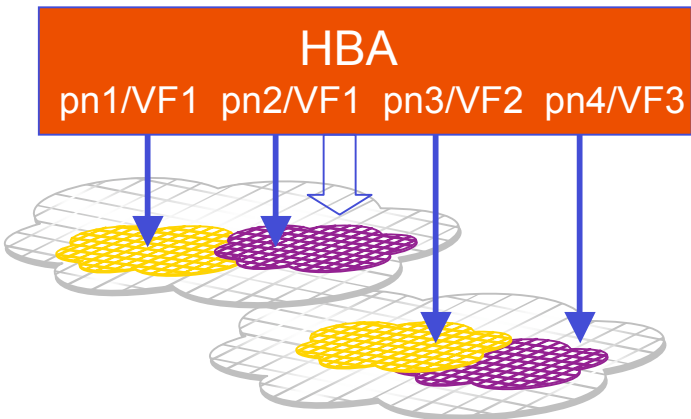


Virtual Ports: FC, cont.



■ Virtual Fabrics (VF) – aka VSAN:

- Initial FLOGI to request Support
- Exchange VF_ID's available
- Single WWPN per VF; FLOGI per VF
- All traffic has an 8byte header which identifies the VF directed to.



■ VF + NPIV:

- FLOGI for 1st instance on an VF
- FDISC for 2nd..N instance on each VF

Why Virtual Ports and XEN ?

- Storage doesn't have performance issues...
- Block Abstraction works fine and is migrate-able...

... It's about Data Center Management and SAN Monitoring

- Users want visibility into the VM-specific data flows
 - Traffic Analysis; Problem Analysis; Charge-Back
 - Opportunities for QoS in the SAN
- SAN Provisioning, Work Flows, and Expertise preserved
 - Controlled storage visibility : Zoning & LUN Masking to the Dom
 - Visibility moves with the Dom, not the server
- HBA Upgrades and Replacements Seamless
- There are some that want direct FC access in the Dom

VPort Abstraction for DomU's

- **If bound to DomU as a resource**
 - I/O path can reflect the Dom
 - SAN visibility and reporting tracks the Dom
 - SAN reconfig only needed if there's a Dom change
- **In Dom0 :**
 - Takes advantage of all the Blktap, SCSItap work, etc
 - Single “services” in Dom0 (vs replication in each DomU)
 - Multipathing, LVM
 - Single Toolsets – based on Dom0 OS
 - Automatic DomU Resource Assignment
 - Negatives: device fencing
- **In DomU :**
 - FC to Dom0 – native SAN view and native OS storage stacks
 - Fencing of devices unnecessary
 - Native tools in the DomU OS
 - Negatives: Migration, IOMMUs, DomU drivers, etc

Emulex Status

- **Implemented NPIV in our Linux device driver**
 - Available on SourceForge since May 3, 2006
 - Refreshed for 2.6.18 and will push upstream
- **Each vport shows up as a new SCSI Host**
- **Simple utilities to create, delete, query**
 - Primitives usable by mgmt tools
- **Working with Standards Bodies**
 - Consistency in DMTF, SMI-S, SMA-HBA
 - Addressing Grey Areas:
 - Example: Virtual Port Migration if HBA “babbling” on system lockup
- **Laying framework for APIs**
 - NPIV w/in Linux

Next Steps

- **Integrate a NPIV API into Linux**
- **Integrate support into Xen Domain Creation and Control**
 - Allocation of WWPN's
 - Specifications of VF_ID's, Roles, Resource Limits, etc
 - Tools to create and manage the Virtual Port

- **We are trying to take a wider view:**
 - **NPIV VPorts are a prelude to other virtual devices**
 - PCI-IOV, Virtual Functions, vNIC, etc
 - Common Mgmt Point - interface consistency
 - **NPIV raises other issues that need to be addressed system-wide**
 - Recognizing what "could" be there
 - Constructing the device and verifying resulting dependencies
 - Creation Policies for HA
 - Introducing resource constraints, QoS policies
 - Fencing of devices for Dom's
 - Multipathing, LVM, Clustered Dom Access, etc