

N_Port ID Virtualization for Solaris/Xen

We're working on an implementation of FibreChannel NPIV (N port ID Virtualization) for Solaris. NPIV is an addition to the FibreChannel standard, which allows one FibreChannel port to represent multiple virtual ports on the Storage Area Network (SAN).

Traditionally, FibreChannel targets and SAN switches use a Fibrechannel WWN to identify to segregate devices for access control. The WWN is stored in the adapter's flash memory and assigned at manufacturing. This access control is called LUN masking (at the target) and zoning (at the switch).

Before NPIV, with virtualized environments such as Xen, the adapter's WWN represented all DOMU (guest operating systems) on the server. This meant there was no per DOMU access control. Additionally, if a DOMU migrates to a new server, the WWN does not migrate. These limitations meant that zoning and LUN masking were ineffective in Xen clusters of any size.

However, with our implementation, each DOMU has its own unique WWN. This WWN migrates with the DOMU.

Our implementation solves some architectural problems in unique ways. The first is where the logic to bind a WWN to a DOMU belongs. One naive implementation would be to divide the implementation between the DOM0 and DOMU. However, we chose to implement most of the work in the DOM0, which is a simpler yet still effective implementation.

Another difficult problem is assigning world wide names. In FC, the WWN is a 64 bit value in an industry standard format. There are many simple choices which have problems. For example, one could use a machine's unique identifier combined with an incrementing number. Or one could use a WWN obtained from a pool maintained on the adapter. Both of these have problems though: the first in retaining the incrementing number across reinstalling the OS, the second in returning a WWN to the adapter's pool when the DOMU has migrated away from the server containing the HBA. We borrow an idea from the Xen networking stack and use a random WWN with WWN collision retry.

This work is being done at Sun as an OpenSolaris project.

Aaron Dailey
aaron.dailey@sun.com
Sun China Engineering and Research Institute
Beijing, China, PRC