

HPSS SAN3P

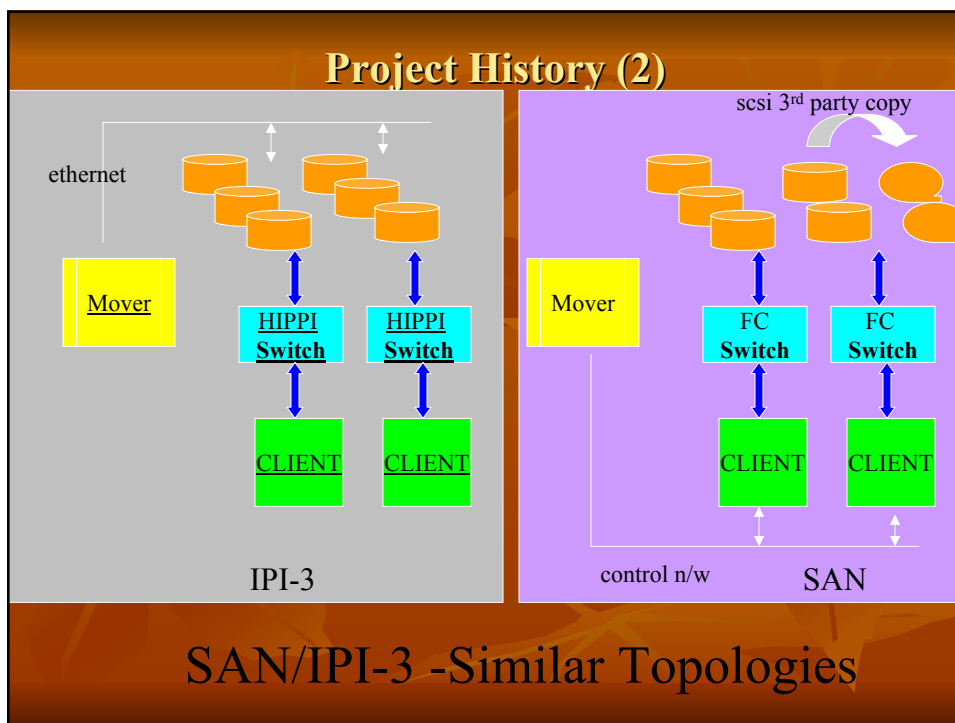
Michael Gleicher
Gleicher Enterprises, LLC

Topics

- Project History/Motivation
- Requirements
- Target Hardware Architecture
- Prototype
- Software Architecture
- IPI-3 vs SAN3P
- Security Issues
- SAN3P Limitations
- Current Status
- Futures

Project History

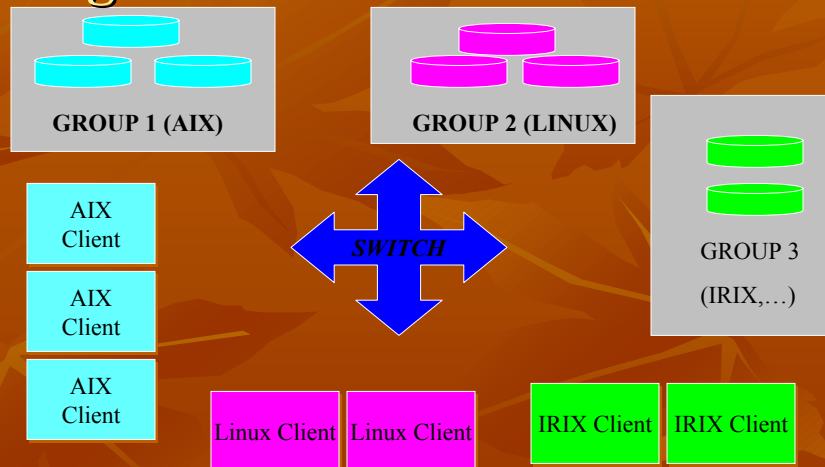
- NSL – IPI-3
 - Separate Control and Data Paths
 - Security
 - Control ethernet
 - Hippi data xfer controlled by IPI device
 - No direct client access to devices



Requirements

- 3rd party transfers to SAN-attached disks (future:tapes)
- No loss in transfer rates vs TCP/IP
- Multiple disparate groups of devices
- Mover selects transfer protocol
- Ability to easily disable for troubleshooting

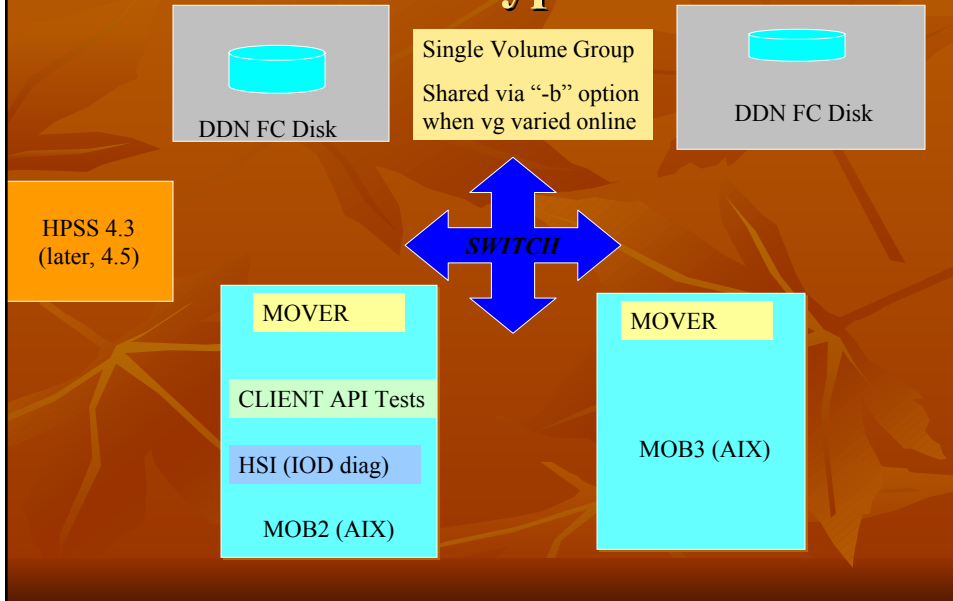
Target Hardware Architecture



•Switch Zoning used to enforce grouping if desired

•HPSS Config file (movers)

SAN3P Prototype at LLNL



Software Architecture

- Mover
 - `hpss_san3p.conf` configuration file
 - Device Groups
 - Device names
 - Hosts / networks allowed to access
 - Hosts / networks restricted from access
 - Choose SAN3P xfer type based on client host IP, client "san3p-capable" flags in IOD
 - Implement "conditional control" flag – choose pdata or mover protocol based on device/host/client flags
 - Add mover-to-mover logic to pass/use SAN Group ID

Software Architecture (2)

- New IOD flags
 - SAN3P transfer type
 - Conditional-Control flag
- different Pdata/Mover Protocol message header delimiters
- new functions to read mover message and determine whether pdata/mover protocol

Software Architecture (4)

- SSM changes:
 - add “san3p-capable” device config flags
 - add “SAN Group ID” field to device config
 - add mover managed object flag to enable/disable SAN3P transfers

Software Architecture (3)

- HPSS Client API library
- upware compatible (API_TRANSFER_TYPE env. variable)
- new transfer type: “mover selects protocol”
 - sets “conditional control” flag in IOD for hpss_Read, hpss_Write
- SAN3P client library
 - Implement direct device I/O (read, write, read/modify/write)

Software Architecture (2)

- New IOD and srcsink flags
- SAN3P address structure
 - Label checking requirement

Software Architecture (3)

- HSI
 - simple change to add SAN3P transfer type to I/O modules
 - “san3p_enabled” global hsirc option to
- PFTP
 - server side issues IOD – must know if client is SAN3P-capable (“feature” exchange)
 - removed “pipi3”, “pdata” commands (user should not have to choose protocol)
 - added support for san3p,tcp/ip,hooks for shared memory xfers in pdata.c

IPI-3 vs SAN3P

- IPI-3 Master (AIX-> kernel process) that performed HIPPI I/O for client processes
- IPI Disk (Max Strat) controlled xfer, using Transfer ID, reading/writing from user buffer
- IPI-3 – private ethernet for mover-device communication
- SAN3P – mover passes device parameters to client
- SAN3P - Client must have read/write access to device

Security Issues

- If host is compromised, entire SAN zone is jeopardized
- For mover-to-mover, sufficient to use SAN zoning
- Security issues for host-resident client apps (HSI,PFTP,...).
 - Set GID,Set UID for trusted Apps
- SAN Security Summit (~2 years ago) – no followon?
- Ongoing efforts (smart disk drives,...)

SAN3P Limitations

- Homogeneous SAN groups (AIX LVM, Linux,...)
- Linux -> Page boundary restrictions for I/O buffers. SAN3P lib will have to be aware of limitations for specific architectures

Current Status

- HPSS R6.1 (Linux only)
 - HSI- basic file xfers tested
 - PFTP Client – in progress
 - Client API – to be tested

Future (?)

- Heterogeneous shared devices (?)
- SAN3P direct-to-tape client access (?)
- SAN Security standard(s) (?)
- Intelligent Disks
 - IPI-3 diagram here