OPENFABRICS ALLIANCE

Linux SRP Initiator Support

Bart Van Assche, Fusion-io

Overview

- About my involvement with SRP
- Linux SRP initiator
- Layers above the SRP initiator
- Current status of the Linux SRP initiator
- Possible future directions

About my involvement with SRP

- Started contributing to Linux SRP initiator and target several years ago
- Joined Fusion-io ION team in April 2012
- ION: SAN software optimized for Fusion-io SSD
- ION = SCST + CLI + HTML GUI + H.A. + Q.A.
- Supports FC, iSCSI and SRP/IB
- InfiniBand offers lower latency and higher bandwidth compared to FC or Ethernet
- Several operating systems provide an SRP initiator
- We are happy with the SRP target but would like to improve the SRP initiator

Linux SRP initiator components

- ib_srp
  - Kernel driver
  - Implemented as a SCSI lower-level driver (LLD)
  - Realizes SRP login
  - SCSI command processing
  - SCSI error handling
    - invokes scsi_remove_host() upon communication error
- srp_daemon
  - user space process
  - Scans fabric for SRP targets
  - Makes ib_srp log in to all detected targets

Layers above the SRP initiator

- SCSI core
  - Invokes srp_queuecommand() to send SCSI commands
- SCSI error handler
  - SCSI timeout handling
  - Invokes srp_abort() if a SCSI command times out
  - Trigger srp_reset_host() if abort fails
  - Offlines paths if error recovery fails
- multipathd + device mapper multipath
  - Monitor all paths
  - Decides which path(s) to use for communication

Current SRP Initiator Status

- Path failover takes longer than desired (two – three minutes)
- Certain failures can cause the SCSI error handler to offline SRP paths and hence cause failover to fail
- Path failure triggers scsi_remove_device()
  - This is problematic in combination with software that does not expect /dev/sdX changes
  - Example: initiator-side mirroring with md
- srp_daemon supports P_Key index 0 only

Faster path failure detection

- Option 1
  - Use the Port Error asynchronous event
  - Low delay, but detects local link failures only
- Option 2
  - Let srp_daemon register for trap type 65 (Port Out of Service)
  - Low delay, but makes ib_srp dependent on srp_daemon for link failure detection
  - Detects some but not all failures when using an unmanaged switch
- Option 3
  - Combine options 1 and 2
- Option 4
  - Periodically send a packet over the link, e.g. a zero-length RDMA write
  - Easy to implement but slightly delays path failure detection
  - Not all SRP targets support zero-length RDMA write
- Option 5 (preferred)
  - Reduce IB RC timeout from 61s to e.g. 10s

Retaining the SCSI host number

- Needed for e.g. initiator-side mirroring with md
- Do not invoke scsi_remove_device() if a communication failure occurs
- Restore communication by reconnecting to target
- Needed: trigger to reconnect
- Trigger generated by srp_daemon or by ib_srp ?
- Preference for ib_srp
- Avoids that restoring communication depends on whether or not srp_daemon is running

SRP initiator FSM proposal

- Similar to Linux FC FSM
- Timers: fast_io_fail_tmo and dev_loss_tmo
- Both timers start after a communication failure or DREQ
- Both timers are reset if communication is restored in time
- Proposed SRP FSM: running → blocked → transport layer failed → lost
- State transitions driven by communication failure / DREQ, fast_io_fail_tmo expired and dev_loss_tmo expired respectively

InfiniBand Partitioning Support

- ib_srp has P_Key support
- srp_daemon only supports P_Key index 0
- Proposal
  - Add command-line option to srp_daemon for specifying P_Key
  - Translate P_Key into P_Key index before each fabric scan
  - Let srp_daemon pass P_Key to ib_srp via login string

Thank You