Implementing High Availability Solutions with OpenFabrics

Olga Shern & Yiftah Shahar

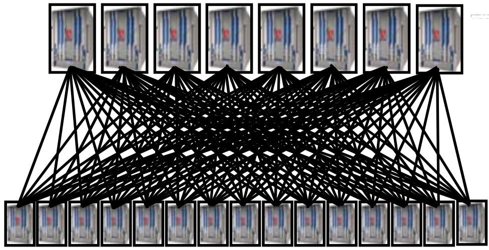
Voltaire

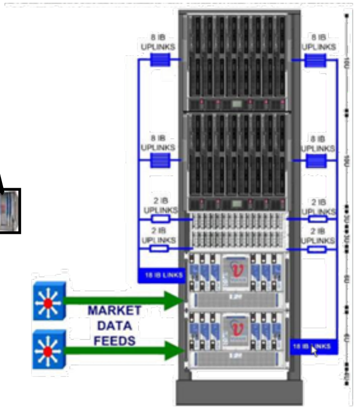
Agenda

- HA & FT - Definitions & Requirements
- System & Components
- Linux stack breakdown

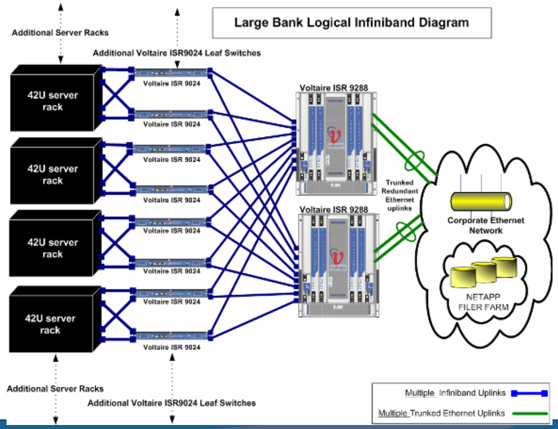
HA & FT - Definitions & Requirements

- Different people have different requirements:
  - HPC
  - Data Center
  - FSI-HPC
  - Cloud
  - File system
  - Storage

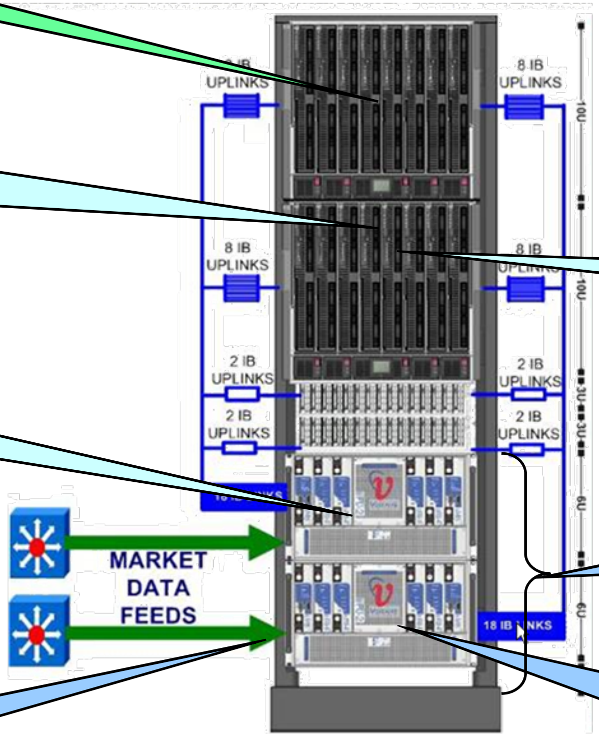




- Single point of failure (?)
- Allow service & application (traffic) continuation on different IB fabric failure events


Large Bank Logical Infiniband Diagram
Additional Server Racks
Additional Voltaire ISR9024 Leaf Switches
42U server rack
42U server rack
42U server rack
42U server rack
Voltaire ISR 9024
Voltaire ISR 9288
Voltaire ISR 9288
Trunked Redundant Ethernet uplinks
Corporate Ethernet Network
NETAPP FILER FARM
Additional Server Racks
Additional Voltaire ISR9024 Leaf Switches
Multiple Infiniband Uplinks
Multiple Trunked Ethernet Uplinks

System Solutions – FSI-HPC Sample



Application

HA & FT in the protocols (VMA/VMS)

SM (Active /Standby)

IP to IB getaway

HA in the stack (IPoIB/Bonding)

Switches connected in FT fabric

HA Switch components

Typical FSI
HA Deployment

System Components

- Hardware & Infrastructure
- IB to IP Getaway
- Subnet Manager
- Host Stack & Protocols

- Applications

Every component (developer) need to “think” it is the most important component in the system (i.e. it can’t fail) and “assume” that all the others will fail

Hardware, Infrastructure & Subnet Manager

HW & Infrastructure

Switch Chassis:

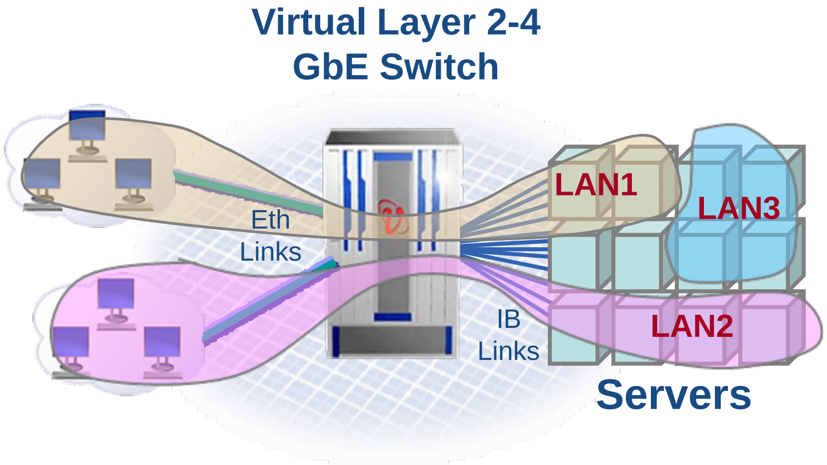
- Redundant fans (& adaptive cooling)
- Redundant power supplies & electricity inlet
- Fully synchronized management boards running in an active/passive clustering configuration
- Out-of-band management communication
- Redundant active-active backplane fabric boards
- Configuration persistency
- Hot swappable components

System:

- Cable & wiring → topology

Ethernet to InfiniBand Gateway

- Working with two or more IB to IP gateways
- Active-Active and Active-Passive mode
- Traffic load distribution (unicast and multicast)
- Gateways synchronize configuration



Subnet Manager

- Need to serve the entire fabric – many different concurrent activities
- Single point of configuration and information
- SM failover & handover (“SMInfo” protocol)
- **“Performance” – work faster :-) ← key element for host HA**

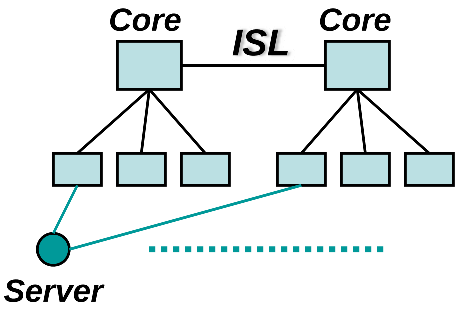
- SM routing consideration:
  - Try to keep the current port LID settings
  - Recalculate & load switch's unicast forwarding tables:
    - Good or bad ?
    - Cache routing mode
  - Can’t keep the multicast forwarding table:
    - Need to have all join/leave information
    - Recalculate and assign

- Host perspective:
  - Path query (distributed SA ?)
  - Multicast join/leave

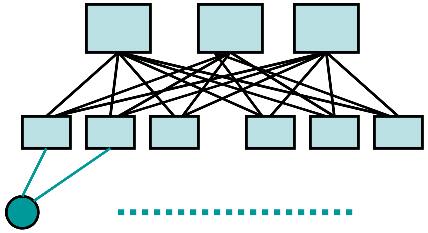
Host-stack

Possible Fabric Topologies for Multi-Rail

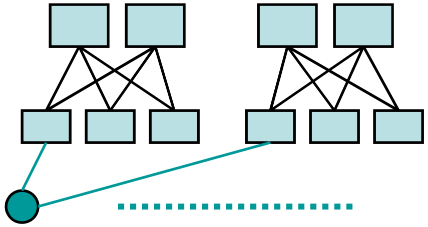
- Two connected fabrics
  - Two islands connected with few wires in between
  - Each server connects to the two islands
  - One SM



- One Clos based fabric
  - Each server connects to two edge switches
  - Symmetric topology



- Two totally independent fabrics
  - Not connected to each other, two SMs
  - Each node connects to the two fabrics



IPoIB & Bonding

- High Availability for IPoIB is achieved through the Linux Bonding driver
- The Linux bonding code was changed in order to support IPoIB
  - Allows bonding to use the HW address of the active slave, as with IPoIB one can’t assign the HW address (GID, QPN)
- Bonding provides HA at the network stack link (L2) level; TCP sessions should not break.
- Port failure would cause the IB RC session of a native IB ULPs (SDP, RDS, iSER, Lustre, rNFS) to break
  - Use APM
  - Bonding allows a new session to be established immediately (as ipoib is the IB stack [rdma_cm] ARP provider)
  - Depending on the ULP, this session breakage may not be even seen by the user!

HA – Bonding (cont')

- Bonding HA mode:
  - Called Active-Backup (has one active slave)
  - Applies link detection mechanisms to trigger fail-over
  - One HW (L2) address is used for the bond, typically the one of the first slave, which is then assigned to the other slaves
- Link detection mechanisms:
  - Local: uses the carrier bit of the slaves
  - Path validation: implemented through an ARP target to which probes are sent
- Bond Fail-over:
  - Bonding sends a Broadcast Gratuitous ARP (originally to update the Ethernet switches tables)
  - Bonding does a "re-play" of all current node multicast join
  - Sends net event to RDMA CM → RDMA CM notifies IB ULP / application.

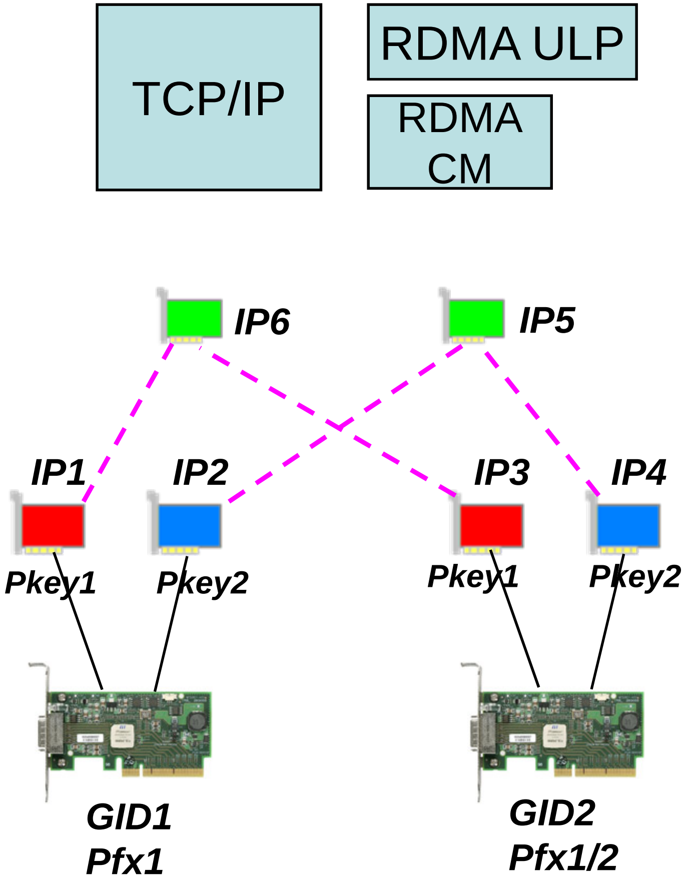
Host Stack and Addressing Overview

TCP/IP and RDMA-CM (NFS-R/iSER/RDS/Lustre/..) leverage IP for addressing

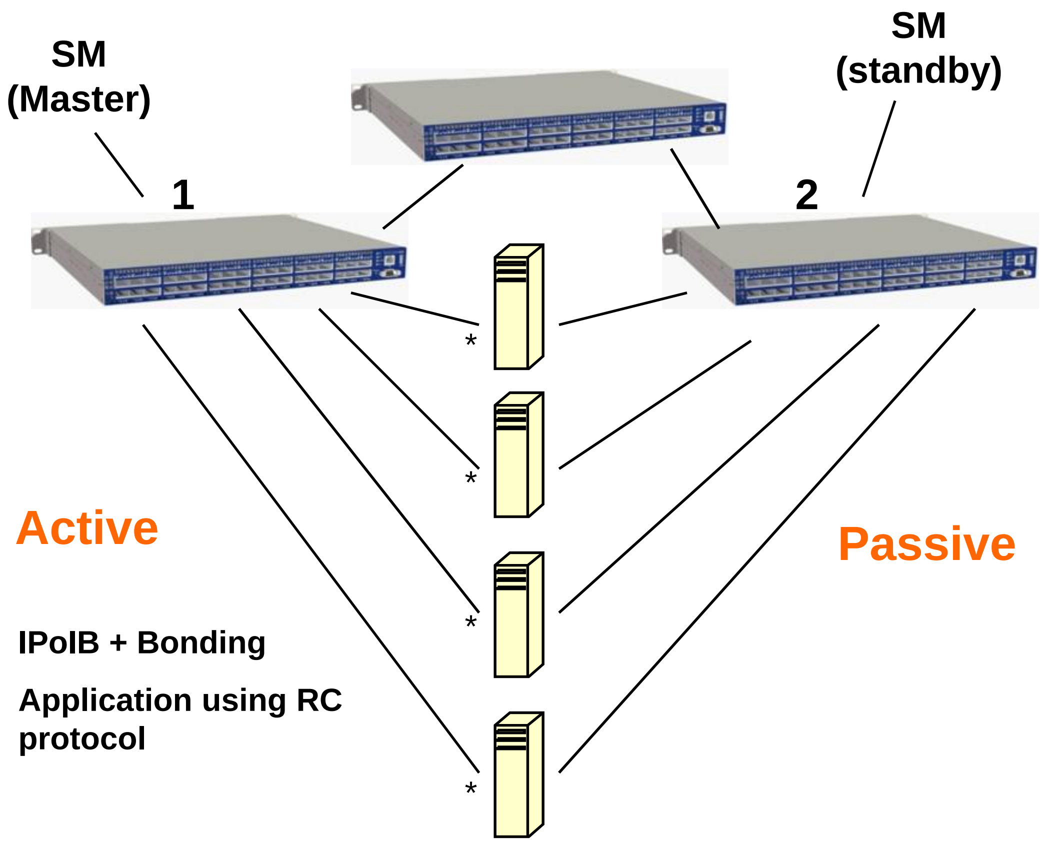
Bounded Interfaces (Optional)

Interfaces (one per port*partitions)

Real IB Ports



System Configuration & Testing



IPoIB + Bonding

Application using RC protocol

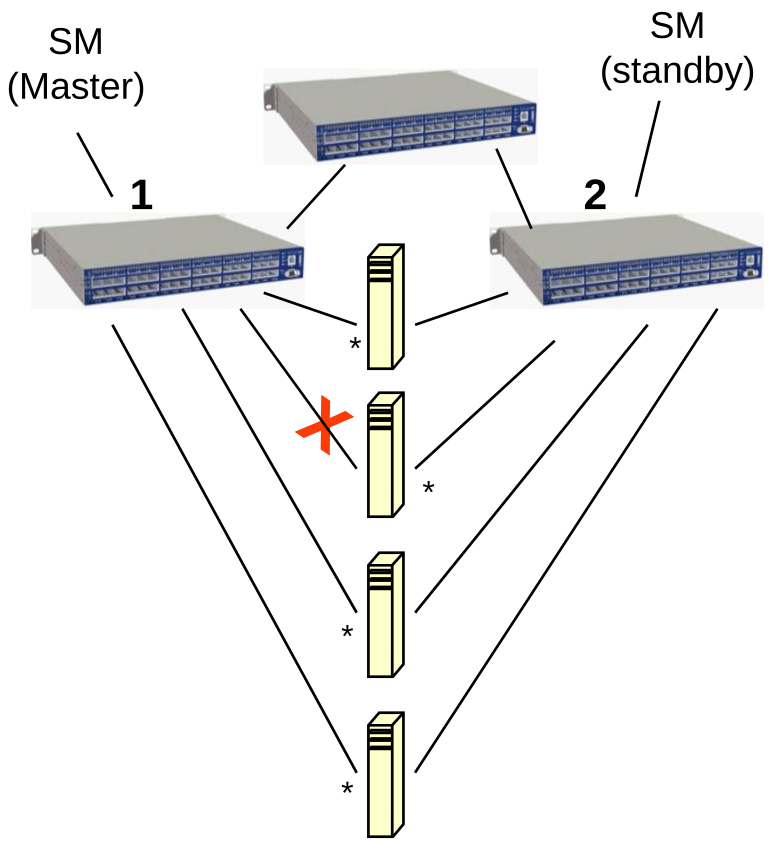
How do we build our systems ?

What do we want to test ?

Does this setup cover all ?

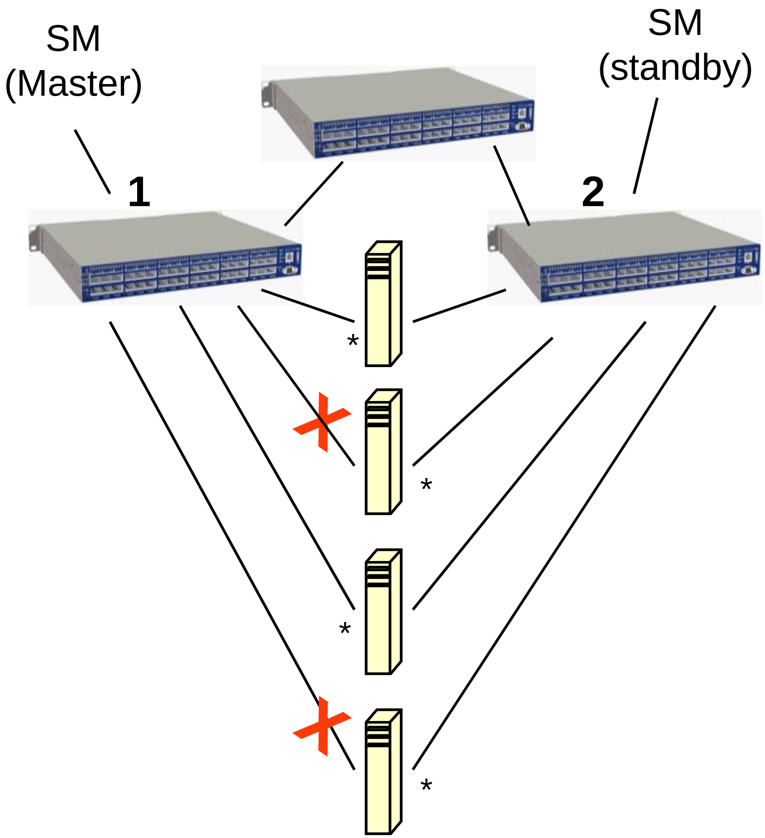
What about scalability ?

#1) Link or Port Failure



- Send gratuitous ARP to notify that GUID was changed
- IPoIB “restart level” - do not flush to all current path (assume same DLID) (1.4)
- Doing path query in any case (thread) and not wait to ARP prob (1.5 pending)
- IPoIB Internal queue for mc traffic during “restart” event (1.3.1)
- Net Event to RDMA CM – notify about bonding failover (Net Event). RDMA CM will notify it’s consumer (1.4).
- Fail back to primary – indication (1.4)
- RDMA CM connection will break (APM ?)
- RDMA CM connection reestablishment (ULP responsibility) – both ISL connections may be used

#2) Link events ^{x}



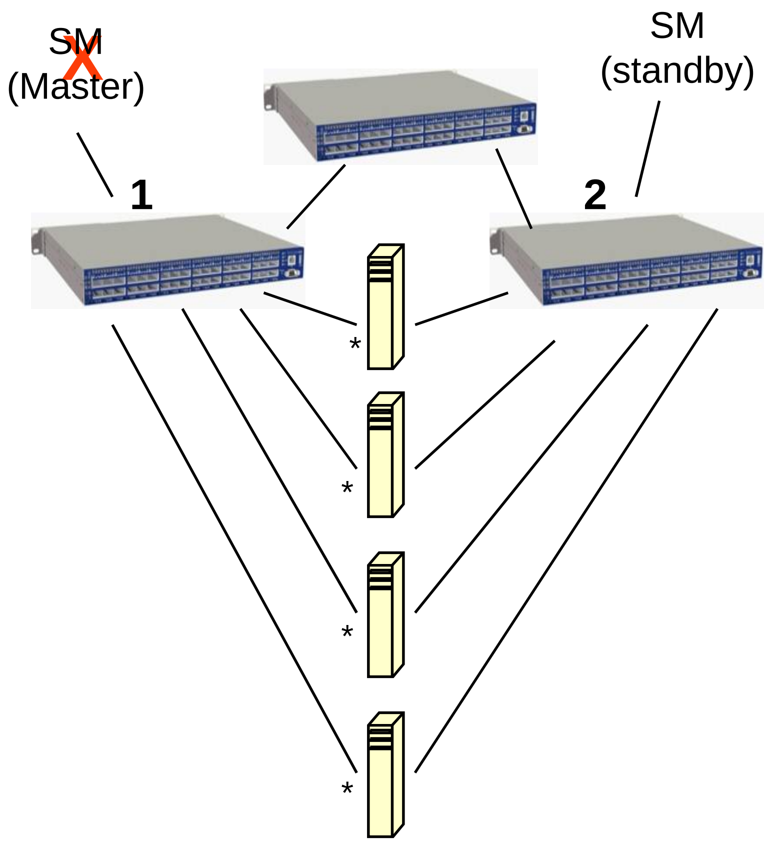
- ➢Bonding failover on two nodes
- ➢Send gratuitous ARP to notify that GUID was changed

What will happen if at the same time the remote port is not active yet ?

>=1.3.1:

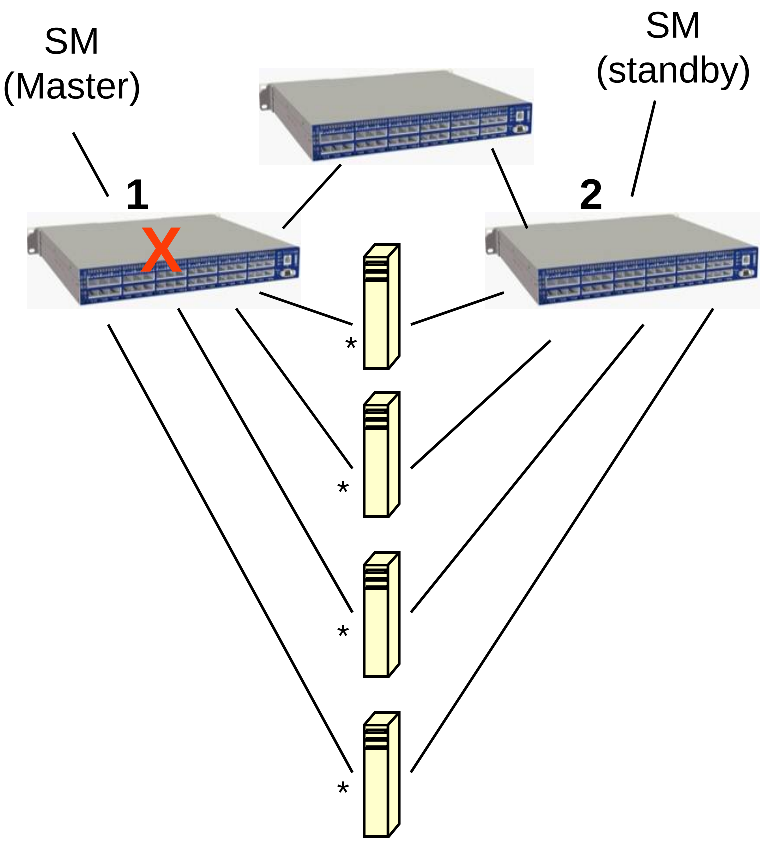
- ➢Send more then one grat ARP
- ➢Add possibility to configure number of grat ARP that can be sent & time interval

#3) SM Failure, Failover, Handover



- ➢After SM timeout standby become master (“SMInfo” protocol)
- ➢New SM validates current unicast LIDs & routing
- ➢New SM sends IB_Client_Reregister async event to all active HCA’s ports in the fabric
- ➢IPoIB “restart level” - do not flush to all current path (assume same DLID) (1.4)
- ➢Resending Join requests for all current joined groups
- ➢No failure for RDMA CM connection

#5) Switch (with master SM) Failure



Switch restart (power failure):

- ➢SM failover to SM2
- ➢May cause bond event
- ➢Host sends join to SM2
- ➢Switch (restart)
- ➢SM1 takeover
- ➢IB_Client_Reregister
- ➢Host sends join to new SM2
- ➢RC connections may not fail (QP timeout & retry settings) – reconnect may take more time

Applications & Protocols

- APM for better RC connection HA (limited to the same HCA)
- HA for different protocols and applications:
  - Many IP protocols know how to leverage multiple interfaces/IPs (e.g. iSCSI, Oracle, MPI*…)
- High Bandwidth (Storage) vs. Latency (heart-beat)
- Timeouts and configuration settings:

Protocol
RDMA_CM
QP parameter
IPoIB & Bonding
SM fail over

Summary

- Understand what the customer needs – not all the customers are the same
- HA & FT is a **System Property**
- Every component matters

Backup & additional info

IBV Events:

- IBV_EVENT_QP_FATAL
- IBV_EVENT_CQ_ERR
- **IBV_EVENT_PORT_ACTIVE**
- **IBV_EVENT_PORT_ERR**
- **IBV_EVENT_LID_CHANGE**
- IBV_EVENT_PKEY_CHANGE
- **IBV_EVENT_SM_CHANGE**
- **IBV_EVENT_CLIENT_REREGISTER**
- IBV_EVENT_DEVICE_FATAL
- IBV_EVENT_QP_REQ_ERR
- IBV_EVENT_QP_ACCESS_ERR
- IBV_EVENT_COMM_EST
- IBV_EVENT_SQ_DRAINED
- IBV_EVENT_PATH_MIG
- IBV_EVENT_PATH_MIG_ERR
- IBV_EVENT_QP_LAST_WQE_REACHED
- IBV_EVENT_SRQ_ERR
- IBV_EVENT_SRQ_LIMIT_REACHED

Linux bonding example

```
# route -n
Destination gateway   mask        flags metric ref use interface
10.10.0.0   0.0.0.0 255.255.0.0    U      0      0   0    bond0

# ip addr show bond0
<BROADCAST,MULTICAST,MASTER,UP> link/infiniband
   80:00:00:48:fe:80:00:00:00:00:00:00:00:02:c9:03:00:02:6b:df
   inet 10.10.5.62/16 brd 10.10.255.255

# ip addr show ib0
<BROADCAST,MULTICAST,SLAVE,UP> link/infiniband
   80:00:00:48:fe:80:00:00:00:00:00:00:00:02:c9:03:00:02:6b:df
# ip addr show ib1
<BROADCAST,MULTICAST,SLAVE,UP> link/infiniband
   80:00:00:48:fe:80:00:00:00:00:00:00:00:02:c9:03:00:02:6b:e0
```

Linux bonding example - cont

→ after **local** fail-over (bond uses secondary slave)

```
# ip addr show bond0
<BROADCAST,MULTICAST,MASTER,UP> link/infiniband
   80:00:00:48:fe:80:00:00:00:00:00:00:00:02:c9:03:00:02:6b:e0
```

→ ping remote node node (172.25.5.157)

```
# ip neigh show 172.25.5.157
10.10.5.157 dev bond0 lladdr
   80:00:00:49:fe:80:00:00:00:00:00:00:00:02:c9:03:00:02:6b:e8
   REACHABLE
```

→ after fail-over at the **remote** node (Grat. ARP updated OS neigh)

```
# ip neigh show 172.25.5.157
10.10.5.157 dev bond0 lladdr
   80:00:00:49:fe:80:00:00:00:00:00:00:00:02:c9:03:00:02:6b:e7
   REACHABLE
```