Overview of InfiniBand Architecture

Dhabaleswar K. (DK) Panda
The Ohio State University
E-mail: panda@cse.ohio-state.edu
http://www.cse.ohio-state.edu/~panda

Processing Bottlenecks in Traditional Protocols

- Ex: TCP/IP, UDP/IP
- Generic architecture for all network interfaces
- Host-handles almost all aspects of communication
  - Data buffering (copies on sender and receiver)
  - Data integrity (checksum)
  - Routing aspects (IP routing)
- Signaling between different layers
  - Hardware interrupt whenever a packet arrives or is sent
  - Software signals between different layers to handle protocol processing in different priority levels

Capabilities of High-Performance Networks

- Intelligent Network Interface Cards
- Support entire protocol processing completely in hardware (hardware protocol offload engines)
- Provide a rich communication interface to applications
  - ***User-level communication capability***
  - Gets rid of intermediate data buffering requirements
- No software signaling between communication layers
  - All layers are implemented on a ***dedicated*** hardware unit, and not on a ***shared*** host CPU

Previous High-Performance Network Stacks

- Virtual Interface Architecture (VIA)
  - Standardized by Intel, Compaq, Microsoft
- Fast Messages (FM)
  - Developed by UIUC
- Myricom GM
  - Proprietary protocol stack from Myricom
- These network stacks set the trend for high-performance communication requirements
  - Hardware offloaded protocol stack
  - Support for fast and secure user-level access to the protocol stack

IB Trade Association

- IB Trade Association was formed with seven industry leaders (Compaq, Dell, HP, IBM, Intel, Microsoft, and Sun)
- Goal: To design a scalable and high performance communication and I/O architecture by taking an integrated view of computing, networking, and storage technologies
- Many other industry participated in the effort to define the IB architecture specification
- IB Architecture (Volume 1, Version 1.0) was released to public on Oct 24, 2000
  - Latest version 1.2.1 released January 2008
- http://www.infinibandta.org

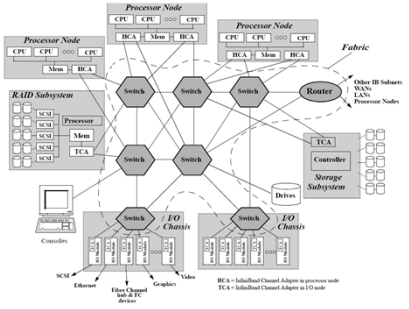
IB Hardware Acceleration

- Some IB models have multiple hardware accelerators
  - E.g., Mellanox IB adapters
- Protocol Offload Engines
  - Completely implement layers 2-4 in hardware
- Additional hardware supported features also present
  - RDMA, Multicast, QoS, Fault Tolerance, and many more

IB Overview

- **InfiniBand**
  - **Architecture and Basic Hardware Components**
  - Communication Model and Semantics
    - Communication Model
    - Memory registration and protection
    - Channel and memory semantics
  - Novel Features
    - Hardware Protocol Offload
      - Link, network and transport layer features
  - Management and Services
    - Subnet Management
    - Hardware support for scalable network management

A Typical IB Network



Three primary components

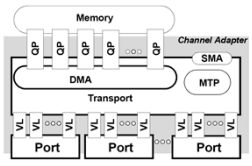
Channel Adapters

Switches/Routers

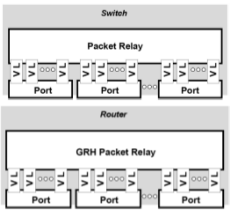
Links and connectors

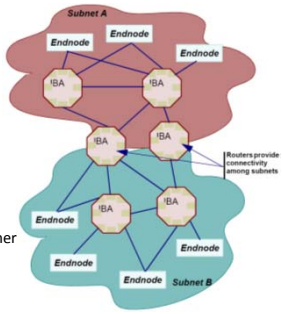
Components: Channel Adapters

- Used by processing and I/O units to connect to fabric
- Consume & generate IB packets
- Programmable DMA engines with protection features
- May have multiple ports
  - Independent buffering channeled through Virtual Lanes
- Host Channel Adapters (HCAs)



Components: Switches and Routers





- Relay packets from a link to another
- Switches: intra-subnet
- Routers: inter-subnet
- May support multicast

Components: Links & Repeaters

- Network Links
  - Copper, Optical, Printed Circuit wiring on Back Plane
  - Not directly addressable
- Traditional adapters built for copper cabling
  - Restricted by cable length (signal integrity)
  - For example, QDR copper cables are restricted to 7m
- Intel Connects: Optical cables with Copper-to-optical conversion hubs (acquired by Emcore)
  - Up to 100m length
  - 550 picoseconds copper-to-optical conversion latency
- Available from other vendors (Luxtera)
- Repeaters (Vol. 2 of InfiniBand specification)

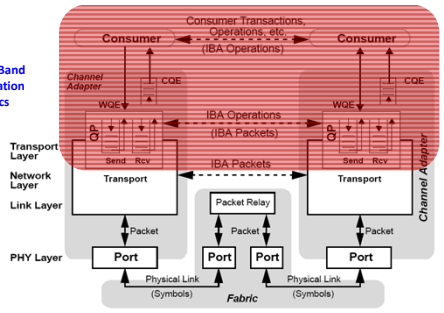
(Courtesy Intel)

IB Overview

- **InfiniBand**
  - Architecture and Basic Hardware Components
  - **Communication Model and Semantics**
    - **Communication Model**
    - **Memory registration and protection**
    - **Channel and memory semantics**
  - Novel Features
    - Hardware Protocol Offload
      - Link, network and transport layer features
  - Management and Services
    - Subnet Management
    - Hardware support for scalable network management

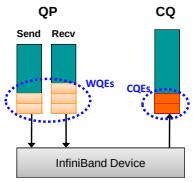
IB Communication Model

Basic InfiniBand Communication Semantics

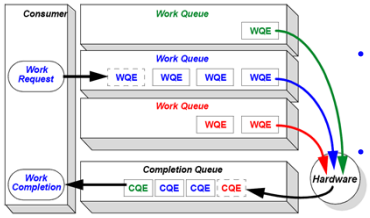


Queue Pair Model

- Each QP has two queues
  - Send Queue (SQ)
  - Receive Queue (RQ)
  - Work requests are queued to the QP (WQEs: "Wookies")
- QP to be linked to a Complete Queue (CQ)
  - Gives notification of operation completion from QPs
  - Completed WQEs are placed in the CQ with additional information (CQEs: "Cookies")



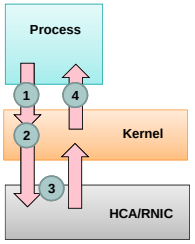
More on WQEs and CQEs



- Send WQEs contain data about what buffer to send from, how much to send, etc.
- Receive WQEs contain data about what buffer to receive into, how much to receive, etc.
- CQEs contain data about which QP the completed WQE was posted on, how much data actually arrived

Memory Registration

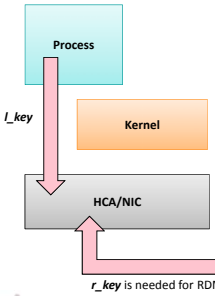
Before we do any communication: All memory used for communication must be registered



1. Registration Request
   - Send virtual address and length
2. Kernel handles virtual->physical mapping and pins region into physical memory
   - Process cannot map memory that it does not own (security !)
3. HCA caches the virtual to physical mapping and issues a handle
   - Includes an *l_key* and *r_key*
4. Handle is returned to application

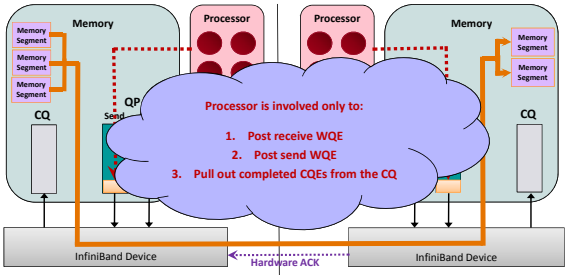
Memory Protection

For security, keys are required for all operations that touch buffers



- To send or receive data the *l_key* must be provided to the HCA
  - HCA verifies access to local memory
- For RDMA, initiator must have the r_key for the remote virtual address
  - Possibly exchanged with a send/recv
  - r_key is not encrypted in IB

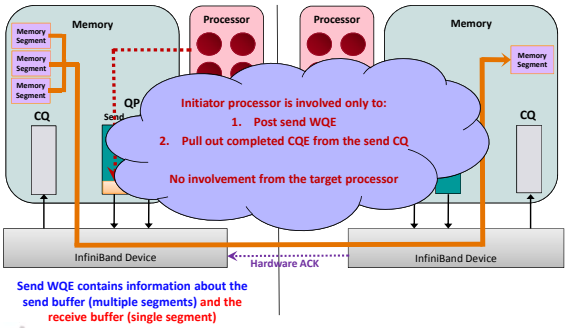
Communication in the Channel Semantics (Send/Receive Model)



Send WQE contains information about the send buffer (multiple non-contiguous segments)

Receive WQE contains information on the receive buffer (multiple non-contiguous segments); Incoming messages have to be matched to a receive WQE to know where to place

Communication in the Memory Semantics (RDMA Model)

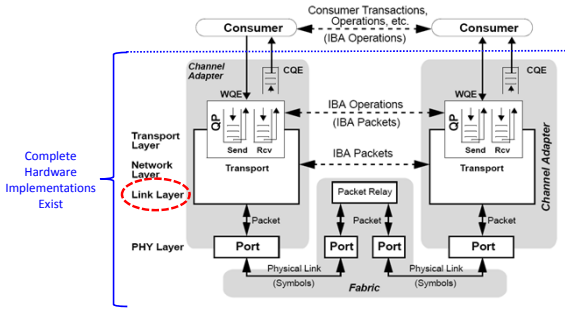


Send WQE contains information about the send buffer (multiple segments) and the receive buffer (single segment)

IB Overview

- **InfiniBand**
  - Architecture and Basic Hardware Components
  - Communication Model and Semantics
    - Communication Model
    - Memory registration and protection
    - Channel and memory semantics
  - **Novel Features**
    - **Hardware Protocol Offload**
      - **Link, network and transport layer features**
  - Management and Services
    - Subnet Management
    - Hardware support for scalable network management

Hardware Protocol Offload



Link Layer Capabilities

- **CRC-based Data Integrity**
- **Buffering and Flow Control**
- Virtual Lanes, Service Levels and QoS
- Switching and Multicast
- IB WAN Capability

CRC-based Data Integrity

- Two forms of CRC to achieve both early error detection and end-to-end reliability
  - Invariant CRC (ICRC) covers fields that do not change per link (per network hop)
    - E.g., routing headers (if there are no routers), transport headers, data payload
    - 32-bit CRC (compatible with Ethernet CRC)
    - End-to-end reliability (does not include I/O bus)
  - Variant CRC (VCRC) covers everything
    - 16-bit CRC
    - Erroneous packets do not have to reach the destination
    - Early error detection

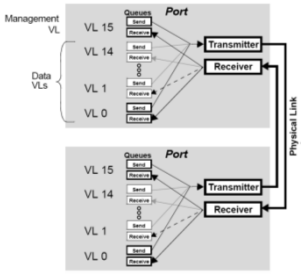
Buffering and Flow Control

- IB provides an absolute credit-based flow-control
  - Receiver guarantees that it has enough space allotted for N blocks of data
  - Occasional update of available credits by the receiver
- Has no relation to the number of messages, but only to the total amount of data being sent
  - One 1MB message is equivalent to 1024 1KB messages (except for rounding off at message boundaries)

Link Layer Capabilities

- CRC-based Data Integrity
- Buffering and Flow Control
- **Virtual Lanes, Service Levels and QoS**
- **Switching and Multicast**
- **IB WAN Capability**

Virtual Lanes

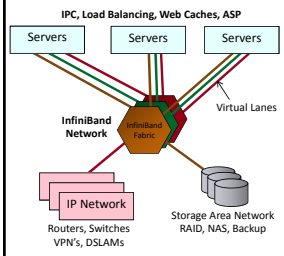


- Multiple virtual links within same physical link
  - Between 2 and 16
- Separate buffers and flow control
  - Avoids Head-of-Line Blocking
- VL15: reserved for management
- Each port supports one or more data VL

Service Levels and QoS

- Service Level (SL):
  - Packets may operate at one of 16 different SLs
  - Meaning not defined by IB
- SL to VL mapping:
  - SL determines which VL on the next link is to be used
  - Each port (switches, routers, end nodes) has a SL to VL mapping table configured by the subnet management
- Partitions:
  - Fabric administration (through Subnet Manager) may assign specific SLs to different partitions to isolate traffic flows

Traffic Segregation Benefits



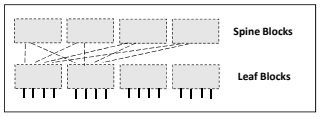
(Courtesy: Mellanox Technologies)

- InfiniBand Virtual Lanes allow the multiplexing of multiple independent logical traffic flows on the same physical link
- Providing the benefits of independent, separate networks while eliminating the cost and difficulties associated with maintaining two or more networks

Switching (Layer-2 Routing) and Multicast

- Each port has one or more associated LIDs (Local Identifiers)
  - Switches look up which port to forward a packet to based on its destination LID (DLID)
  - This information is maintained at the switch
- For multicast packets, the switch needs to maintain multiple output ports to forward the packet to
  - Packet is replicated to each appropriate output port
  - Ensures at-most once delivery & loop-free forwarding
  - There is an interface for a group management protocol
    - Create, join/leave, prune, delete group

Destination-based Switching/Routing



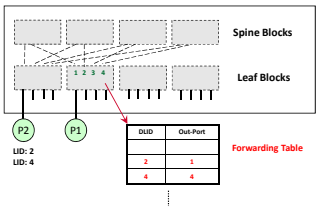
An Example IB Switch Block Diagram (Mellanox 144-Port)

Switching: IB supports Virtual Cut Through (VCT)

Routing: Unspecified by IB SPEC Up*/Down*, Shift are popular routing engines supported by OFED

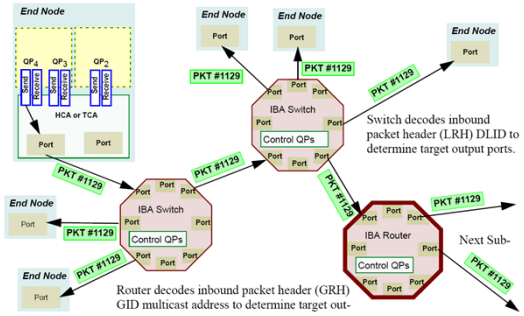
- Fat-Tree is a popular topology for IB Clusters
- Different over-subscription ratio may be used

IB Switching/Routing: An Example



- Someone has to setup these tables and give every port an LID
  - "Subnet Manager" does this work (more discussion on this later)
- Different routing algorithms may give different paths

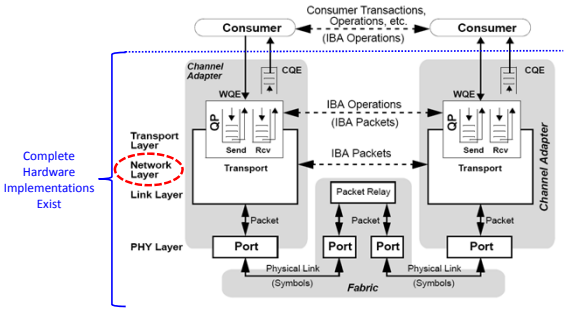
IB Multicast Example



IB WAN Capability

- Getting increased attention for:
  - Remote Storage, Remote Visualization
  - Cluster Aggregation (Cluster-of-clusters)
- IB-Optical switches by multiple vendors
  - Obsidian Research Corporation: www.obsidianresearch.com
  - Network Equipment Technology (NET): www.net.com
  - Layer-1 changes from copper to optical; everything else stays the same
    - Low-latency copper-optical-copper conversion
- Large link-level buffers for flow-control
  - Data messages do not have to wait for round-trip hops
  - Important in the wide-area network

Hardware Protocol Offload



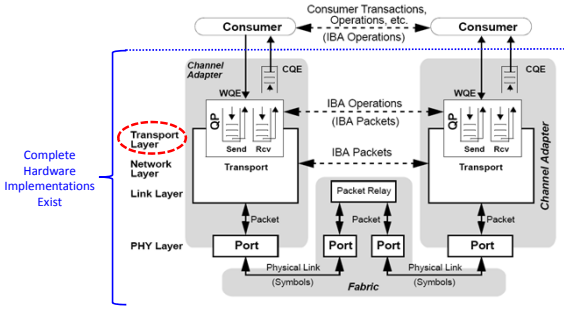
IB Network Layer Capabilities

- Most capabilities are similar to that of the link layer, but as applied to IB routers
  - Routers can send packets across subnets (subnet are management domains, not administrative domains)
  - Subnet management packets are consumed by routers, not forwarded to the next subnet
- Several additional features as well
  - E.g., routing and flow labels

Routing and Flow Labels

- Routing follows the IPv6 packet format
  - Easy interoperability with Wide-area translations
  - Link layer might still need to be translated to the appropriate layer-2 protocol (e.g., Ethernet, SONET)
- Flow Labels allow routers to specify which packets belong to the same connection
  - Switches can optimize communication by sending packets with the same label in order
  - Flow labels can change in the router, but packets belonging to one label will always do so

Hardware Protocol Offload



IB Transport Services

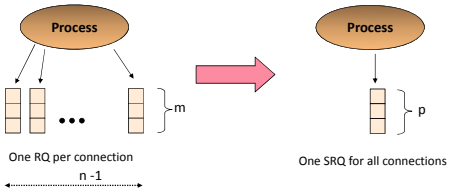
Service Type	Connection Oriented	Acknowledged	Transport
Reliable Connection	Yes	Yes	IBA
Unreliable Connection	Yes	No	IBA
Reliable Datagram	No	Yes	IBA
Unreliable Datagram	No	No	IBA
RAW Datagram	No	No	Raw

- Each transport service can have zero or more QPs associated with it
  - E.g., you can have four QPs based on RC and one QP based on UD

Trade-offs in Different Transport Types

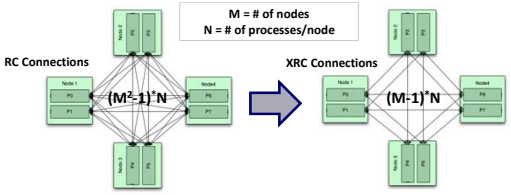
	Attribute	Reliable Connection	Reliable Datagram	Unreliable Datagram	Unreliable Connection	Raw Datagram (both IPv6 & ethertype)
	Scalability (M processes on N Processor nodes communicating with all processes on all nodes)	M^2*N QPs required on each processor node, per CA	M QPs required on each processor node, per CA.	M QPs required on each processor node, per CA.	M^2*N QPs required on each processor node, per CA.	1 QP required on each end node, per CA.
Reliability	Corrupt data detected	Yes				
	Data delivery guarantee	Data delivered exactly once		No guarantees		
	Data order guaranteed	Yes, per connection	Yes, packets from any one source QP are ordered to multiple destination QPs.	No	Unordered and duplicate packets are detected.	No
	Data loss detected	Yes		No	Yes	No
	Error recovery	**Reliable**. Errors are detected at both the requestor and the responder. The requestor can transparently recover from errors (retransmission, alternate path, etc.) without any involvement of the client application. QP processing is halted only if the destination is inoperable or all fabric paths between the channel adapters have failed.		**Unreliable**. Packets with some types of errors may not be delivered. Neither source nor destination QPs are informed of dropped packets.	**Unreliable**. Packets with errors, including sequence errors, are detected and may be logged by the responder. The requestor is not informed.	**Unreliable**. Packets with errors are not delivered. The requestor and responder are not informed of dropped packets.

Shared Receive Queue (SRQ)



- SRQ is a hardware mechanism for a process to share receive resources (memory) across multiple connections
  - Introduced in specification v1.2
- $0 < p << m*(n-1)$

eXtended Reliable Connection (XRC)

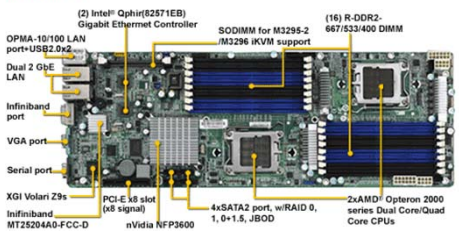


- Each QP takes at least one page of memory
  - Connections between all processes is very costly for RC
- New IB Transport added: eXtended Reliable Connection
  - Allows connections between nodes instead of processes

IB Hardware Products

- Many IB vendors: Mellanox, Voltaire and Qlogic
  - Aligned with many server vendors: Intel, IBM, SUN, Dell
  - And many integrators: Appro, Advanced Clustering, Microway
- Broadly two kinds of adapters
  - Offloading (Mellanox) and Onloading (Qlogic)
- Adapters with different interfaces:
  - Dual port 4X with PCI-X (64 bit/133 MHz), PCIe x8, PCIe 2.0 and HT
- MemFree Adapter
  - No memory on HCA → Uses System memory (through PCIe)
  - Good for LOM designs (Tyan S2935, Supermicro 6015T-INFB)
- Different speeds
  - SDR (8 Gbps), DDR (16 Gbps) and QDR (32 Gbps)
- Some 12X SDR adapters exist as well (24 Gbps each way)

Tyan Thunder S2935 Board



(Courtesy Tyan)

Similar boards from Supermicro with LOM features are also available

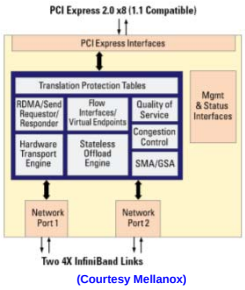
IB Hardware Products (contd.)

- Customized adapters to work with IB switches
  - Cray XD1 (formerly by Octigabay), Cray CX1
- Switches:
  - 4X SDR and DDR (8-288 ports); 12X SDR (small sizes)
  - 3456-port "Magnum" switch from SUN → used at TACC
    - 72-port "nano magnum"
  - 36-port Mellanox InfiniScale IV QDR switch silicon in early 2008
    - Up to 648-port QDR switch by SUN
  - New IB switch silicon from Qlogic introduced at SC '08
    - Up to 846-port QDR switch by Qlogic
- Switch Routers with Gateways
  - IB-to-FC; IB-to-IP

IB Software Products

- Low-level software stacks
  - VAPI (Verbs-Level API) from Mellanox
  - Modified and customized VAPI from other vendors
  - New initiative: Open Fabrics (formerly OpenIB)
    - http://www.openfabrics.org
    - Open-source code available with Linux distributions
    - Initially IB; later extended to incorporate iWARP
- High-level software stacks
  - MPI, SDP, IPoIB, SRP, iSER, DAPL, NFS, PVFS on various stacks (primarily VAPI and OpenFabrics)

Mellanox ConnectX Architecture



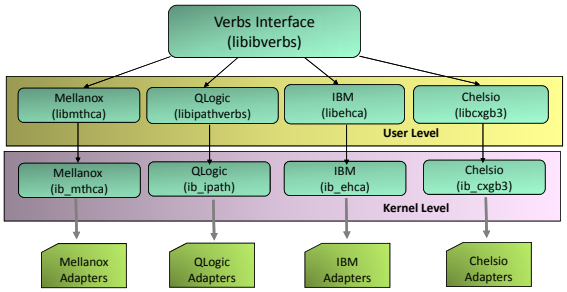
(Courtesy Mellanox)

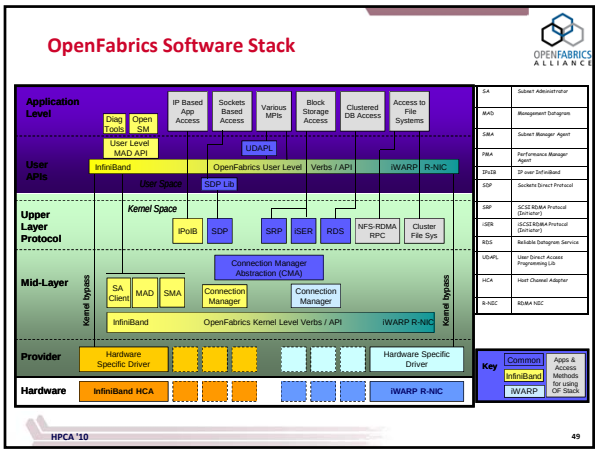
- Early adapter supporting IB/10GE convergence
  - Support for VPI and IBoE
- Includes other features as well
  - Hardware support for Virtualization
  - Quality of Service
  - Stateless Offloads

OpenFabrics

- Open source organization (formerly OpenIB)
  - www.openfabrics.org
- Incorporates both IB and iWARP in a unified manner
  - Support for Linux and Windows
  - Design of complete stack with `best of breed' components
    - Gen1
    - Gen2 (current focus)
- Users can download the entire stack and run
  - Latest release is OFED 1.4.3
  - OFED 1.5 is underway

OpenFabrics Stack with Unified Verbs Interface




OpenFabrics Software Stack
OPENFABRICS ALLIANCE
Application Level
Diag Tools
Open SM
IP Based App Access
Sockets Based Access
Various MPIs
Block Storage Access
Clustered DB Access
Access to File Systems
User Level MAD API
UDAPL
User APIs
InfiniBand
OpenFabrics User Level Verbs / API
iWARP R-NIC
User Space
SDP Lib
Kernel Space
Upper Layer Protocol
IPoIB
SDP
SRP
iSER
RDS
NFS-RDMA RPC
Cluster File Sys
Mid-Layer
Connection Manager Abstraction (CMA)
SA Client
MAD
SMA
Connection Manager
Connection Manager
Kernel bypass
Kernel bypass
InfiniBand
OpenFabrics Kernel Level Verbs / API
iWARP R-NIC
Provider
Hardware Specific Driver
Hardware Specific Driver
Hardware
InfiniBand HCA
iWARP R-NIC
SA Subnet Administrator
MAD Management Datagram
SMA Subnet Manager Agent
PMA Performance Manager Agent
IPoIB IP over InfiniBand
SDP Sockets Direct Protocol
SRP SCSI RDMA Protocol (Initiator)
iSER iSCSI RDMA Protocol (Initiator)
RDS Reliable Datagram Service
UDAPL User Direct Access Programming Lib
HCA Host Channel Adapter
R-NIC RDMA NIC
Key
Common
InfiniBand
iWARP
Apps & Access Methods for using OF Stack
HPCA '10
49