OFA Training Programs

Programming and Fabric Admin

Author: Rupert Dance
Date: 03/26/2012

Agenda – OFA Training Programs

- Writing Applications for RDMA using OFA Software
  - Program Goals and Instructors
  - UNH-IOL facilities & cluster equipment
  - Programming course format
  - Course requirements & syllabus
  - RDMA Benefits
  - Programming course examples
- Fabric Administration
- Future courses
- Course availability

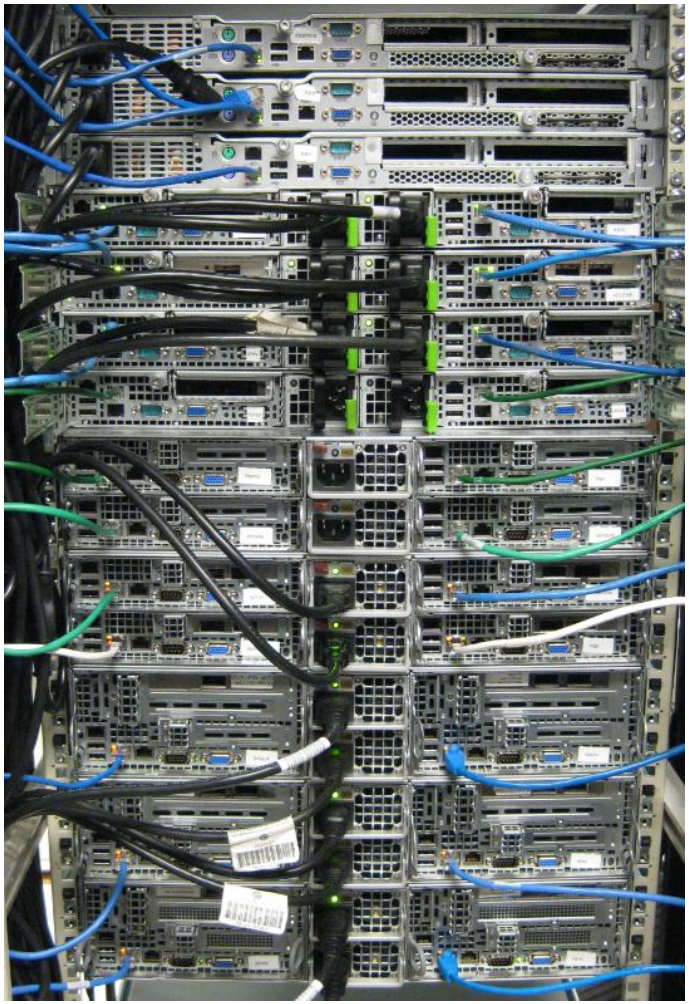
OFA Training Program - Overall Goals

- Provide application developers with classroom instruction and hands on experience writing, compiling and executing an application using OFED verbs API
- Illustrate how RDMA programming is different from sockets programming and provide the rationale for using RDMA.
- Focus on the OFED API, RDMA concepts and common design patterns
- Opportunity to develop applications on the OFA cluster at the University of New Hampshire – includes the latest hardware from Chelsio, DDN, Intel, Mellanox, NetApp & QLogic

Instructors

- **Dr. Robert D. Russell**: Professor in the CS Department at UNH
  - Dr. Russell has been an esteemed member of the University of New Hampshire faculty for more than 30 years and has worked with the InterOperability Laboratory's iSCSI consortium, iWARP consortium and the OpenFabrics Interoperability Logo Program.
- **Paul Grun**: Chief Scientist for System Fabric Works
  - Paul has worked for more than 30 years on server I/O architecture and design, ranging from large scale disk storage subsystems to high performance networks. He served as chair of the IBTA's Technical Working Group, contributed to many IBTA specifications and chaired the working group responsible for creating the RoCE specification.
- **Rupert Dance:** Co-Chair of the OFA Interoperability Working Group
  - Rupert helped to form and has led both the IBTA Compliance and Interoperability and OFA Interoperability programs since their inception. His company, Software Forge, worked with the OFA to create and provide the OFA Training Program.

UNH Interoperability Lab



Resources

- 32,000 Square Feet
- 100+ staff & students
- IB, iWARP and Windows Clusters

Programming Course Format

- Part One - Introduction to RDMA
  - I/O Architecture and RDMA Architecture
  - Address translation and network operations
  - Verbs Introduction and the OFED Stack
  - Introduction to wire protocols
  - 138 Slides
- Part Two - Programming with RDMA
  - Hardware resources: HCAs, RNICs, etc
  - Protection Domains and Memory Registration keys
  - Connection Management
  - Explicit Queue Manipulation & Event Handling
  - Explicit Asynchronous Operation
  - Explicit Manipulation of System Data Structures
  - 425 Slides and 20 complete client/server program examples

Programming Course Requirements

- Requirements
  - Knowledge of “C” programming including concepts such as structures, memory management, pointers, threads and asynchronous programming
  - Knowledge of Linux since this course does not include Windows programming
- Helpful
  - Knowledge of Event Handlers
  - Knowledge of sockets or network programming
  - Familiarity with storage protocols

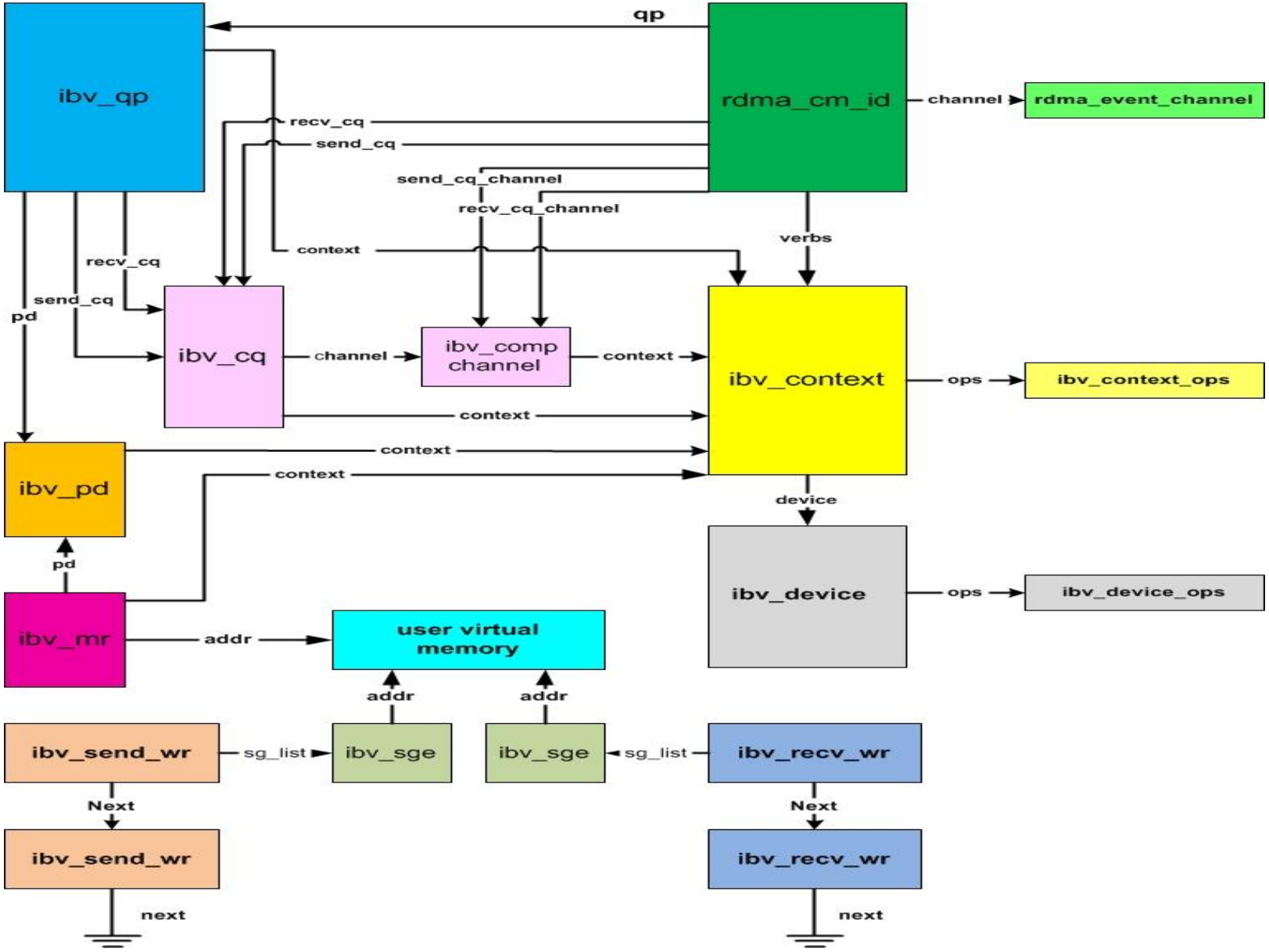
Programming Course Syllabus

- Introduction to OFA architecture
  - Verbs and the verbs API
  - A Network perspective
  - RDMA Operations – SEND/RECEIVE, RDMA READ & WRITE
  - RDMA Services
  - Isolation and Protection Mechanisms
  - A brief overview of InfiniBand Management
  - A quick introduction to the OFED stack
  - Host perspective
    - Asynchronous processing
    - Channel vs. RDMA semantics
- Basic Data Structures
  - Connection Manager IDs
  - Connection Manager Events
  - Queue Pairs
  - Completion Queues
  - Completion Channels
  - Protection Domains
  - Memory Registration Keys
  - Work Requests
  - Work Completions
- Connection management basics
  - Establishing connections using RDMACM
  - RDMACM API
- Basic RDMA programming
  - Memory registration
  - Object creation
  - Posting requests
  - Polling
  - Waiting for completions using events
  - Common practices for implementing blocking wait
- Design patterns
  - Send-receive
  - RDMA cyclic buffers
  - Rendezvous
- Advanced topics
  - Work Request chaining
  - Multicast
  - Unsignaled Completions
- RDMA ecosystems
  - Native InfiniBand
  - iWARP
  - RoCE

Programming Course Sample

- OFA Software Benefits
- Conventional I/O versus RDMA I/O
- Address Translation and Transport Independence
- Description of the Verbs & Data Structures
- Preparation for posting a send operation
- Create the work request
- Gathering data from memory
- Putting gathered elements on the wire
- Multicast concept
- The Big Picture

Programming Course – The Big Picture

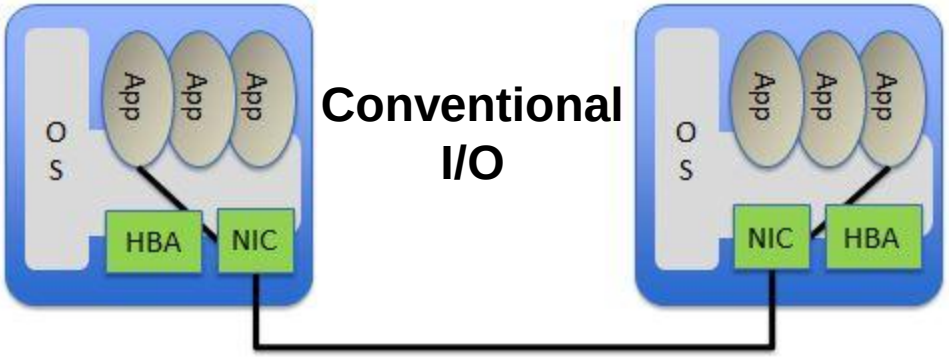


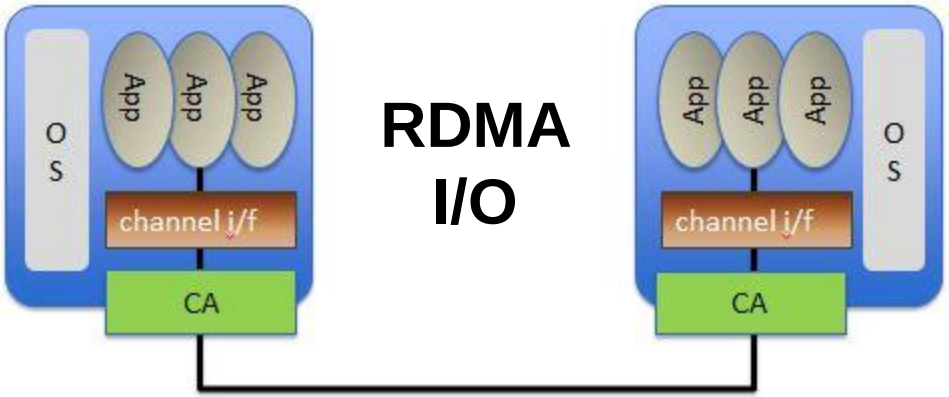
OFA Software Benefits

- Remote Direct Memory Access provides
  - Low latency – stack bypass and copy avoidance
  - Kernel bypass – reduces CPU utilization
  - Reduces memory bandwidth bottlenecks
  - High bandwidth utilization
- Cross Platform support
  - InfiniBand
  - iWARP
  - RoCE

Conventional I/O versus RDMA I/O

OS involved in all operations

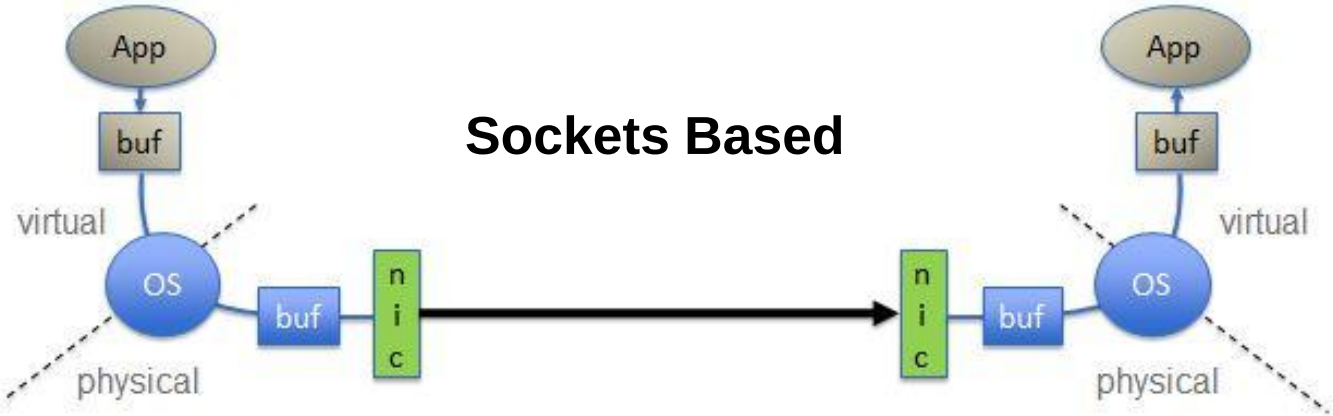


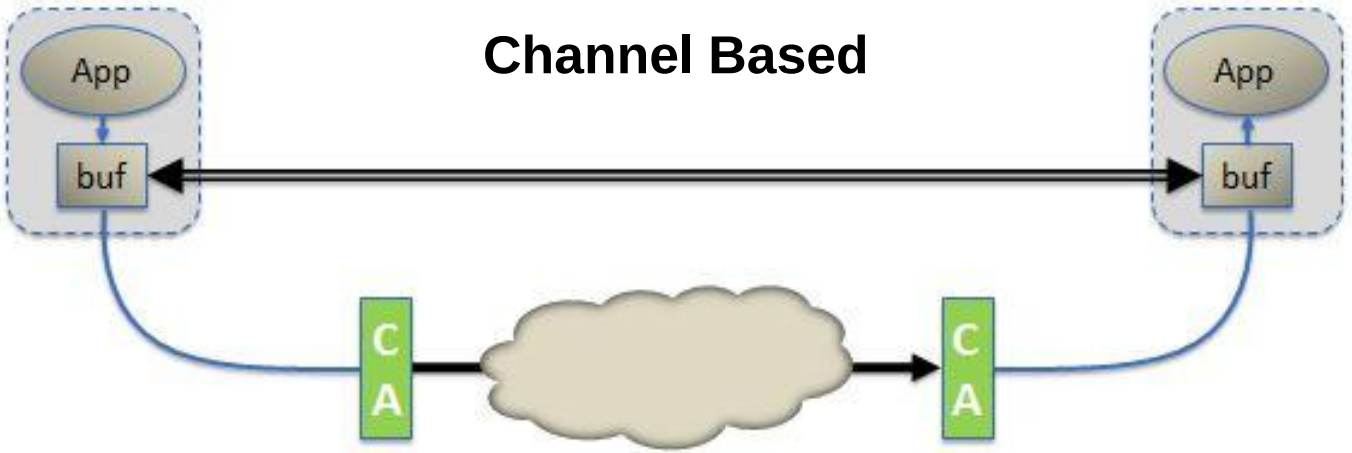


Channel interface runs in user space

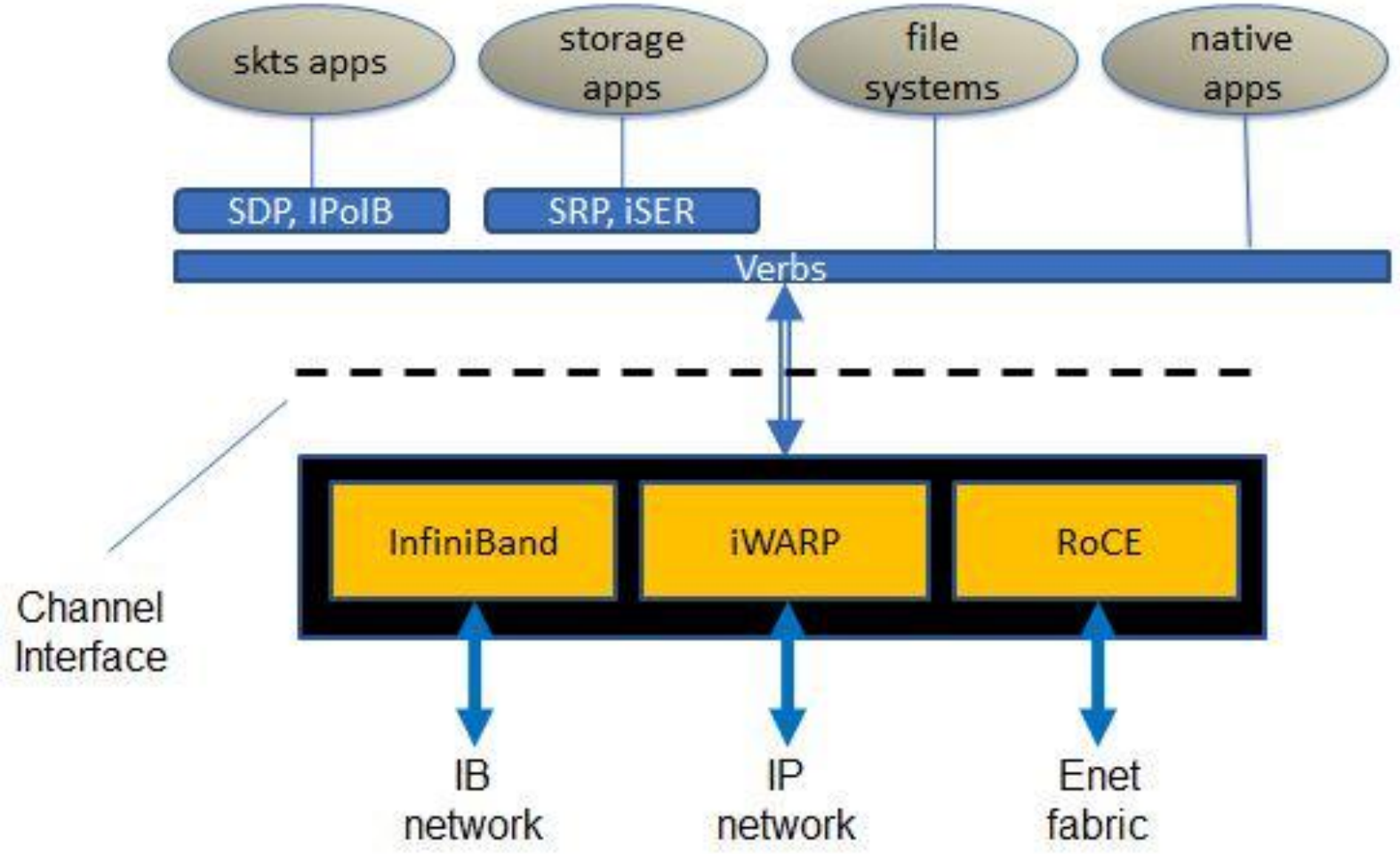
No need to access the kernel

Address Translation

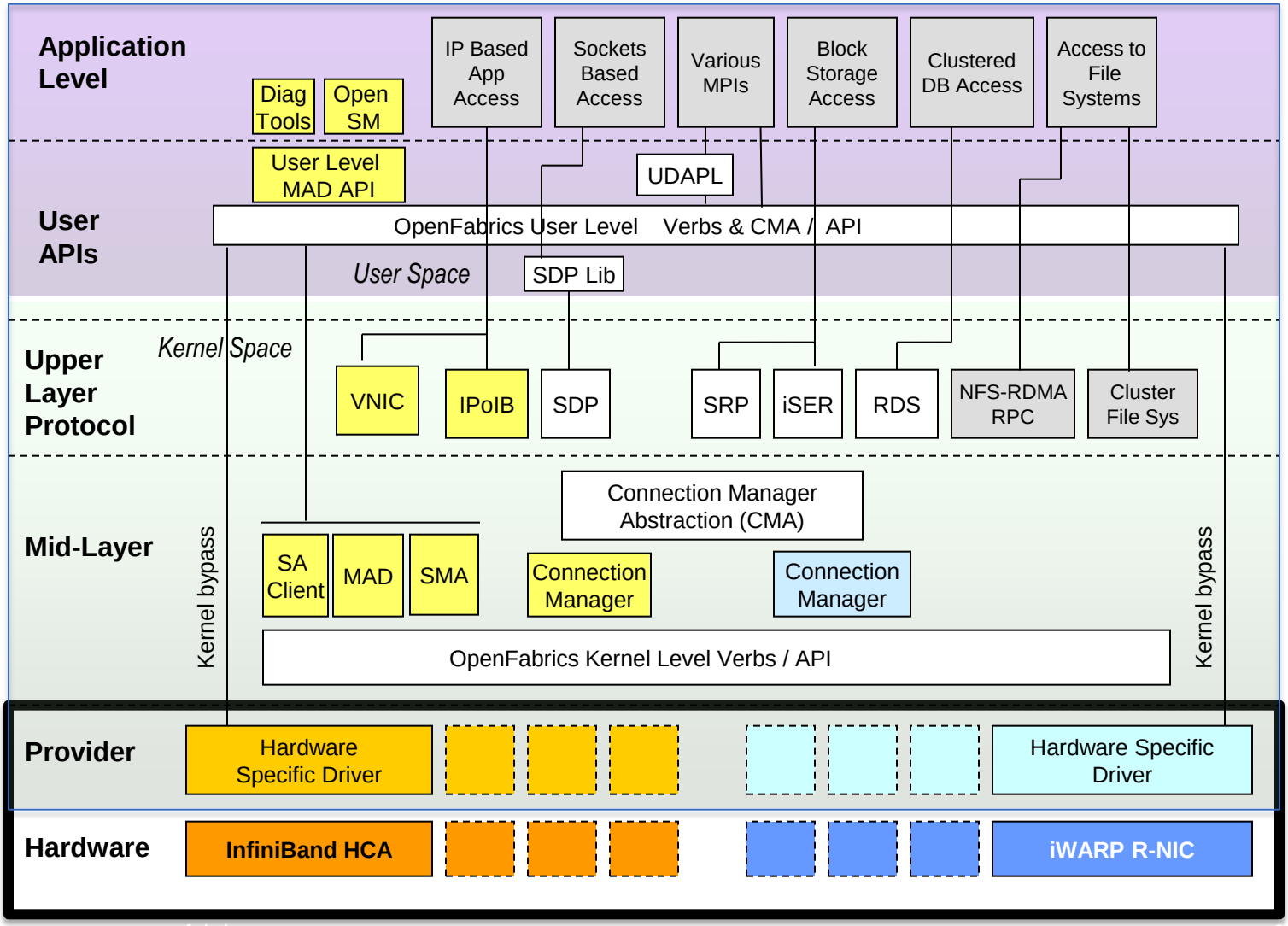




Many apps, one interface, three wires



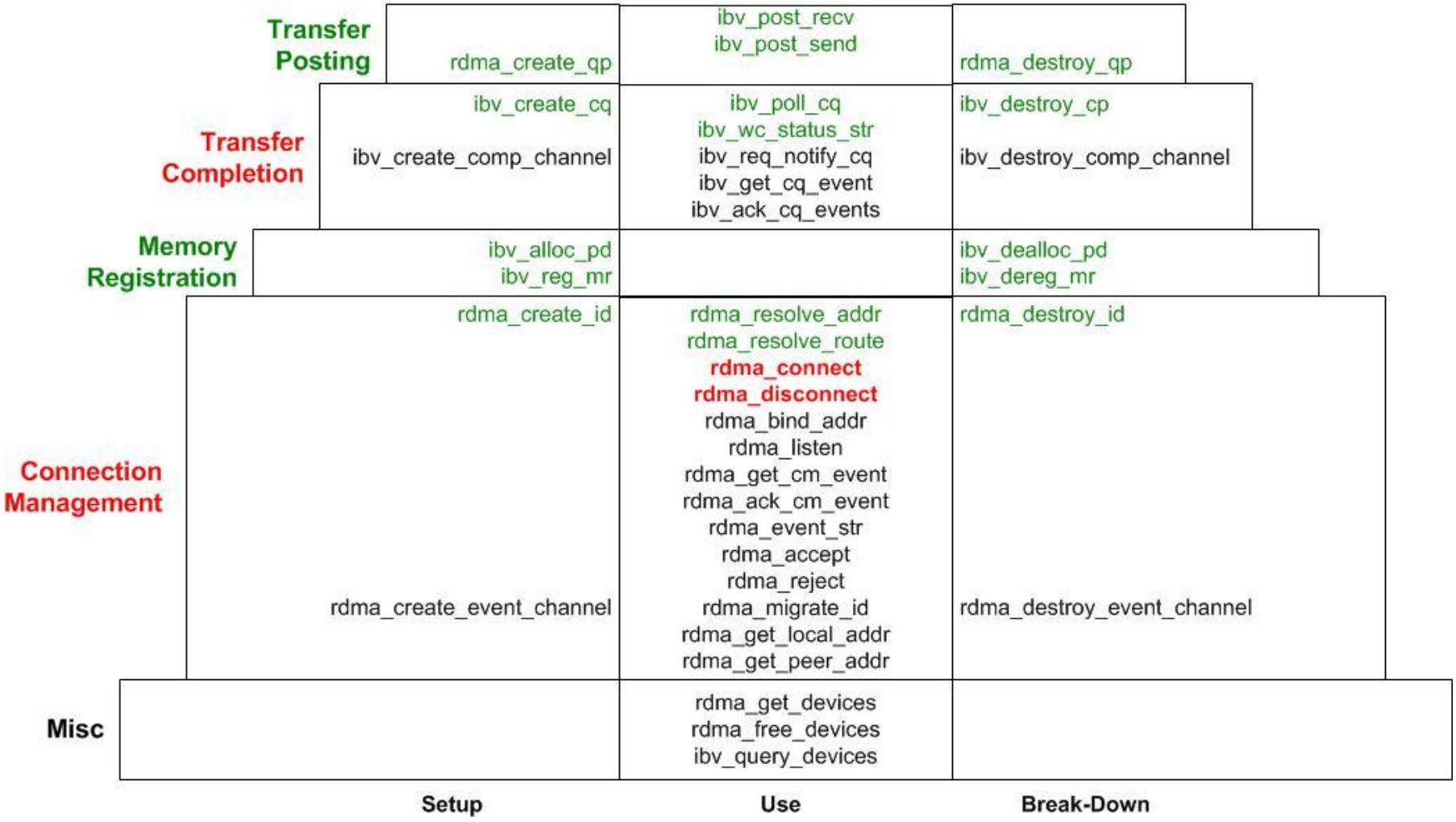
OFED – the whole picture



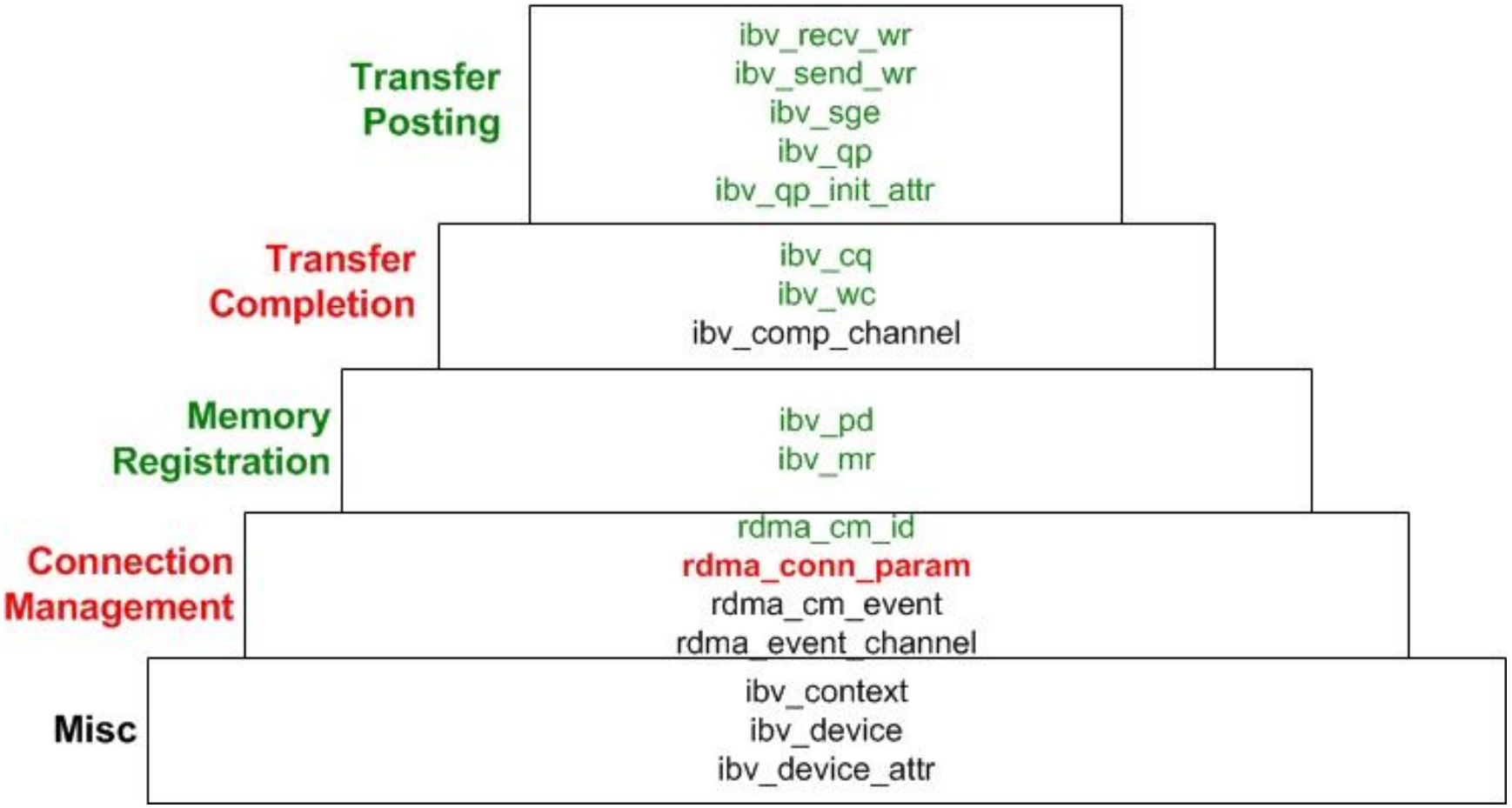
SA	Subnet Administrator
MAD	Management Datagram
SMA	Subnet Manager Agent
PMA	Performance Manager Agent
IPoIB	IP over InfiniBand
SDP	Sockets Direct Protocol
SRP	SCSI RDMA Protocol (Initiator)
iSER	iSCSI RDMA Protocol (Initiator)
RDS	Reliable Datagram Service
VNIC	Virtual NIC
UDAPL	User Direct Access Programming Lib
HCA	Host Channel Adapter
R-NIC	RDMA NIC

Key: Common; InfiniBand; iWARP; Apps & Access Methods for using OF Stack

Programming Course - OFED Verbs



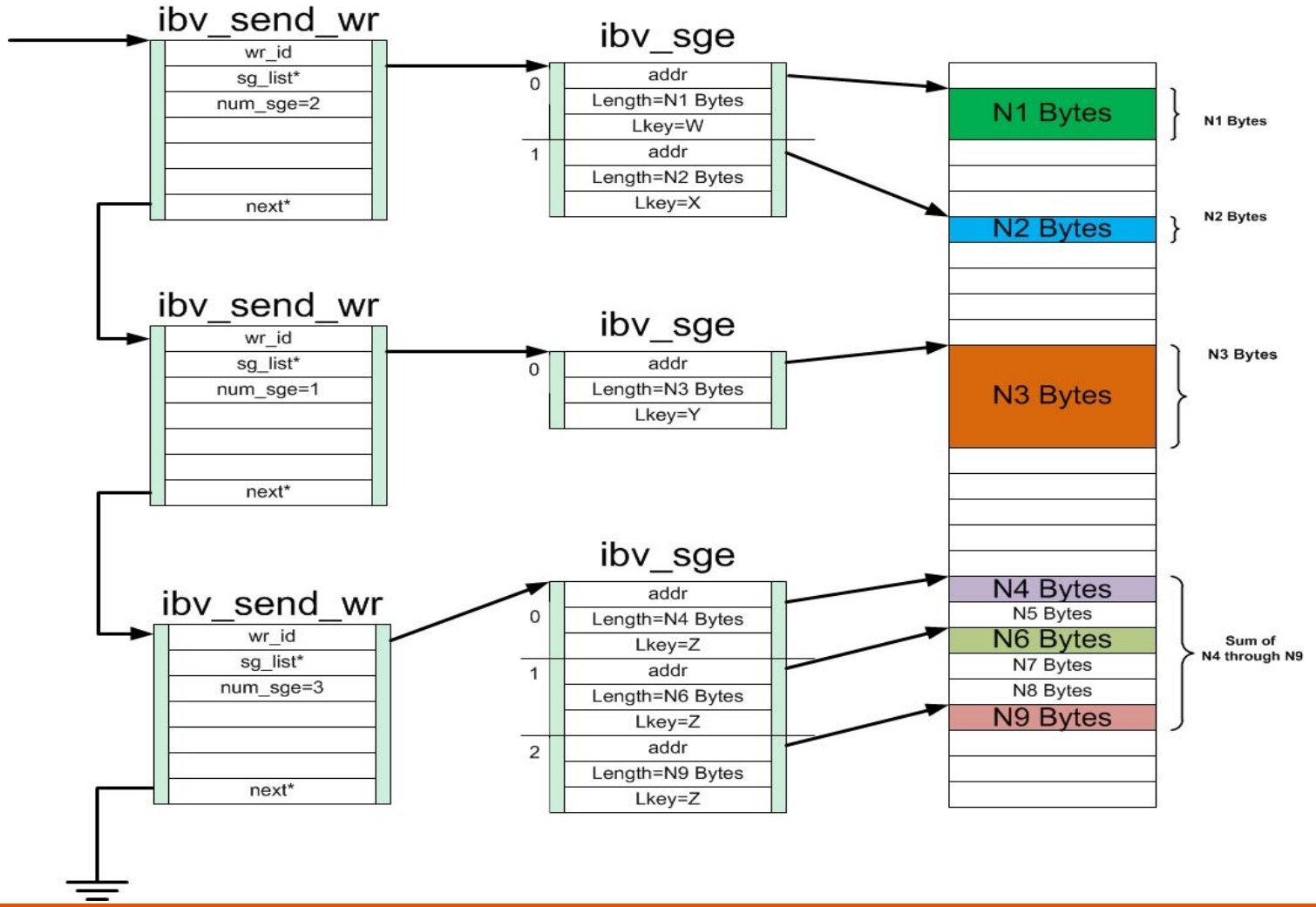
Programming Course – Data Structures



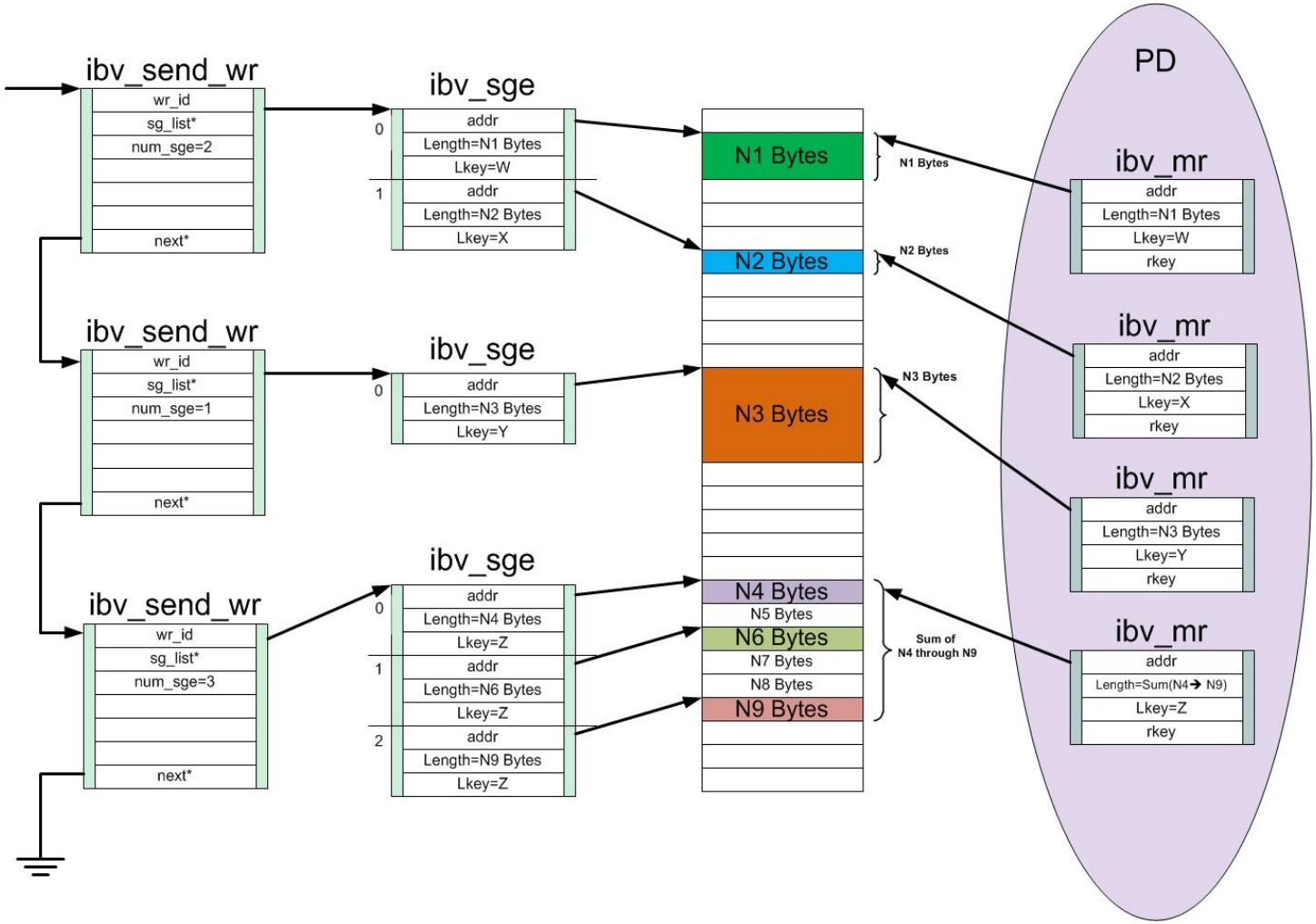
Bottom-up client setup phase

- **rdma_create_id()** - create **struct rdma_cm_id** – identifier
- **rdma_resolve_addr()** - bind **struct rdma_cm_id** to local device
- **rdma_resolve_route()** - resolve route to remote server
- **ibv_alloc_pd()** - create **struct ibv_pd** – protection domain
- **ibv_create_cq()** - create **struct ibv_cq** – completion queue
- **rdma_create_qp()** - create **struct ibv_qp** – queue pair
- **ibv_reg_mr()** - create **struct ibv_mr** – memory region
- **rdma_connect()** - create connection to remote server

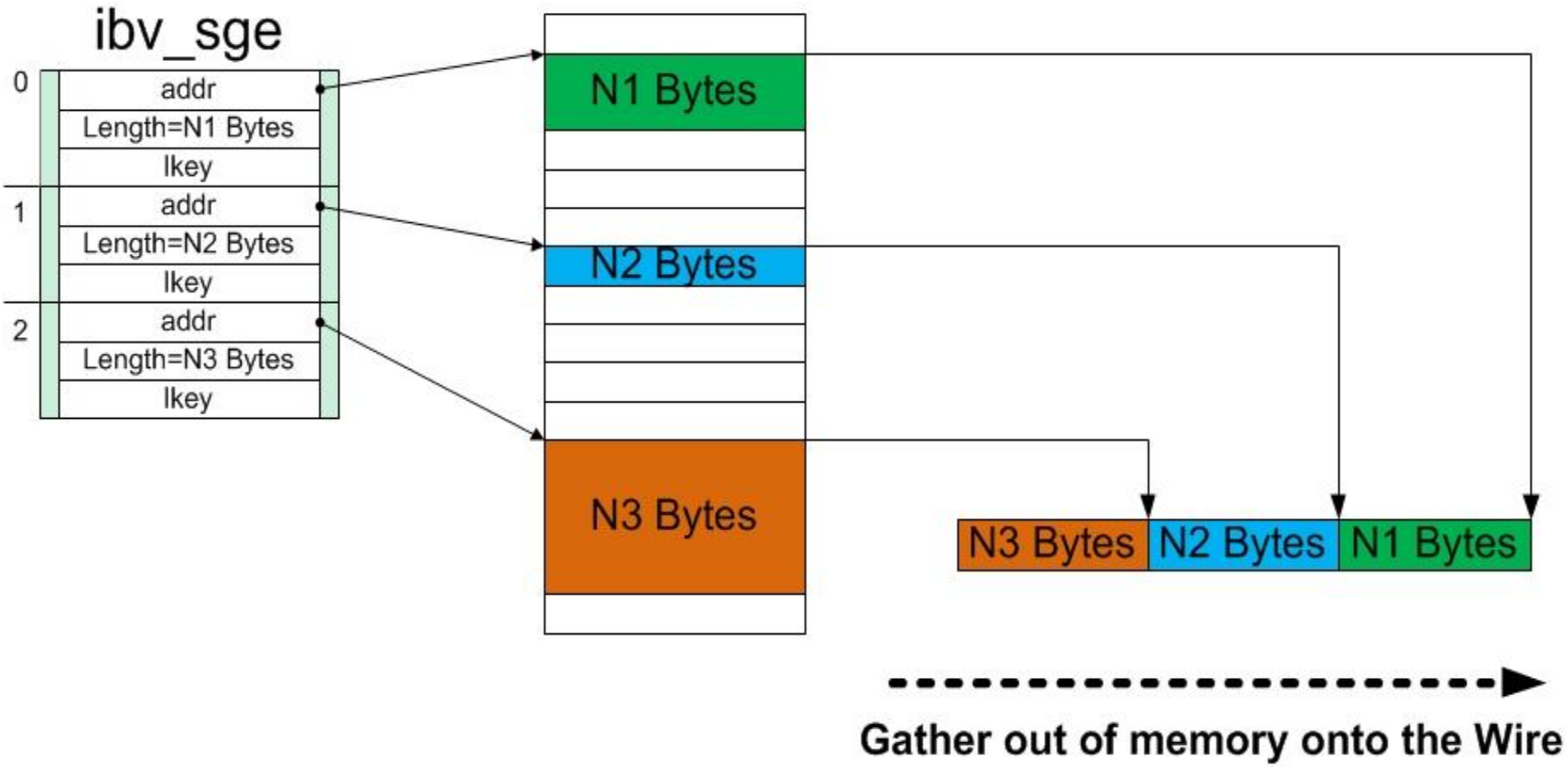
Creating Scatter Gather Elements



Protection Domains – Memory Regions



Gather during ibv_post_send()



Send Work Request (SWR)

- Purpose: tell network adaptor what data to send
- Data structure: **struct ibv_send_wr**
- Fields visible to programmer:

next	pointer to next SWR in linked list
wr_id	user-defined identification of this SWR
sg_list	array of scatter-gather elements (SGE)
opcode	**IBV_WR_SEND**
num_sge	number of elements in **sg_list** array
send_flags	**IBV_SEND_SIGNALED**

- Programmer must fill in these fields before calling **ibv_post_send()**

Posting to send data

- Verb: **ibv_post_send()**
- Parameters:
  - Queue Pair - QP
  - Pointer to linked list of Send Work Requests – SWR
  - Pointer to bad SWR in list in case of error
- Return value:

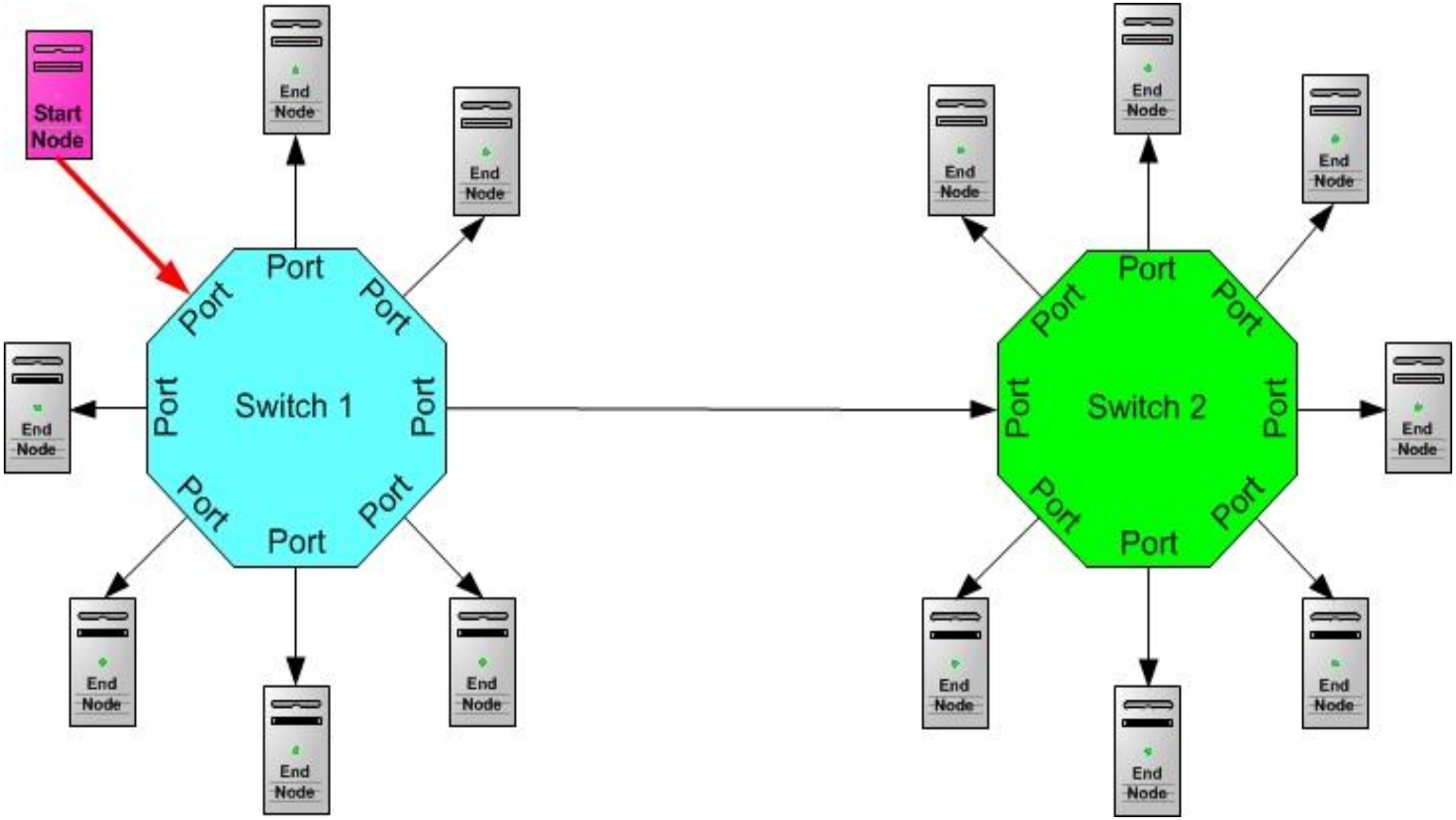
  == 0 all SWRs successfully added to send queue (SQ)

  != 0 error code

Bottom-up client break-down phase

- **rdma_disconnect()** - destroy connection to remote server
- **ibv_dereg_mr()** - destroy **struct ibv_mr** – memory region
- **rdma_destroy_qp()** - destroy **struct ibv_qp** – queue pair
- **ibv_destroy_cp()** - destroy **struct ibv_cq** – completion queue
- **ibv_dealloc_pd()** - deallocate **struct ibv_pd** – protection domain
- **rdma_destroy_id()** - destroy **struct rdma_cm_id** – identifier

Multicast concept



Multicast

- Optional to implement in IB CAs and switches
- Uses **Unreliable Datagram (UD)** mode
  - Only **Send/Recv** operations allowed
  - Both sides must actively participate in data transfers
- Receiver must have RECV posted for next SEND
- Receiver must process each RECV completion
- Only possible with IB, not iWARP

OPENFABRICS ALLIANCE

OFA Training Programs

RDMA Fabric Administration

IB, iWARP and RoCE

- IB, RoCE and iWARP all present the same APIs
  - Above RDMA-CM there is little or no difference
- Below RDMA-CM, two distinct management paradigms
  - The IB management paradigm
  - The Ethernet/IP management paradigm (iWARP, RoCE)
  - Below RDMA-CM there is no commonality
- Recommendation to prepare separate modules
  - IB Fabric Admin
  - iWARP Fabric Admin
  - RoCE
- Benefit is the ability to deliver targeted information to the end user

Delivery Venue

- This course is intended to be delivered in a dedicated facility (e.g. UNH-IOL)
  - It can also be delivered in a customer classroom, or remotely
  - Face-to-face delivery is preferred
- For lab purposes:
  - Access to a dedicated (non-production) cluster is required
  - Students can SSH into a remote cluster (e.g. UNH-IOL)
  - There is no specific requirement for an on-site cluster to support the class
  - Course scheduling should be coordinated with UNH-IOL activities

OPENFABRICS ALLIANCE

IB Fabric Administration using the OFED Mid-Layer

Syllabus

A System Admin's Guide to IB – 4 hrs

1. Introduction to InfiniBand for System Administrators – 1 hour
   - IB goals and concepts (leverage material from RDMA Programming Course)
   - IB as a layer 2 fabric – RoCE fabrics, iWARP
   - The role of IB management in the overall RDMA architecture
   - Introduction to OFED architecture and ULPs (e.g. IPoIB)

2. IB physical components – :30 min
   - HCAs/TCAs, switches, routers, gateways, cables

3. IB Management Concepts – 2.5 hours
   - Centralized management architecture
   - Manager/agent communication
   - Fabric management concept
   - Connection Management concept
   - Traffic engineering features (partitioning, VLs, QoS)
   - IB multicast compared to Ethernet

Planning and Installation – 4 hrs

1. Planning an IB Installation – 1 hr
   - Introduction to topologies
   - Deadlocks and deadlock avoidance
   - Traffic engineering considerations: SLs → VLs, Partitioning, QoS

2. Installing IB – 3 hrs
   - Combination lab / lecture
   - OFED installation – Linux and Windows
   - OFED compile tips and tricks
     - Use pre-compiled binaries
     - Hardware installation, cabling, BKMs
   - Optical versus copper cables, connector types, distances
   - Verifying basic connectivity and operation, BKMs
   - Initial power-on and debugging

IB Management Architecture – 4 hrs

OFED mid-layer topics

1. Fabric Management – 1.5 hrs
   - SMPs, GMPs
   - End-to-end path concepts – MTU, PMTU
   - Fabric discovery and configuration (directed routed, routed discovery, VL15)
   - Fabric partitioning, P_Keys, Q_Keys
   - QoS considerations
2. Identifiers, Addresses and Address Resolution – 1.5 hrs
   - Global addresses: GUIDs, GIDs, IP addresses
   - Local addresses – LIDs, MLIDs,
   - Address resolution
3. Managers and agents - :30 min
   - Introduction to SM, SMA, SA - discuss OpenSM, introduce 3rd party managers
   - Interacting with IB managers - SA query, Path Records, PM etc.
4. ULPs & connection management - :30 min
   - IPoIB , RDMA-CM, IB-CM
5. Multicast management - :30 min
   - Establishing /joining/leaving multicast groups
   - IB multicast and the IB SM

Operating an IB Fabric – 4 hrs

1. Configuring the fabric using IB management methods - :30 min
   - Working with SMs, SAs
   - Switch-based versus host-based SMs
   - Discovering SMs, resolving conflicts

2. OFED Tools - :30 min
   - List of IB tools
   - Fabric discovery tools, connectivity tools, debugging tools
   - The ‘Top 10’ tools – practical advice on using the tools
   - Getting more information
     - Man pages
     - IB specifications

3. Interacting with the system, system tools – 3 hrs
   - Verifying correct installation and basic operation
   - Interacting with the system via OFED software
   - OFED tools (ib_rdma_bw…etc)

Advanced Topics – not included in course

1. Performance analysis
   - Tools
   - Metrics
2. Traffic engineering
   - TC → SL → VL
3. Advanced topologies
4. Storage applications & management
   - SRP, iSER, NFS/RDMA, etc.
5. Bridging technologies
   - EoIB, FCoIB…
6. IB over the WAN
7. MPI

OPENFABRICS ALLIANCE

Managing an IP Fabric for iWARP

Syllabus

Fabric Admin's Guide to iWARP

1. Introduction to iWARP for Network Admins
   - Contrasting and Comparing iWARP with IB and RoCE fabrics
   - Introduction to OFED architecture and ULPs
2. iWARP physical components
   - Ethernet switches and routers.
   - Ethernet cables
3. Software
   - RNIC Verbs
   - MPI APIs – Intel MPI, HP-MPI, Scali MPI, MVAPICH2
   - Non-MPI APIs AMQP Voltaire VMA, NYSE Data Fabric
   - uDAPL v1.2, v1.3

iWARP Planning and Installation

1. Planning an iWARP Installation
   - iWARP considerations in an Ethernet networking environment
   - Ethernet network topologies
   - Ethernet Switch features and configuration
2. Installing iWARP – 3 hrs
   - Combination lab / lecture
   - BIOS and System settings
   - OFED installation – Linux and Windows
     - OFED compile tips and tricks
     - Use pre-compiled binaries
   - Hardware installation, cabling, BKMs
     - Optical vs copper cables, connector types, distances
   - Initial power-on and debugging
   - Verifying basic connectivity and operation, BKMs
3. Switch Management
   - Enable Flow Control, Transmit and Receive
   - Enable DCBx and Priority Flow Control (RoCE)
   - Configure default VLAN 0

Future OFA Software Training Course

- Advanced Programming topics
  - Kernel level programming
- ULP Training
  - MPI
  - RDS
  - SRP
- Other

OFA Training Course availability

Writing Applications for RDMA using OFA Software

- Next course: May 22-23 2012 – OFA Programming Course
- The course can be presented at your company location
  - Europe and Asia supported in addition to USA
  - Minimum of 8 attendees required
- The course is available via Webinar
  - Four half day course available for developers in Asia to avoid travel expense
  - Minimum of 8 attendees required

Fabric Admin

- Projected availability - Q3 2012
- Available quarterly at the UNH-IOL

For more information contact: rsdance@soft-forge.com