Main Page

From OpenFabrics Alliance Wiki

General information

- How do I get started? (pointers to InfiniBand documentation)
- RDMA stacks (OFED, Mellanox OFED, Linux RDMA...) presentation (https://openfabrics.org/images/eventpresos/workshops2014/IBUG/presos/Wednesday/PDF/04_RDMA_Stacks.pdf)
- Inside an OFED distribution
- Links to OFED software (https://downloads.openfabrics.org)
- (InfiniBand User's Group) 2014 workshop presentations (https://www.openfabrics.org/2014-ibug-workshop/)
- IBUG (InfiniBand User's Group) 2013 workshop presentations (https://www.openfabrics.org/2013-ibug-workshop/)

General Tools and Tips

- Overview of Error Counters
- Basic Commands
- Diagnostics examples using OFED tools (ibnetdiscover,ibdiagnet,etc.)
- Routing Verification Tools, Dave McMillen (https://www.openfabrics.org/images/docs/user_day_2013/2013_UserDay_Fri_1400_McMillenRoutingVerificationTools.pdf)

Subnet Manager

- OpenSM Per-Module Logging
- Common error messages
- OpenSM performance manager

User contributed tools

- **User tools discussion page**

- ibmon (2013_UserDay_presentation (https://www.openfabrics.org/images/docs/user_day_2013/2013_UserDay_Fri_1130_CoulterOFA_Monitoring.pdf))
- Pragmatic IB Utilities (https://computing.llnl.gov/linux/piu.html)
- MPI-launched native IB performance tests (https://github.com/skcoulter/mpiib)
- Python RDMA (https://github.com/jgunthorpe/python-rdma)

Advanced topics

- IB partitioning (Pkeys and Mkeys)
- IB in OpenStack

- This page was last modified on 9 January 2019, at 10:11.
- This page has been accessed 151,403 times.

Learning Infiniband

From OpenFabrics Alliance Wiki

- Introduction to InfiniBand for End Users (https://cw.infinibandta.org/document/dl/7268)
- InfiniBand Specifications from the IBTA (InfiniBand Trade Association) (http://www.infinibandta.org/content/pages.php?pg=technology_download)
- InfiniBand Guide from Bull (http://support.bull.com/documentation/byproduct/infra/sw-extremcomp/sw-extremcomp-com/g/86Y242FD03)
- InfiniBand Network Troubleshooting Guidelines and Methodologies from Oracle (http://www.oracle.com/technetwork/database/availability/infiniband-network-troubleshooting-1863251.pdf)

OPENFABRICS ALLIANCE

RDMA Stacks

MOFED, OFED & Linux Kernel

Fernando Garcia (fg@gentwo.org)
and
Christoph Lameter (cl@gentwo.org)

RDMA libraries and stacks

Problem: Numerous RDMA / IB stacks from multiple sources with various features

What does one use for a production environment? We have a custom built kernel on top of Ubuntu 10.04.

One particular issue were the kernel modules and therefore Linux kernel dependencies. With that came fragility and lack of integration into Linux distributions.

What we need:

1. Extreme low latency
2. Multicast issues
3. Ethernet support

RDMA stacks

Stack	*Versions*	*Characteristic*
Linux Kernel **kernel.org**	*2.6.32-3.15*	*No external module build*
OFED **openfabrics.org**	*1.1.x-1.5.x* *3.2/3.5/3.12*	*OpenFabrics Releases*
MLNX_OFED **mellanox.com**	*1.4.X/1.5.X* *2.0/2.1*	*Mellanox enhanced OFED releases*
OFED-MIC *openfabrics.org*	*3.5*	*Intel Xeon Phi Ofed stack*
OFED-VMA	*1.5.X?*	*Early flow steering implementation*

Non Linux OFED stacks

- Windows “WinOFED”
- AIX 7.1 OFED
- Solaris/Ilumos OFED
- FreeBSD OFED

Linux Distributions supported

RHEL 6.X	Needs OFED/MOFED
RHEL 7 beta	Works out of box
Ubuntu 10.04	No OFED support, custom hacks
Ubuntu 12.04	OFED/MOFED available
Ubuntu 14.04	Works out of box
SLES11/SLES10	OFED/MOFED required
OLE	Not dealt with it
Debian	Usually requires customization work

History with RDMA Stacks

- 2008 SDR/DDR OFED 1.2/1.3 with VMA
- 2009 DDR with IPoIB OFED-1.4
- 2010 QDR OFED-1.5.X -> RDMA apps
- 2014 QDR Linux Kernel 3.14 IB stack

OFED API breakage

Binaries built against OFED 1.X break with strange errors when run with OFED 3.X or the Linux IB stack

Mismatch in data structures. Checks on symbol versioning of the linker do not trigger.

Other issues with header changes but those are to be expected when new features are introduced:

Ethernet support changes of MOFED/OFED vs Linux IB.

Flow steering APIs vs. earlier hacks

How to deploy an upstream RDMA stack

3.12 kernel with extra patches or vanilla 3.14

https://www.kernel.org/pub/scm/libs/infiniband/

https://git.kernel.org/cgit/libs/infiniband/

libibverbs verbs extensions patches

Missing send flags

Device controlled flow steering

    log_num_mgm_entry_size=-1

Questions?
Suggestions?

OPENFABRICS
ALLIANCE

Inside an OFED distribution

From OpenFabrics Alliance Wiki

This list is based on the source RPMs found in OFED-3.12.

	What does it provide?	Should be installed if ...
ibutils	advanced diagnostics commands	ibdiagnet, ibdmchk commands
libcxgb3		
libibmad		
infiniband-diags	Common network utilities	iblinkinfo, ibportstate, indiagnet, perfquery ...
ofed-docs		
ofed-scripts		
libmlx5		
libipathverbs		
perftest		
libmthca		
dapl		
ibsim		
compat-rdma		
srptools		
libibcm		
libnes		
infinipath-psm		
qperf		
libcxgb4		
libocrdma		
ibacm		
libibverbs		
opensm		
libehca		
rds-tools		
librdmacm		
libmlx4		
qlvnictools		
libibumad		
mstflint		

references

- BULL InfiniBand Guide http://support.bull.com/documentation/byproduct/infra/sw-extremcomp/sw-extremcomp-com/g/86Y242FD03

Retrieved from "https://www.openfabrics.org/mediawiki/index.php?title=Inside_an_OFED_distribution&oldid=105"

2014 IBUG Workshop

The **2nd Annual IBUG Workshop** was held April 2-3 in Monterey, California and offered a 2-day event with sessions related to understanding, implementing, and administering OpenFabrics Software (OFS) and the underlying hardware. This unique event brought together users of InfiniBand, RoCE and all RDMA technologies bundled in the OpenFabrics Software suite, providing them with a setting to talk together about the challenges and different opportunities using OFS.

Presentations

Wednesday, April 2nd

2:30 PM
Welcome / Opening Remarks
Susan Coulter/LANL

3:00 PM
Optimizing & Tuning Techniques for Running MVAPICH2 over IB
Dr. DK Panda & Hari Subramoni/OSU

4:00 PM
Distro/Testing & Integration
John Jolly/SUSE

4:30 PM
Distro/Testing & Integration
Doug Ledford/RedHat

5:00 PM
RDMA Stacks (RHEL 6, RHEL 7, MOFED, OFED & Linux Kernel
Christoph Lameter/GenTwo

5:30 PM
Verbs 2.0 / Open Framework
Sean Hefty/Intel

Thursday, April 3, 2014

8:00 AM

Welcome / Opening Remarks

Susan Coulter/LANL

8:30 AM

Scalable Subnet Administration

Hal Rosenstock/Mellanox

9:00 AM

Monitoring: Errors/Performance

Jesse Martinez/LANL

9:30 AM

Monitoring: A Case Study

Florent Parent/Calcul Quebec

10:00 AM

Subnet Manager Logs Explained

Hal Rosenstock/Mellanox

11:00 AM

SMC-R/RoCE Update

Jerry Stevens/IBM

11:30 AM

Optimizing Open MPI Parameters for IB

Nathan Hjelm/LANL

1:00 PM

Taming LNET

Doug Oucharek/Intel

1:30 PM

OpenStack & IB

Blake Caldwell/ORNL

2:00 PM

SRP

Bart Van Assche/Fusion-io

2:30 PM

Implementing TCP Sockets over RDMA

Patrick MacArthur/UNH-IOL

3:30 PM

Mkey Usage / IB Partitions

Susan Coulter/LANL

4:00 PM

Delivering RoCE for Converged Enterprise & Cloud Deployments

Joe Gervais/Emulex

4:30 PM

Virtualization & IB

Michel Riviere/Oracle

5:00 PM

Closing Remarks

Susan Coulter/LANL

Privacy Statement **Membership Information** **Contact**

2013 IBUG Workshop

The OFA User Day Workshop held on April 18-19, offered a 2-day workshop with sessions related to understanding, implementing and administering OpenFabrics Software and the underlying hardware. This unique event brought together OpenFabrics Software (OFS) users and provided them with a setting to talk with each other about the challenges, and different opportunities in using OFS.

Susan Coulter, HPC Network Administrator at Los Alamos National Laboratory, spear-headed this year's event and brought together an impressive line-up of speakers that included, Dave McMillen, Storage Architect at Cray, Inc., Blake Caldwell, HPC Systems Administrator with Oak Ridge National Laboratory, and, Florent Parent, Director of Operations, Site de l'Université Laval, Calcul Québec, as well as, other past participants and OFS design experts.

Download all the presentations.

Presentations

Thursday April 18th

Welcome / Opening Remarks
Susan Coulter, LANL

RDMA Architectural presentation
Rupert Dance, Software Forge
1:00 - 2:00 p.m.

RDMA Programming Concepts
Dr. Bob Russell, UNH-IOL
2:00 - 3:00 p.m.

Fabric Administration training classes, opportunities
Rupert Dance, Software Forge
3:30 - 4:30 p.m.

OFILG - OFA Interop Program
Edward Mossman, UNH-IOL
4:30 - 5:00 p.m.

iWARP - RDMA over Ethernet

Tom Reu, Chelsio

Friday, April 19th

Storage Area Networks

Blake Caldwell, ORNL

9:00 - 9:30 a.m.

NFS over RDMA

Jeff Becker,NASA

9:30 - 10:00 a.m.

Storage at a Distance

Jason Hick, NERSC

10:00 - 10:30 a.m.

Performance Metrics/Testing

Susan Coulter, LANL

11:00 - 11:30 a.m.

Practical Monitoring

Susan Coulter, Blake Caldwell, DOE

11:30 a.m. - 12:00 p.m.

SM/SA Functionality

Hal Rosenstock, Mellanox

12:00 - 12:30 p.m.

Routing verification tools - ibdmchk, ibsim, etc.

Dave McMillen, Cray

2:00 - 2:30 p.m.

DHCP/PXE Boot over IB

Florent Parent, Calcul Quebec

2:30 - 3:00 p.m.

Sockets over RDMA - Rsockets

Sean Hefty, Intel

3:00 - 3:20 p.m.

Sockets over RDMA - EXS

Dr. Bob Russell, UNH-IOL

3:20 - 3:40 p.m.

Sockets over RDMA - SMC-r

Jerry Stevens, IBM

3:40 - 4:00 p.m.

IB Multicast - uses, IPoIB arp

Dave McMillen, Cray

4:30 - 5:00 p.m.

Big Data

Eyal Gutkind, Mellanox

5:00 - 5:30 p.m.

Partitions - what applications need partitions

Dave McMillen, Cray

5:30 - 6:00 p.m.

Cloud and OFS

Narayan Desai, ANL

6:00 - 6:30 p.m.

OSM PML

From OpenFabrics Alliance Wiki

Contents

- 1 OpenSM Per-Module Logging
  - 1.1 Purpose
  - 1.2 How to enable
  - 1.3 Example

OpenSM Per-Module Logging

References:

- Subnet Manager logs explained, Hal Rosenstock (https://openfabrics.org/images/eventpresos/workshops2014/IBUG/presos/Thursday/PDF/04_opensm-pml.pdf)
-

Purpose

How to enable

- Enable via per_module_logging_file option in options file
- syntax:
- Module name are found is source code: opensm/osm_subnet.c

Example

(add example of an per_module_logging_file file)

OSM ErrorMsg

From OpenFabrics Alliance Wiki

OpenSM Error Messages

- Subnet Manager logs explained, Hal Rosenstock (https://openfabrics.org/images/eventpresos/workshops2014/IBUG/presos/Thursday/PDF/04_opensm-pml.pdf)
  - add examples from 10.00_2014_OFA_IBUG_opensm-pml.pdf
- InfiniBand Volume 1 IB management related chapters (http://www.infinibandta.org/content/pages.php?pg=technology_download)
- Source code: error numbers are unique

OSM PerfManager

From OpenFabrics Alliance Wiki

Contents

- 1 OpenSM Performance Manager
  - 1.1 Purpose
  - 1.2 How to enable
  - 1.3 Example

OpenSM Performance Manager

references:

- IB Monitoring Through the Console. J. Martinez, IBUG 2014 (https://openfabrics.org/images/eventpresos/workshops2014/IBUG/presos/Thursday/PDF/02_IB_Monitoring.pdf)

Purpose

How to enable

- Enable via options file
- syntax:
-

Example

add example

Partitions

From OpenFabrics Alliance Wiki

Contents

- 1 InfiniBand Partitions
  - 1.1 Purpose
  - 1.2 How to enable
  - 1.3 Example

InfiniBand Partitions

reference:

- IBUG 2014 presentation, S. Coulter (https://openfabrics.org/images/eventpresos/workshops2014/IBUG/presos/Thursday/PDF/10_MKeys.pdf)

Purpose

How to enable

Example

Retrieved from "https://www.openfabrics.org/mediawiki/index.php?title=Partitions&oldid=128"

OpenStack

From OpenFabrics Alliance Wiki

InfiniBand in Virtual Environments

sources of information

- OpenStack and IB, Blake Caldwell/ORNL (https://openfabrics.org/images/eventpresos/workshops2014/IBUG/presos/Thursday/PDF/07_OpenStack_IB.pdf)
- Magellan: Building high-performance Openstack Clouds, Narayan Desai, ANL (https://openfabrics.org/images/eventpresos/workshops2013/IBUG/2013_UserDay_Fri_1800_CloudIB_Magellan.pdf)

Examples

Overview of Error Counters

From OpenFabrics Alliance Wiki

Contents

- 1 IB Error Counter Definitions and Examples
  - 1.1 LinkDowned
  - 1.2 Linkspeed not at maximum
  - 1.3 Linkwidth not at maximum
  - 1.4 PortRcvErrors
  - 1.5 PortRcvRemotePhysicalErrors
  - 1.6 PortRcvSwitchRelayErrors
  - 1.7 Port[Rcv|Xmit]ConstraintErrors
  - 1.8 PortXmitWait
  - 1.9 RcvRemotePhys(ical)Errors
  - 1.10 SymbolErrors
  - 1.11 VL15Drop
  - 1.12 XmtDiscards

IB Error Counter Definitions and Examples

LinkDowned

Just like it says. Usually associated with a node reboot.

If not associated with a reboot, could be a failing connection. (like port flapping)

Linkspeed not at maximum

Link is not operating at full speed. (i.e. 2.5 Gbps, 5.0Gbps, 10.0Gbps)

Usually a reseat of cable/card resolves the issue.

Linkwidth not at maximum

Link is not operating at full width. (i.e. 4x, 8x, 12x)

Usually a reseat of cable/card resolves the issue.

PortRcvErrors

These errors can be due to local physical errors, local buffer overruns, or receiving a malformed packet.

If a malformed packet is received - this indicates a problem somewhere else on the fabric. Somebody is putting bad messages on the wire.

PortRcvRemotePhysicalErrors

Similar to !PortRcvErrors, the end bad packet EBP flag is set. Usually a problem between the physical and logical layers.

PortRcvSwitchRelayErrors

This field counts the number of packets that could not be forwarded by the switch.

The reasons for this include

1. VL mapping errors. (LANL has not implemented VLs (yet)).

2. Looping; input port and output port are the same

3. DLID errors; It is a Multicast DLID (0xC000 to 0xFFFE) not configured for this CA, or DLID is outside the LFTS range or greater than the LinearFDBTop, or Port associated with this DLID in the LFTS file does not exist.

Usually this is due to the poor implementation of multicast on IB and therefore can be ignored.

Port[Rcv|Xmit]ConstraintErrors

This is the number of packets [received and discarded on | not transmitted by] a port in the fabric.

There are 2 general reasons for this.

1. The filter for raw packets [inbound | outbound] is turned on and these are raw packets

2. The partition key or IP version check has failed.

PortXmitWait

This field counts the number of packets that had to wait before being transmitted.

It is almost always non-zero.

Really large numbers indicate congestion. If the congestion gets really bad, you will see !XmitDiscards.

RcvRemotePhys(ical)Errors

This field counts "Total number of packets marked with the EBP delimiter received on the port."

The idea is that an "End Bad Packet" can be used instead of EGP (End Good Packet) whenever you know there is something wrong with the packet. So, if a packet is passing through the fabric and some port notices a problem (i.e. bad CRC), it will end it with EBP instead of EGP. If the packet progress requires store-and-forward, an option would be to just drop it and not waste bandwidth sending EBP packets. The CA that reports this error is NOT where the corruption occurred. It occurred elsewhere in the fabric.

SymbolErrors

The interpretation of symbols within the packet is done on the HCA/CA. If the translation or interpretation fails, it creates a minor event called a symbol error. 99% of all !SymbolErrors are hardware related. If the counts are small (small being a relative term that is up for interpretation) they can be ignored. If the numbers are large and/or the same CA is reporting this error regularly it should be looked into. On a node, the HCA and/or cable should be reseated. If the reseat is unsuccessful it should be replaced. On a switch, reseat the cable or replace the cable.

VL15Drop

VL15 is the default virtual lane for management packets. They are the first to be dropped when there are resource limitations on the port. This is usually related to not enough space in the buffers. In many instances these errors can be ignored. There have been instances, however, when these messages were very closely correlated to user problems in time and fabric space. Obviously, if they are being dropped the buffers are being kept very busy with other data and therefore could indicate congestion.

XmtDiscards

This counter tracks packets that were discarded instead of transmitted. This usually indicates congestion in the fabric. The CA this packet was supposed to be sent to cannot accept it. After so many retries and/or too many incoming packets, the packet to be transmitted gets dropped. If the fabric is being routed well, without deadlocks or credit loops, these should be transient.

Basic Commands

From OpenFabrics Alliance Wiki

Contents

- 1 IB Basic Commands
  - 1.1 ibstat
  - 1.2 ibv_devinfo
  - 1.3 sminfo
  - 1.4 ping hostname-ib0
  - 1.5 ibping
  - 1.6 ib_tracert

IB Basic Commands

ibstat

```
Displays operational information about one or more InfiniBand network devices. This command displays InfiniBand operation
information pertaining to a specified Host Channel Adapter Device (HCAD). If an HCAD device name is not entered, status f
all available HCADs are displayed. Select a flag to narrow down your search results. You can display specific categories
information, including Node, Port, Interface, and Debug information. You can also choose to display all of the informatio
categories.

Flags:
-d     Displays current debug setting.
-h     Displays ibstat command usage.
-i     Displays network interface information.
-n     Displays IB node information.
-p     Displays IB port information.
-v     Displays all IB device information.
```

Example output:

```
[markus]# ibstat
CA 'mlx4_0'
      CA type: MT26428
      Number of ports: 1
      Firmware version: 2.9.1000
      Hardware version: b0
      Node GUID: 0x0002c903004af586
      System image GUID: 0x0002c903004af589
      Port 1:
              State: Active
              Physical state: LinkUp
              Rate: 40
              Base lid: 250
              LMC: 0
              SM lid: 250
              Capability mask: 0x0251086a
              Port GUID: 0x0002c903004af587
              Link layer: InfiniBand
```

ibv_devinfo

```
Print information about RDMA devices available for use from user space.

Flags:
-d, --ib-dev=DEVICE   use IB device DEVICE (default first device found)
-i, --ib-port=PORT    query port PORT (default all ports)
-l, --list            only list names of RDMA devices
-v, --verbose         print all available
```

Example output:

```
[markus]# ibv_devinfo
hca_id:        mlx4_0
      transport:                      InfiniBand (0)
      fw_ver:                         2.9.1000
      node_guid:                      0002:c903:004a:f586
      sys_image_guid:                 0002:c903:004a:f589
      vendor_id:                      0x02c9
      vendor_part_id:                 26428
      hw_ver:                         0xB0
      board_id:                       MT_0D90110009
      phys_port_cnt:                  1
              port:    1
                      state:                  PORT_ACTIVE (4)
                      max_mtu:        4096 (5)
                      active_mtu:             4096 (5)
                      sm_lid:                 250
                      port_lid:               250
                      port_lmc:               0x00
                      link_layer:             InfiniBand
```

sminfo

```
Query InfiniBand SMInfo attribute.
Optionally set and display the output of a sminfo query in human readable format.
This command indicates where the running SM is.
```

Example output:

```
sminfo: sm lid 250 sm guid 0x2c903004af587, activity count 1828732680 priority 15 state 3 SMINFO_MASTER
```

ping hostname-ib0

ibping

```
ibping uses vendor mads to validate connectivity between IB nodes.
The destination is specified by lid.
On exit, (IP) ping like output is show. ibping is run as client/server. Default is to run as client.
Note also that a default ping server is implemented within the kernel.

Flags:
-c            stop after count packets
-f, --flood   flood destination: send packets back to back without delay
-o, --oui     use specified OUI number to multiplex vendor mads
-S, --Server  start in server mode (do not return)
```

Server: (server has a lid of 250)

```
[markus]# ibping -S
```

Client:

```
[mu0007 ~]# ibping 250
Pong from mu-master.lanl.gov.(none) (Lid 250): time 0.196 ms
Pong from mu-master.lanl.gov.(none) (Lid 250): time 0.102 ms
Pong from mu-master.lanl.gov.(none) (Lid 250): time 0.141 ms
```

ib_tracert

```
Traces the path from Source GID/LID to Destinatino GID/LID
Each hop along the path is displayed until the destination is reached or a hop does not respond.
By using the -m option, multicast path tracing can be performed between source and destination nodes.

Flags:
-n, --no_info                   simple format; don't show additional information
-m                              show the multicast trace of the specified mlid
--node-name-map <node-name-map> Specify a node name map. The node name map file maps GUIDs to more user friendly names.
```

Example output:

```
[root@mu-master markus]# ibtracert 250 199
From ca {0x0002c903004af586} portnum 1 lid 250-250 "mu-master"
[1] -> switch port {0x0002c902004484c0}[36] lid 427-427 "muib91"
[7] -> switch port {0x0002c90200449680}[2] lid 2-2 "muibcore1 Line 32"
[19] -> switch port {0x0002c902004486f0}[32] lid 449-449 "muibcore1 Spine 01"
[3] -> switch port {0x0002c90200421750}[19] lid 65-65 "muibcore1 Line 03"
[1] -> switch port {0x0002c902004492c0}[5] lid 433-433 "muib1"
[14] -> ca port {0x002590ffff167f8d}[1] lid 199-199 "mu0007"
To ca {0x002590ffff167f8c} portnum 1 lid 199-199 "mu0007"
```

Routing Verification Tools

ibutils – e.g. ibdmchk

infiniband-diags – e.g. ibsim, etc.

Dave McMillen

What do you verify?

- Did it work?
- Is it deadlock free?
- Does it distribute routes as expected?
- What happens when pieces break?

Simulate and Automate

- ibsim and friends allow real programs to be used
- Symmetry is very helpful
- opensm produces files with useful information
- ibdmchk is needed to answer tough questions
- Routing “failures” are often position dependent

What If?

- You should really know what the routing engines do
- … But …
- You want to see what a routing engine does with a topology
- You want to know if a change will break things
- You want to know how routes distribute when parallel links are added/removed

Caveats

- Simulation is only for MAD traffic
- Timing under simulation is very different
- Be aware of version differences

Routing Verification Tools

Discussion?

Dave McMillen

IB Multicast

What are the uses?

IPoIB and arp activity

Dave McMillen

What is Multicast?

- One source sending to any number of destinations
- Datagram Service Only
- Normally messages, can be RDMA Write
- Multicast Create/Join required to get subnet routing
- Standard Infiniband packet delivery

What is Different About IB Multicast?

- High performance
- Data loss in fabric only on hardware error (mostly)
- Receipt into QP queues
- Destinations are MLIDs instead of LIDs

Why Use IB Multicast?

- Multiple destinations need the same information
- Status / Statistics updates
- Well known address
- Distributed and/or Parallel Servers
- ARP (specific case of above)
- Fault tolerance
- Potentially deep queues
- Pretty good idea of delivery deadline
- But don’t forget it is datagram service

IB Multicast

Discussion?

Dave McMillen

Partitions

What are they?

What applications need partitions?

Dave McMillen

Partitions and Normal Activity

- P_Key is a 16 bit value specifying a partition
- A collection of endnodes with the same P_Key in their P_Key Tables are referred to as being *members of a partition*, or *in a partition*.
- The high-order bit of the partition key is used to record the type of membership in a partition table: 0 for Limited, and 1 for Full.
- Limited members cannot accept information from other Limited members, but communication is allowed between every other combination of membership types.
- Two P_Keys have special meaning: the default partition key (0xFFFF), and the invalid partition keys (low-order 15 bits are all zero).
- The maximum number of entries the P_Key Table can hold must be >= to one and <=to 65535.
- You might only have one, but there are always partitions.

Partitions and Subnet Management

Every IBA port has a QP dedicated to subnet management. This is QP0. QP0 has special features that make it unique compared to other QPs.

- QP0 is permanently configured for Unreliable Datagram class of service.
- Each port of an IBA device has a QP0 that sends and receives packets.
- QP0 is a member of all partitions (i.e., can accept any packet specifying any partition).
- Only subnet management packets (SMPs) are valid
- Traffic for QP0 (i.e., SMPs) exclusively uses VL15, which is not subject to link-level flow control.

Partitions and General Services

Every IBA channel adapter has a QP dedicated to general fabric services. This is QP1. QP1 has special features that make it unique compared to other QPs.

- QP1 is permanently configured for Unreliable Datagram class of service.
- Each port of an IBA device has a QP1 that sends and receives packets.
- QP1 is a member of all of the port's partitions (i.e., can accept any packet specifying a P_Key contained in the port's P_Key table).
- Only management datagrams (MADs) are valid
- Traffic for QP1 does not use VL15

Where are partitions used?

- VLANs are mapped to partitions
- VLANs only exist under IPoIB
- Isolate different sets of attachments
  1) Grouped by system
  2) Grouped by interface
  3) Arbitrary
- Security in the sense of no accidental connections
- QoS
- Multicast domains are different
- IPoIB bonding in one partition can only be active/passive but if the interfaces are in different partitions it can be active/active. Note individual connections do not use both paths at the same time.

How are partitions defined?

- Overlapping is allowed
- Each partition is defined by either a complete list of all GUIDs participating, or the special keyword “ALL”
- --Pconfig, -P, or partition_config_file option
- Default=0x7fff,ipoib:ALL=full;
- 0 for Limited, and 1 for Full results in 0xffff
- Partition flags are:

1) ipoib - indicates that this partition may be used for IPoIB, as result IPoIB capable MC group will be created.
2) rate=<val> - specifies rate for this IPoIB MC group (default is 3 (10GBps))
3) mtu=<val> - specifies MTU for this IPoIB MC group (default is 4 (2048))
4) sl=<val> - specifies SL for this IPoIB MC group (default is 0)
5) scope=<val> - specifies scope for this IPoIB MC group (default is 2 (link local))

Caveats

- All of the parts you care about need to handle partitions
- Most code has only ever been run in default partition
- VLAN mapping may constrain choices

Partitions

Discussion?

Dave McMillen

User Tools Page

From OpenFabrics Alliance Wiki

Goals

- Build inventory and promote on user tools available
- Discussion on features (what tools offer today, what is required by users)
- Encourage collaborative development

Members

Meetings

- 2015-03-30

InfiniBand Practical Monitoring

Susan Coulter
Los Alamos National Laboratory

High Performance Computing Division
HPC-3 Production Systems
skc@lanl.gov

April 19, 2013

So many choices … so much data …

- ibdiagnet
- ibcheckfabric
- ibqueryerrors
- ibtrackerrors
- saquery
- smpquery
- perfquery
- ibtraceroute
- lbnetdiscover
- OpenSM Console ← future development for LANL

Current LANL implementation

RPM

```
ibmon2-1.0.0-9.noarch
[root@mu-master infiniband]# rpm -ql ibmon2
/etc/cron.d/zz_run_ibmon
/etc/ibmon
/etc/ibmon/README
/etc/ibmon/data
/etc/ibmon/data/grid.dest
/etc/ibmon/grid.pl
/etc/ibmon/grid.sh
/etc/ibmon/ib_cookietrail.sh
/etc/ibmon/ib_ct_helper.pl
/etc/ibmon/ib_links.pl
/etc/ibmon/ib_rosetta.pl
/etc/ibmon/ibmon.sh
/etc/ibmon/ibmon_cleaner.pl
```

cronjob

```
[root@mu-master infiniband]# cat /etc/cron.d/zz_run_ibmon
# $Header:
03,33 * * * * root /etc/ibmon/ibmon.sh &> /dev/null
09,39 * * * * root /etc/ibmon/grid.sh
07 7 * * 0 root /etc/ibmon/ibmon_cleaner.pl
```

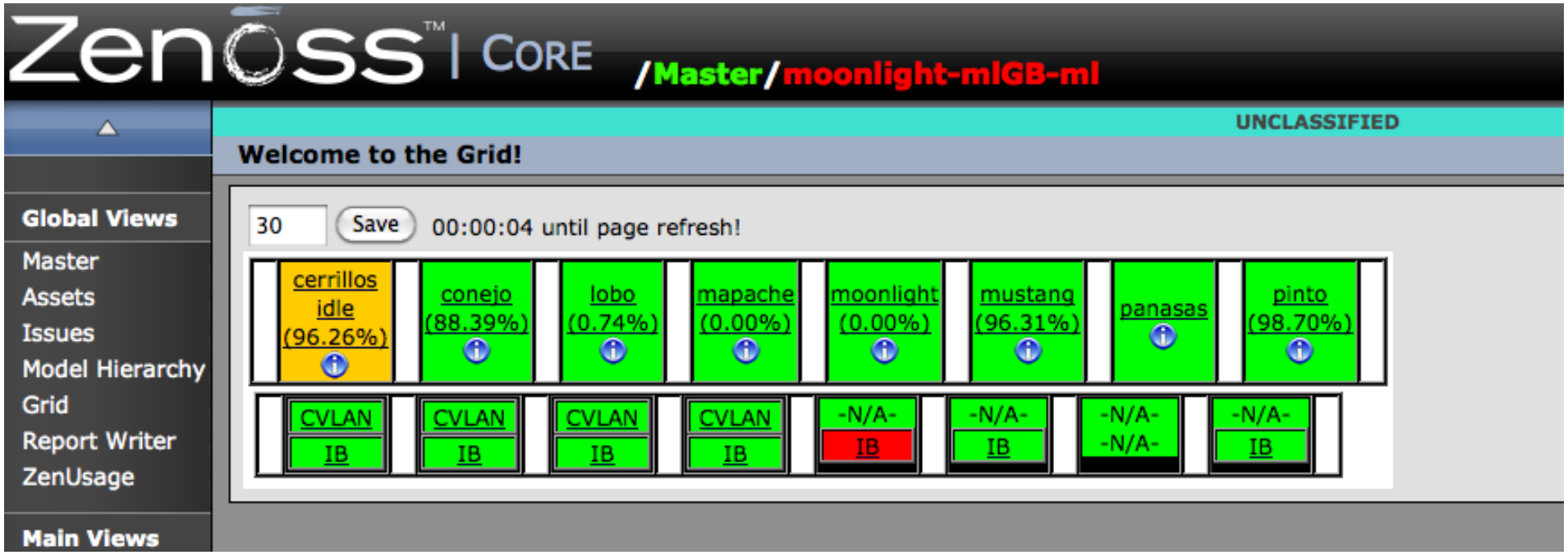
Under the covers of ibmon.sh

```
#!/bin/bash

# new ibnet map
/usr/sbin/ibnetdiscover -g --node-name-map /etc/opensm/ib-node-name-map &> /etc/ibmon/data/ibnet_map
/bin/cp /etc/ibmon/data/ibnet_map /etc/ibmon/data/ibnet_map.`date +%Y%m%d%H%M`

# HCA list with lids
/usr/sbin/ibnetdiscover | egrep "^\[.*H-" | sort -k 4 | cut -d " " -f 1-6 > /etc/ibmon/data/hca_list

# grab counters then reset
/usr/sbin/ibqueryerrors -c -s PortXmitWait | grep -v "##" | grep -v ALL | sed 's/.*GUID .* port/  Port/'
                                                                        | /etc/ibmon/ib_rosetta.pl >& /dev/null
/usr/sbin/ibclearerrors >& /dev/null

# check links
grep -i sdr /etc/ibmon/data/ibnet_map | sed 's/#//' | sed 's/ lid.*//' | /etc/ibmon/ib_links.pl SDR >& /dev/null
grep -i 1x /etc/ibmon/data/ibnet_map | sed 's/#//' | sed 's/ lid.*//' | /etc/ibmon/ib_links.pl 1X >& /dev/null
~
```

Syslog output

```
Apr  2 10:33:04 mu-master ibmon2[12939]: mu1352  Port 1: [PortRcvRemotePhysicalErrors == 1] - (muib77 port 8)
Apr  2 10:33:04 mu-master ibmon2[12939]: muib77  Port 25: [PortRcvRemotePhysicalErrors == 1] - (muibcore3 Line 25 Port 5)
Apr  2 10:33:04 mu-master ibmon2[12939]: muib76  Port 26: [SymbolErrorCounter == 1] [PortRcvErrors == 1] - (mu1343 Port 1)
Apr  2 10:33:04 mu-master ibmon2[12939]: muibcore3 Line 25  Port 4: [PortRcvRemotePhysicalErrors == 1] - (muib76 Port 25)
Apr  2 10:33:23 mu-master ibmon2[14871]: No InfiniBand SDR Link Problems this run
Apr  2 10:33:23 mu-master ibmon2[14875]: No InfiniBand 1X Link Problems this run
Apr  2 11:03:04 mu-master ibmon2[23509]: muib41  Port 16: [SymbolErrorCounter == 1] [PortRcvErrors == 1] - (mu0716 Port 1)
Apr  2 11:03:23 mu-master ibmon2[25946]: No InfiniBand SDR Link Problems this run
Apr  2 11:03:23 mu-master ibmon2[25950]: No InfiniBand 1X Link Problems this run
Apr  2 11:33:04 mu-master ibmon2[9193]: No InfiniBand Errors this run
Apr  2 11:33:23 mu-master ibmon2[11080]: No InfiniBand SDR Link Problems this run
Apr  2 11:33:23 mu-master ibmon2[11084]: No InfiniBand 1X Link Problems this run
Apr  2 12:03:04 mu-master ibmon2[24356]: No InfiniBand Errors this run
Apr  2 12:03:23 mu-master ibmon2[26790]: No InfiniBand SDR Link Problems this run
Apr  2 12:03:23 mu-master ibmon2[26794]: No InfiniBand 1X Link Problems this run
```

Interface with Zenoss

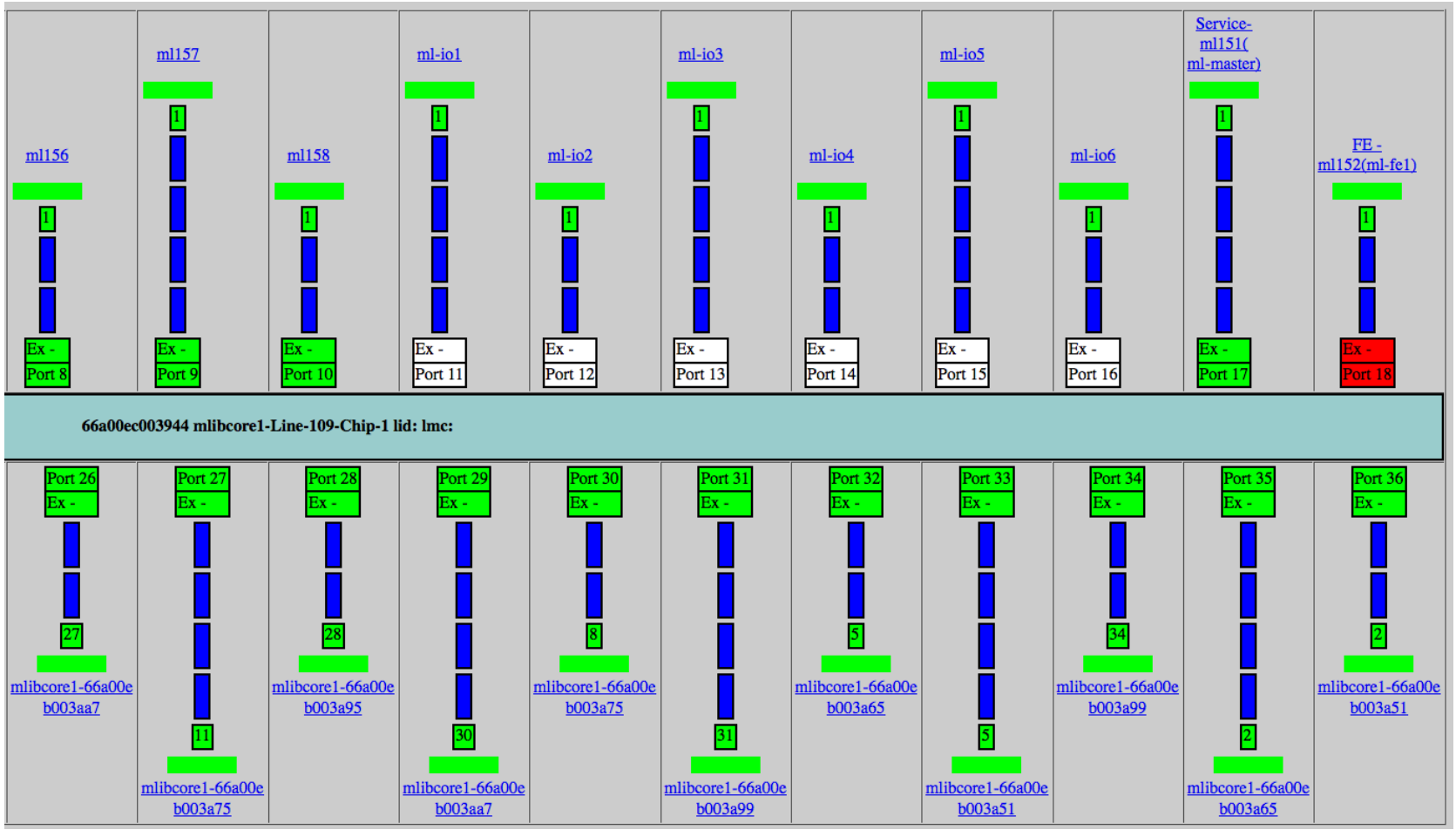


Grid details

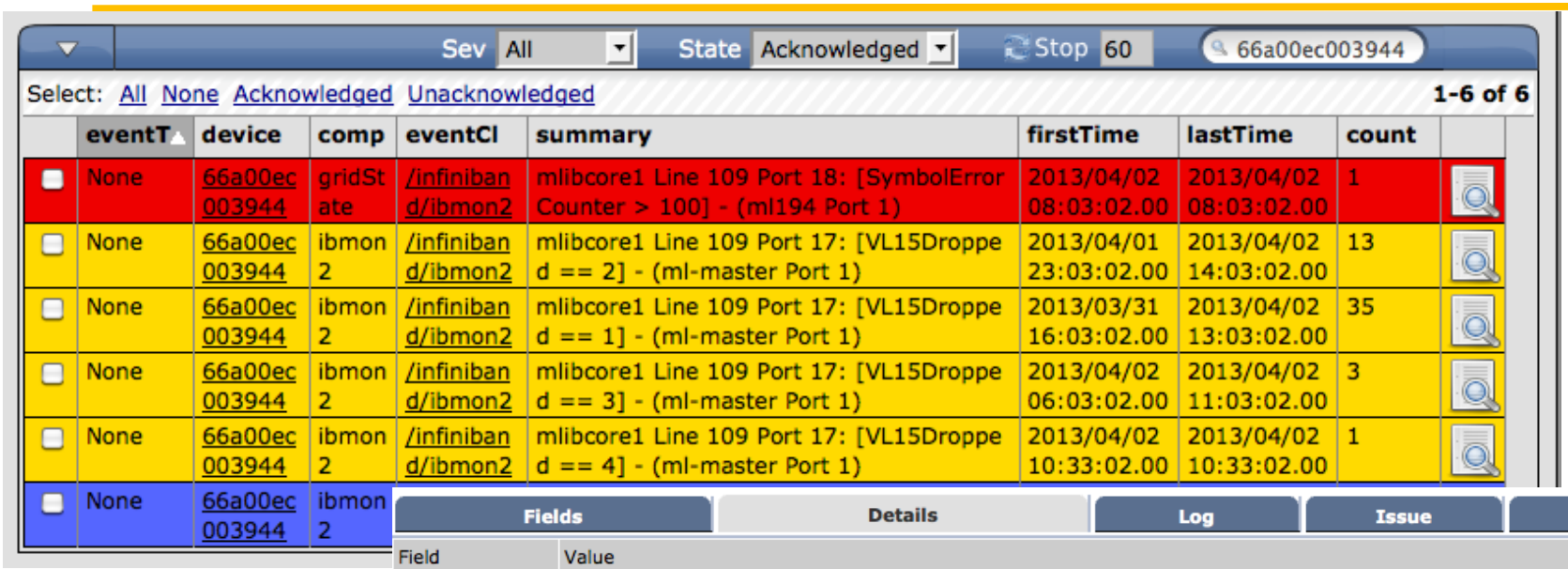
mlibcore1

Spine		Line	
Spine-201A-Chip-1	66a00eb003a51	Line-101-Chip-1	66a02e800132f
Spine-201B-Chip-1	66a00eb003a95	Line-102-Chip-1	66a01e800132f
Spine-203A-Chip-1	66a00eb003a55	Line-103-Chip-1	66a00ec003943
Spine-203B-Chip-1	66a00eb003a99	Line-104-Chip-1	66a00ec00392b
Spine-205A-Chip-1	66a00eb003a75	Line-105-Chip-1	66a00ec00391e
Spine-205B-Chip-1	66a00eb003ab7	Line-106-Chip-1	66a00ec003947
Spine-207A-Chip-1	66a00eb003a65	Line-107-Chip-1	66a00ec003932
Spine-207B-Chip-1	66a00eb003aa7	Line-108-Chip-1	66a00ec00394e
Spine-209A-Chip-1	66a00eb00381f	Line-109-Chip-1	66a00ec003944
		Line-110-Chip-1	66a00ec003919
		Line-111-Chip-1	66a00ec00391f
		Line-112-Chip-1	66a00ec003921
		Line-113-Chip-1	66a00ec003909
		Line-114-Chip-1	66a00ec00393b
		Line-115-Chip-1	66a00ec003945
		Line-116-Chip-1	66a00ec003925
		Line-117-Chip-1	66a00ec003934
		Line-118-Chip-1	66a00ec003940

More grid details



Message as displayed by Zenoss



Fields	Details	Log	Issue	Rollup

Field	Value
explanation	If SymbolErrorCounter > 100 Then email system oncall and hpcnet-day-oncall, turn grid red. Regex catches numbers >= 100. Trigger command(resendMessageForError) to resend a message to trigger filter to increment count for this local device(catchLocalDevi
LLABEL	mlibcore1 Line 109
LLOC	18
LOCAL_DEVICE	mlibcore1 Line 109 Port 18
MESG	[SymbolErrorCounter == 65535] [LinkDownedCounter == 1]
NAME	SymbolErrorCounter
originalTime	Apr 2 08:03:02
pid	89835
REMOTE_DEVICE	ml194 Port 1
SEC	65535

Wiki pages / documentation …

AWOL Link

This is not a counter, but an error as identified by LANL processes. When ibnetdiscover is run, any link that is *live* but not responding throws an error. ibmon and logged to syslog.

```
Example:

Aug  9 12:33:15 mu-master ibmon2[32424]: AWOL Link: (DR path slid 0; dlid 0; 0,1,1,20,25,16,34 Attr 0x11:0)

[root@mu-master ~]# smpquery portinfo -D 0,1,1,20,25,16,34
This should result in an error.  If it does not you will see information like what is in the next example.
If this works - it means the node/port was probably coming up but not yet able to respond to a MAD packet.
If it fails, remove the last port number and run again, grep'ing for the LID.

[root@mu-master ~]# smpquery pi -D 0,1,1,20,25,16,34 | grep -i lid
# Port info: DR path slid 65535; dlid 65535; 0,1,1,20,25,16,34 port 0
Lid:.............................1095
SMLid:...........................250
```

-- SymbolErrorCounter

```
Mmm Dd Hh:Mm:Ss Host      Class         Local Side:     Error                                                    - Remote Side
-------------------------------------------------------------------------------------------------------------------------------
May 25 08:03:03 mu-master ibmon2[15906]: mu1246  Port 1: [SymbolErrorCounter == 1] [PortRcvErrors == 1] - (ib71 port 4)
```

Action: If SymbolErrorCounter > 100 Then email system oncall and hpcnet-day-oncall, turn grid red. Trigger command(resendMessageForError) to resend a message to trigger filter to increment count for this local device(catchLocalDevice)

Regex: (.*): .*[SymbolErrorCounter == (.*)] .* - \((.*)\) Where $1 is local device, $2 is SymbolErrorCounter value, $3 is remote device

Regex(Python): (?P<LOCAL_DEVICE>(?P<LLABEL>[A-Za-z0-9]+)\s*(?P<LLOC>.*)): (?P<MESG>.*\[(?P<NAME>SymbolErrorCounter) == (?P<SEC>\d{3,})\].*) - \((?P<REMOTE_DEVICE>.*)\)

End

Questions?