Screencast: Openib BTL v1.3 Sneak Peak

Jeff Squyres
May 2008
v1.3 Upcoming Features

• This presentation is a “sneak peak”
  ▪ …and is therefore subject to change
  ▪ These slides show what is **likely** to be included
  ▪ **But nothing is definite until v1.3 ships 😊**

• Features shown here are in addition to all the other Goodness coming in v1.3…
  ▪ Performance improvements
  ▪ Tool integration
  ▪ …much more
New Hardware Support

- iWARP supported
  - Tested with Chelsio T3 adapters
- Support for Mellanox ConnectX XRC
  - Reduce number of QPs, increase performance
- OpenFabrics Connection Managers
  - RDMA CM: works with both IB and iWARP
  - IB CM: “better” connection wireup over IB
v1.2 Long Message Params

- \text{eager\_limit}
- \text{min\_rdma\_size}
- Fragments of size up to \text{max\_send\_size}
- Fragments of size up to \text{max\_rdma\_size}
v1.3 Long Message Params

- eager_limit
- Fragments of size up to max_send_size
- Fragments of size up to rdma_pipeline_frag_size
- rdma_pipeline_send_length
Include / Exclude Interfaces

- if_include / if_exclude
  - Comma-delimited list of devices / ports to use or not use

```
mpirun --mca btl_openib_if_include \mthca0:1,mthca1 ...
```

```
mpirun --mca btl_openib_if_exclude \mthca0 ...
```
New Receive Queue System: “Bucket” SRQ (BSRQ)

- Based on idea from Cray Portals
  - Different SRQ message sizes allow for much more efficient use of registered memory
  - BSRQ + XRC = fewer QPs, better memory utilization = better performance

 Lots of small message buffering

 Multiple medium size bufferings (all smaller than RDMA/large messages)
Specifying the BSRQ List

- **receive_queues:**
  - Comma-delimited list of RQs for each peer
  - Specifying queue sizes and types for “smaller than large” (RDMA) messages
  - Replaces “use_srq” and “rd_num” (and others)

- Default value for some IB HCAs
  - P,128,256,192,128:S,2048,256,128,32:
  - S,12288,256,128,32:S,65536,256,128,32
BSRQ Parameter List

• **P**: Per-peer queues (precious)
  - Size of buffers
  - Number of buffers
  - *Optional*: Low watermark buffer count
  - *Optional*: Credit window size
  - *Optional*: Credit “reserve” buffers

• **S**: Shared receive queues
  - Size of buffers
  - Number of buffers
  - *Optional*: Low watermark buffer count
  - *Optional*: Max number of outstanding sends
Flow Control

• IB/iWARP are “lossless” networks
  ▪ Must have [hardware] credits to send
  ▪ However, receivers can still be overwhelmed
  ▪ Packets can be dropped due to congestion
  ▪ Or receivers might not be ready (not enough posted receiver buffers)

• Open MPI has software flow control
  ▪ Explicit FC for per-peer receive queues
  ▪ Implicit FC for SRQs (relies on RNR; excellent performance when SRQ not filled)

• Sum of all “reserve” buffers added to smallest PP QP for flow control messages
Small Message Coalescing

- `use_message_coalescing`:
  - Boolean enabling small message coalescing
- Defaults to 1
  - Only effective if sending many short messages of same MPI signature very rapidly (i.e., faster than HCA can transmit)
  - Some benchmarks show performance gain
  - Only applicable to some real-world apps
NUMA-Aware Device Selection

• In NUMA architectures (e.g., AMD servers)
  ▪ Choose the HCAs / NICs that are “closest”
  ▪ Prevents crossing extra busses
  ▪ Makes the most sense when enabled with processor affinity

• NUMA architecture specified by text config file
  ▪ Can “fake” a NUMA configuration to share devices in high-core count servers
More Information

- Open MPI FAQ
  - General tuning
    [http://www.open-mpi.org/faq/?category=tuning](http://www.open-mpi.org/faq/?category=tuning)
  - OpenFabrics tuning
    [http://www.open-mpi.org/faq/?category=openfabrics](http://www.open-mpi.org/faq/?category=openfabrics)