Optimization and Tuning of Hybrid, Multirail, 3D Torus Support and QoS in MVAPICH2

MVAPICH2 User Group (MUG) Meeting

by

Hari Subramoni
The Ohio State University
E-mail: subramon@cse.ohio-state.edu
http://www.cse.ohio-state.edu/~subramon
Outline

• **Memory overheads in large scale systems**
• Optimizations in MVAPICH2 to address overheads
• Multirail Clusters
• 3D Torus Support
• Quality of Service
Memory overheads in large-scale systems

- Reliable Connection (RC) is the most common transport protocol in IB
- Connections need to be established between every pair of processes
  - Each connection requires a certain amount of memory for handling related data structures
  - Memory required for all connections can increase with system size
- Buffers need to be posted at each receiver to receive messages from any sender
  - Buffer requirement can increase with system size
- Both issues have become critical as large-scale IB deployments have taken place
  - Being addressed by both IB specification and upper-level middleware
- IB offers alternate mechanisms and transport protocols for scalability (XRC, UD, SRQ)
Outline

- Memory overheads in large scale systems
- Optimizations in MVAPICH2 to address overheads
  - Shared Receive Queue (SRQ)
    - Hybrid Communication Channels
      - eXtended Reliable Connection (XRC)
      - Unreliable Datagram (UD) Transport
      - Integrated Hybrid UD-RC/XRC Design
- Multirail Clusters
- 3D Torus Support
- Quality of Service
Optimizations in MVAPICH2 to address overheads

- MVAPICH2 has optimizations to address these overheads
  - Support for
    - Shared Receive Queue
      - Enables use of common buffer pool for receiving data
    - eXtended Reliable Connect transport protocol
      - Reduce number of connections
    - UD transport protocol
      - Reduce QP cache trashing
      - Only requires one connection
  - Hybrid of RC, XRC and UD gives best of everything
### 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
- \[ 0 < Q << P \times ((M \times N) - 1) \]
Using Shared Receive Queues with MVAPICH2

- SRQ reduces the memory used by $\frac{1}{6}$th at 64,000 processes

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Significance</th>
<th>Default</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MV2_USE_SRQ</td>
<td>• Enable / Disable use of SRQ in MVAPICH2</td>
<td>Enabled</td>
<td>• Always Enable</td>
</tr>
<tr>
<td>MV2_SRQ_MAX_SIZE</td>
<td>• Limits the maximum size of the SRQ</td>
<td>4096</td>
<td>• Increase to 8192 for large scale runs</td>
</tr>
<tr>
<td>MV2_SRQ_SIZE</td>
<td>• Number of buffers posted to the SRQ</td>
<td>256</td>
<td>• Upper Bound: MV2_SRQ_MAX_SIZE</td>
</tr>
<tr>
<td></td>
<td>• Automatically doubled by MVAPICH2 on receiving SRQ LIMIT EVENT from IB HCA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Refer to Shared Receive Queue (SRQ) Tuning section of MVAPICH2 user guide for more information
- [http://mvapich.cse.ohio-state.edu/support/user_guide_mvapich2-2.0a.html#x1-1010008.5]
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
• Memory usage for 32K processes with 8-cores per node can be 54 MB/process (for connections)
• NAMD performance improves when there is frequent communication to many peers
• Enabled by setting MV2_USE_XRC to 1 (Default: Disabled)
• Requires OFED version > 1.3
  – Unsupported in earlier versions (< 1.3), OFED-3.x and MLNX_OFED-2.0
  – MVAPICH2 build process will automatically disable XRC if unsupported by OFED
• Automatically enables SRQ and ON-DEMAND connection establishment

• Refer to eXtended Reliable Connection (XRC) section of MVAPICH2 user guide for more information
  • http://mvapich.cse.ohio-state.edu/support/user_guide_mvapich2-2.0a.html#x1-1020008.6
Unreliable Datagram (UD) Transport

• Connectionless unreliable communication
• Avoid QP trashing
• Light weight reliability layer in the library
• Zero-copy large message transfer
Using UD Transport with MVAPICH2

Memory Footprint of MVAPICH2

<table>
<thead>
<tr>
<th>Number of Processes</th>
<th>Conn.</th>
<th>Buffers</th>
<th>Struct</th>
<th>Total</th>
<th>Buffers</th>
<th>Struct</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>512</td>
<td>22.9</td>
<td>24</td>
<td>0.3</td>
<td>47.2</td>
<td>24</td>
<td>0.2</td>
<td>24.2</td>
</tr>
<tr>
<td>1024</td>
<td>29.5</td>
<td>24</td>
<td>0.6</td>
<td>54.1</td>
<td>24</td>
<td>0.4</td>
<td>24.4</td>
</tr>
<tr>
<td>2048</td>
<td>42.4</td>
<td>24</td>
<td>1.2</td>
<td>67.6</td>
<td>24</td>
<td>0.9</td>
<td>24.9</td>
</tr>
</tbody>
</table>

Performance with SMG2000

- Can use UD transport by configuring MVAPICH2 with the `--enable-hybrid`
  - Reduces QP cache trashing and memory footprint at large scale

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Significance</th>
<th>Default</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MV2_USE_ONLY_UD</td>
<td>• Enable only UD transport in hybrid configuration mode</td>
<td>Disabled</td>
<td>• RC/XRC not used</td>
</tr>
<tr>
<td>MV2_USE_UD_ZCOPY</td>
<td>• Enables zero-copy transfers for large messages on UD</td>
<td>Enabled</td>
<td>• Always Enable when UD enabled</td>
</tr>
<tr>
<td>MV2_UD_RETRY_TIMEOUT</td>
<td>• Time (in usec) after which an unacknowledged message will be retried</td>
<td>500000</td>
<td>• Increase appropriately on large / congested systems</td>
</tr>
<tr>
<td>MV2_UD_RETRY_COUNT</td>
<td>• Number of retries before job is aborted</td>
<td>1000</td>
<td>• Increase appropriately on large / congested systems</td>
</tr>
</tbody>
</table>

- Refer to Running with scalable UD transport section of MVAPICH2 user guide for more information
- [http://mvapich.cse.ohio-state.edu/support/user_guide_mvlapich2-2.0a.html#x1-640006.11](http://mvapich.cse.ohio-state.edu/support/user_guide_mvlapich2-2.0a.html#x1-640006.11)
Hybrid (UD/RC/XRC) Mode in MVAPICH2

- Both UD and RC/XRC have benefits
  - Hybrid for the best of both
- Enabled by configuring MVAPICH2 with the `--enable-hybrid`
- Available since MVAPICH2 1.7 as integrated interface

### Performance with HPCC Random Ring

<table>
<thead>
<tr>
<th>Number of Processes</th>
<th>Time (us)</th>
<th>UD</th>
<th>Hybrid</th>
<th>RC</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>26%</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>256</td>
<td>40%</td>
<td>4</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>512</td>
<td>38%</td>
<td>4</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>1024</td>
<td>30%</td>
<td>6</td>
<td>12</td>
<td>2</td>
</tr>
</tbody>
</table>

### Parameter Significance

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Significance</th>
<th>Default</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MV2_USE_UD_HYBRID</td>
<td>• Enable / Disable use of UD transport in Hybrid mode</td>
<td>Enabled</td>
<td>• Always Enable</td>
</tr>
<tr>
<td>MV2_HYBRID_ENABLE_THRESHOLD_SIZE</td>
<td>• Job size in number of processes beyond which hybrid mode will be enabled</td>
<td>1024</td>
<td>• Uses RC/XRC connection until job size &lt; threshold</td>
</tr>
<tr>
<td>MV2_HYBRID_MAX_RC_CONN</td>
<td>• Maximum number of RC or XRC connections created per process&lt;br&gt;• Limits the amount of connection memory</td>
<td>64</td>
<td>• Prevents HCA QP cache thrashing</td>
</tr>
</tbody>
</table>

- Refer to Running with Hybrid UD-RC/XRC section of MVAPICH2 user guide for more information
- [http://mvapich.cse.ohio-state.edu/support/user_guide_mvapich2-2.0a.html#x1-650006.12](http://mvapich.cse.ohio-state.edu/support/user_guide_mvapich2-2.0a.html#x1-650006.12)
Outline

- Memory overheads in large scale systems
- Optimizations in MVAPICH2 to address overheads
  - Multirail Clusters
  - 3D Torus Support
  - Quality of Service
MVAPICH2 Multi-Rail Design

- What is a rail?
  - HCA, Port, Queue Pair

- Automatically detects and uses all active HCAs in a system
  - Automatically handles heterogeneity

- Supports multiple rail usage policies
  - Rail Sharing – Processes share all available rails
  - Rail Binding – Specific processes are bound to specific rails
Performance Tuning on Multi-Rail Clusters

Impact of Default Message Stripping on Bandwidth

Impact of Default Rail Binding on Message Rate

Impact of Advanced Multi-rail Tuning on Message Rate

Two 24-core Magny Cours nodes with two Mellanox ConnectX QDR adapters
Six pairs with OSU Multi-Pair bandwidth and messaging rate benchmark

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Significance</th>
<th>Default</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MV2_IBA_HCA</td>
<td>• Manually set the HCA to be used</td>
<td>Unset</td>
<td>• To get names of HCA ibstat</td>
</tr>
<tr>
<td>MV2_DEFAULT_PORT</td>
<td>• Select the port to use on a active multi port HCA</td>
<td>0</td>
<td>• Set to use different port</td>
</tr>
<tr>
<td>MV2_RAIL_SHARING_LARGE_MSG_THRESHOLD</td>
<td>• Threshold beyond which striping will take place</td>
<td>16 Kbyte</td>
<td></td>
</tr>
</tbody>
</table>
| MV2_RAIL_SHARING_POLICY | • Choose multi-rail rail sharing / binding policy
• For Rail Sharing set to USE_FIRST or ROUND_ROBIN
• Set to FIXED_MAPPING for advanced rail binding options | Rail Binding in Round Robin mode | • Advanced tuning can result in better performance |
| MV2_PROCESS_TO_RAIL_MAPPING | • Determines how HCAs will be mapped to the rails | BUNCH | • Options: SCATTER and custom list |

• Refer to Enhanced design for Multiple-Rail section of MVAPICH2 user guide for more information

• [http://mvapich.cse.ohio-state.edu/support/user_guide_mvapich2-2.0a.html#x1-670006.14](http://mvapich.cse.ohio-state.edu/support/user_guide_mvapich2-2.0a.html#x1-670006.14)
Outline

• Memory overheads in large scale systems
• Optimizations in MVAPICH2 to address overheads
• Multirail Clusters
  • 3D Torus Support
• Quality of Service
Support for 3D Torus Networks in MVAPICH2

- Deadlocks possible with common routing algorithms in 3D Torus InfiniBand networks
  - Need special routing algorithm for OpenSM
- Users need to interact with OpenSM
  - Use appropriate SL to prevent deadlock
- MVAPICH2 supports 3D Torus Topology
  - Queries OpenSM at runtime to obtain appropriate SL
- Usage
  - Enabled at configure time
    - `--enable-3dtorus-support`
  - `MV2_NUM_SA_QUERY_RETRIES`
    - Control number of retries if PathRecord query fails
Outline

- Memory overheads in large scale systems
- Optimizations in MVAPICH2 to address overheads
- Multirail Clusters
- 3D Torus Support
- **Quality of Service**
Exploiting QoS Support in MVAPICH2

- IB is capable of providing network level differentiated service – QoS
- Uses Service Levels (SL) and Virtual Lanes (VL) to classify traffic
- Enabled at configure time using CFLAG ENABLE_QOS_SUPPORT
- Check with System administrator before enabling
  - Can affect performance of other jobs in system

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Significance</th>
<th>Default</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MV2_USE_QOS</td>
<td>• Enable / Disable use QoS</td>
<td>Disabled</td>
<td>• Check with System administrator</td>
</tr>
<tr>
<td>MV2_NUM_SLS</td>
<td>• Number of Service Levels user requested</td>
<td>8</td>
<td>• Use to see benefits of Intra-Job QoS</td>
</tr>
<tr>
<td>MV2_DEFAULT_SERVICE_LEVEL</td>
<td>• Indicates the default Service Level to be used by job</td>
<td>0</td>
<td>• Set to different values for different jobs to enable Inter-Job QoS</td>
</tr>
</tbody>
</table>

- How can QoS be used to isolate Checkpoint Restart traffic from Application Traffic ???
- “Fault-Tolerance Support (CR, SCR and Migration) in MVAPICH2”; 11:50 – 12:20; Tuesday, August 27th

MVAPICH2 User Group Meeting 2013
Web Pointers

NOWLAB Web Page
http://nowlab.cse.ohio-state.edu

MVAPICH Web Page
http://mvapich.cse.ohio-state.edu