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 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 message 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)

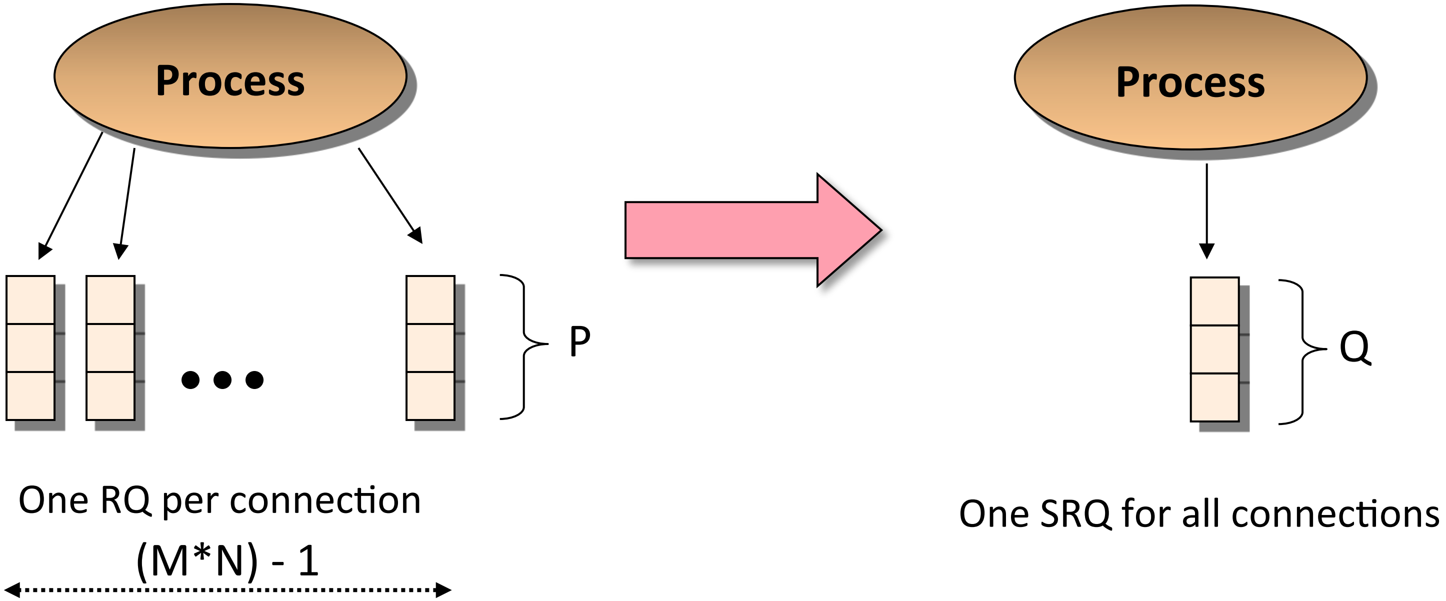
Outline

- 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

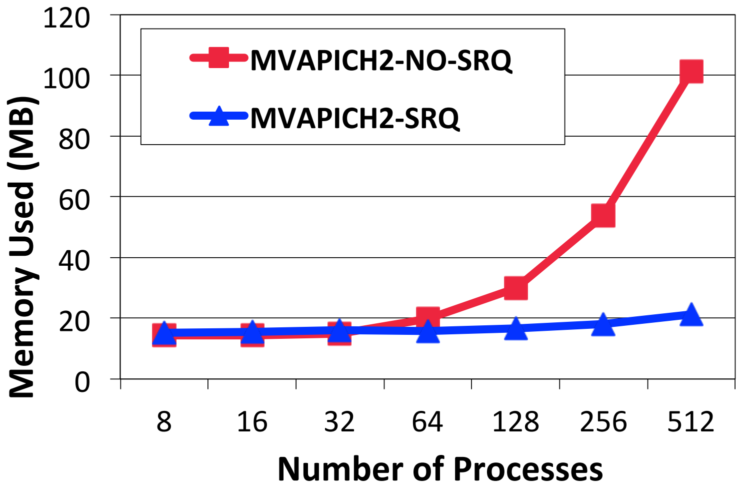
- MVAPICH2 has optimizations to address these overheads
  - Support for
    - Shared Receive Queue
      - Enables use of common buffer pool for receiving data
    - eXtended Reliable Connect transport protocol
      - Reduce number of connections
    - UD transport protocol
      - Reduce QP cache trashing
      - Only requires one connection
- Hybrid of RC, XRC and UD gives best of everything

Shared Receive Queue (SRQ)

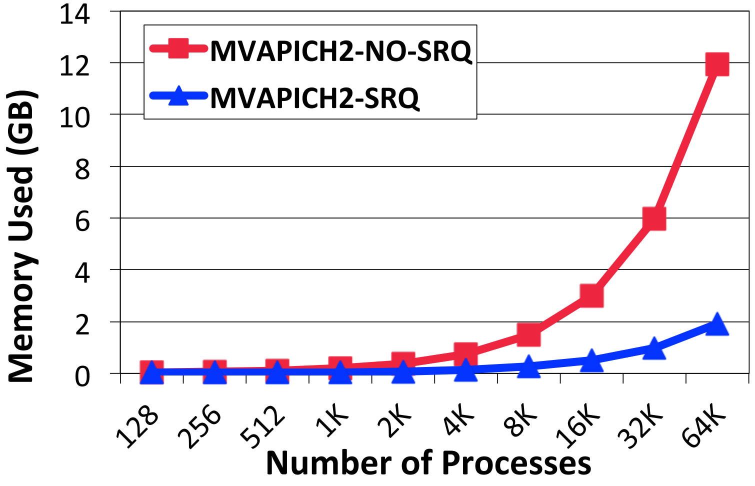


- SRQ is a hardware mechanism for a process to share receive resources (memory) across multiple connections
  - Introduced in specification v1.2
- 0 < Q << P*((M*N)-1)

Using Shared Receive Queues with MVAPICH2



MPI_Init memory utilization



Analytical model

- **SRQ reduces the memory used by 1/6th at 64,000 processes**

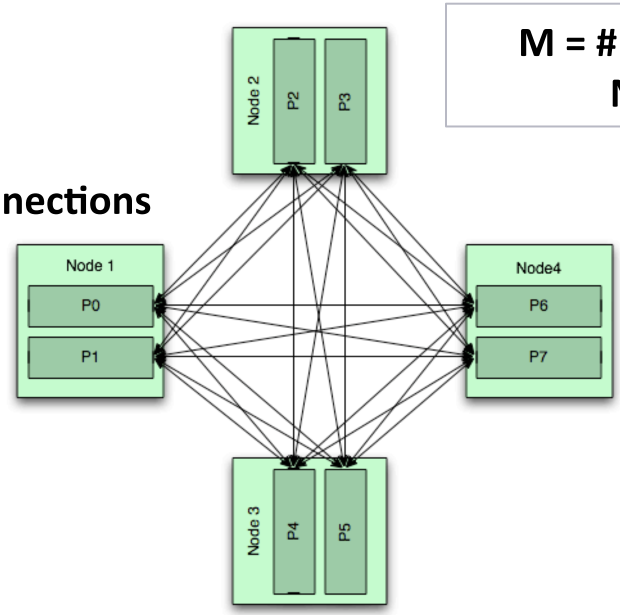
Parameter	**Significance**	**Default**	**Notes**
MV2_USE_SRQ	• Enable / Disable use of SRQ in MVAPICH2	Enabled	• Always Enable
MV2_SRQ_MAX_SIZE	• Limits the maximum size of the SRQ • Places upper bound on amount of memory used for SRQ	4096	• Increase to 8192 for large scale runs
MV2_SRQ_SIZE	• Number of buffers posted to the SRQ • Automatically doubled by MVAPICH2 on receiving SRQ LIMIT EVENT from IB HCA	256	• Upper Bound: MV2_SRQ_MAX_SIZE

- **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)

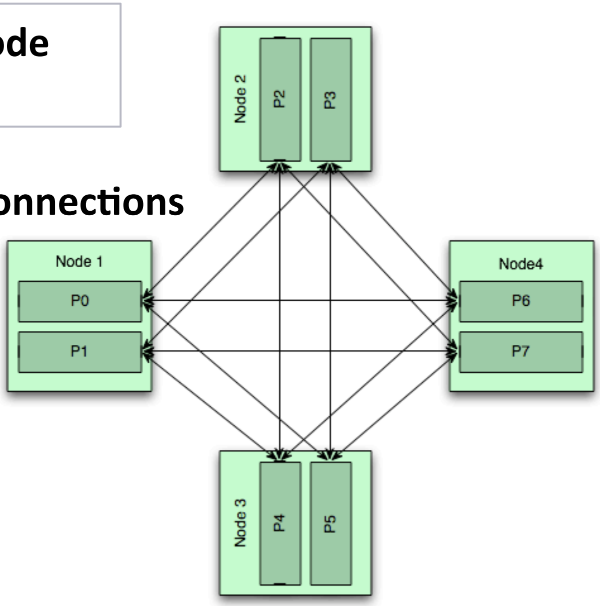
M = # of processes/node
N = # of nodes

RC Connections



M^2 x (N – 1) connections/node

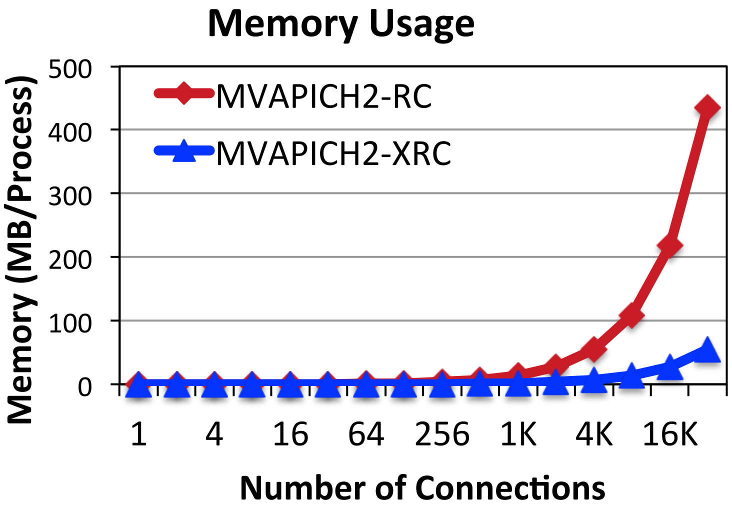
XRC Connections

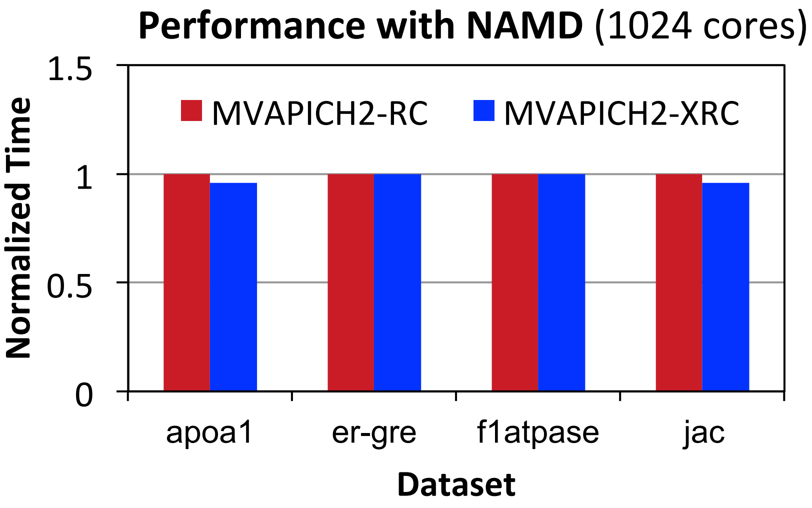


M x (N – 1) connections/node

- 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

Using eXtended Reliable Connection (XRC) in MVAPICH2





- 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

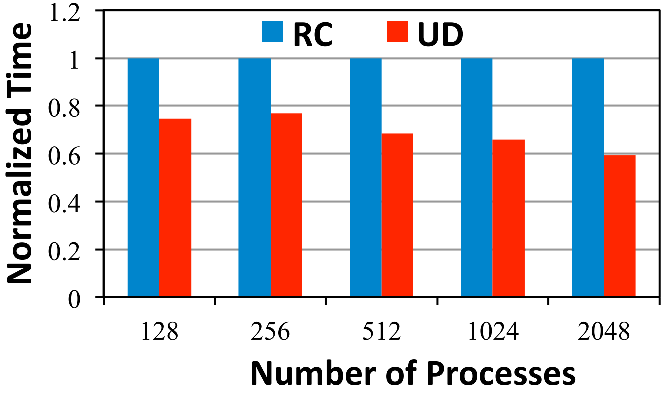
- Connectionless unreliable communication
- Avoid QP trashing
- Light weight reliability layer in the library
- Zero-copy large message transfer

Using UD Transport with MVAPICH2

Memory Footprint of MVAPICH2

	RC (MVAPICH2 2.0a)				UD (MVAPICH2 2.0a)		
Number of Processes	**Conn.**	**Buffers**	**Struct**	**Total**	**Buffers**	**Struct**	**Total**
512	22.9	24	0.3	47.2	24	0.2	24.2
1024	29.5	24	0.6	54.1	24	0.4	24.4
2048	42.4	24	1.2	67.6	24	0.9	24.9

Performance with SMG2000

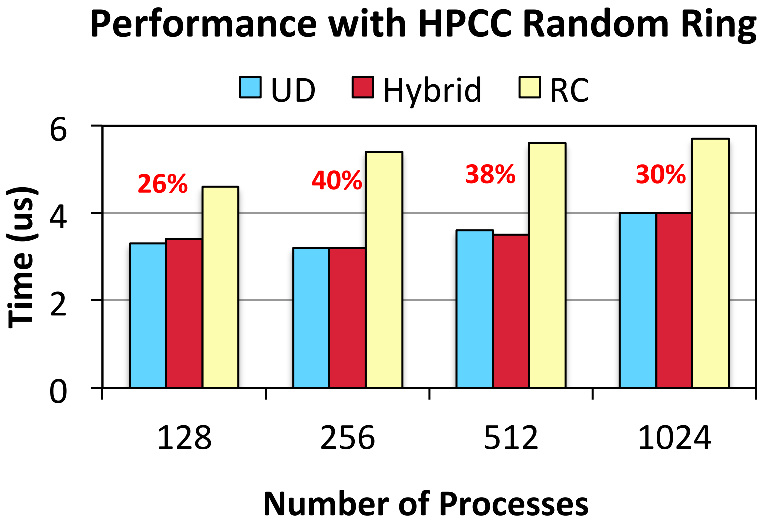


- Can use UD transport by configuring MVAPICH2 with the **–enable-hybrid**
  - Reduces QP cache trashing and memory footprint at large scale

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

- **Refer to Running with scalable UD transport section of MVAPICH2 user guide for more information**
- **http://mvapich.cse.ohio-state.edu/support/user_guide_mvapich2-2.0a.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

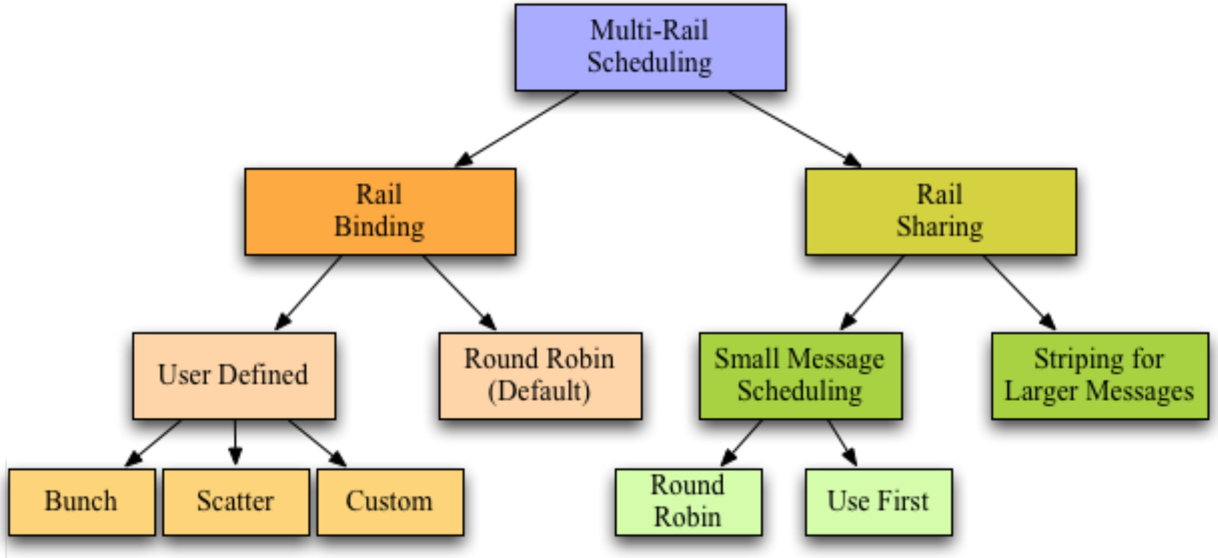
Parameter	Significance	Default	Notes
MV2_USE_UD_HYBRID	• Enable / Disable use of UD transport in Hybrid mode	Enabled	• Always Enable
MV2_HYBRID_ENABLE_THRESHOLD_SIZE	• Job size in number of processes beyond which hybrid mode will be enabled	1024	• Uses RC/XRC connection until job size < threshold
MV2_HYBRID_MAX_RC_CONN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

- **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**

Outline

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

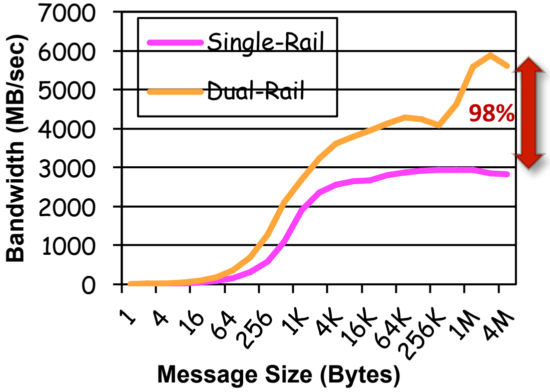
MVAPICH2 Multi-Rail Design



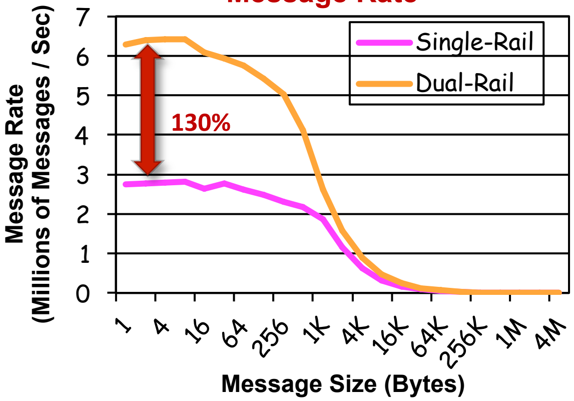
- What is a rail?
  - **HCA, Port, Queue Pair**
- Automatically detects and uses all active HCAs in a system
  - Automatically handles heterogeneity
- Supports multiple rail usage policies
  - Rail Sharing – Processes share all available rails
  - Rail Binding – Specific processes are bound to specific rails

Performance Tuning on Multi-Rail Clusters

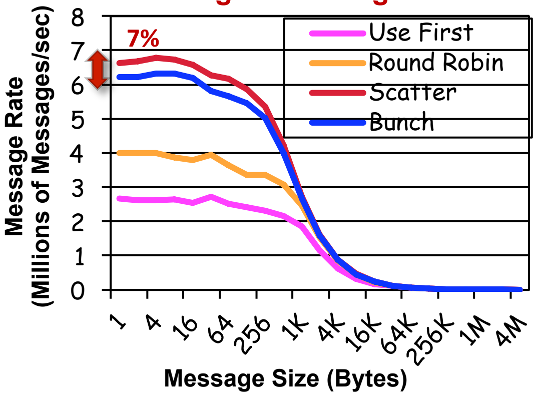
Impact of Default Message Striping on Bandwidth



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

Parameter	Significance	Default	Notes
MV2_IBA_HCA	• Manually set the HCA to be used	Unset	• To get names of HCA ibstat | grep "^CA"
MV2_DEFAULT_PORT	• Select the port to use on a active multi port HCA	0	• Set to use different port
MV2_RAIL_SHARING_LARGE_MSG_THRESHOLD	• Threshold beyond which striping will take place	16 Kbyte	
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**

Outline

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

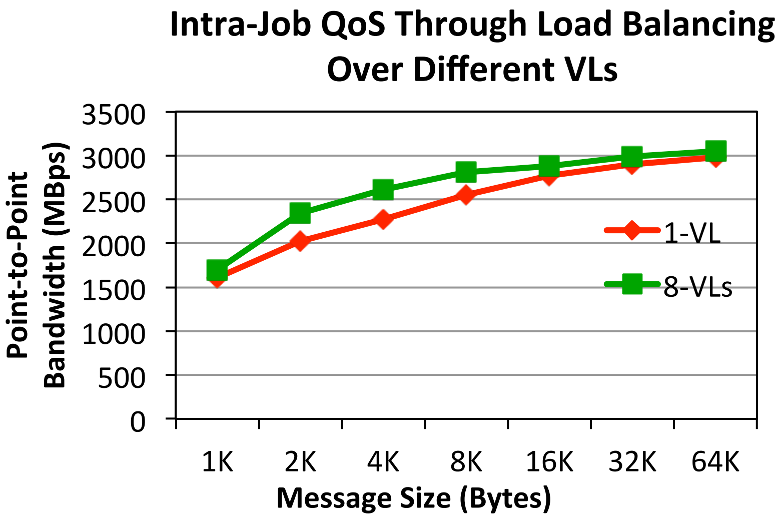
Support for 3D Torus Networks in MVAPICH2

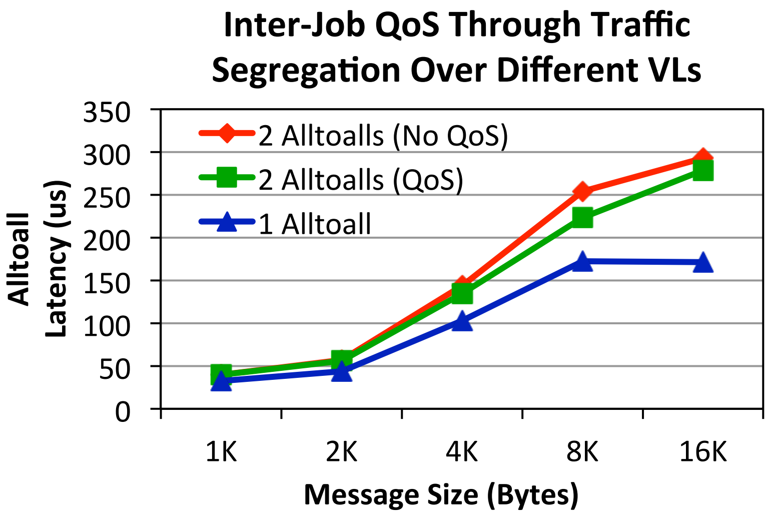
- Deadlocks possible with common routing algorithms in 3D Torus InfiniBand networks
  - Need special routing algorithm for OpenSM
- Users need to interact with OpenSM
  - Use appropriate SL to prevent deadlock
- MVAPICH2 supports 3D Torus Topology
  - Queries OpenSM at runtime to obtain appropriate SL
- 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

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

- **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**

Web Pointers

NOWLAB Web Page

http://nowlab.cse.ohio-state.edu

MVAPICH Web Page

http://mvapich.cse.ohio-state.edu

MVAPICH