

---

4e-4g: Scalability Issues  
788.08P Winter 2008

Jaidev Sridhar

---

## Papers

---

- S. Sur, M. Koop, and D. K. Panda, High-Performance and Scalable MPI over InfiniBand with Reduced Memory Usage: An In-Depth Performance Analysis
  - M. Koop, S. Sur and D. K. Panda, Zero-Copy Protocol for MPI using InfiniBand Unreliable Datagram
  - M. Koop, T. Jones and D. K. Panda, MVAPICH-Aptus: Scalable High-Performance Multi-Transport MPI over InfiniBand
- 

## Introduction

---

- In-depth analysis of memory utilization for communication
  - Eager protocol
    - Low latency for small messages
  - Rendezvous protocol
    - Avoid buffering large messages
    - Control messages over Eager protocol
  - Only eager protocol needs additional communication buffers
- 

## Communication Buffers

---

- Buffers pre-posted for low latency
    - Pinned to physical memory
  - Send / Receive channels – buffers posted on QP at sender and receiver
    - SRQ can be used to reduce number of buffers
  - RDMA channels for eager protocol
    - Not zero-copy, but low latency due to hardware
- 

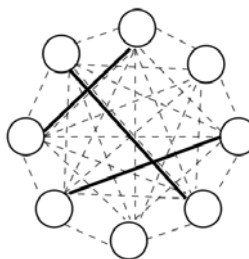
## Eager Protocol Implementation

---

- Goals: Low latency, reduced memory utilization
  - Notes:
    - Not all pairs communicate with each other (frequently)
    - Some pairs communicate frequently
  - Implementations
    - Adaptive RDMA with Send/Receive Channel
    - Adaptive RDMA with SRQ Channel
    - SRQ Channel
- 

## ARDMA-SR

---

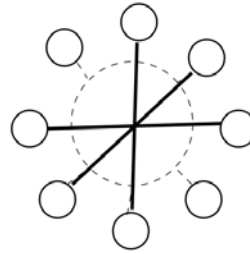


- Init: SR channel with limited number of buffers per remote process
  - Switch to RDMA write channel when message count breaches threshold
  - RDMA buffers made available for every process that may send a message
-

## SRQ

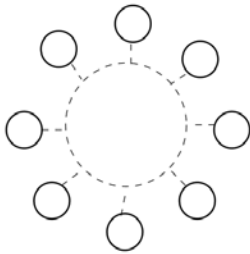
- Hardware feature – allows receive buffers to be shared on a single receive queue
- Buffers consumed in FCFS order
- No flow control
- MVAPICH –
  - Receiver driven “low-watermark” based flow control
  - Post additional buffers when triggered by SRQ\_LIMIT\_REACHED interrupt

## ARDMA-SRQ



- Init: All processes have full SRQs
- When certain number of buffers are consumed, RDMA channel created for the sender
- Similar to ARDMA-SR

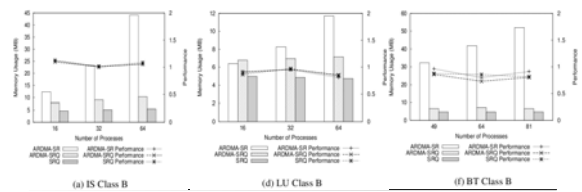
## SRQ



- Use only SRQ
- Reduced memory consumption (vs RDMA) at the cost of latency  $\sim 1\mu s$

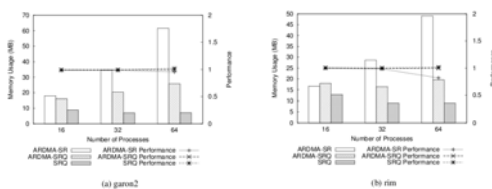
## Experimental Evaluation

	IS	MG	CG	FT	LU	BT	SP	NAMD	HPL
Avg. RDMA Connections	6.14	9.0	3.09	0.98	3.92	3.89	1.17	53.15	6.26
Avg. Low-Watermark events	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.03	0.0
Unexpected Messages (%)	2.7	10.2	13.5	11.9	38.1	0.3	0.7	48.2	13.6
Total Messages	1.9e5	3.1e5	2.7e6	3.6e5	5.3e6	1.6e6	4.7e6	3.7e6	7.3e5
MPI Time (%)	47.25	9.16	33.87	37.85	14.23	10.17	11.88	23.54	24.68



## SuperLU

Processes	garon2			rim		
	16	32	64	16	32	64
Avg. RDMA Connections	12.44	25.75	40.25	7.25	12.06	14.25
Avg. Low-Watermark events	1.56	0.06	0.12	1.56	0.66	0.64
Unexpected Messages (%)	33.5	22.0	31.6	29.4	24.2	30.0
Total Messages	2.9e5	4.8e5	7.5e5	3.8e5	7.4e5	1.1e6



## Conclusion

- SRQ approach able to scale while maintaining performance
- 5-10MB internal memory needed per process

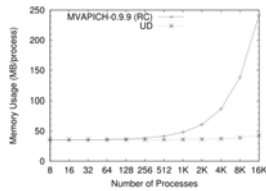
## Papers

- S. Sur, M. Koop, and D. K. Panda, High-Performance and Scalable MPI over InfiniBand with Reduced Memory Usage: An In-Depth Performance Analysis
- M. Koop, S. Sur and D. K. Panda, Zero-Copy Protocol for MPI using InfiniBand Unreliable Datagram
- M. Koop, T. Jones and D. K. Panda, MVAPICH-Aptus: Scalable High-Performance Multi-Transport MPI over InfiniBand

## Introduction

- RC used in most MPI implementations over Infiniband
  - One QP per communicating pair
  - 68KB per QP
- Optimizations – SRQ, On Demand connection setup
  - MPI Library still needs around 140 MB for 8K processes

## UD – Scalable Alternative

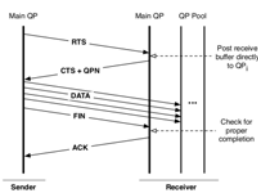


- Needs single QP per node
- Better ICM cache utilization
- Better fabric utilization
  - MPI doesn't require ordered message delivery
  - Lazy ACKs can be used

## UD - Limitations

- Connection-less, unreliable
  - No ordering guarantees, possible loss of packets
- Messages limited to MTU – currently 2KB, max 4KB.
  - Fragmentation at upper layer
- No dedicated receive buffers
- No RDMA - Send / Receive bandwidth around 50% of RDMA+RC for 1MB messages
- Extra header (GRH) in UD packets

## Zero-Copy UD Design



- Zero-Copy with serialized communication
- Buffers posted in 2KB chunks
- QP Pool – Dedicated QP for zero-copy
- Sequence number in immediate data field
  - Out of order / dropped packets trigger full retransmit
- Scatter list used to map GRH to temporary buffer

## Experimental Evaluation

- 2.8 Ghz Opteron 254 single-core, 4MB Memory
- Mellanox MT25208 Memfree HCA

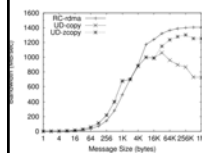


Fig. 5. Uni-Directional Bandwidth

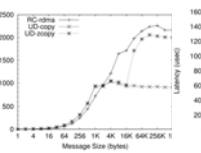


Fig. 6. Bi-Directional Bandwidth

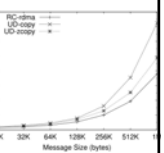
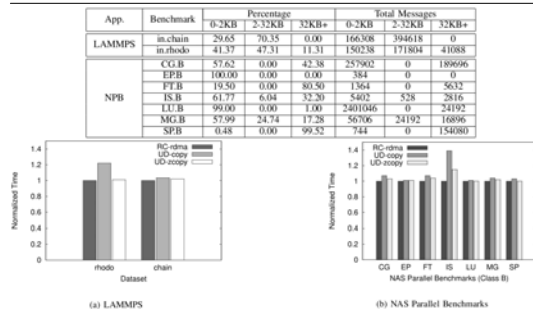


Fig. 4. One-way Latency

## Application Benchmarks



## Conclusion

- High performance systems continue to scale, MPI libraries need to be scalable
- Previous work established UD as a promising alternative but for zero copy
- Zero-copy protocol over UD can achieve significant speedup over copy based approach keeping bandwidth close to RC-RDMA

## Papers

- S. Sur, M. Koop, and D. K. Panda, High-Performance and Scalable MPI over InfiniBand with Reduced Memory Usage: An In-Depth Performance Analysis.
- M. Koop, S. Sur and D. K. Panda, Zero-Copy Protocol for MPI using InfiniBand Unreliable Datagram
- M. Koop, T. Jones and D. K. Panda, MVAPICH-Aptus: Scalable High-Performance Multi-Transport MPI over InfiniBand

## Introduction

- UD – Scalable, RC – Low latency, higher bandwidth
- MVAPICH-Aptus Multi-transport MPI for Infiniband
- Dynamically select underlying transport
  - Scalability and Performance

## Message Channels - Eager

- Reliable Connection Send/Receive (RC-SR)
  - SRQ based design used for scalability
- Reliable Connection Fast Path (RC-FP)
  - Small message RDMA write – low latency
  - 300KB per RC-FP channel
- Unreliable Datagram Send/Receive (UD-SR)
  - Better scalability

## Message Channels - Rendezvous

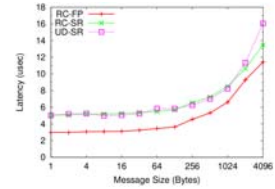
- Reliable Connection RDMA (RC-RDMA)
  - RDMA Write
  - No intermediate copies for large messages
- Unreliable Datagram Zero-Copy (UD-Zcopy)
  - High scalability and good bandwidth
- Copy-Based Send
  - If former methods unavailable
  - Segment and send over eager protocol channel

## Shared Memory

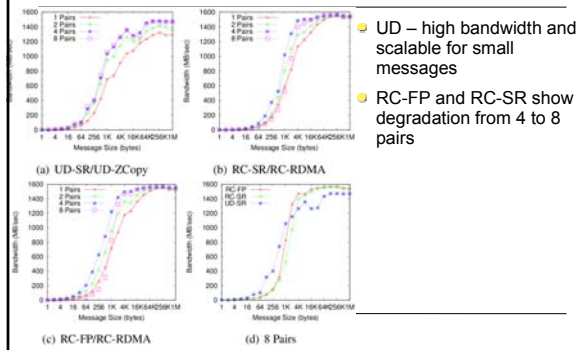
- Intra-node communication use shared memory channel
  - Reduces contention over network device, lower latency
  - Both eager and rendezvous

## Channel Evaluation

- 70 nodes, 2.33GHz Intel Clovertown quad-core.
- Mellanox MT25208 dual-port memfree HCA

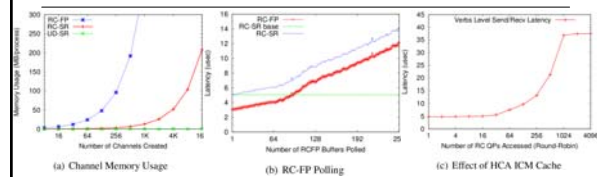


## Channel Evaluation (cont'd)



- UD – high bandwidth and scalable for small messages
- RC-FP and RC-SR show degradation from 4 to 8 pairs

## Channel Scalability



- UD-SR maintains constant memory utilization
- RC-FP polling delays message detection on other channels

## Channel Characteristics

Type	Channel	Transport	Latency	Throughput	Scalability
Eager	RC Send/Receive (RC-SR)	RC	Good	Fair	Fair
	RC Fast-Path (RC-FP)	RC	Best	Good	Poor
	UD Send/Receive (UD-SR)	UD	< 2KB, Good ≥ 2KB, Poor	< 2KB, Best ≥ 2KB, Poor	Best
Rendezvous	RC-RDMA	RC	-	Best	Fair
	UD Zero-Copy (UD-ZCopy)	UD	-	Good	Best
	Copy-Based	UD or RC	-	Poor	-

## Initialization

- Create UD QPs (UD-SR + UD-ZCopy) on all tasks
  - Intra-node SMP channels
- Frequently communicating pairs allocated resource intensive channels
  - RC-SR / RC-FP
- Decision based on messaging statistics and architecture

## Channel Multiplexing

- Multiple active channels
  - Messages may be re-ordered
- General framework to re-order messages from different channels
  - Reliability engine required for UD

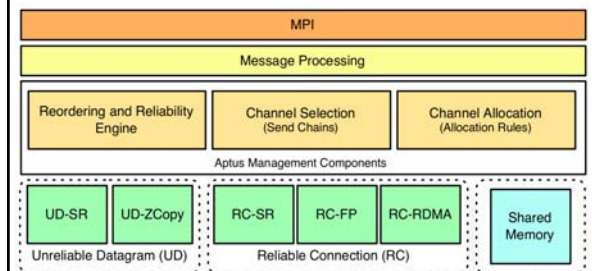
## Channel Selection

- Characteristics of channels vary with architecture, number of ranks, etc.
- Send rule chain method for channel selection
  - { COND, MESSG\_CHANNEL }
- Default
  - { MSG\_SIZE <= 2048, RC-FP },
  - { MSG\_SIZE <= 2008, RC-SR },
  - { MSG\_SIZE <= 8192, RC-SR },
  - { MSG\_SIZE <= 8192, UD-SR },
  - { TRUE, RC-RDMA }, { TRUE, UD-ZCOPY }

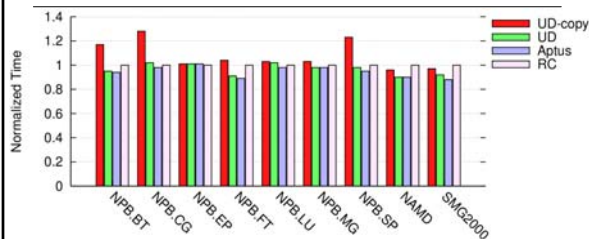
## Channel Allocation

- Need to limit certain kinds of channels (RC-FP)
- RC connections aren't scalable
- Adaptively create channels based on counts over rules
  - Limit RC-SR/RC-RDMA to 16 per task
  - RC-FP to 8 per task

## Design Overview



## Application Benchmarks



## Conclusion

- Aptus seeks to get best of both worlds
  - RC benefits for frequently communicating peers
  - UD for others - scalability
- SMG2000: 12% improvement over pure RC and 4% better than UD
- NAMD: 10% improvement over RC