NETWORK-BASED COMPUTING LABORATORY

MVAPICH-Aptus: Scalable High-Performance Multi-Transport MPI over InfiniBand

Matthew Koop[1,2] Terry Jones[2] D. K. Panda[1]

{koop, panda}@cse.ohio-state.edu

trj@llnl.gov

[1]Network-Based Computing Lab,
The Ohio State University
Columbus, OH USA

[2]Lawrence Livermore National Lab
Livermore, CA USA

OHIO STATE

Introduction

- Scientific applications consume ever-increasing levels of computing power and memory
  - Increased resolution
  - 2D vs. 3D
- To keep up with this demand, parallel machines are increasing in scale
- Commodity clusters are scaling to thousands of processors/cores
  - TACC *Ranger*, LLNL *Atlas*, Sandia *Thunderbird*, …
  - Larger clusters with tens-of-thousands of cores are planned
- MPI is programming model of choice on large clusters for scientific applications

InfiniBand Overview

- InfiniBand is an increasingly popular HPC interconnect
  - Industry Standard
- Very good performance with many features
  - Minimum Latency: ~1-2us
  - Peak Bandwidth: ~1500MB/s
  - Remote Data Memory Access (RDMA), Hardware multicast, Quality of Service …
  - Variety of transport modes

Courtesy TACC

TACC Ranger:
•3936 compute nodes
•62,976 processing cores
•InfiniBand interconnect fabric

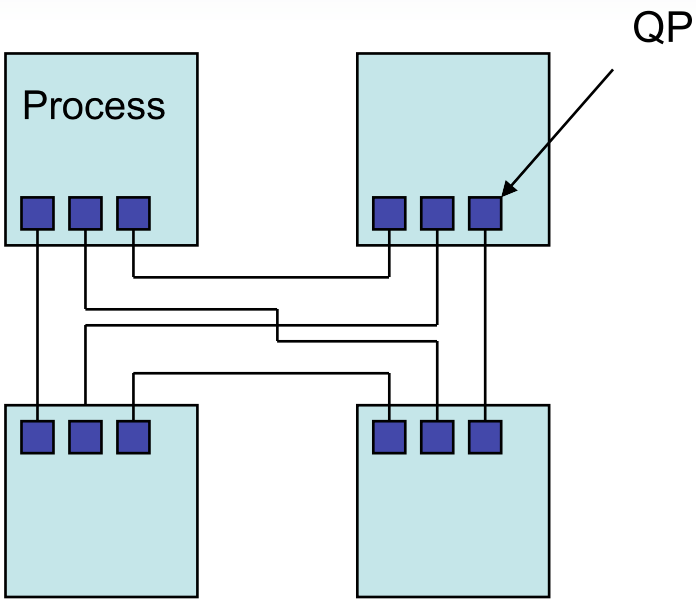
InfiniBand Communication

- Queue Pair (QP) Model
  - Each QP consists of two queues:
    - Send Queue (SQ)
    - Receive Queue (RQ)
  - A QP must be linked to a Completion Queue (CQ) which gives notification of operation completion from QPs
    - Polling
    - Event-based
- Memory and Channel Semantics
  - Memory: Remote Data Memory Access (RDMA)
  - Channel: Receive buffers are posted to the QP Receive Queue
    - Can be shared among QPs using a Shared Receive Queue (SRQ)

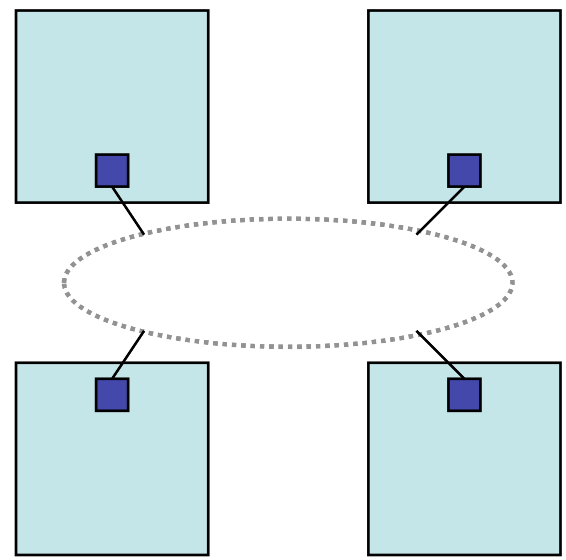
InfiniBand Transports

- **Reliable Connection (RC)**
  - Used as the primary transport for MVAPICH, OpenMPI, and other MPIs over InfiniBand
  - Most feature-rich -- supports RDMA and provides reliable service
  - Dedicated QP must be created for each communicating peer
- **Reliable Datagram (RD)**
  - Most of the same features as RC, however, a dedicated QP is not required.
  - *Not implemented on any current hardware*
- **Unreliable Connection (UC)**
  - Provides RDMA capability
  - No guarantees on ordering or reliability
  - Dedicated QP must be created for each communicating peer
- **Unreliable Datagram (UD)**
  - Connection-less. Single QP can communicate with any other peer QP
  - Limited message size
  - No guarantees on ordering or reliability

UD vs. RC



RC Communication Model



UD Communication Model

- UD has lower resource requirements since only one QP is required regardless of the number of peers

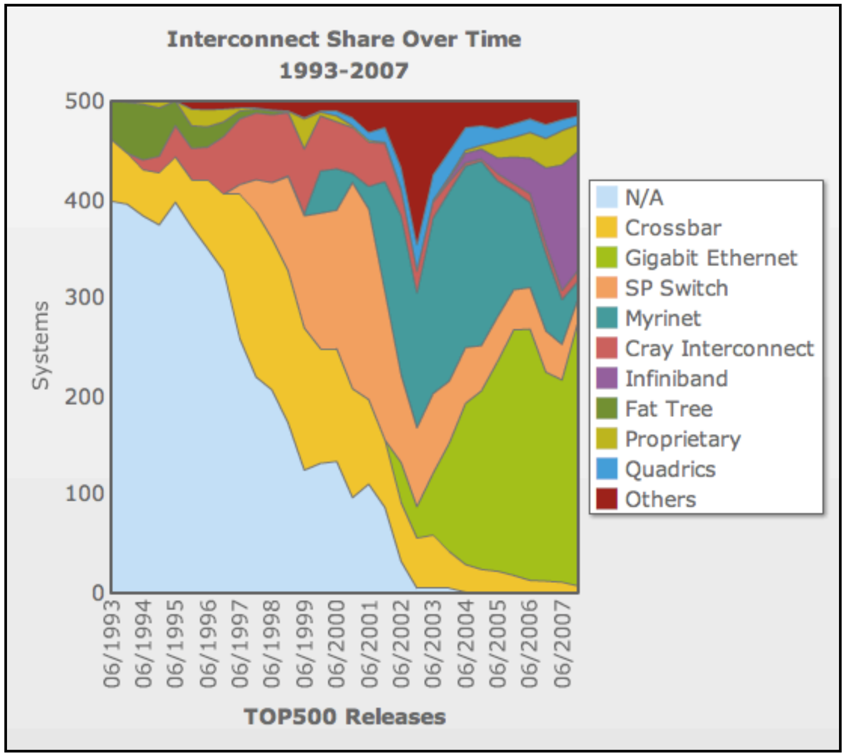
Presentation Outline

- Introduction
- Recent Advances and Problem Statement
- Message Channels
- MVAPICH-Aptus Design
- Experimental Evaluation
- Conclusions and Future Work

Recent Advances:

InfiniBand Cluster Deployments

- InfiniBand has grown *significantly* in popularity and deployment scale
- Top500 List
  - First appearance in 2003 on a 128 processor cluster
  - Now deployed on TACC Ranger with 62,976 cores
  - 25% now use InfiniBand as the primary interconnect



Recent Advances:

InfiniBand MPI Developments

- **Multiple Message Channels**
  - *Many* different methods of transferring messages have been proposed
- **Shared Receive Queue (SRQ)**
  - Scalable posting of receive buffers to Queue Pairs
  - Memory usage can still grow to hundreds of MB/process
- **Unreliable Datagram (UD) based MPI***
  - Lower memory requirements
  - Host Channel Adapter (HCA) caching efficiency
  - Fabric utilization

*Additional details can be found in:

M. Koop, S. Sur, Q. Gao, D.K. Panda, "High Performance MPI Design Using Unreliable Datagram for Ultra-Scale InfiniBand Clusters", International Conference on Supercomputing (ICS2007)

Problem Statement

- This work seeks to address two main questions:

What are the different protocols developed for MPI over InfiniBand and how do they perform at scale?

Given this knowledge, can the MPI library be designed to dynamically select protocols to optimize for performance and scalability?

Presentation Outline

- Introduction
- Recent Advances and Problem Statement
- Message Channels
- MVAPICH-Aptus Design
- Experimental Evaluation
- Conclusions and Future Work

Message Channels

- Message passing is generally implemented with two modes:
  - **Eager Protocol**: Small messages (<8K)
  - **Rendezvous Protocol**: Large messages

- *Multiple designs* of both protocols have been implemented for InfiniBand
  - Describe and evaluate each of them to determine performance and scalability characteristics

Message Channels:

InfiniBand Transports

- **Reliable Connection (RC)**
  - Used as the primary transport for MVAPICH, OpenMPI, and other MPIs over InfiniBand
  - Most feature-rich -- supports RDMA and provides reliable service
  - Dedicated QP must be created for each communicating peer
- **Reliable Datagram (RD)**

On these two transports various eager and rendezvous protocols have been implemented

  - No guarantees on ordering or reliability
  - Dedicated QP must be created for each communicating peer
- **Unreliable Datagram (UD)**
  - Connection-less. Single QP can communicate with any other peer QP
  - Limited message size
  - No guarantees on ordering or reliability

Message Channels:

Eager Channels

- **Reliable Connection Send/Receive** (*RC-SR*)
  - Channel built directly on the channel semantics of the RC transport of InfiniBand
  - Use of the Shared Receive Queue (SRQ) allows pooling of receive buffers to achieve better scalability
- **Reliable Connection Fast Path** (*RC-FP*)
  - Current adapters only reach their lowest latency using RDMA Write operations
  - This approach uses paired queues and last-byte polling to achieve low latency (at the cost of memory usage)
- **Unreliable Datagram Send/Receive** (*UD-SR*)
  - Built on the channel semantics of the UD transport of InfiniBand
  - Must take care of reliability, however, it is very scalable

Message Channels:

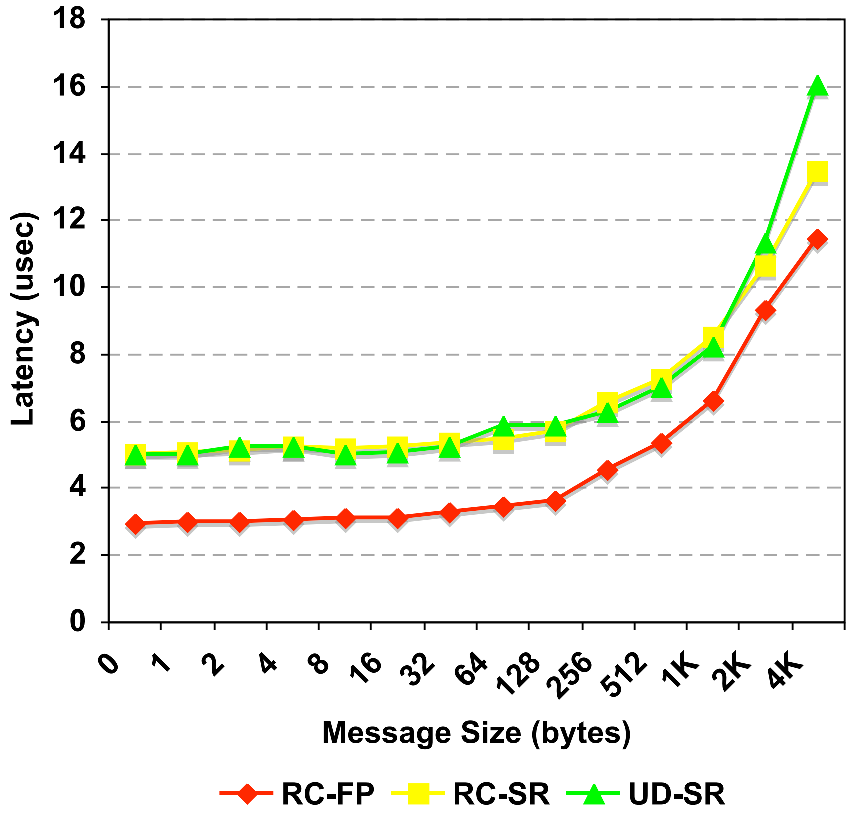
Rendezvous Channels

- **Reliable Connection RDMA** (*RC-RDMA*)
  - Using this method an RDMA write operation is used to write directly into the application buffer without intermediate copy operations
- **Unreliable Datagram Zero-Copy** (*UD-ZCopy*)
  - Using a pool of QPs and a novel approach, data can be transferred over UD -- preventing the requirement that RC connections be created
- **Copy-Based Send**
  - Negotiate buffer availability, but then use the eager channels to push the data to the receiver

Message Channels:

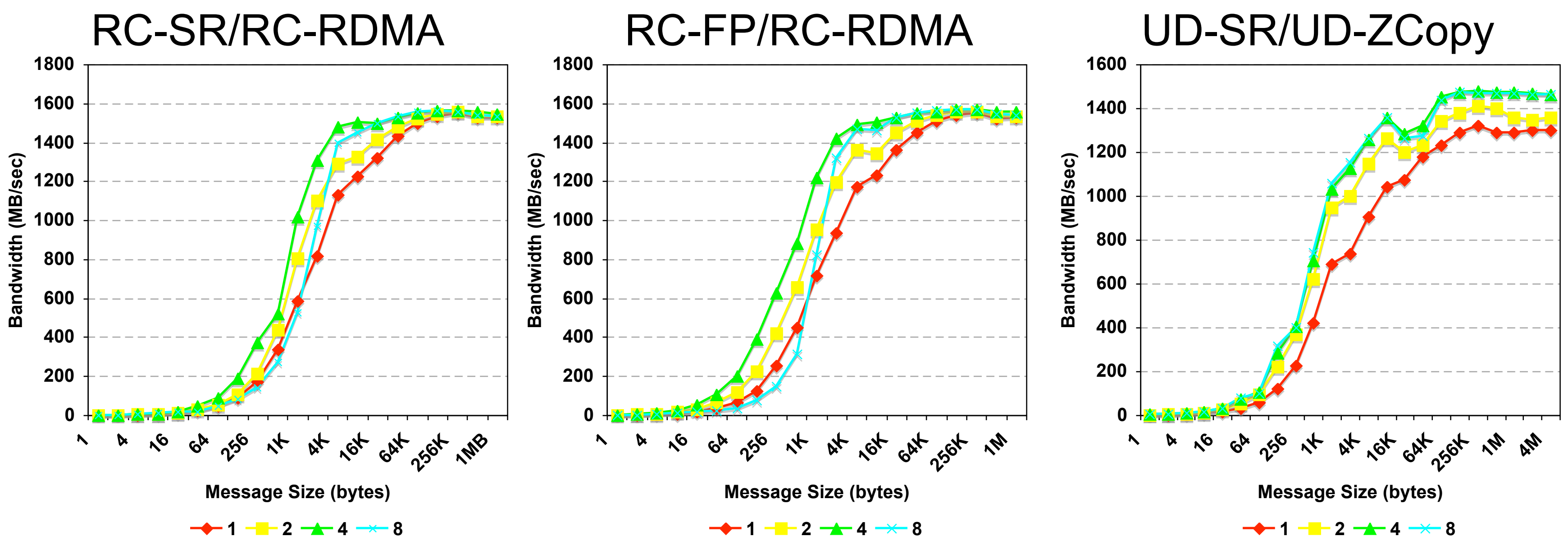
Performance: Eager Latency

- Classic ping-pong latency test (`osu_latency`)
- *RC-FP delivers lowest latency*
- RC-SR and UD-SR perform similarly until 2K and beyond where UD-SR requires software packetization



Message Channels:

Performance: Bandwidth

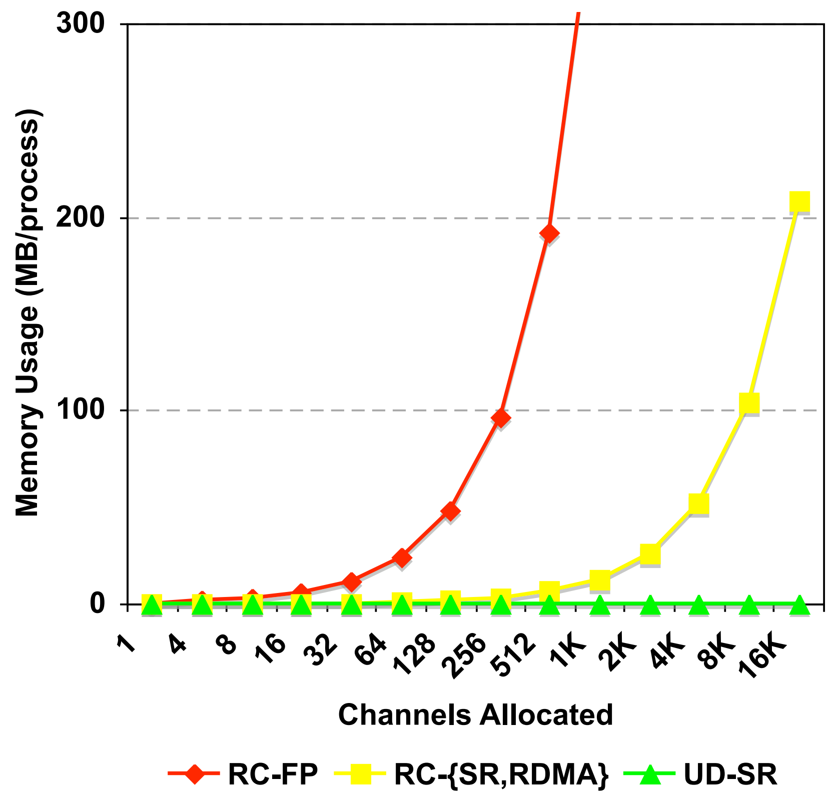


- Throughput for RC-based channels performs poorly when the number of communicating pairs increases
- UD-SR remains scalable in performance

Message Channels:

Scalability: Memory Usage

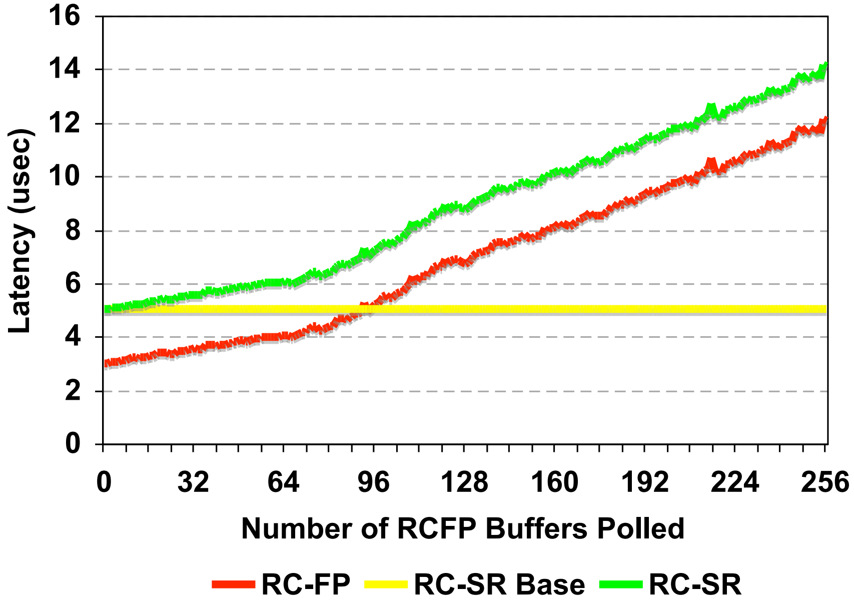
- *RC-FP* requires a significant amount of memory resources
- *RC-SR* is much more scalable in memory, but can still have issues at scale
- UD-SR remains very scalable with near-constant memory usage

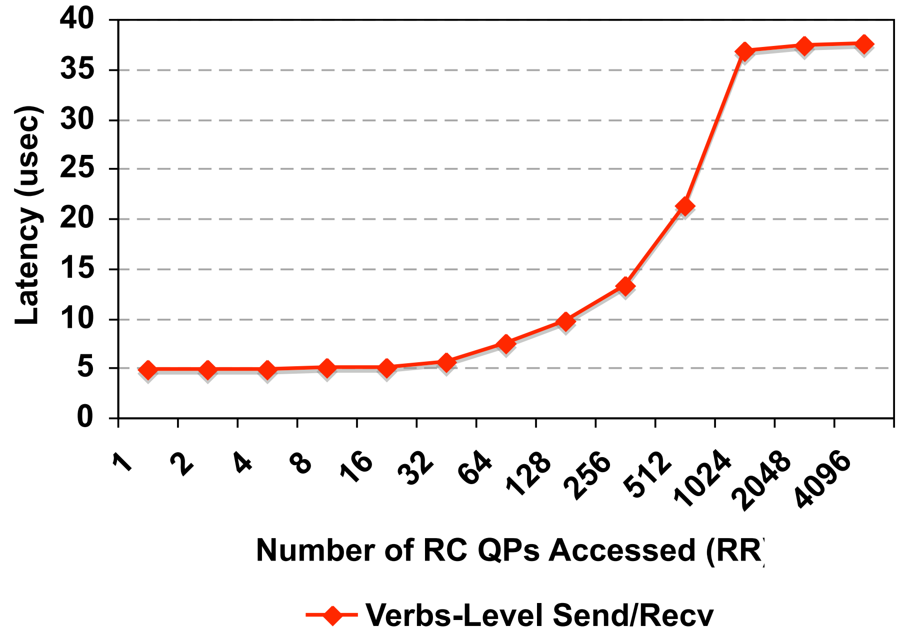


NETWORK-BASED COMPUTING LABORATORY

Message Channels:

Scalability: Latency





- Due to the memory polling used in RC-FP only a few channels can be allocated before latency increases
- The InfiniBand HCA has only a limited number of QPs that can be active in the on-card cache

OHIO STATE

Message Channels:

Summary

Type	Channel	Transport	Latency	Throughput	Scalability
Eager	RC-SR	RC	Good	**Fair**	**Fair**
	RC-FP	RC	**Best***	**Good**	**Poor**
	UD-SR	UD	<2K, **Good** >2K, **Poor**	< 2K, **Best** > 2K, **Poor**	**Best**
Rendezvous	RC-RDMA	RC	-	**Best**	**Fair**
	UD-ZCopy	UD	-	**Good**	**Best**
	Copy-Based	RC/UD	-	**Poor**	-

No eager or rendezvous channel has all of the desired features

Presentation Outline

- Introduction
- Recent Advances and Problem Statement
- Message Channels
- MVAPICH-Aptus Design
- Experimental Evaluation
- Conclusions and Future Work

Design Overview

- As seen from the previous evaluation results, no single channel for either eager or rendezvous is always best
- General Goal:

Use a combination of message channels and transports to optimize for **performance** and **scalability**

- Design Challenges:
  - *When should a channel be* ***created?***
  - *When should a channel be* ***used?***

Design:

Channel Allocation

- Some channels perform well when only a limited number of them are created, but quickly deteriorate
  - RC Transports (RC-SR/RC-FP/RC-RDMA)
    - Each RC connection requires additional memory usage
    - Cache on HCA can be overflowed quickly
  - RC-FP:
    - Too many channels increases polling time
    - Memory scalability is poor
- Strategy:
  - Create up to a configurable number of channels of each type
    - 16 RC QPs
    - 8 RC-FP connections
  - Setup after a certain number of “qualified” messages are transferred

Design:

Channel Usage

- As found earlier, some channels also perform differently given message size and other features
- We allow a flexible form of matching when sending a message:

Sample Configuration

```
MSG_SIZE <= 2048, RC-FP,
MSG_SIZE <= 2008, UD-SR,
MSG_SIZE <= 8192, RC-SR,
MSG_SIZE <= 8192, UD-SR,
TRUE, RC-RDMA,
TRUE, UD-ZCopy,
TRUE, Copy-Based
```

- Take the first match where both the conditional is true and the channel is allocated to the destination peer

Presentation Outline

- Introduction
- Recent Advances and Problem Statement
- Message Channels
- MVAPICH-Aptus Design
- Experimental Evaluation
- Conclusions and Future Work

Evaluation

- We implement our design in MVAPICH:
  - High-performance MPI over InfiniBand
  - Used by over 660 organizations worldwide
  - Available as part of Open Fabrics Enterprise Distribution (OFED)
  - http://mvapich.cse.ohio-state.edu

- Evaluated Configurations:

	RC-SR	RC-RDMA	RC-FP	UD-SR	UD-ZCopy
RC MVAPICH0.9.9	Available	Available	Available		
UD MVAPICH-UD				Available	Available
Aptus	Available	Available	Available	Available	Available

- We implement our Aptus design by extending the `ch_gen2_ud` device of MVAPICH

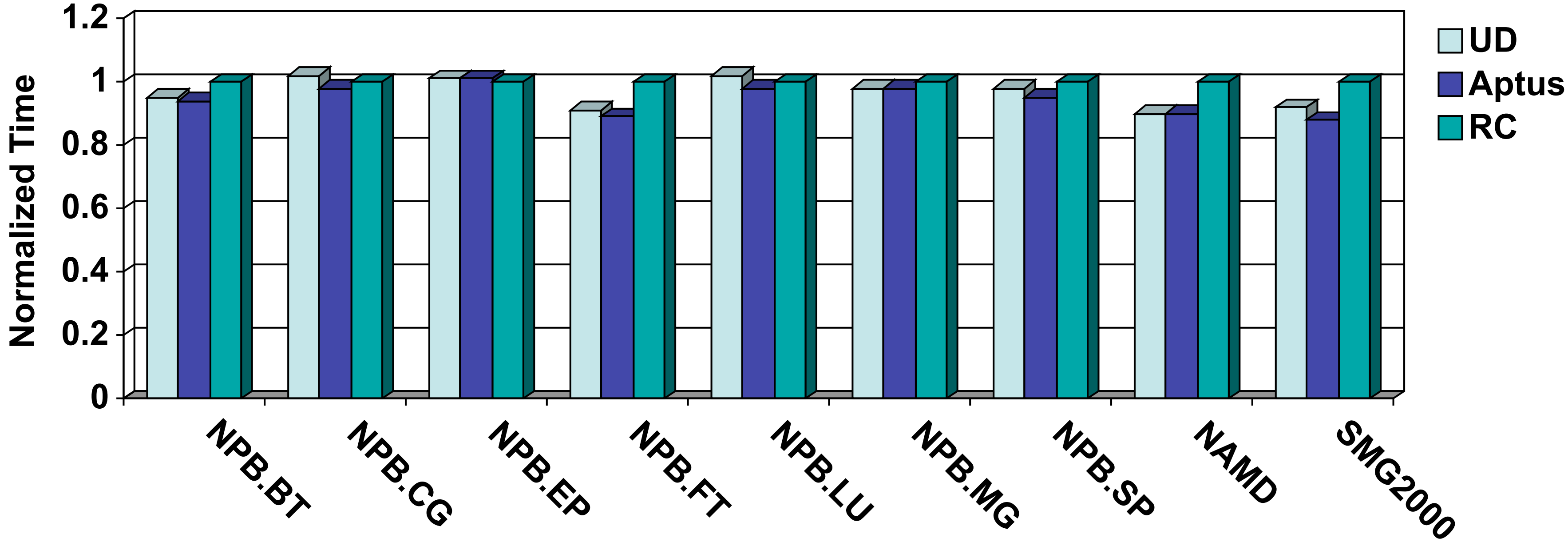
Evaluation:

Experimental Method

- Experimental Testbed:
  - 70 node, 560-core InfiniBand Linux cluster
  - Dual 2.3GHz “Clovertown” quad-core processors
  - Mellanox MT25208 DDR HCA
  - OpenFabrics OFED 1.2
- We evaluate the following application benchmarks
  - NAS Parallel Benchmarks: CFD application kernels
  - NAMD: Molecular dynamics application
  - SMG2000: Multigrid solver (ASC Benchmark)
- In addition to collecting the wallclock performance measurement, we also evaluate other characteristics:
  - Channels created
  - Message and data volume over each channel

Evaluation:

Performance Results



- In all results we see that the hybrid UD/RC design is able to outperform or match either mode used exclusively
- 512/484 processes

Evaluation:

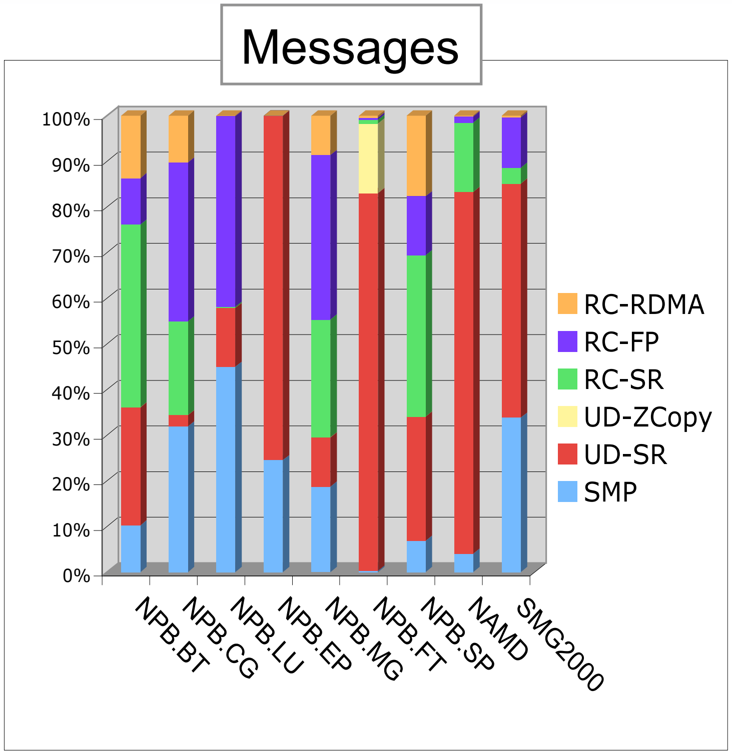
Avg. Channels Allocated / Process

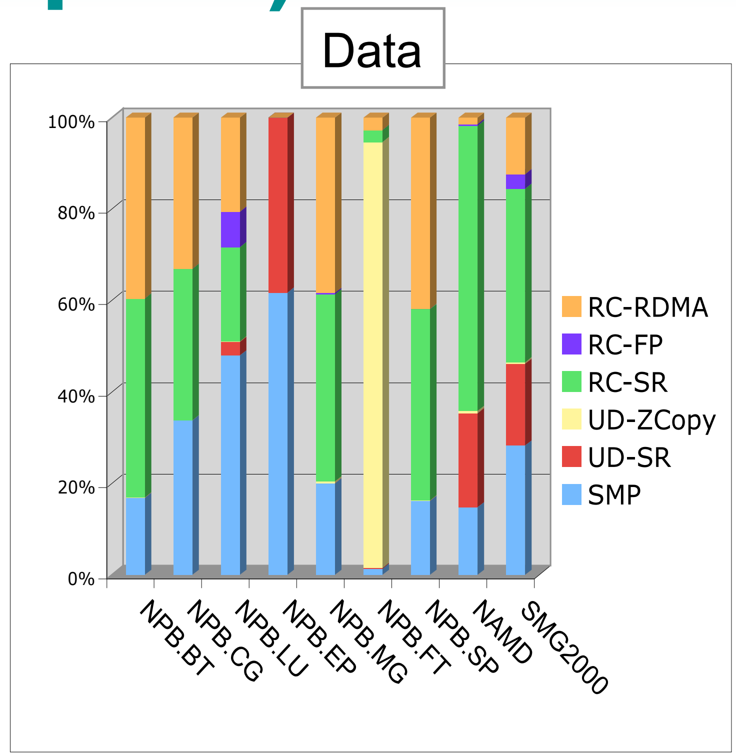
	SMP	UD-{SR,Zcopy}	RC-{SR,RDMA}	RC-FP
NPB.BT	4.11	20.17	10.60	7.88
NPB.CG	3.00	6.94	2.94	2.94
NPB.EP	3.00	6.00	0.00	0.00
NPB.FT	7.00	504.00	16.00	8.00
NPB.MG	4.31	9.00	5.63	5.63
NPB.LU	3.75	7.06	2.23	2.23
NPB.SP	4.11	20.17	10.62	7.88
NAMD	6.30	120.80	16.47	8.00
SMG2000	4.25	120.19	16.34	8.00

Breakdown shows Aptus dynamically has setup the fewest channels needed

Evaluation:

Channel Volume (Aptus)





- Breakdown of message transfers by channel show *good utilization* of “expensive” channels, despite allocating only a few of them

Conclusions and Future Work

- As clusters continue to scale, the MPI library must be scalable in memory as well as performance
- Previously a UD-based MPI showed superior scalability, but lower performance in some applications
- In this work we bridge the gap between RC and UD designs
- We are working towards
  - Looking into the new eXtended Reliable Connection (XRC) transport provided in ConnectX adapters
  - Release of the Aptus (UD/RC) design in an upcoming version of MVAPICH
  - Investigate support for dynamic communication patterns

NETWORK-BASED COMPUTING LABORATORY

Acknowledgements

Our research is supported by the following organizations

- Current Funding support by

- Current Equipment support by

MVAPICH

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

Questions?