MPI/CTP: A Reconfigurable MPI for HPC Applications

Manjunath Gorentla Venkata, Patrick Bridges

Scalable Systems Lab
University of New Mexico

Static configuration of protocol is not enough

Static configuration of communication protocol cannot optimize service in all cases

- Varying static protocol demands
  - HPC applications have diverse communication requirements
  - HPC machines have variety of network interface capabilities
- Dynamic communication characteristics

Solution: MPI/CTP - A reconfigurable implementation of MPI implementation

A fine grained reconfigurable protocol driven by application and hardware specific protocol optimizations

Outline

MPI/CTP features

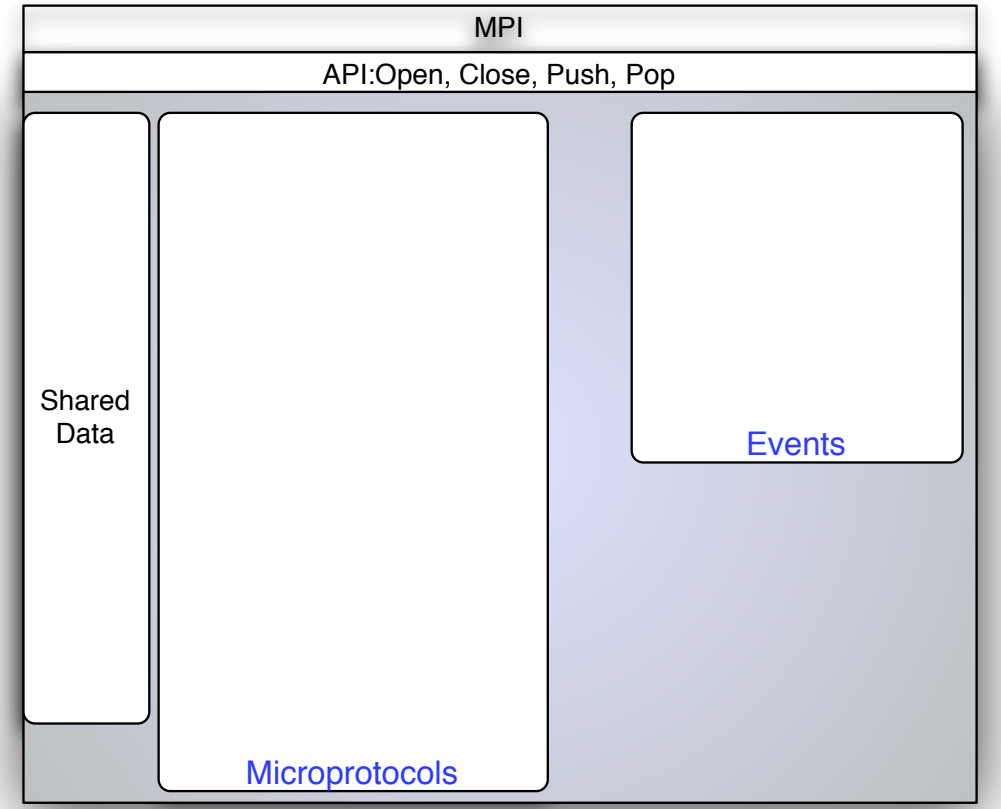
- Supports various granularities of configuration
  - Fine grained - functional properties and protocol behavior
  - Coarse grained - selecting network interfaces for message transfer during component start time.
- Supports all types(sync, async,buffered) of point-to-point operations

MPI/CTP - Birds eye view of design

- MPI/CTP built using Cactus framework and CTP
- MPI functionality implemented as microprotocols
  - Example microprotocols MPI message transfer, Demultiplexing, Reliability, Ordering, Congestion Control
- Events and event handlers

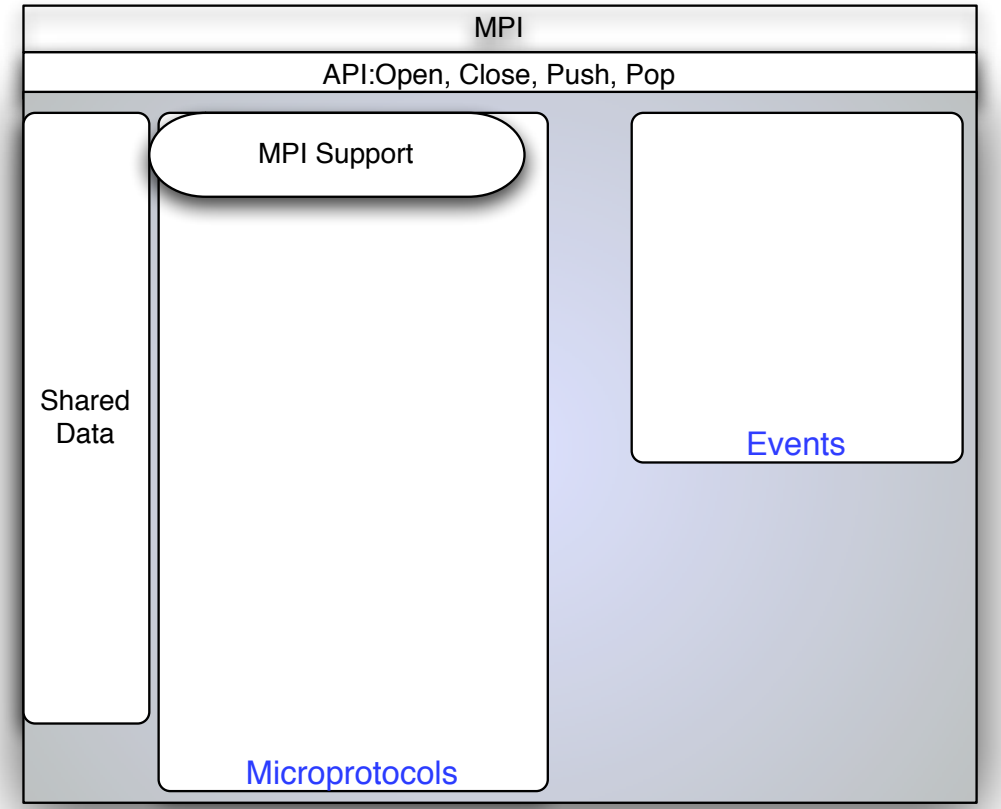
Relevant design details

MPI layer to match semantics of Cactus and MPI applications



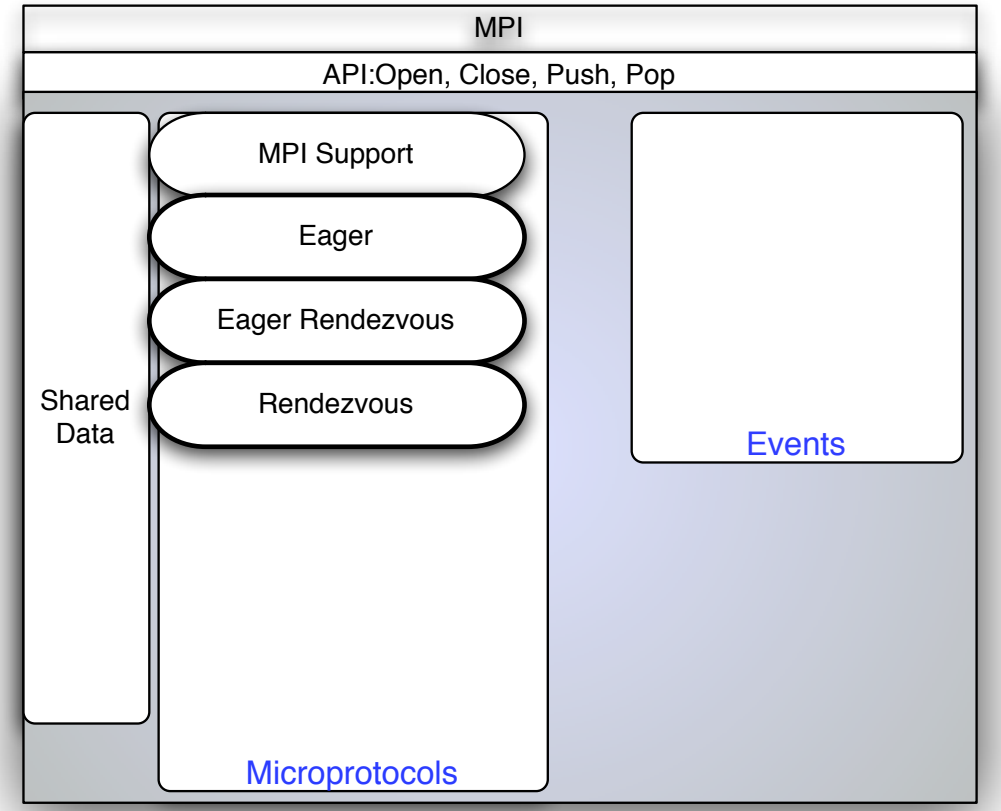
MPI/CTP - Relevant design details

MPI/CTP matching/demultiplexing



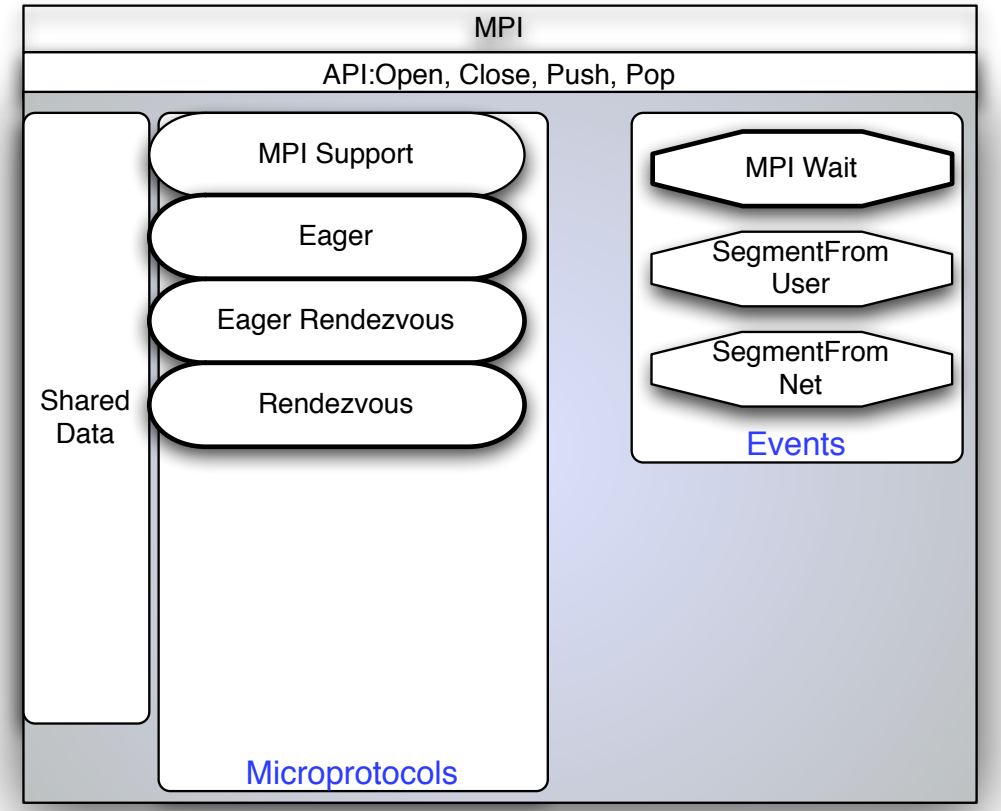
MPI/CTP - Relevant design details

Message transfer protocols



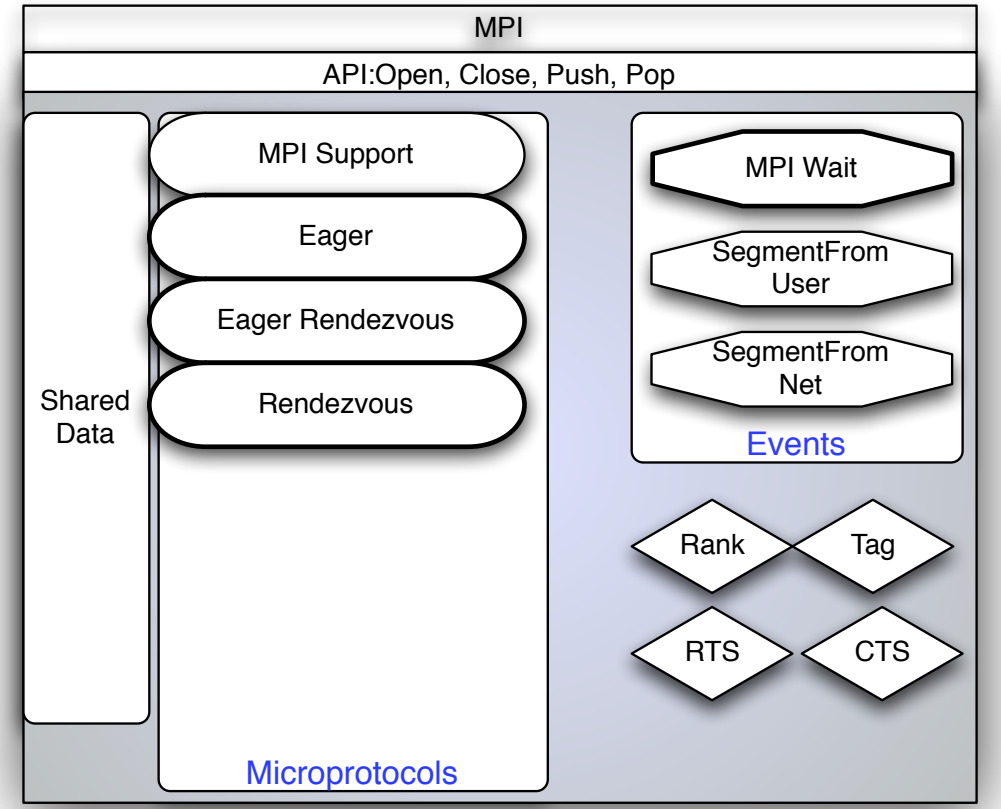
MPI/CTP - Relevant design details

MPI related Events



MPI/CTP - Relevant design details

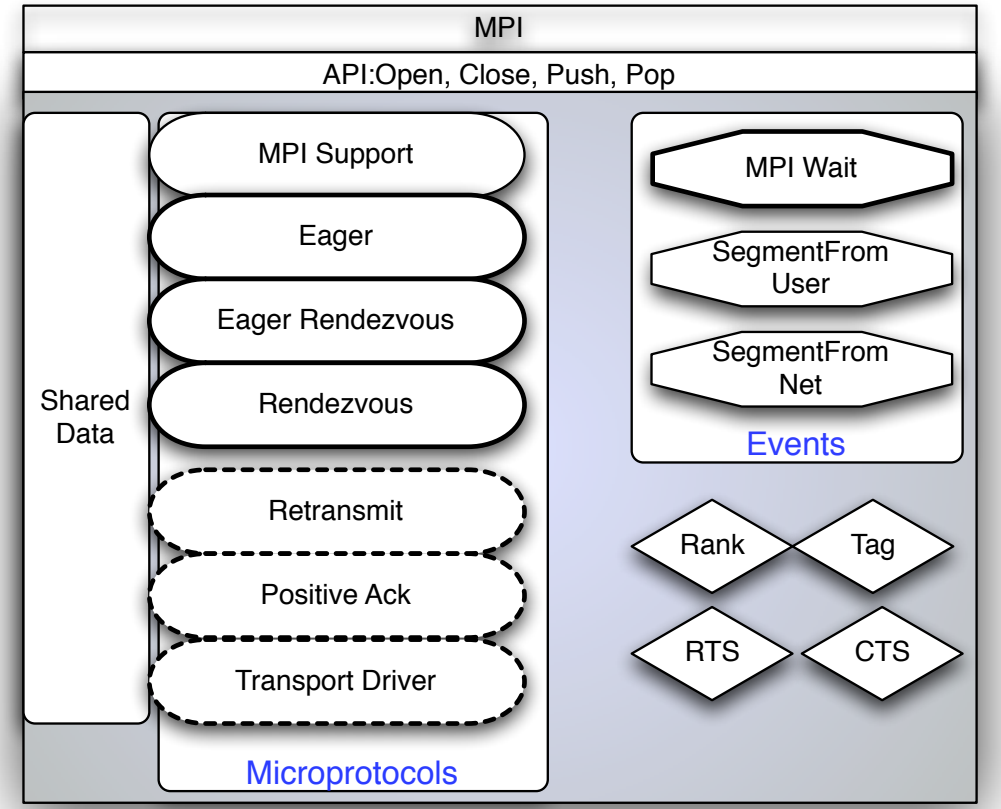
MPI/CTP Message headers



MPI/CTP - Relevant design details

Compatible with CTP functions

- Reliability
- Congestion Control
- Flow control



Reconfiguration opportunities

- Network-based
  - Reliability protocols
- Application-based
  - Message list management
  - Message transfer protocols
- Collectives

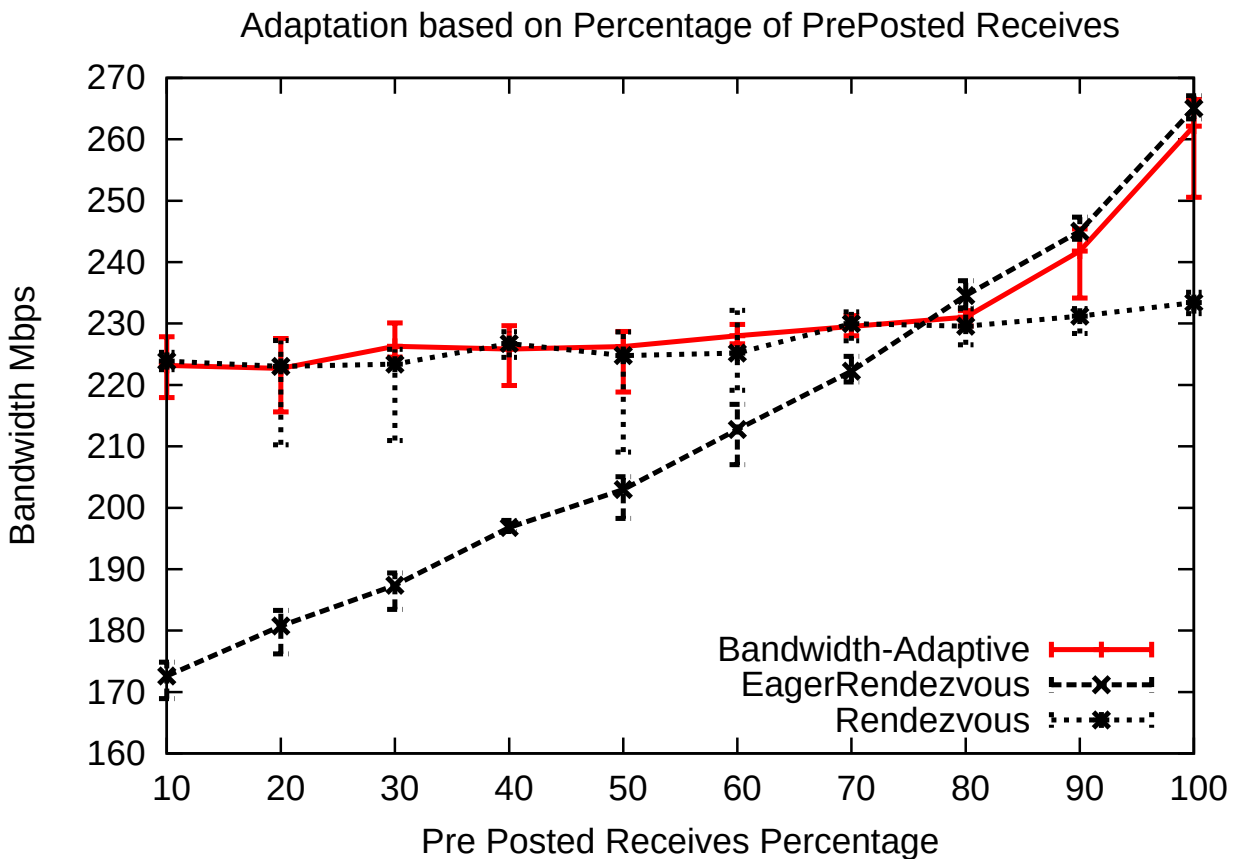
Case study: Adapting protocol behavior

- MPI/CTP message transfer microprotocols Eager Rendezvous, Rendezvous,Eager
- Preposted receives percentage dictate message send protocols
- Per-message, per-peer protocol reconfiguration

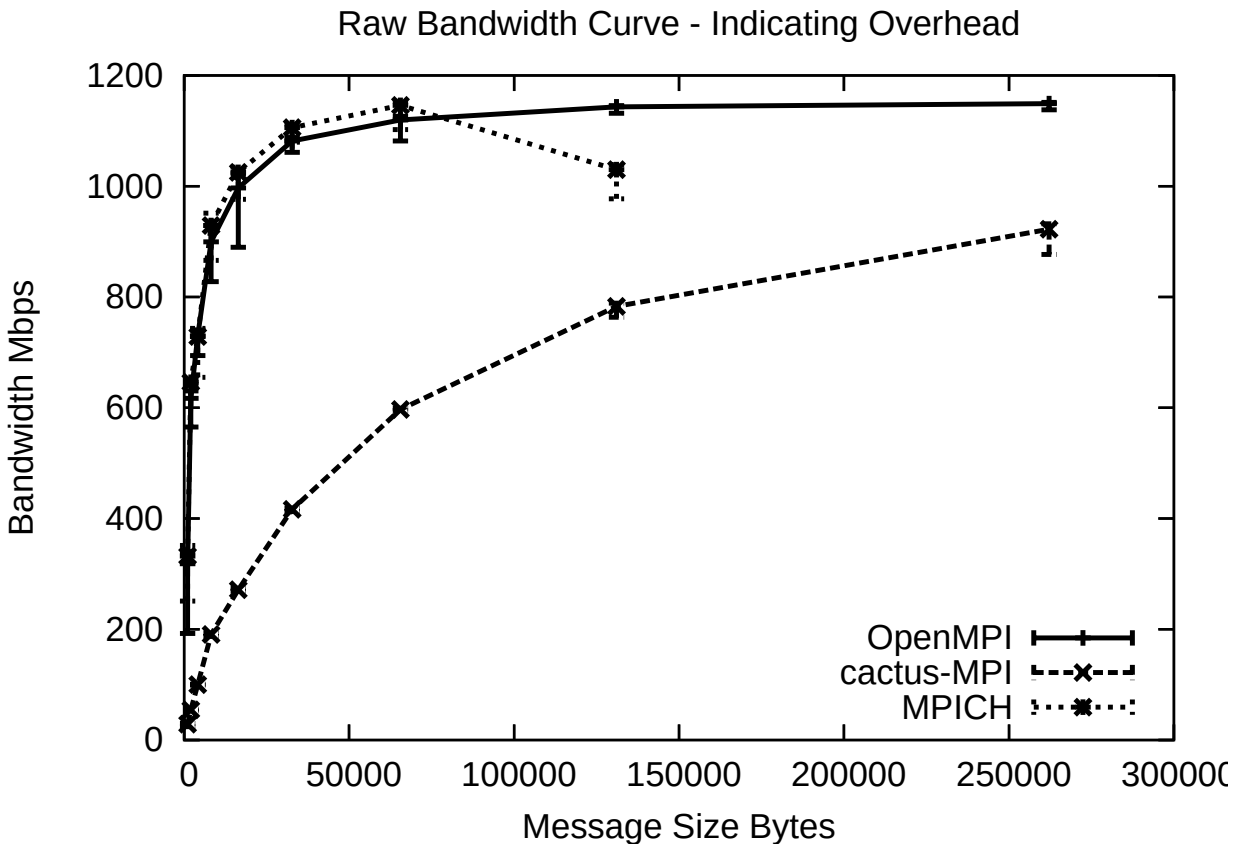
Experiment setup

- Hardware - 2.3 Ghz Pentium III Xeon, Myrinet NIC with Lanai 7 processor
- Software - Linux Kernel version 2.4.2, GM 2.1.1, MPICH 1.2.6, OpenMPI 1.0.2
- Benchmark - derived from SNL benchmarks
  - Measure effective throughput
  - Vary percent of receives preposted

Selecting protocol dynamically provides more bandwidth



Overhead in current prototype



- Zero copy
- Offload

Related Work

	MPI/CTP
Open-MPI	
Coarse grained reconfiguration	Fine grained
MPICH	
Compile time reconfiguration	Run time
H-CTP	
Designed for Grid systems	HPC systems

Conclusion

- MPI/CTP recovers lost performance
  - Flexibility to provide more bandwidth to application

Contributions

- Protocol architecture for application and hardware specific protocol reconfiguration in MPI
- Prototype MPI implementation supporting reconfiguration at compile time and runtime

Future Work

- Support for zero-copy in MPI/CTP
- Demonstrate advantages of hardware specific protocol reconfiguration in MPI
- Collective and single sided MPI operations reconfigurable to application requirements.

Thanks

Acknowledgments

- UNM - Prof.Patrick Bridges, Prof. Barney Maccabe
- SNL- Ron Brightwell, Rolf Riesen
- SSL members
- DOE - Office of Science grant DE-FG02-05ER25662
- Sandia University Research Program contract number 190576