AMR (Adaptive Mesh Refinement) Performance Benchmark and Profiling

July 2011

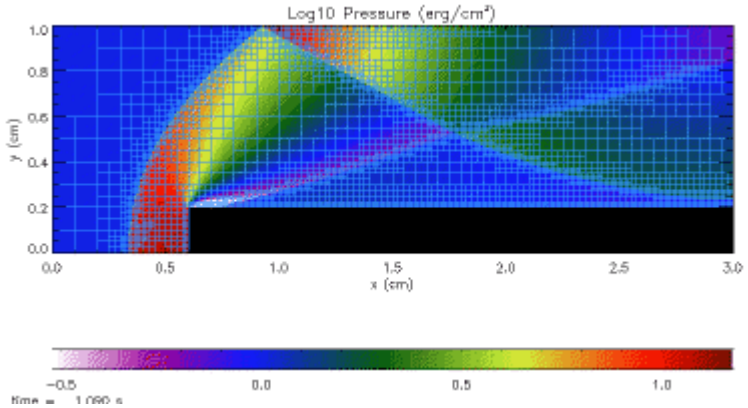
Acknowledgment:
- The DoD High Performance Computing Modernization Program
- John Bell from Lawrence Berkeley Laboratory

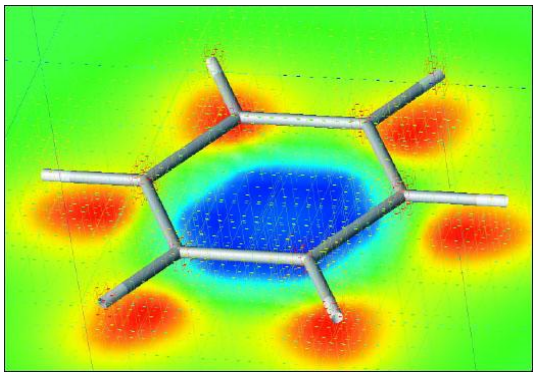
Note

- **The following research was performed under the HPC Advisory Council HPC|works working group activities**
  - Participating vendors: HP, Intel, Mellanox
  - Compute resource - HPC Advisory Council Cluster Center
- **We would like to acknowledge**
  - The DoD High Performance Computing Modernization Program for providing access to the FY 2009 benchmark suite
  - John Bell from Lawrence Berkeley Laboratory for developing the application
- **For more info please refer to**
  - http://www.hp.com/go/hpc
  - www.intel.com
  - www.mellanox.com

AMR - Adaptive Mesh Refinement

- **Adaptive Mesh Refinement (AMR) is a collection of 3 applications for solving problems that benefit from grids with adaptive, inhomogeneous spatial resolution**
- **AMR is developed at Lawrence Berkeley National Laboratory**
- **This particular benchmark makes use of the HyperClaw application for solving a gasdynamic problem**
- **The AMR source code supplied with the ABTP benchmarking distribution is the revision of AMR that shall be used in ABTP benchmarking**





Objectives

- **The presented research was done to provide best practices**
  - MPI libraries comparisons
  - Interconnect performance benchmarking
  - AMR Application profiling
  - Understanding AMR communication patterns

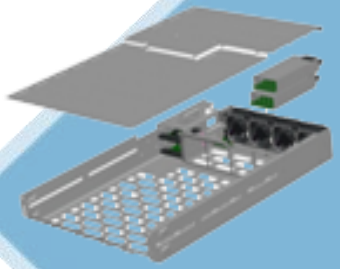
- **The presented results will demonstrate**
  - Balanced compute environment determines application performance

Test Cluster Configuration

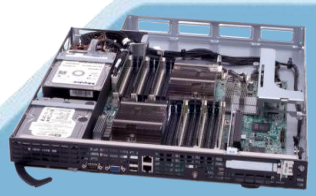
- **HP ProLiant SL2x170z G6 16-node cluster**
  - Six-Core Intel X5670 @ 2.93 GHz CPUs
  - Memory: 24GB per node
  - OS: CentOS5U5, OFED 1.5.3 InfiniBand SW stack
- **Mellanox ConnectX-2 InfiniBand QDR adapters and switches**
- **Fulcrum based 10Gb/s Ethernet switch**
- **MPI**
  - Intel MPI 4, Open MPI 1.5.3 with KNEM 0.9.6, Platform MPI 8.1.1, MVAPICH2-1.6rc1
- **Compilers: Intel Compilers 11.1.064**
- **Application: AMR (2009 May 11)**
- **Benchmark workload**
  - AMR cell = 1024x64x64

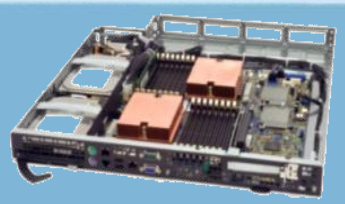
About HP ProLiant SL6000 Scalable System

- **Solution-optimized for extreme scale out**



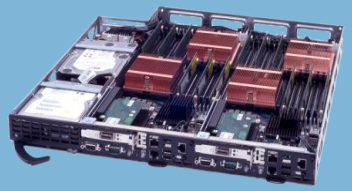
ProLiant z6000 chassis
Shared infrastructure
– fans, chassis, power





ProLiant SL160z G6 ProLiant SL165z G7
Large memory
-memory-cache apps

ProLiant SL170z G6
Large storage
-Web search and database apps



ProLiant SL2x170z G6
Highly dense
- HPC compute and
web front-end apps

Save on cost and energy -- per node, rack and data center

Mix and match configurations

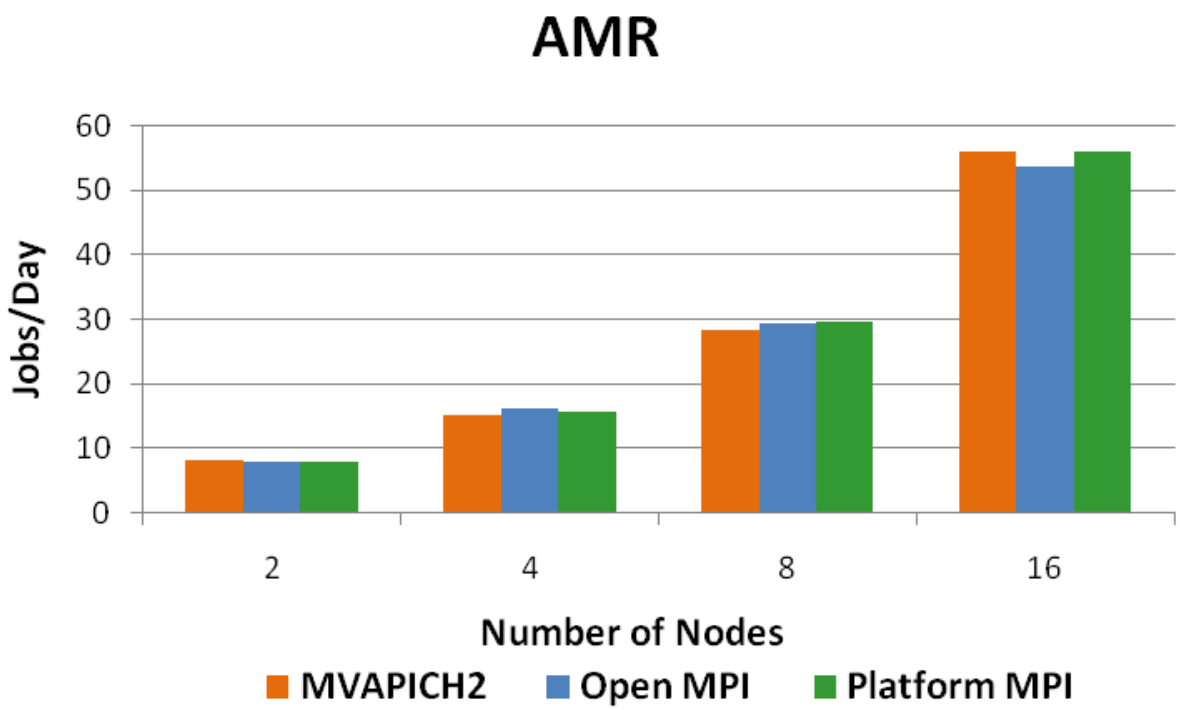
Deploy with confidence

#1 Power Efficiency*

* SPECpower_ssj2008
www.spec.org
17 June 2010, 13:28

AMR Benchmark Results – MPI Libraries

- **Input Dataset**
  - Cell: 1024x64x64
- **AMR scales with all three MPIs over InfiniBand**

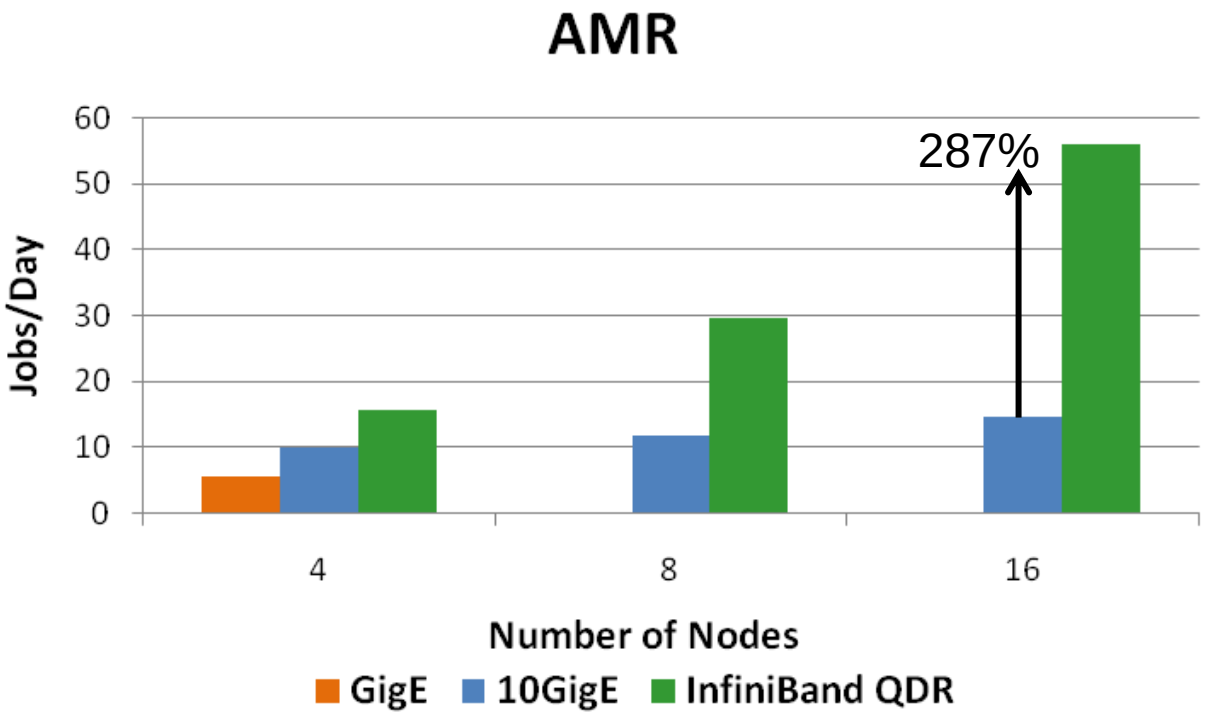


Higher is better

12-cores per node
InfiniBand QDR

AMR Benchmark Results – Interconnects

- **InfiniBand enables highest performance and scalability for AMR**
  - 287% higher than 10GigE at 16 nodes
- **GigE stops scaling after 2 nodes, 10GigE doesn't scale beyond 4 nodes**

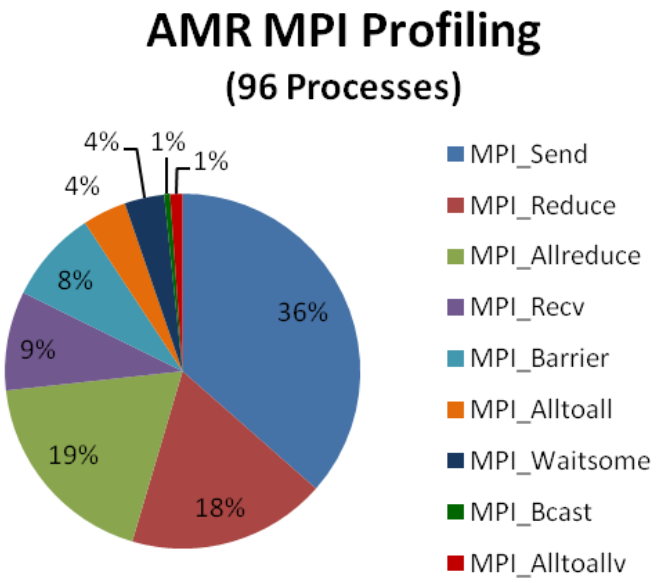


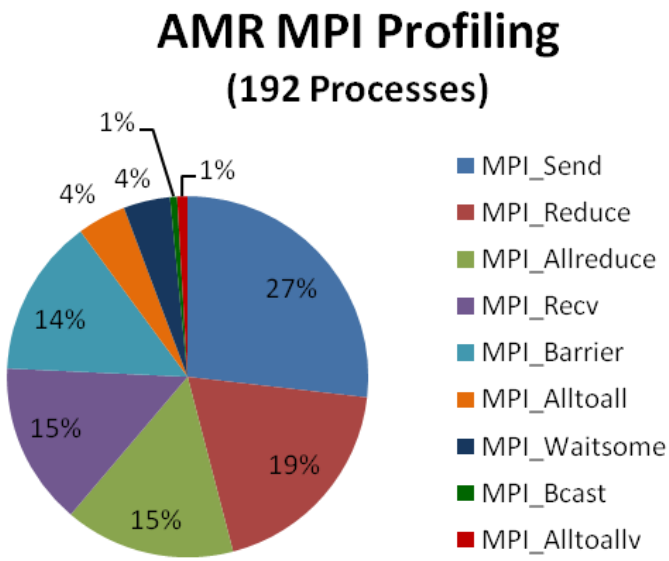
Higher is better

12-cores per node

AMR MPI Profiling – MPI Functions

- **MPI collectives and point-to-point have similar communication overhead**
  - Collectives: MPI_Reduce, MPI_Allreduce, and MPI_Barrier
  - Point-to-point: MPI_Send/Recv

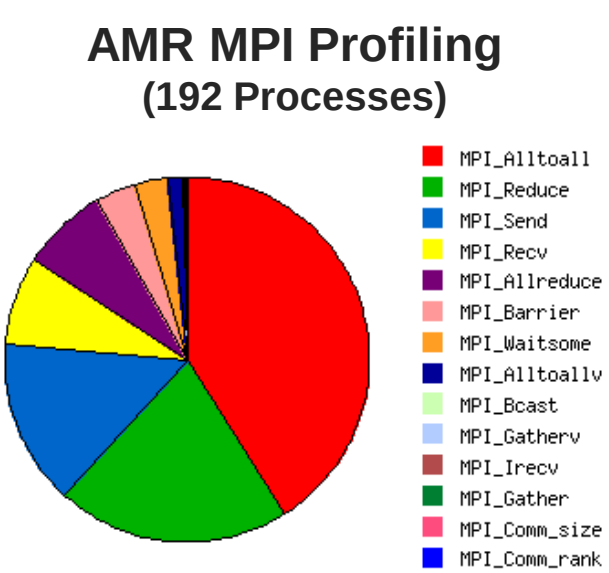




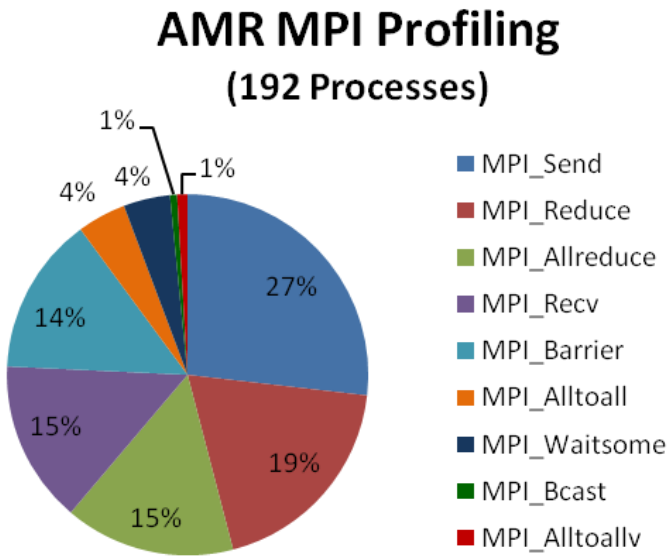
Platform MPI

AMR MPI Profiling – Open MPI vs Platform MPI

- **Open MPI has larger Alltoall overhead**
  - This causes AMR to run slightly slower with Open MPI comparing to Platform MPI
  - MPI_Alltoall optimization with Open MPI could enhance application performance



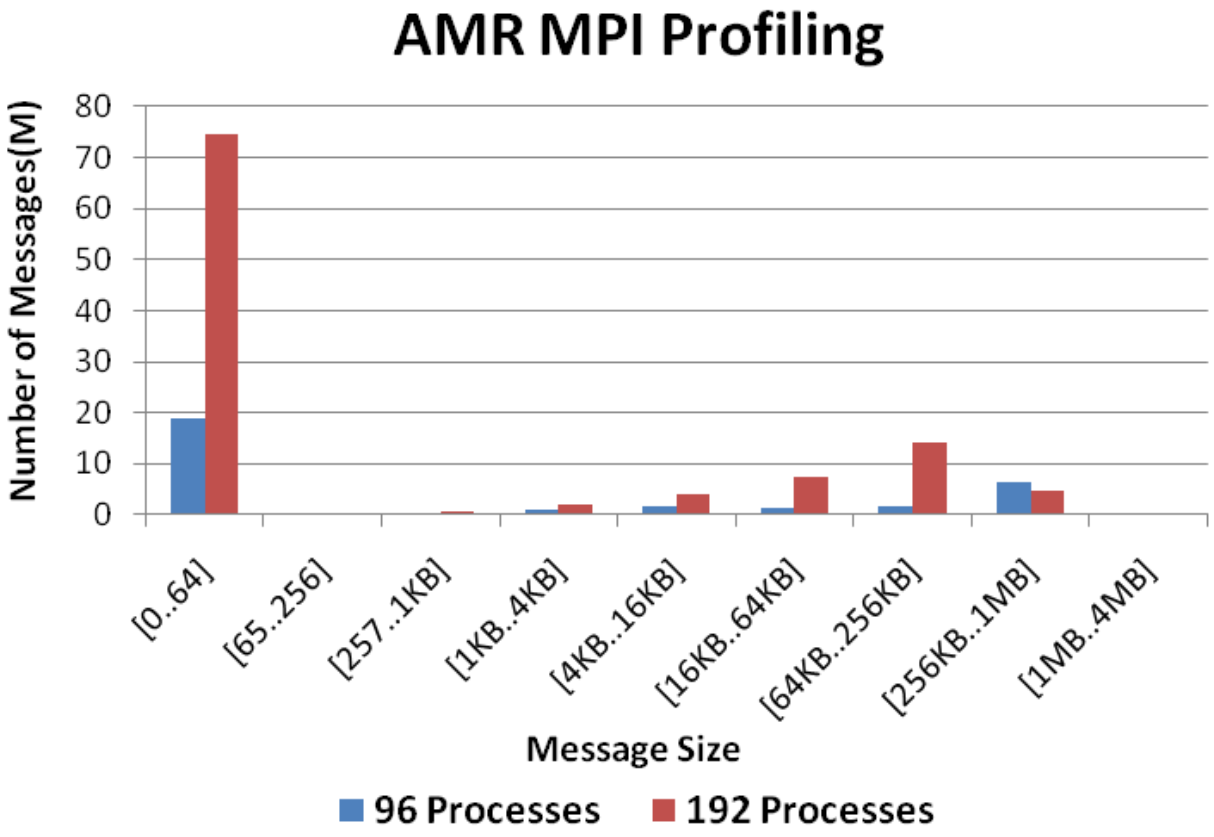
Open MPI



Platform MPI

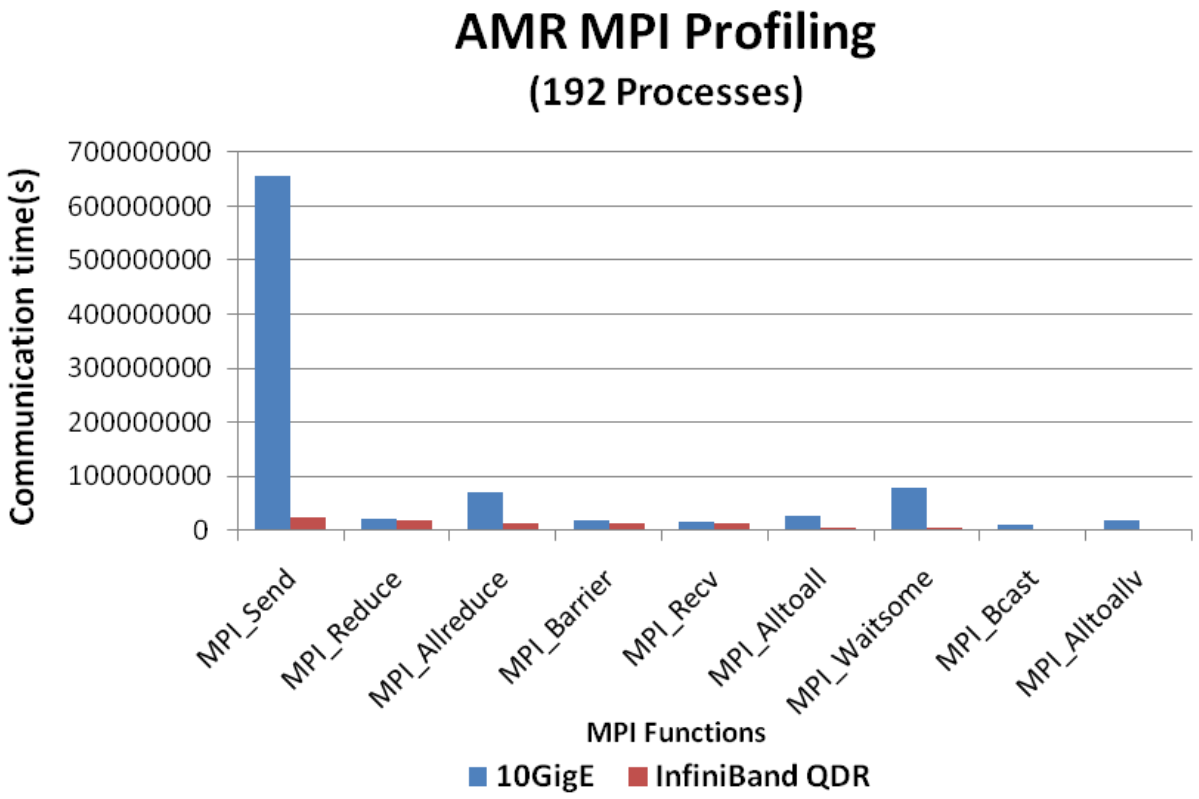
AMR MPI Profiling – Message Size

- **Both large and small messages increases significantly**
  - Small message < 64B
  - Large message >16KB



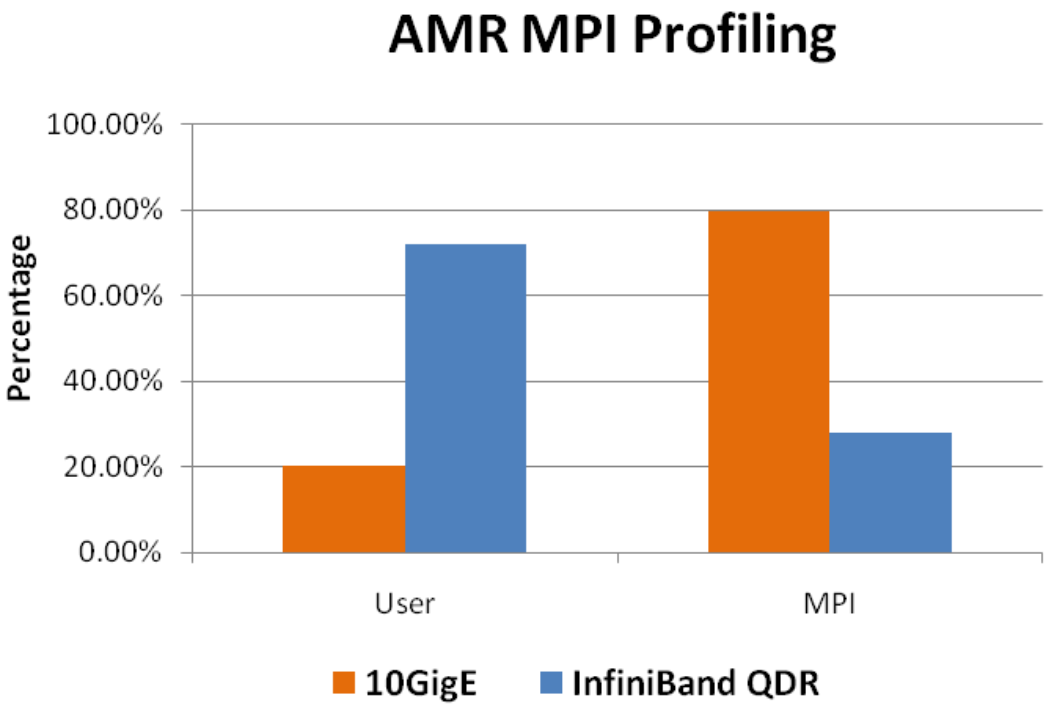
AMR MPI Profiling – MPI Time (10GigE vs InfiniBand)

- **MPI send takes much longer time to communicate**
  - Interconnect bandwidth is critical to this application at any cluster size
  - Latency is crucial for AMR to scale to larger cluster size



AMR MPI Profiling – Runtime Distribution (10GigE vs IB)

- **Application spends most time in communication with 10GigE**
- **InfiniBand QDR has much lower communication overhead**



192 Ranks

AMR Benchmark Summary

- **AMR performance benchmark demonstrates**
  - InifiniBand QDR enables higher application performance and scalability
  - AMR can't scale over neither GigE nor 10GigE
  - AMR is file I/O intensive
    - Lustre file system over InfiniBand meet application file I/O requirement

- **AMR MPI profiling**
  - Both MPI point-to-point and collectives create big communication overhead
  - Both large and small message are used by AMR
  - 10GigE has much bigger communication overhead versus InfiniBand QDR
  - Interconnect latency and bandwidth are crucial to AMR performance

Thank You

HPC Advisory Council