
# On the Current State of Open MPI on Cray Systems

Nathan Hjelm - HPC-5 LANL

Samuel Gutierrez - CCS-7 LANL

Manjunath Gorentla Venkata - ORNL

Cray Users Group (CUG) - May 8, 2014

UNCLASSIFIED LA-UR 14-23080


cielo
Alliance for Computing at Extreme Scale

# Outline

- Overview of Open MPI
- Overview of the Modular Component Architecture
- Whats Changed?
- Performance Results
- Conclusions
- Ongoing/Future work

# Overview of Open MPI

## 13 members, 15 contributors, 2 partners

























# Overview of Open MPI

- Started as an evolution of several prior MPI implementations
  - LA-MPI (Los Alamos), LAM/MPI (Indiana), FT-MPI (Tennessee)
- Follows an even-odd release cycle
  - “Feature” releases - 1.<odd> - Last release 1.7.5, next 1.9 (Est Summer/Fall 2014)
  - “Stable” releases - 1.<even> - Current release 1.8.1 (April 23, 2014)
  - Each release is extensively QAed
- Open source implementation of the MPI-2.1 (1.6.x) and MPI-3.0 + errata (1.8.x) standards
- Supports both simple byte transport networks and matching networks
- Supports a number of high-performance interconnect APIs: verbs, Cray uGNI, MXM, PSM, Cisco usNIC, Portals4

# Overview of Open MPI (MCA)

- Framework components allow easy addition of new hardware support, vendor specific APIs, algorithms, etc
- Modules represent specific instances of framework components
- Support for configuration and performance variables
- Specify configuration variables to mpirun using -mca <variable> <value> or by setting environment variables: MCA_OMPI_<variable>=<value>
- All MCA control variables exposed via the MPI Tool Information Interface (MPI_T) introduced in MPI-3 (Sept, 2012)

# Open MPI 1.8 Changes From 1.6

- New hierarchal collective algorithms
  - MPI_Allreduce, MPI_Allgather, MPI_Reduce
- “New” Transports
  - scif, ugni, usNIC, vader (xpmem, CMA)
- Cray XE, XK, and XC support
  - Building for XC still needs work
- MPI-3/MPI-2.2 conformant
  - MPI_T Tools Information Interface
    - Control Variables and performance variables
  - Shared Memory Windows
  - New fortran bindings (includes changes from MPI-3 errata)
- Java language bindings (non-standard)

# Overview of Cray XE/XK/XC Support in 1.8

- Support for Cray Gemini and Aries networks via ugni BTL
  - Lazy modex and connection establishment
  - Supports both FMA and BTE transport mechanisms
- Support for XPMEM via vader BTL
  - Utilizes lock-free message queues with fast-box support
  - Single copy send mechanism for medium/large messages
- Support for multiple resource managers
  - Alps, slurm, etc
- Support for direct launching via aprun or srun commands

# Whats Changed since CUG2012?

- As of Open MPI 1.8 Cray XE/XK/XC systems supported with a “super stable” (even-release) build
- Updated default SMSG limits to match Cray MPICH 6.3.0
- Added support for udreg to get memory notification
  - Removes requirement of ptmalloc2
  - Default for uGNI in 1.8
- Support for MPI-2 dynamic process management
  - MPI_Comm_spawn
  - Caveat: Only supported with mpirun
  - Not currently supported by vendor MPI (Cray MPICH 6.3)

# Performance Evaluation - Setup

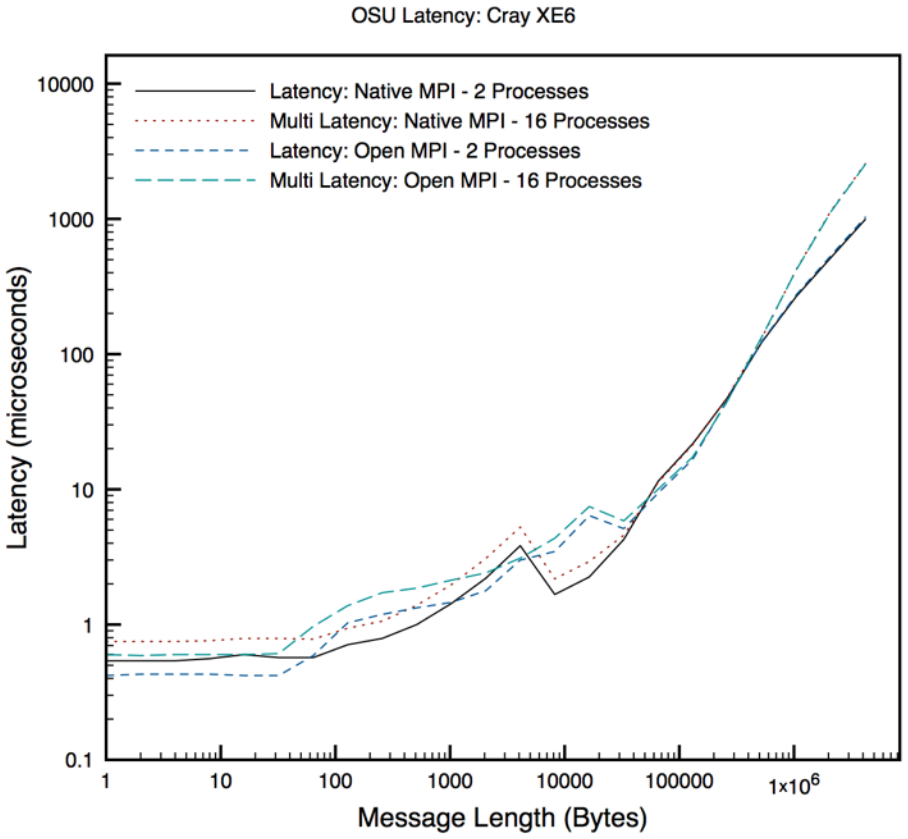
- Production systems with normal job mixes
- Test Beds
  - Cielo - 142,304 core Cray XE6
  - Edison - 133,824 core Cray XC30
- PrgEnv-gnu
  - CLE 4.1.40 (XE6), 5.1.29 (XC30)
  - gcc 4.7.2
  - Cray MPICH 6.3.0
- Open MPI 1.9 pre-release r31308
  - Contained fixes not yet in the 1.8 series
  - Default uGNI and vader BTL parameters

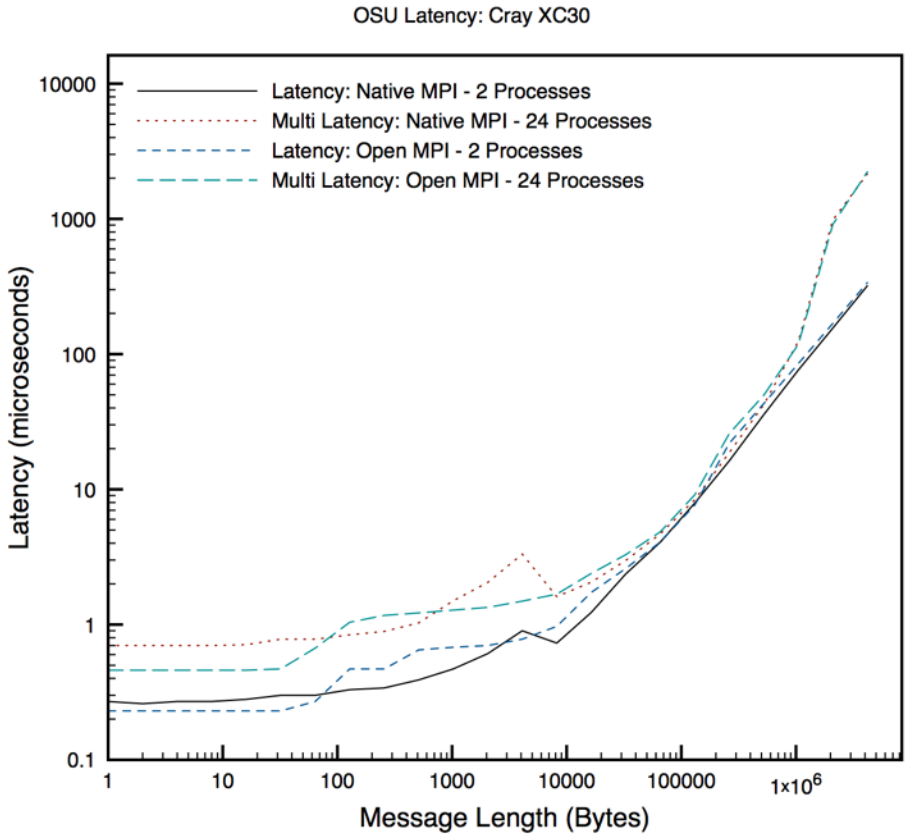
# Performance Evaluation - Benchmarks

- OSU Micro-Benchmark Suite v4.3
- Point-to-point latency
  - osu_latency, osu_multi_latency
- Point-to-point bandwidth
  - osu_bibw, osu_mbw_mr
- One-sided performance
  - osu_get_latency, osu_put_latency, osu_get_bw, osu_put_bw

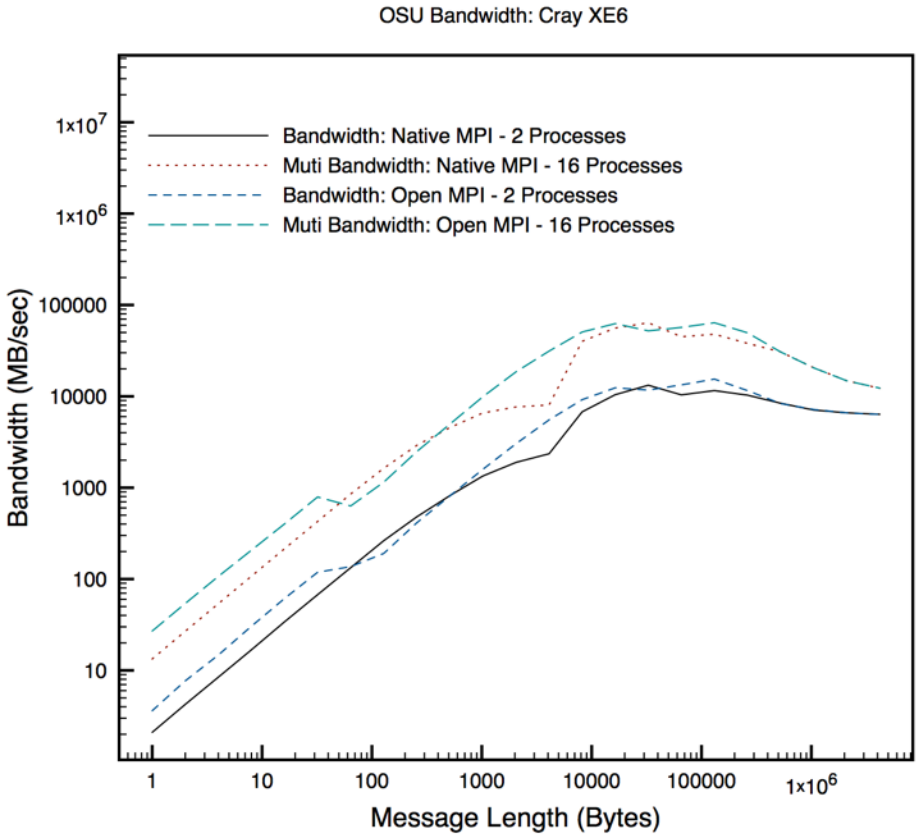
# Performance - Two Sided

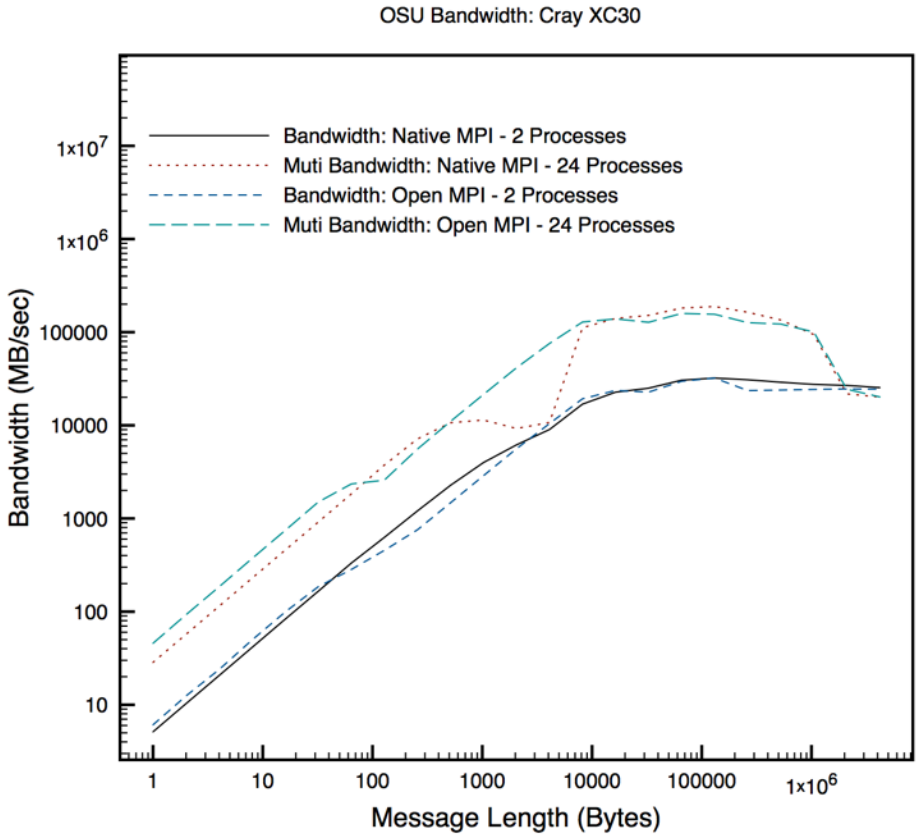
# Shared Memory P2P Latency



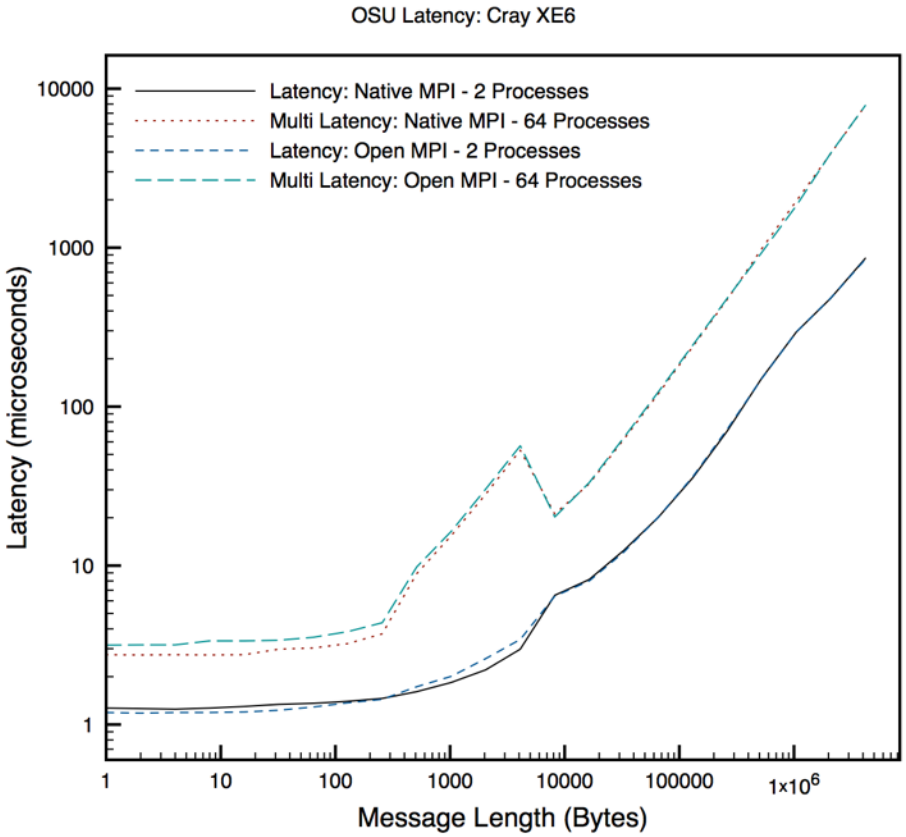



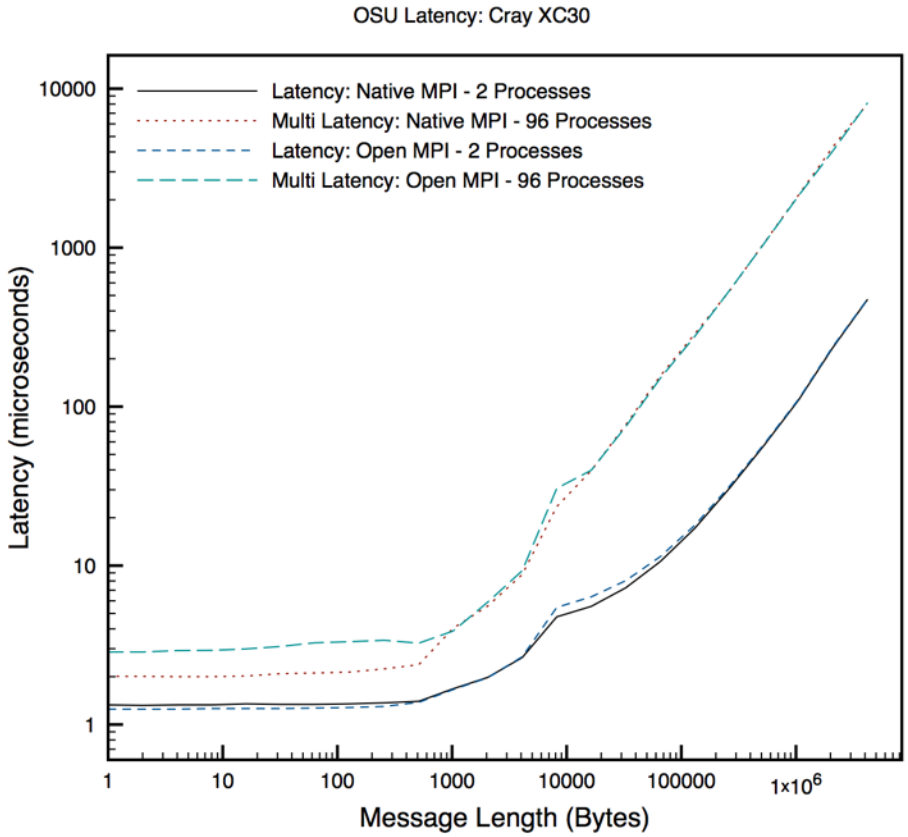
# Shared Memory P2P Bandwidth



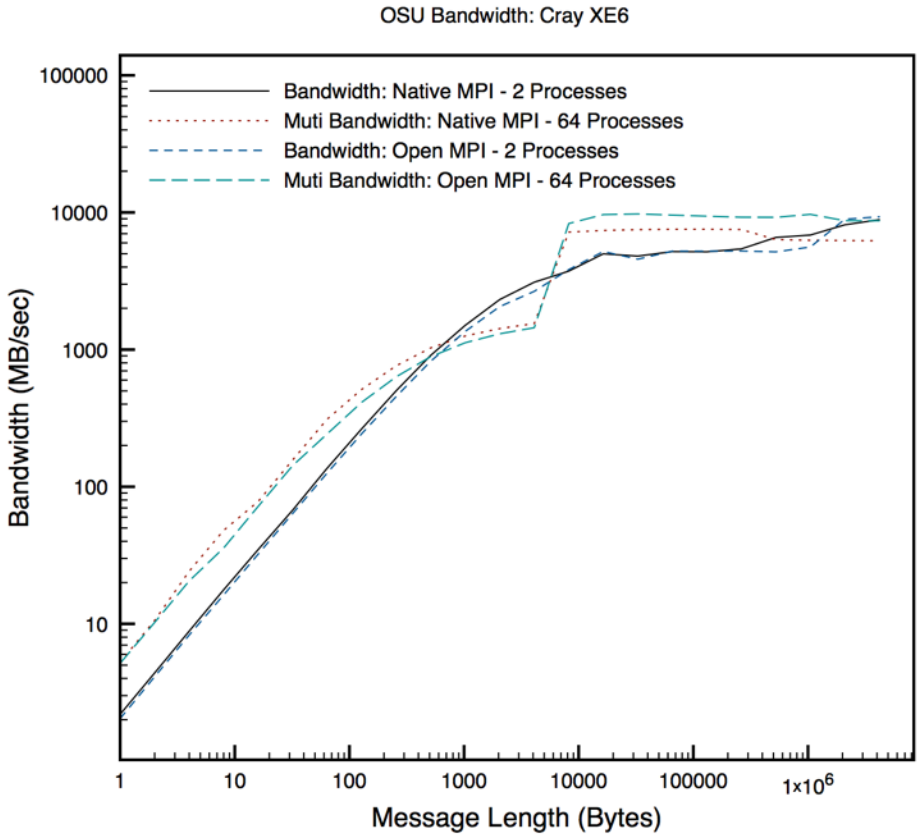



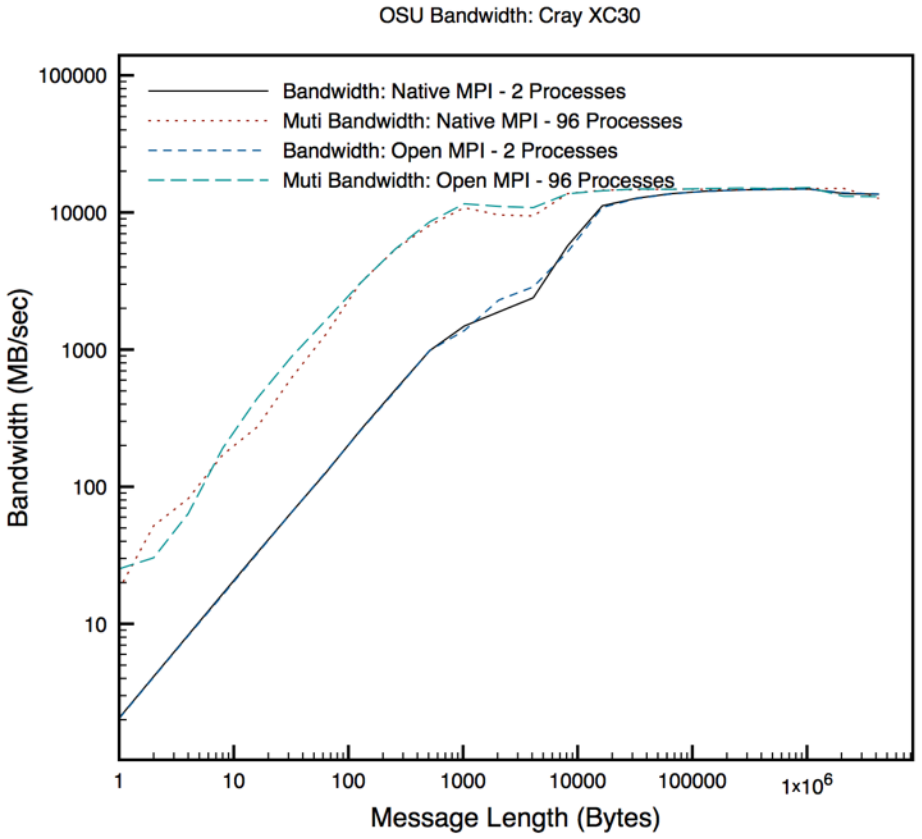
# uGNI P2P Latency



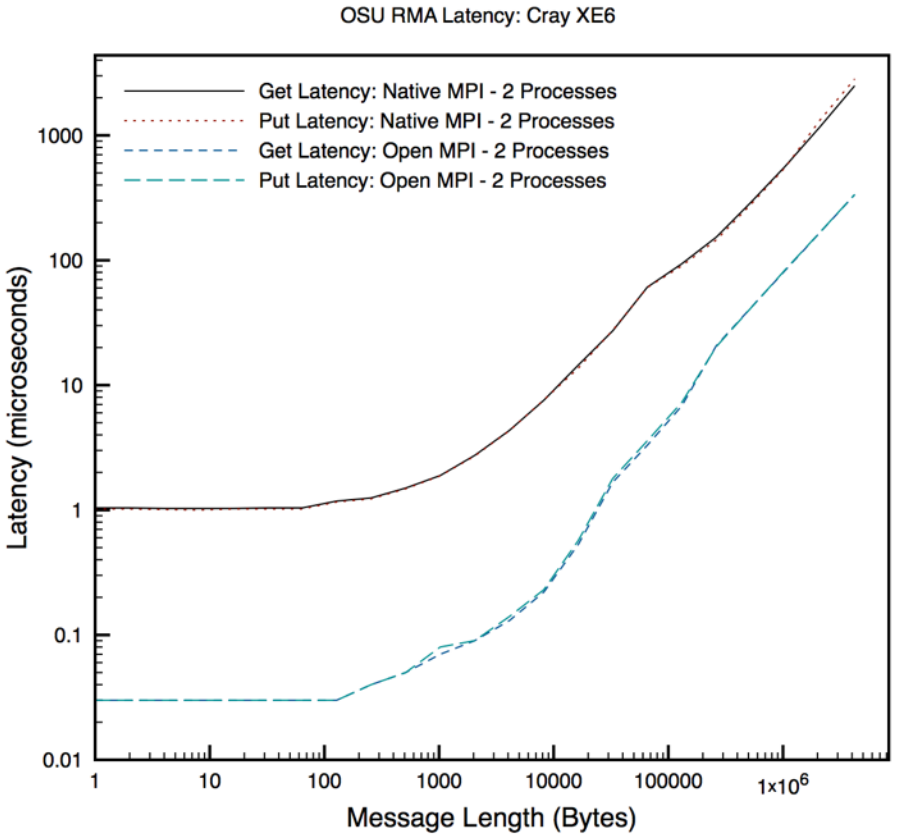
# uGNI P2P Bandwidth

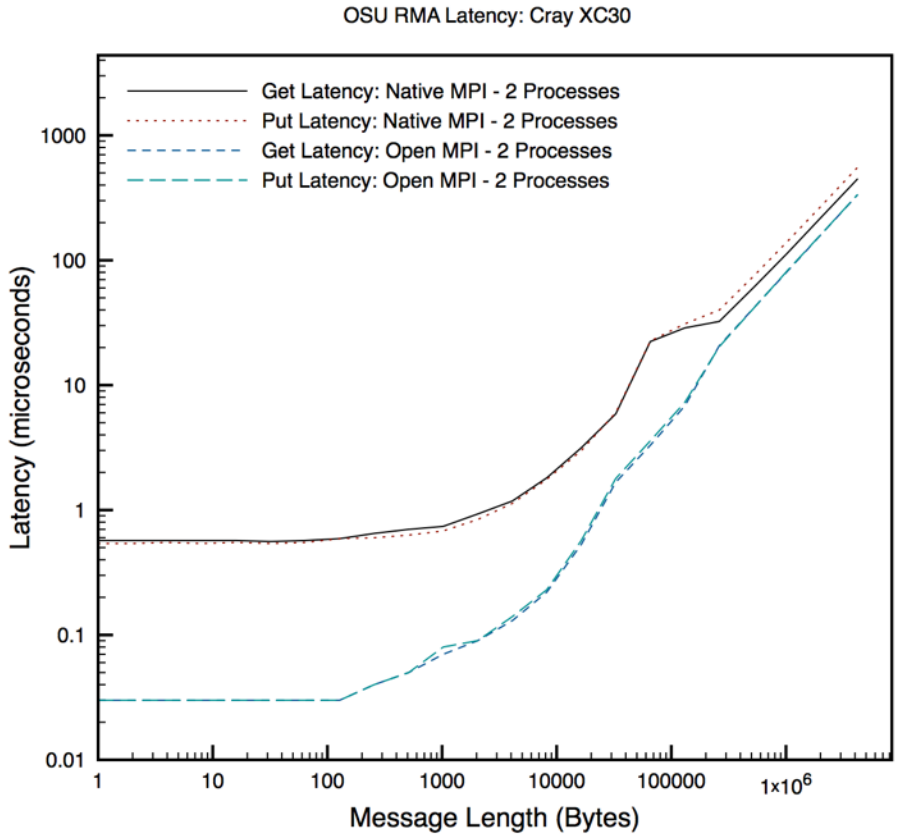


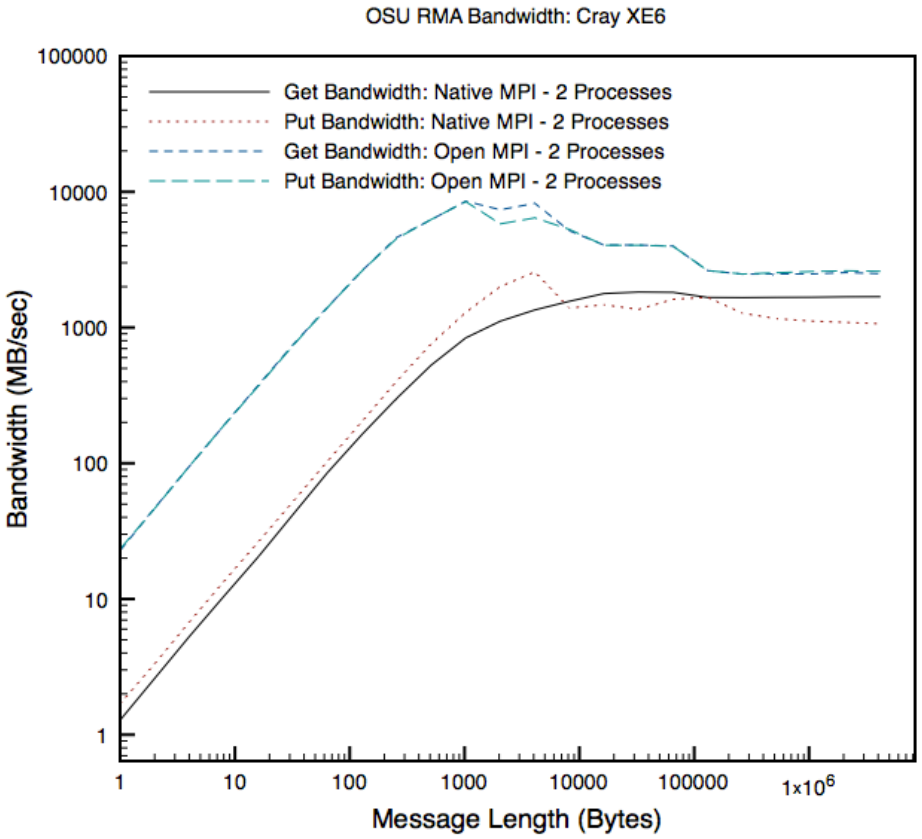
# Performance - One-sided

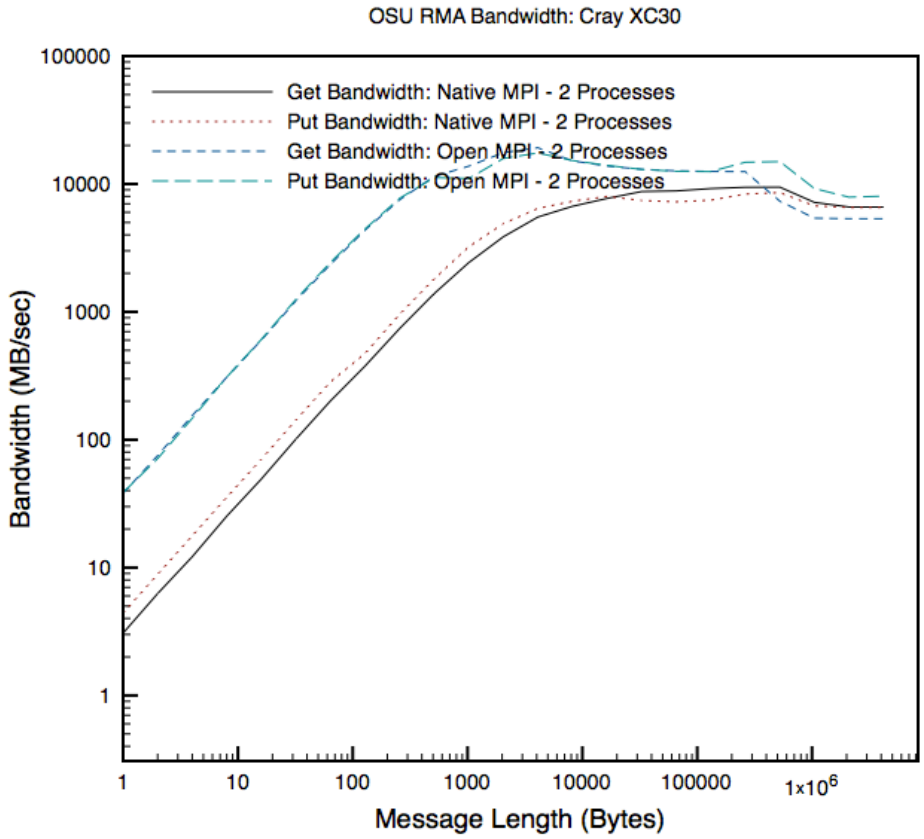
# Shared Memory RMA Latency



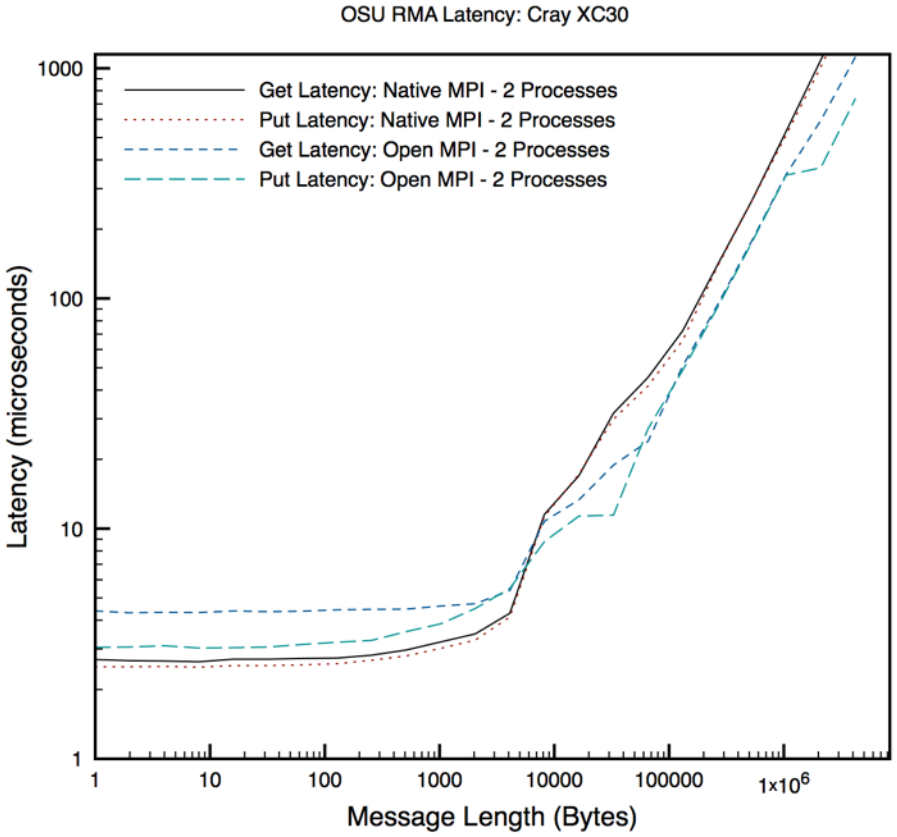



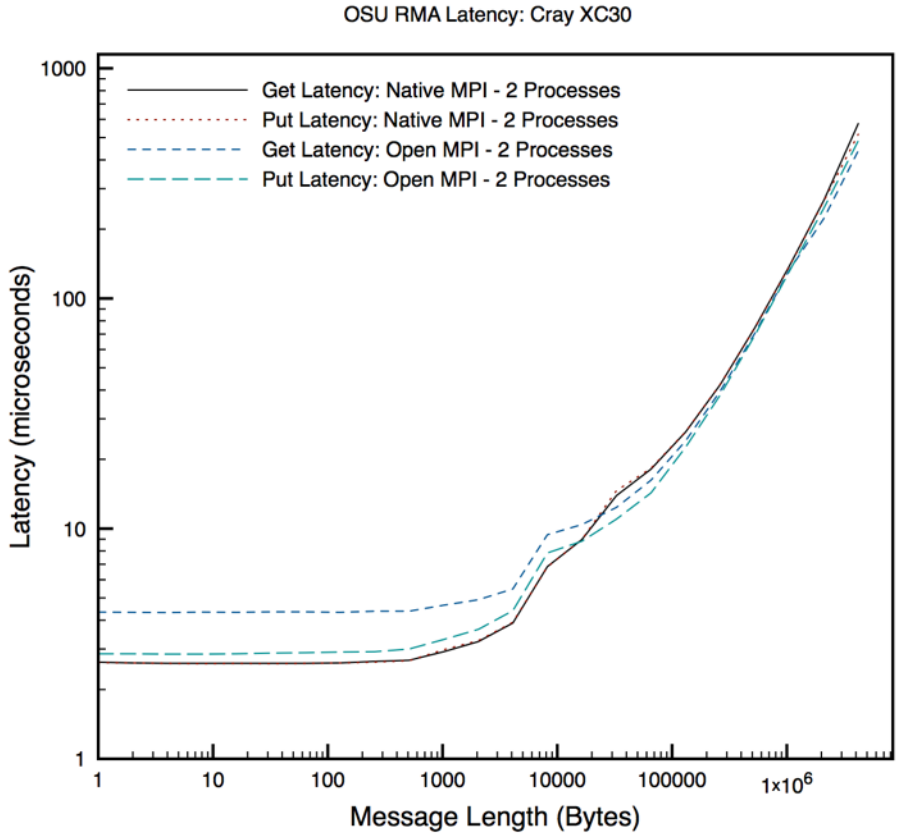
# Shared Memory RMA Bandwidth



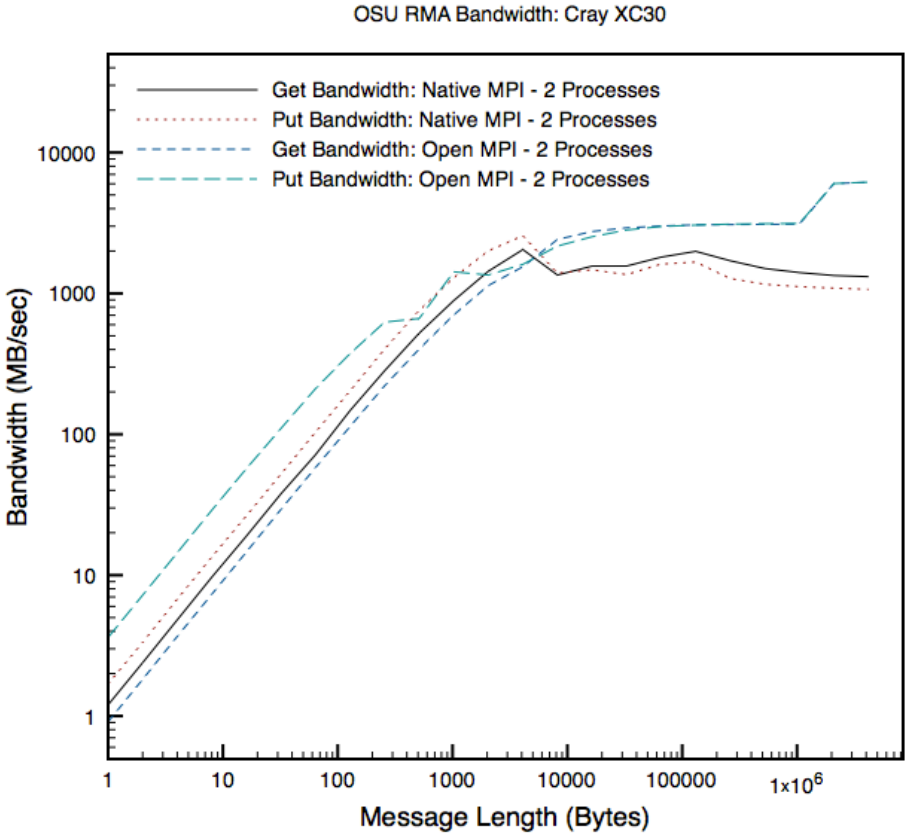



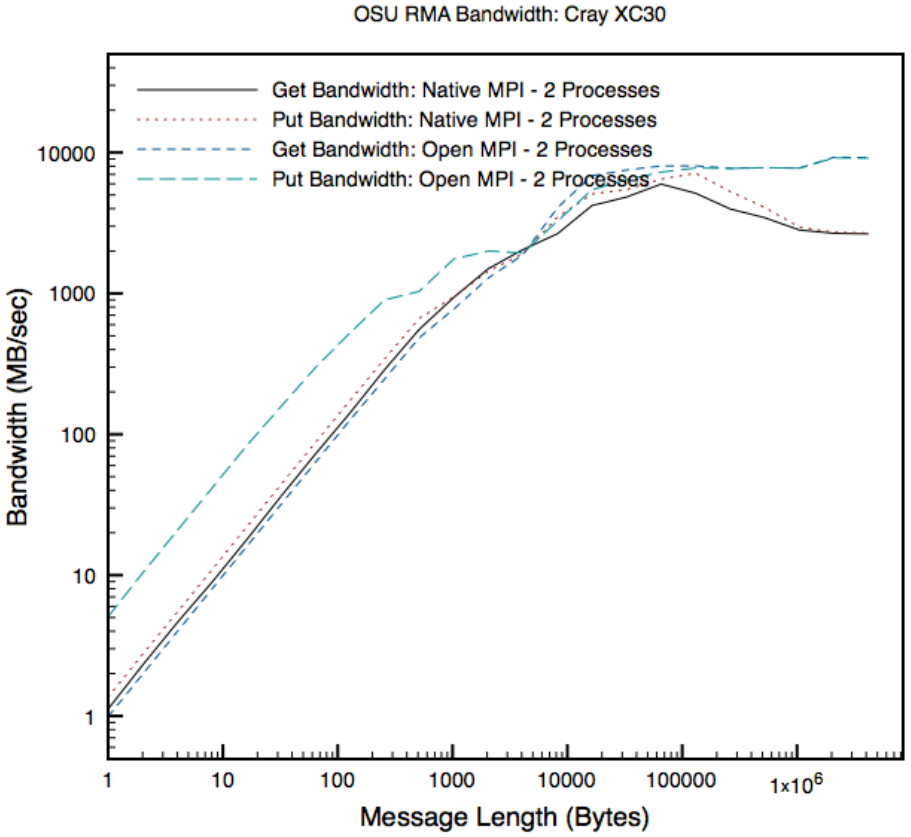
# uGNI RMA Latency



# uGNI RMA Bandwidth



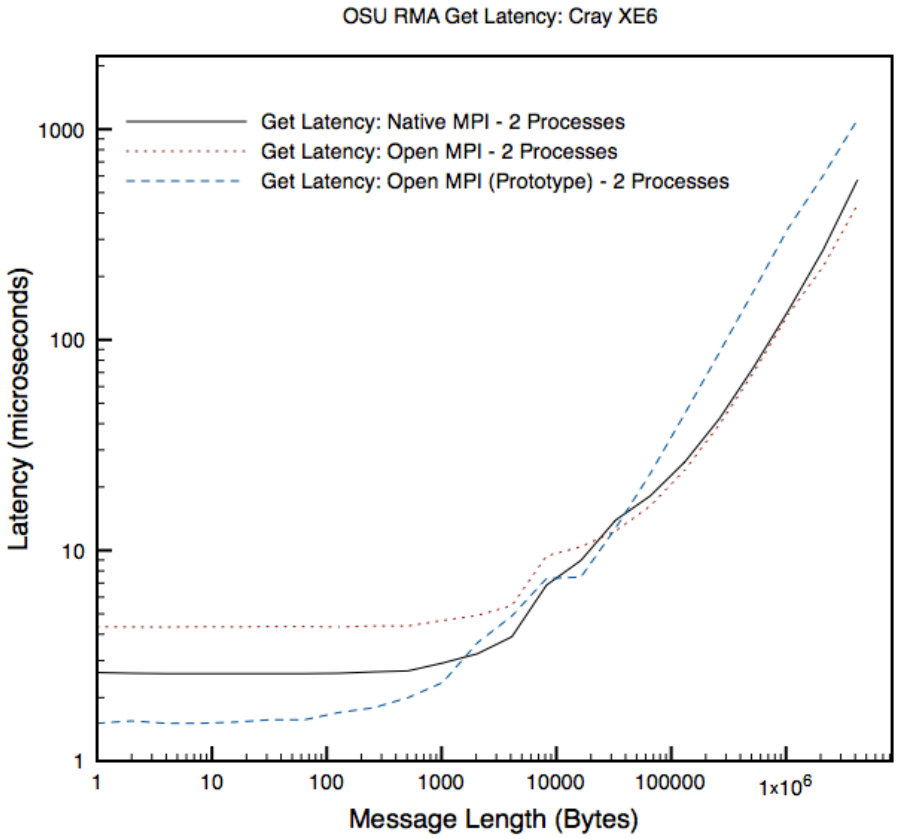
# Conclusions

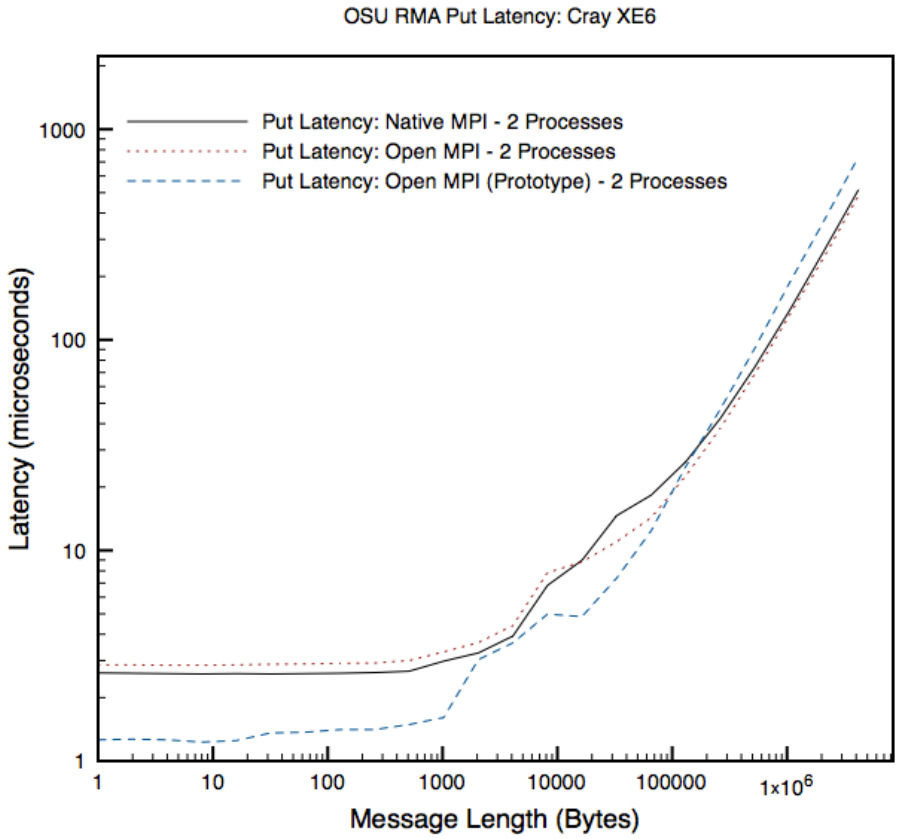
- Similar performance to the native MPI
- Fully supports both Gemini and Aries networks

# Ongoing/Future Work

- Improve launch scalability
  - Reduce memory requirements
  - Improve launch times with both mpirun and aprun
- Enhanced one-sided support for Gemini/Aries
  - Directly make use of RDMA and atomics in uGNI
  - Make use of XPMEM for on-node one-sided
- Better integration with Cray programming environment
- Bug fixes

# Ongoing/Future Work



# Acknowledgements


- The authors would like to thank Alliance for Computing at Extreme Scale (ACES) management and staff for their support. Work supported by the Advanced Simulation and Computing program of the U.S. Department of Energy's NNSA. Los Alamos National Laboratory is operated by Los Alamos National Security, LLC for the NNSA. The authors would also like to thank the Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. Additionally, the authors would like the thank National Energy Research Scientific Computing Center for use of their Edison system.

# Thanks!

# Questions?

- Questions?
- Comments?

# References


- [1] Open MPI. Apr. 28, 2014 <www.open-mpi.org>
- [2] S. Gutierrez, N. Hjelm, M. Venkata, and R. Graham, “Performance evaluation of open mpi on cray xe/xk systems,” in High-Performance Interconnects (HOTI), 2012 IEEE 20th Annual Symposium on, Aug 2012, pp. 40–47.