# Recent progress on the Open MPI implementation

George Bosilca

University of Tennessee, Knoxville

the OMPI-X team and

the OMPI community

All over


NNSA National Nuclear Security Administration

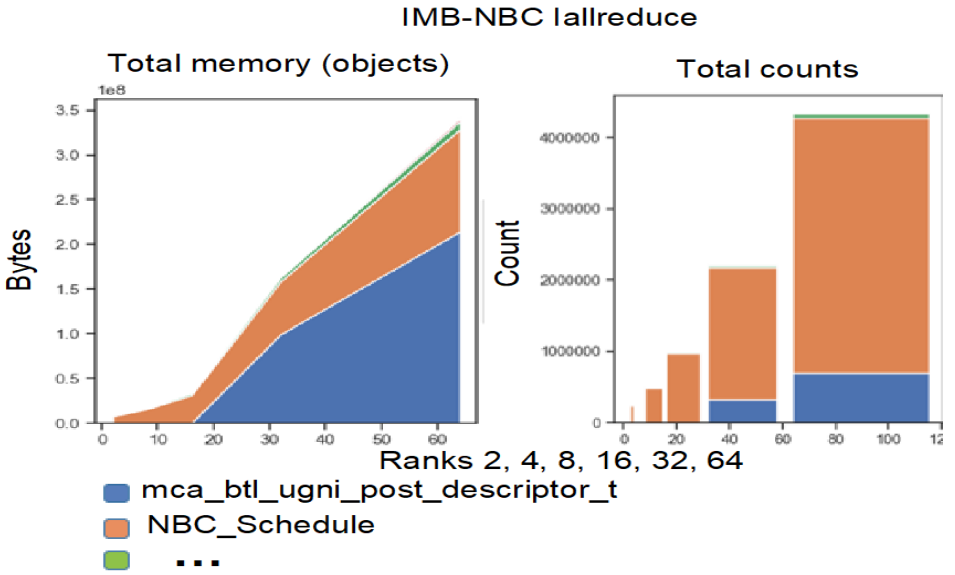
U.S. DEPARTMENT OF ENERGY | Office of Science

# OMPI-X overview

- The OMPI-X project ensures that the Message Passing Interface (MPI) standard, and its specific implementation in Open MPI meet the needs of the ECP community in terms of performance, scalability, and capabilities or features.
  - The OMPI-X team is active in both the MPI Forum and the Open MPI community
- The OMPI-X project is focusing on prototyping and demonstrating exascale-relevant proposals under consideration by the MPI Forum, as well as improving the fundamental performance and scalability of Open MPI, particularly for exascale-relevant platforms and job sizes.
  - MPI users will be able to take advantage of these enhancements simply by linking against recent builds of the Open MPI library.

# Progress so far

- Visible goodies
  - MPI_T extensions
    - Memory allocations
  - Software Performance Counters
    - Out-of-sequence messages, time to match, number of unexpected, instant bandwidth, collective bins
    - Exposed via MPI_T, or PAPI SDE or shared file via PMIx plugins
  - Quality control (CI, nightly testing, …)
- Invisible goodies
  - Added communication support for UCX both as a PML (message layer) and as a BTL (byte transfer layer).
    - OpenUCX (openucx.org) is now the preferred method for InfiniBand support
  - RMA improvements: Faster, more scalable and thread safe
  - Architecture benefits: datatype engine (taskfied, AVX), improved MPI ops (architecture aware, AVX)
  - Performance and scalability of Open MPI
    - Improved MPI matching, sparse groups, lazy initialization

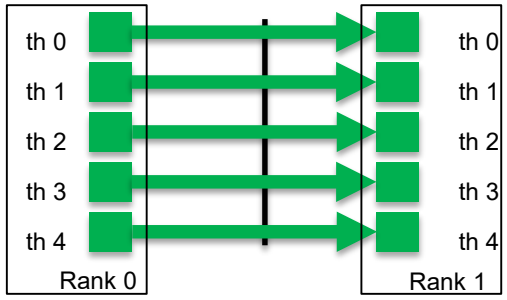


# Ongoing efforts

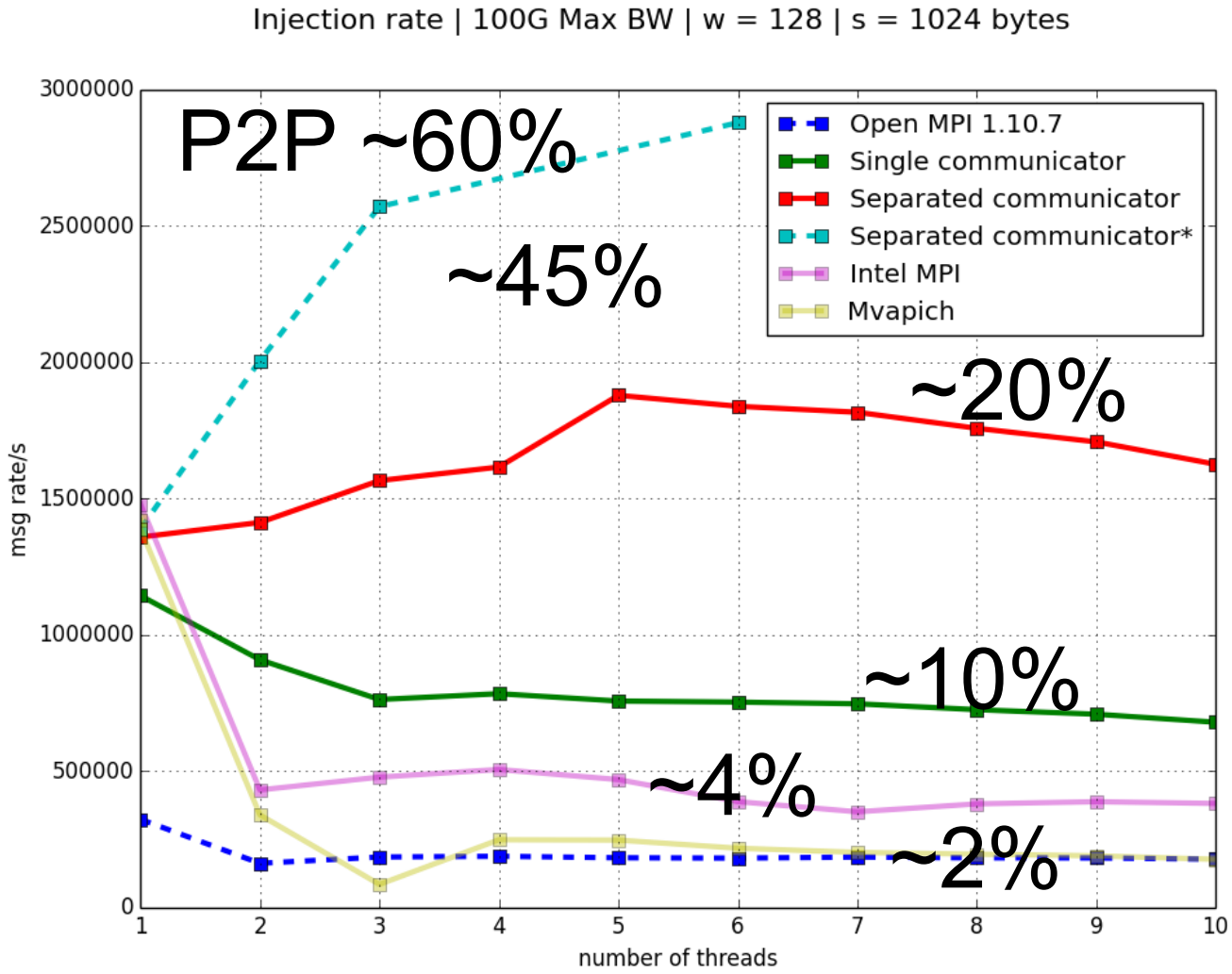
- MPI Explorations (cool things not yet blessed by the MPI standard)
  - Persistent communications
  - Better threading support (more to come)
  - MPI_Finepoints
  - Resilience
    - MPI_Reinit and ULFM 2.0 implementation (in sync with master)
    - More scalable revocation, agreement
    - Non-blocking constructs for resilience, process management
  - Noise resistant, topology aware collective communication

# MPI + X

- Runtime level
  - Involvement in PMIx
  - Involvement in the MPI Forum topology WG
  - Topology awareness plus improved support for process/thread placement
- Better isolation on the same process - MPI_Session
- Communication level
  - Threading improvements
  - Better completion
    - Exploration: Build other completion mechanisms than MPI_Wait* / MPI_Test*
    - Exploration: Callbacks




# Threading support




Open MPI 4.1 (with different communicators)

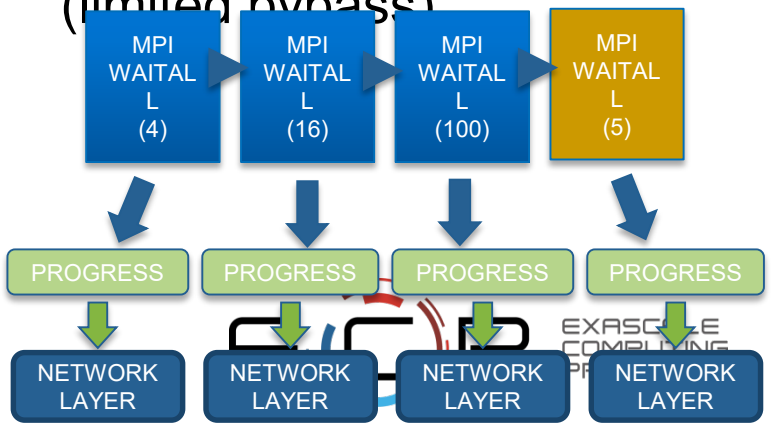
Open MPI 4.0 (with different communicators)

Open MPI 4.0

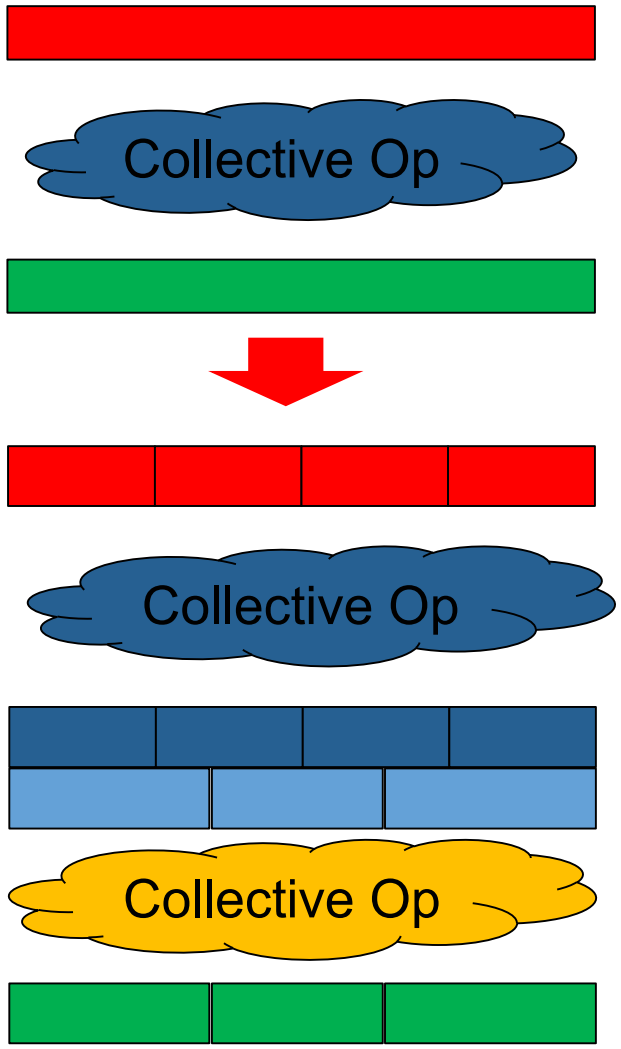
Intel MPI

Open MPI 1.10.7
MVAPICH

- Improvements:
  - Synchronization primitive
  - Unrestricted progress (protections done at the lowest level)
  - Credit management
  - Requests memory management
  - Out-of-sequence management (limited bypass)



# Collective communications




- Dataflow collective: different algorithms compose naturally (using a dynamic granularity for the pipelining fragments)
- Architecture aware: Each level reshape tuned collective to account for architecture capabilities
- The algorithm automatically adapts to network conditions
- Resistant to system noise
- MCA allows vendor specific collective libraries (hear SHARP, NCCL)

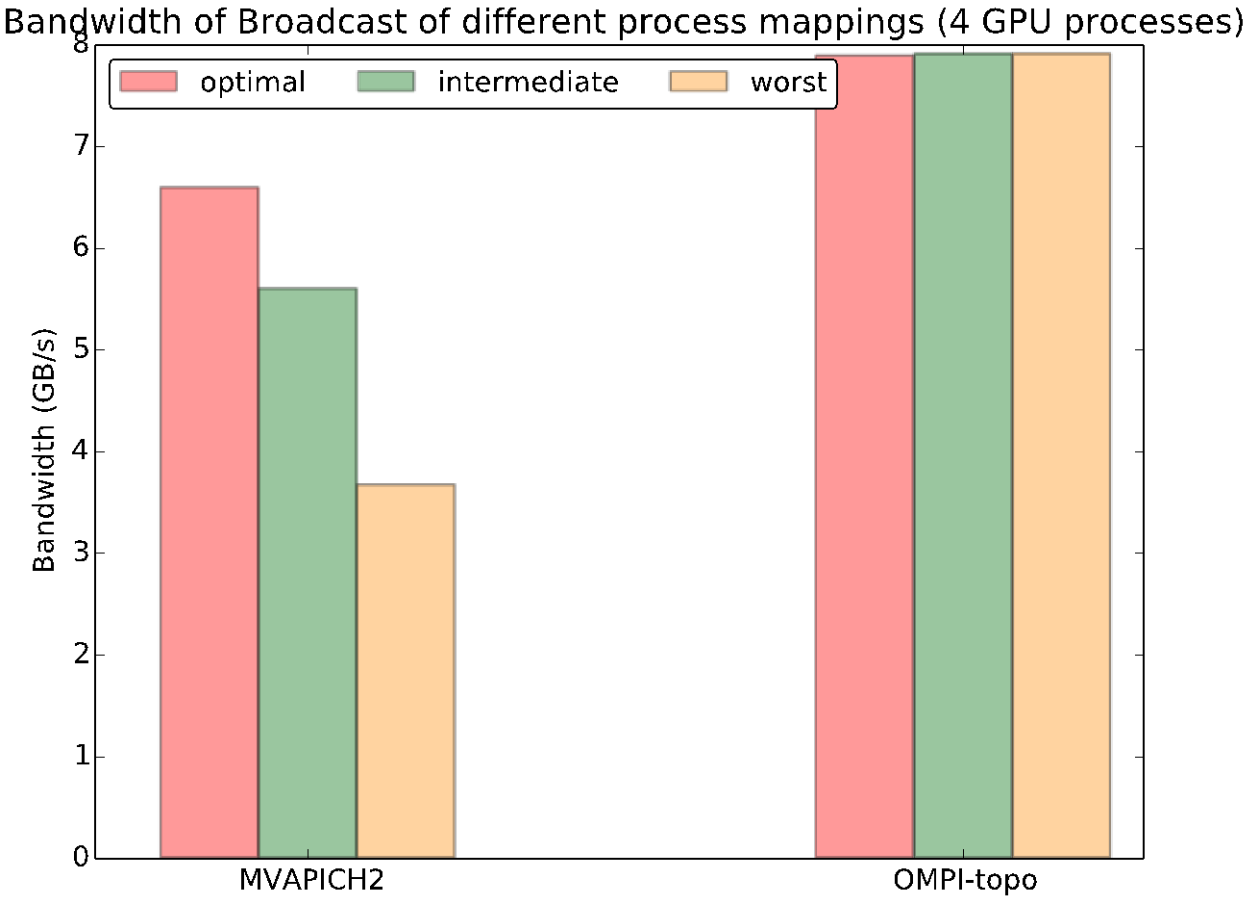
# Collective Communication

Process location

Noise Reduction

Shared Memory

Hybrid Architecture



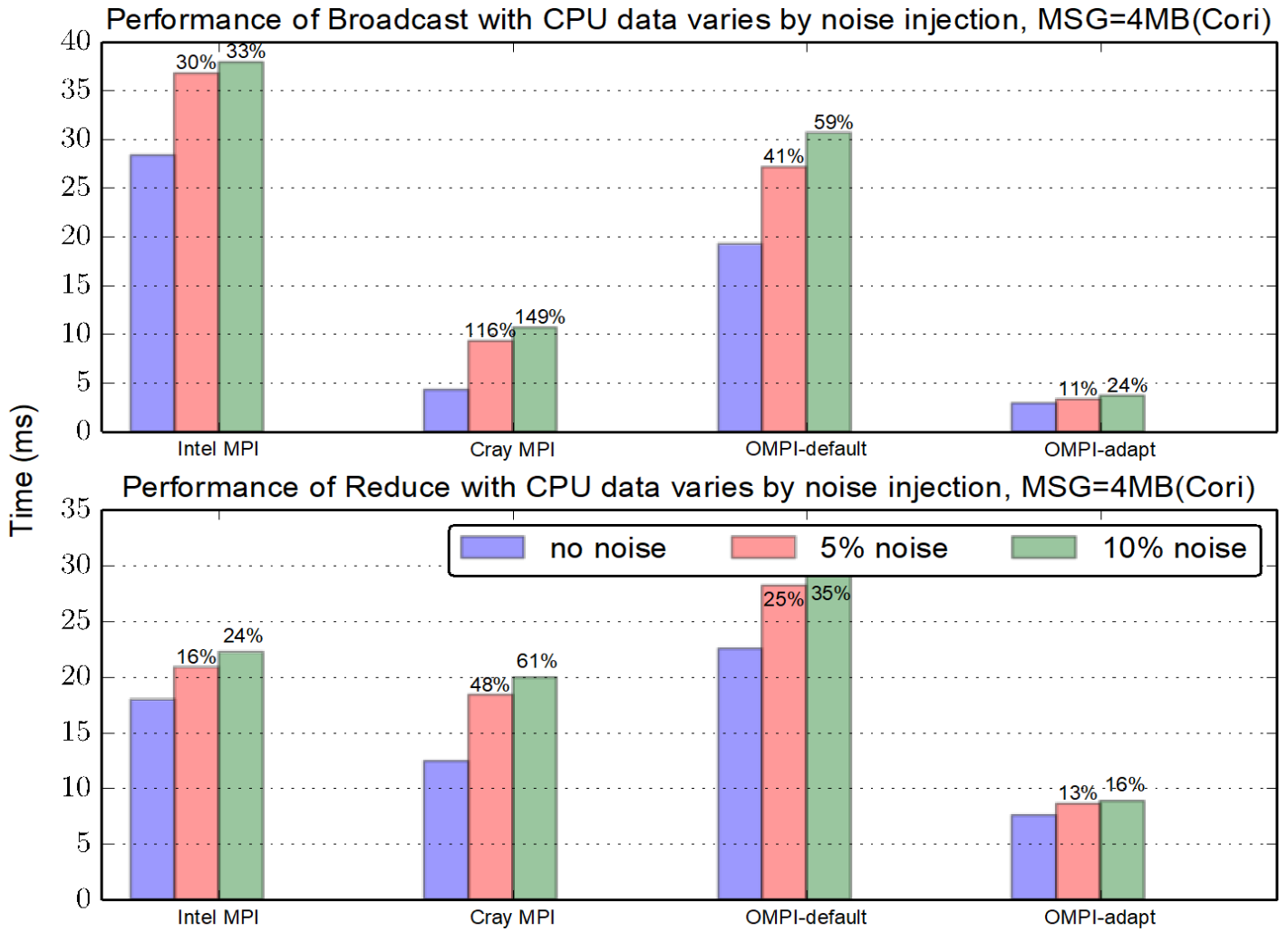
# Collective Communication

Process location

Noise Reduction

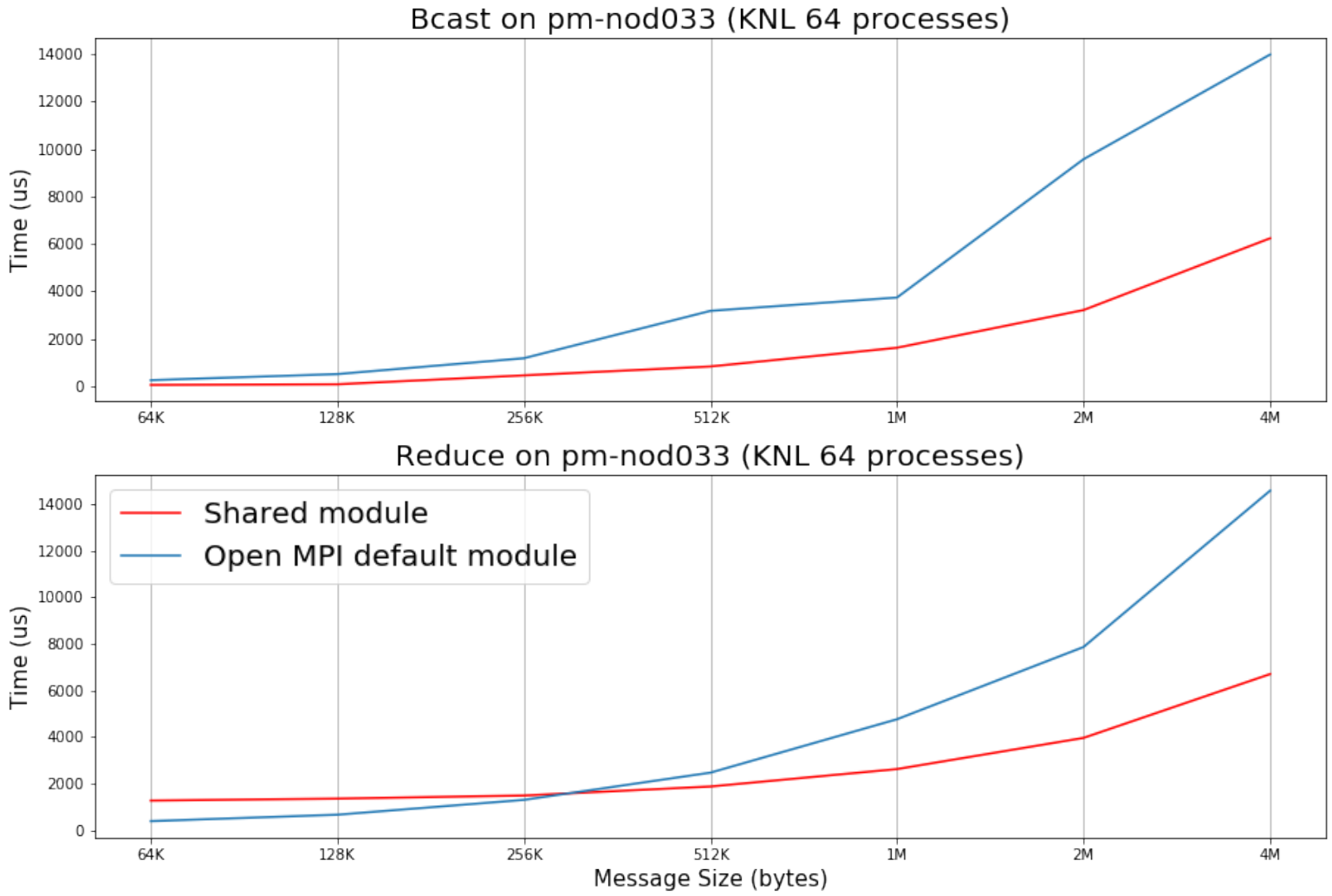
Shared Memory

Hybrid Architecture



# Collective Communication

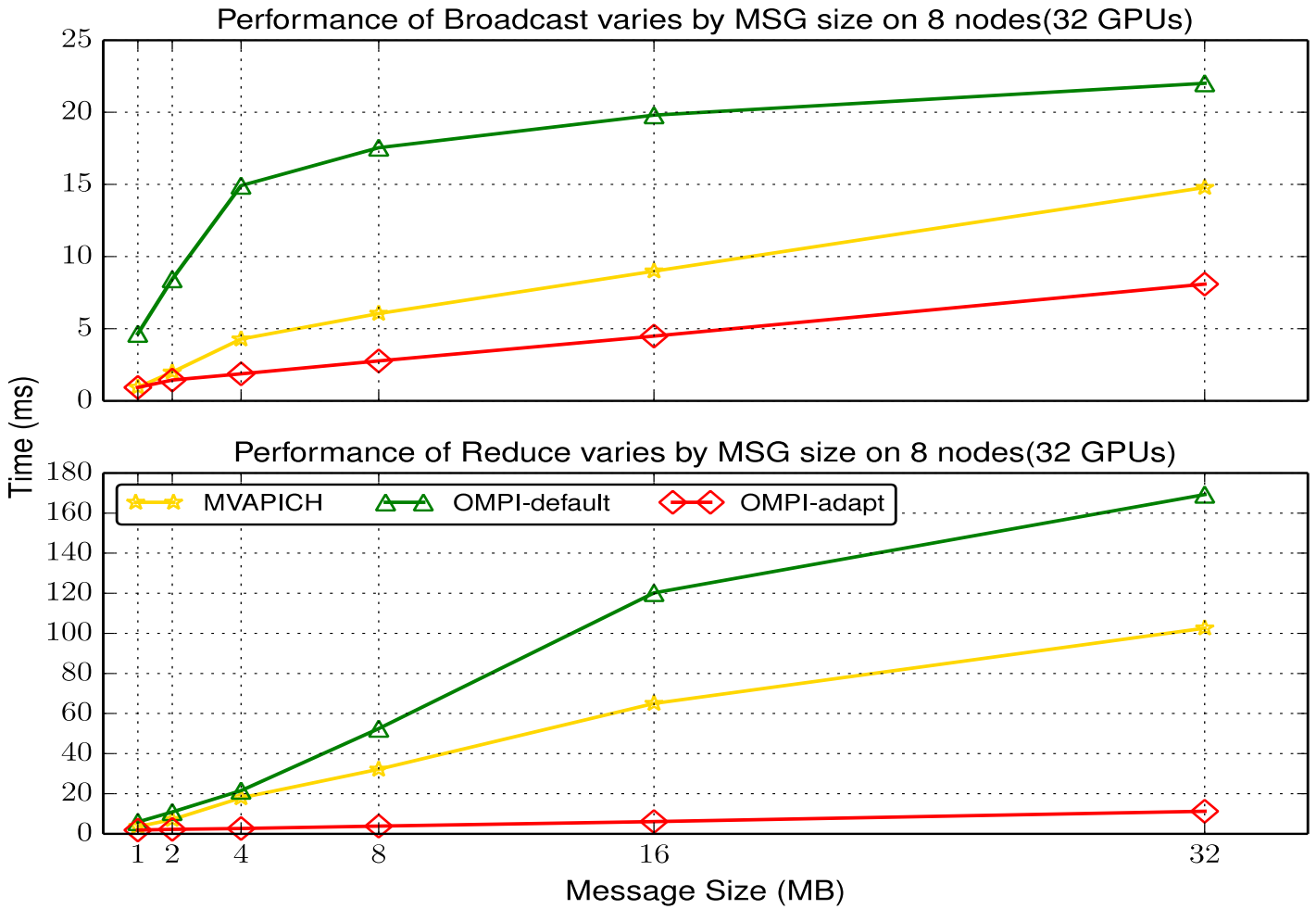
Process location
Noise Reduction
Shared Memory
Hybrid Architecture



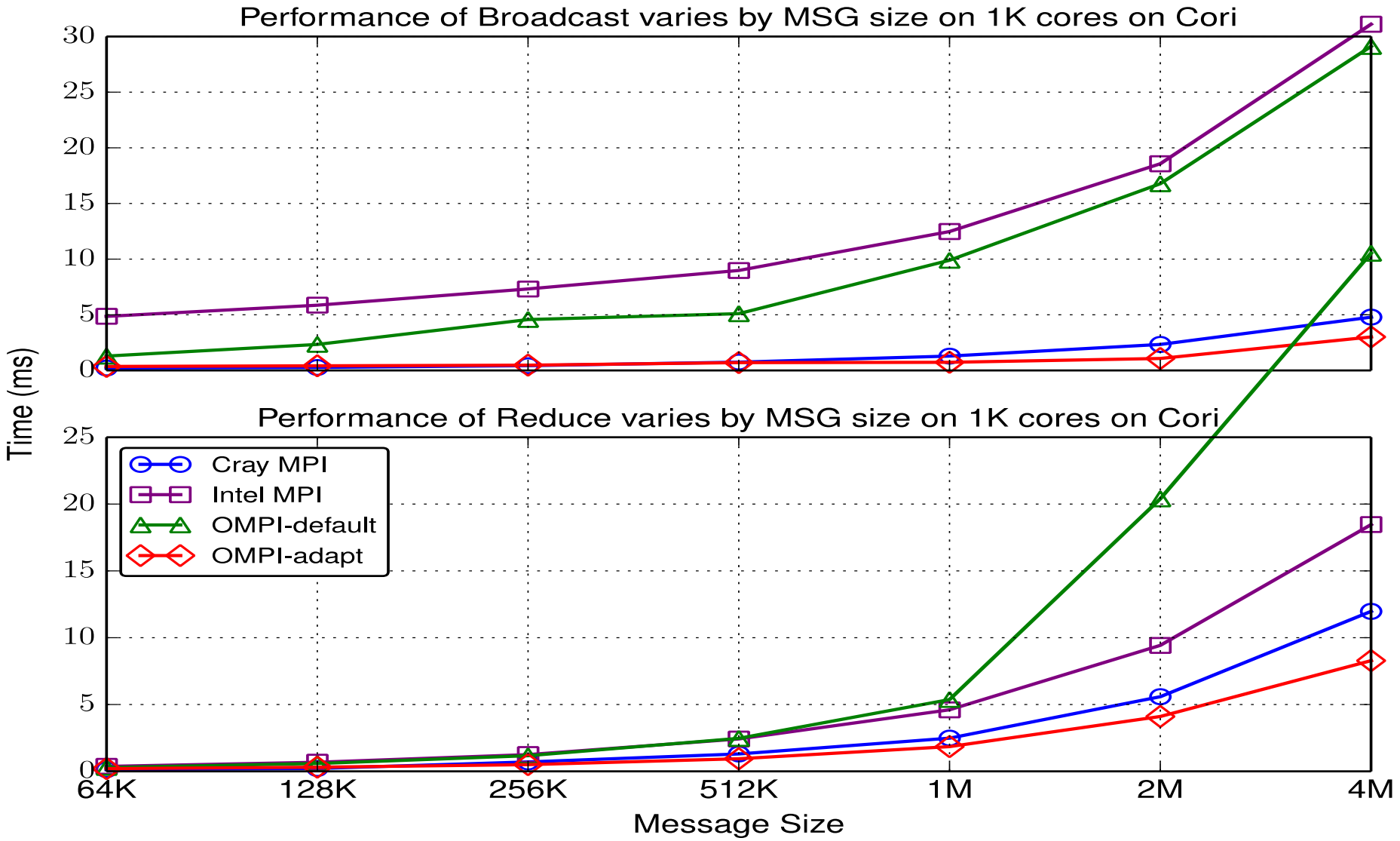
# Collective Communication

Process location
Noise Reduction
Shared Memory
Hybrid Architecture

PSG Cluster:
4*K40/node
FDR IB



# Collective Communication



# Resilience - User Level Failure Mitigation (U' FM)

- Move the underlying resilient mechanisms outside ULFM/OMPI
  - Failure detector and reliable broadcast in PMIx
  - Used in OMPI ULFM and SUNY OpenSHMEM
- ULFM 2.1 released
  - Based on OMPI master (will remain in sync)
  - Transition to integrate ULFM in OMPI master
- Scalable fault tolerant algorithms demonstrated in practice for revoke, agreement, and failure detection (SC'14, EuroMPI'15, SC'15, SC'16)