

Fujitsu Development Status and Some Topics towards Next MPI development

2015/06/23

Shinji Sumimoto

Fujitsu Ltd.

- Third Party Contribution Agreement Status
- PRIMEHPC FX100 Overview
- Some Topics of Next MPI Library Development

- We are still working on internal negotiation process, because we have to apply new process including request for decision process.
 - We need a couple of month to process. Sorry for late.
- After the internal negotiation process, Fujitsu will join the Open MPI Development Team.
 - Soon... (I hope by the end of Sept.2015)

PRIMEHPC FX100 OVERVIEW

K computer and Fujitsu PRIMEHPC series

- Massively parallel machines with single-socket per node
 - Increased core-count and enhanced HPC instruction extension
 - Eight memory channels for high throughput
- PRIMEHPC FX100 is the successor of FX10
 - Realizing 100PFlops Class Post-Petascale Systems

K computer



8 core
HPC-ACE
8 DDR3-DIMM
Tofu interconnect

FX10



16 core
HPC-ACE
8 DDR3-DIMM
Tofu interconnect

FX100



32 core
HPC-ACE2
8 Hybrid Memory Cube
Tofu interconnect 2

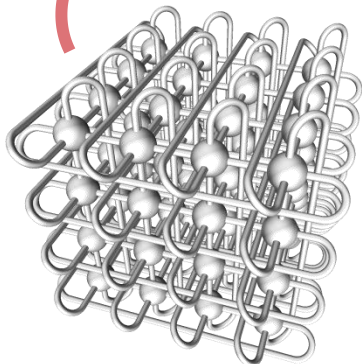
2010

2012

2015

Fujitsu designed SPARC64™ XIfx

- ◆ 32 + 2 core CPU
- ◆ HPC-ACE2 support
- ◆ Tofu2 integrated
- ◆ Over 1TFlops

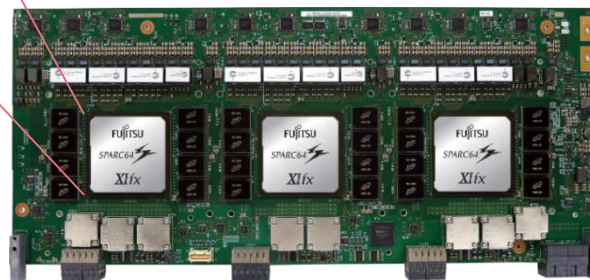
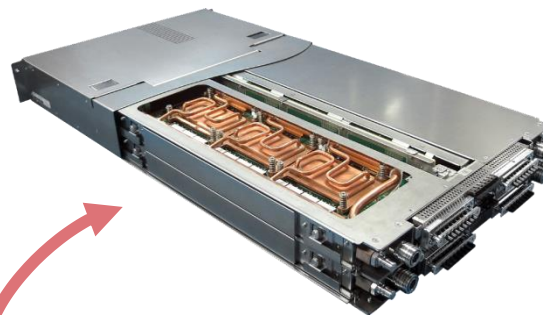


Tofu Interconnect 2

- ◆ 12.5 GB/s X 2(in/out)/link
- ◆ 10 links/node
- ◆ Optical technology

Chassis (12 CPUs)

- ◆ 1 CPU/1 node
- ◆ 12 nodes/2U Chassis
- ◆ Water cooled



CPU Memory Board

- ◆ CPU x 3
- ◆ 3 x 8 Micron's HMCs
- ◆ 8 Finisar's opt modules, BOA, for inter-chassis connections

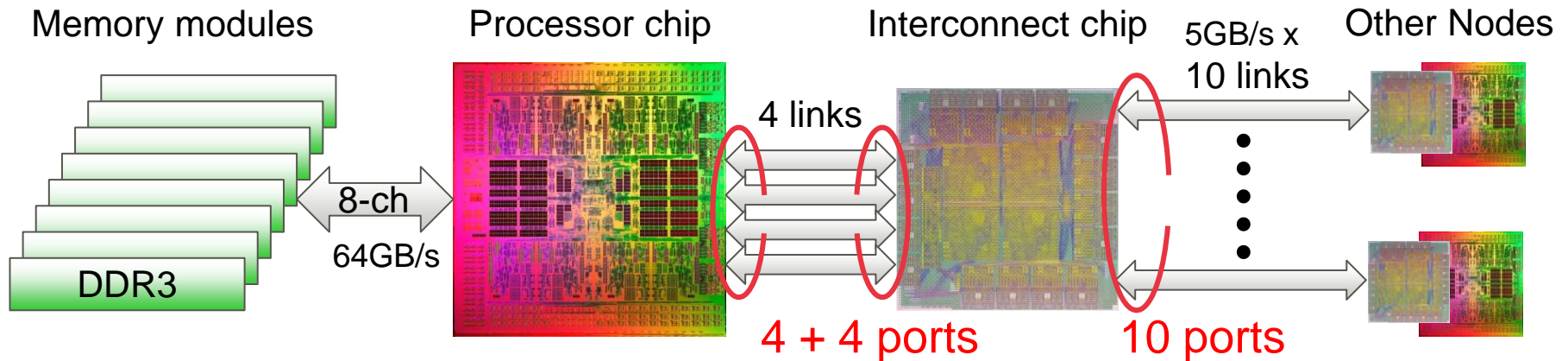


Cabinet

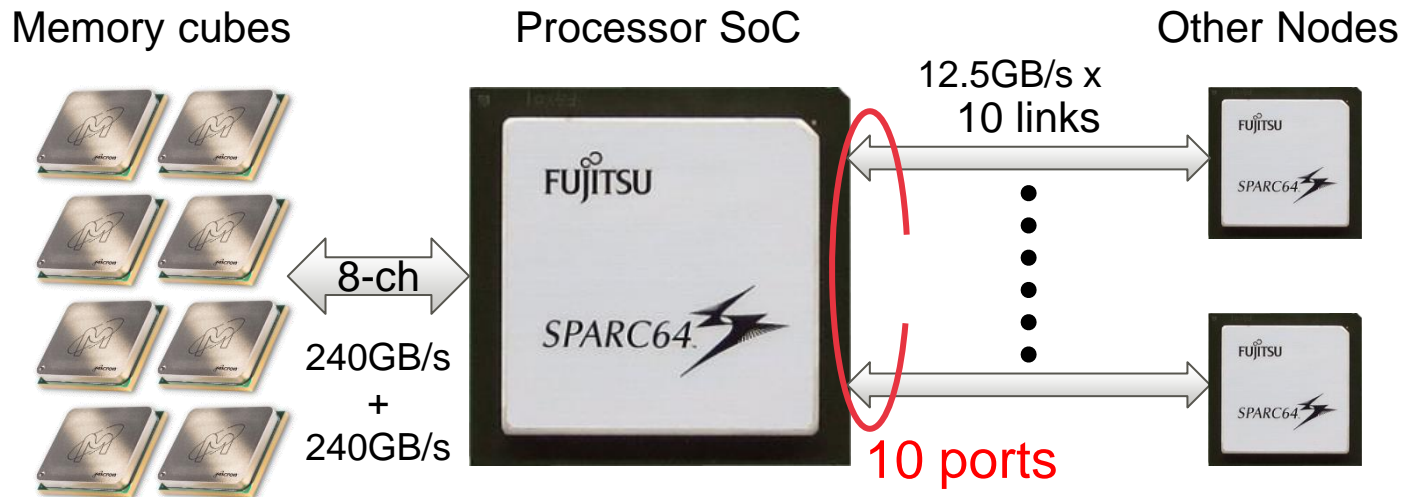
- ◆ 216 nodes/cabinet
- ◆ High-density
- ◆ 100% water cooled with EXCU (option)

PRIMEHPC FX100: System-on-Chip Integration

- Tofu1 was implemented as a discrete chip



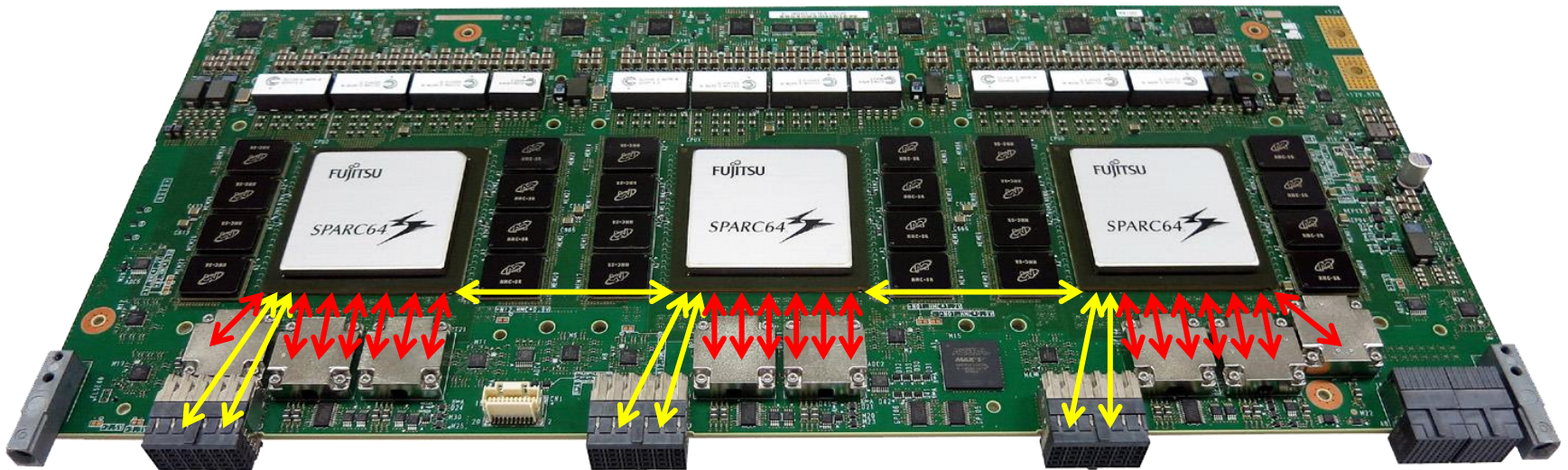
- Tofu2 is integrated into a processor SoC



- Number of link ports per node decreased from 18 to 10

PRIMEHPC FX100: Transmission Technology

- Link speed increases from 40 Gbps to 100 Gbps
 - Tofu1: 8 lanes \times 6.25 Gbps \times 8b/10b = 40 Gbps
 - Tofu2: 4 lanes \times 25.78125 Gbps \times 64b/66b = 100 Gbps
- Number of lanes per node decreases from 144 to 40
- 2/3 of links uses optical transceivers
 - 1 out of 3 nodes uses **6 optical links** + **4 electrical links**
 - 2 out of 3 nodes use **7 optical links** + **3 electrical links**



Some of Slides from SC14 ExaMPI Workshop Slides

SOME TOPICS OF NEXT MPI LIBRARY DEVELOPMENT

- Higher Bandwidth and Lower Latency on many core systems
 - Cache Injection for Lower Latency
 - Single Core CPU Copy Performance Issue
- Memory Usage of MPI Library must be reduced dramatically
 - Memory Saving Technologies of K computer is not enough:
I talked the last OMPI Developer Meeting
- Better Power Consumption Saving
 - Reducing Number of MPI Processing Instructions:
Simplified MPI Architecture

- Tofu2 supports Injecting received data into L2 cache directly

	Tofu1	Tofu2
Memory bypass (sender)	✓	✓
Memory bypass (receiver)		✓

- Injection flag On/Off is indicated by the sender
- Communication latency is reduced by 0.16 usec

Injection flag	Half round-trip latency
Off	0.87 usec
On	0.71 usec

- The evaluations used the Verilog RTL codes for the production
- The estimated communication pattern is Ping-Pong of Put transfer

Background of High Performance Communication on Many Core Processor for Exa-scale

- Communication Bandwidth of Interconnect continues to increase:
 - Tofu(5+5)GB/s x 4 → Tofu2(12.5+12.5)GB/s x 4
 - Multiple RDMA engines: 4 RDMA engines on Tofu and Tofu2
- Single CPU Core Processing Frequency will not increase dramatically because of Power Wall
 - This is because many core CPUs become to use widely
 - Xeon Phi: around 1GHz, FX10: 1.8GHz
- The problem is memory copy performance of single CPU core will not increase dramatically!

Issues of Single Core Memory Copy Performance Limitation

- Two Major Issues in case of Network Bandwidth > Single CPU Copy Bandwidth:
 - Simple Intra-Node Communication
 - Simple Communication such as MPI_Send, MPI_Recv
 - Collective Communication especially handling multi-rail network
 - Simple Communication such as MPI_Bcast
 - Communication with Arithmetic Computation such as MPI_Reduce
- Solution
 - Hardware Offload: DMA Engine, Loopback Data Transfer
 - Multi Threaded Communication Processing

- MVAPICH2-MIC: A High-Performance MPI Library for Xeon Phi Clusters with InfiniBand, by **Sreeram Potluri** at Extreme Scaling Workshop, August 2013.

[http://nowlab.cse.ohio-state.edu/publications/
conf-presentations/2013/Sreeram-XSCALE13.pdf](http://nowlab.cse.ohio-state.edu/publications/conf-presentations/2013/Sreeram-XSCALE13.pdf)

■ Approach:

- Short Message Size: Using CPU Copy
- Long Message Size: Using SCIF(DMA Engine)

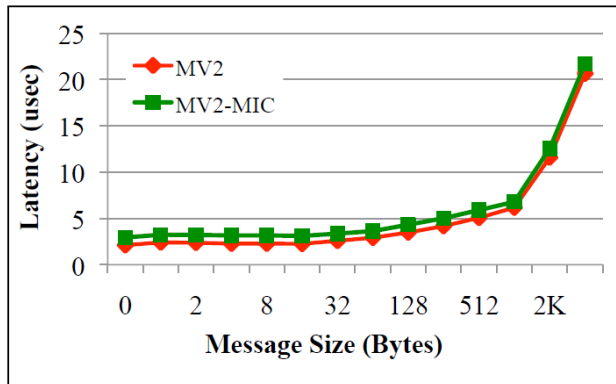
■ Results:

- Pros: Simple Intra-Node Performance
- Cons: DMA engine Resource Bottleneck in case of number of processes
Communication with Arithmetic Computation
- Multi Threaded Communication Processing will be needed

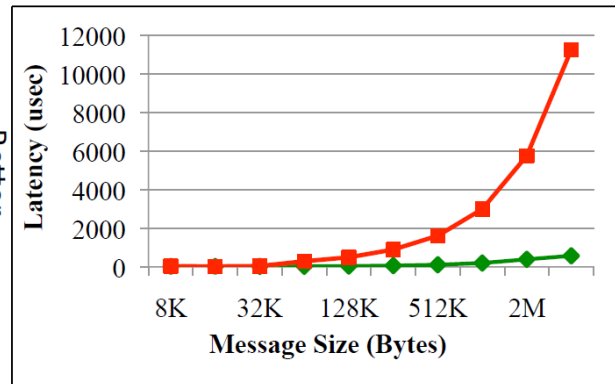
Xeon Phi Case: MVAPICH2-MIC Solution

<http://nowlab.cse.ohio-state.edu/publications/conf-presentations/2013/Sreeram-XSCALE13.p>

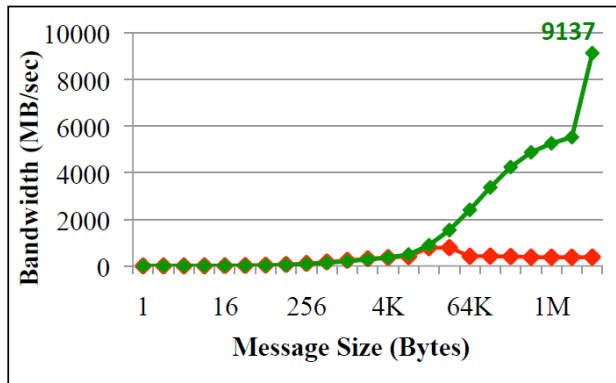
Intra-MIC - Point-to-Point Communication



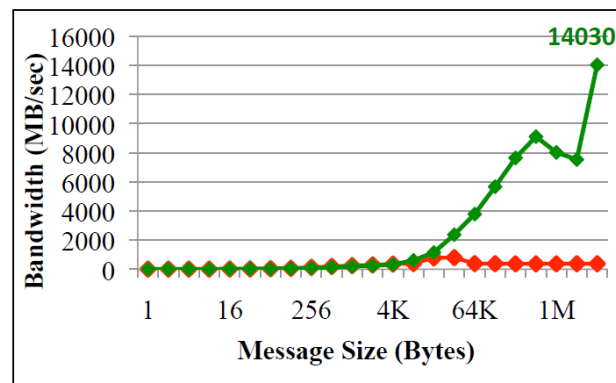
Better



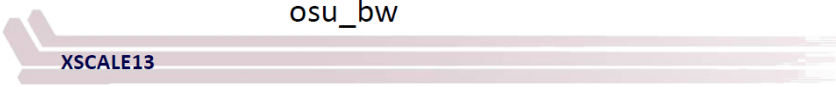
Better



Better



Better



- MT-MPI: Multithreaded MPI for Many-core Environments, Min-Si at ICS-14, June 2014

<http://www.il.is.s.u-tokyo.ac.jp/~msi/pdf/ics2014-mtmpi.pdf>

- Approach: Multi Threaded Communication Processing by OpenMP
 - Derived Data Type Processing for Pack, Un-pack
 - Shared Memory Communication for Long Messages
- Results:
 - Pros: Higher Performance
 - Cons: Depending on number of Idle Threads

MULTI-THREAD OPTIMIZATION IN MPI LIBRARIES

- Assuming to Implement in `MPI_THREAD_FUNNEL` or `MPI_THREAD_SERIALIZED` Environment

- For Simple Intra-Node Communication
 - Increasing Communication Performance by using Multi Threaded CPU Copy.

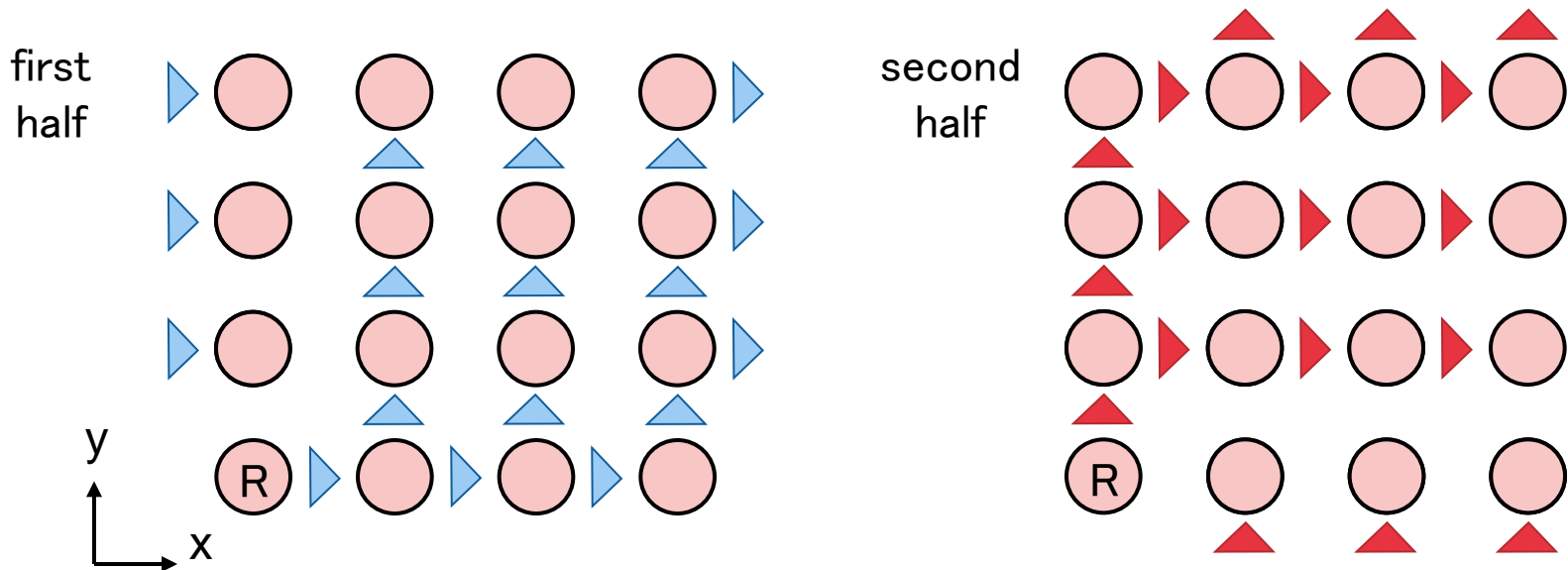
- For Development of Collective Communication especially Handling Multi-Rail Network
 - For Simple Communication such as `MPI_Bcast`
 - Optimization by Multi-Threaded CPU Copies
 - For Communication with Arithmetic Computation such as `MPI_Reduce`
 - Increasing Arithmetic Performance by using multi threads

For General Optimization of Collective Communication

■ Collective Communication In MPI_THREAD_MULTIPLE Environment

- Simple Implementation using MPI_isend, MPI_irecv for each rail in multi-rail network
- Realizing Portable Implementation

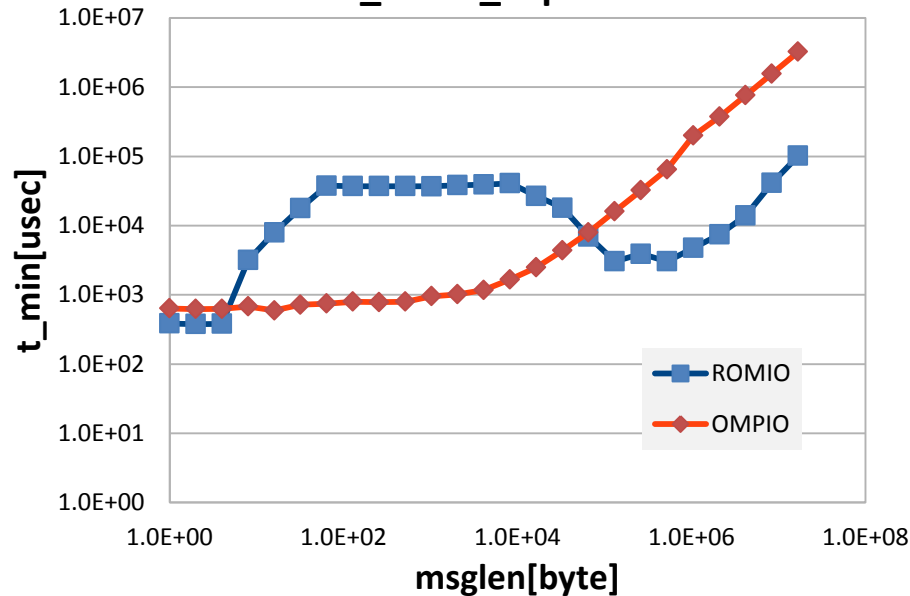
MPI_Bcast by using Long-message algorithm: “Trinaryx3”



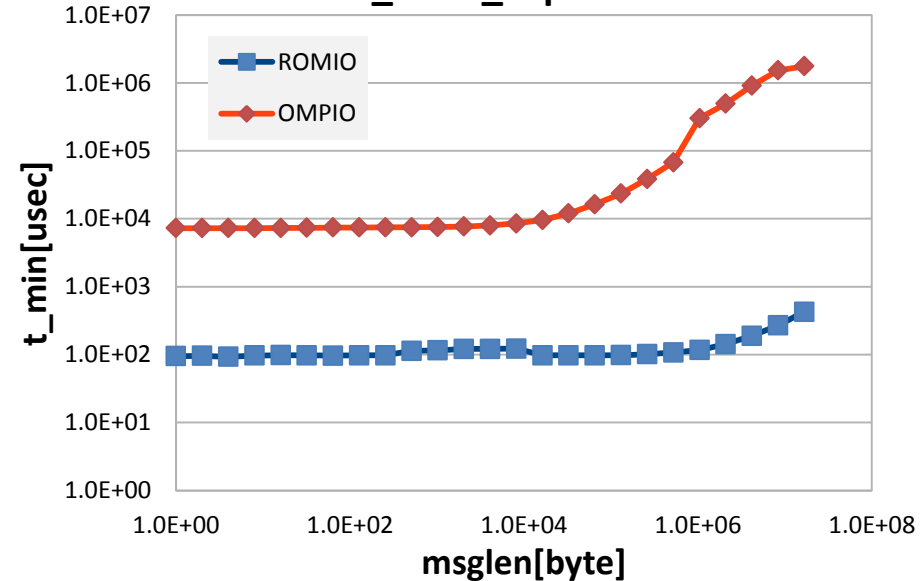
EVALUATION OF MPI-IO OMPIO VS ROMIO

Evaluation Results Using IMB-IO

C_Write_Expl




C_Read_Expl



■ Evaluation Results on Fujitsu MPI Based on 1.8 series on PRIMEHPC FX10 384 nodes

- C_Write_Expl: Better Performance in Short Message
- C_Read_Expl: Worse Performance than ROMIO
- A lot of Debug Message in some benchmarks because of un-implemented functions

■ We also evaluated using IOR, but IOR did not complete in OMPIO



FUJITSU

shaping tomorrow with you