Topology Awareness in the Tofu Interconnect Series

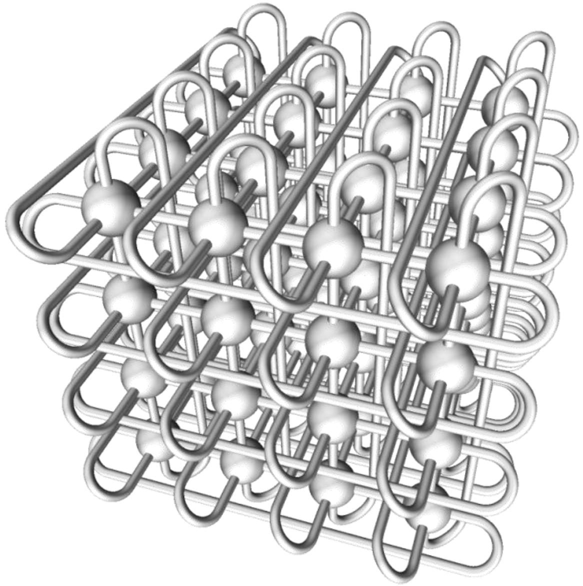
Yuichiro Ajima
Senior Architect
Next Generation Technical Computing Unit
Fujitsu Limited

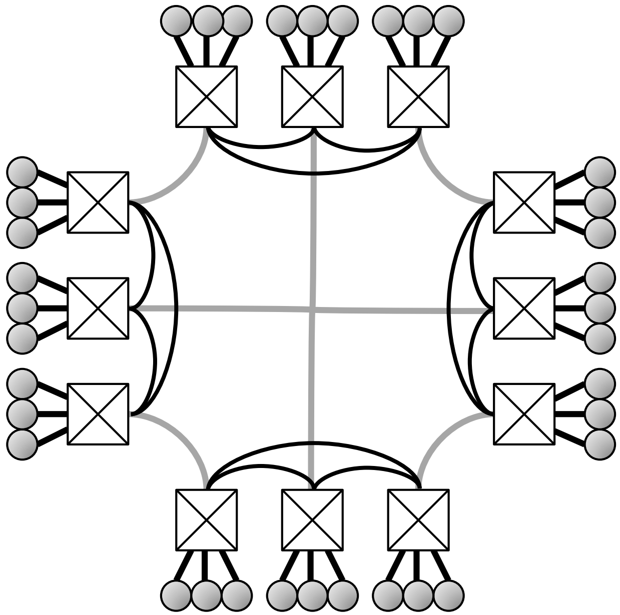
Introduction

- Networks are getting larger
  - Systems have tens of thousands of nodes
- Highly scalable network topologies

  e.g. multi-dimensional torus, dragonfly
  - Channel bisection < 1/2 node count
  - Bisection bandwidth < injection bandwidth
- Issue: communication algorithms
  - Existing general algorithms will be inefficient

    (video) MPI_Bcast on the K computer
- Topology-aware optimization is required
- This talk presents the topology-awareness design of the Tofu interconnect series, and visualizes the achievements





Tofu Interconnect Series

- Highly scalable 6D mesh/torus network
- Tofu interconnect
  - Developed for the K computer
- Tofu interconnect 2
  - SoC integration and optical transceiver
- Another version is being developed for the Post-K machine

Tofu interconnect

K computer

PRIMEHPC FX10

Tofu interconnect 2

PRIMEHPC FX100

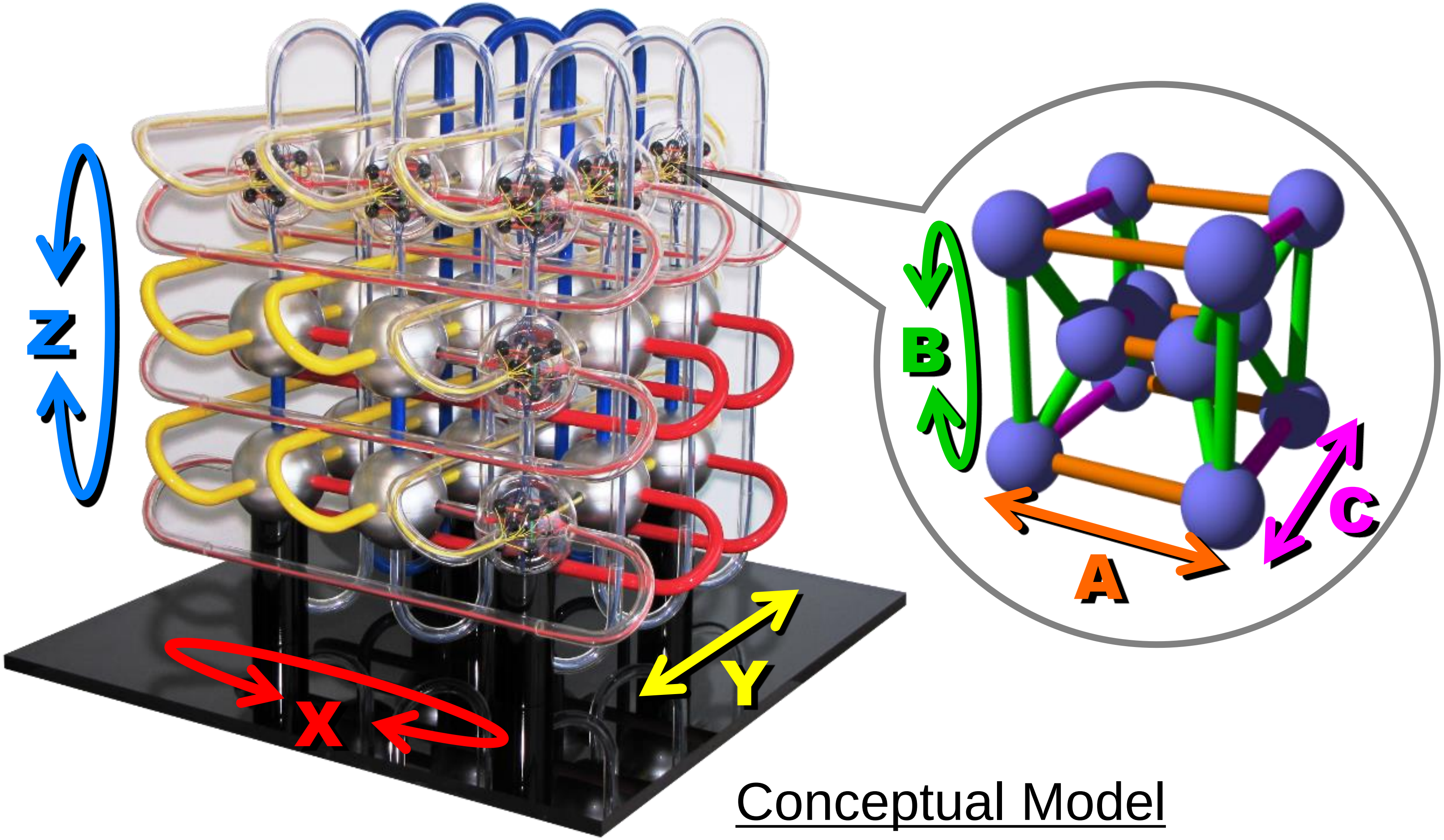
2010 **2012** **2015**

Index

- Topology-aware task allocation
- Topology-aware optimization
- Tuned collective communication library
  - Low-level features of the network interface
  - Topology-aware algorithms (for long messages)

6D Mesh/Torus Network

- Dimension labels: XYZABC
- Lengths of A-, B-, and C-axes are fixed; 2, 3 and 2



Conceptual Model

Task Allocation and Rank Mapping

FUJITSU

- A rectangular region in the physical 6D network for each task
  - Contiguous in the XYZ-axes and not divided in the ABC-axes
- Virtual torus rank mapping
  - Users defined the logical shape of the task as a virtual 1D/2D/3D torus
  - The length for each dimension is defined in the batch script

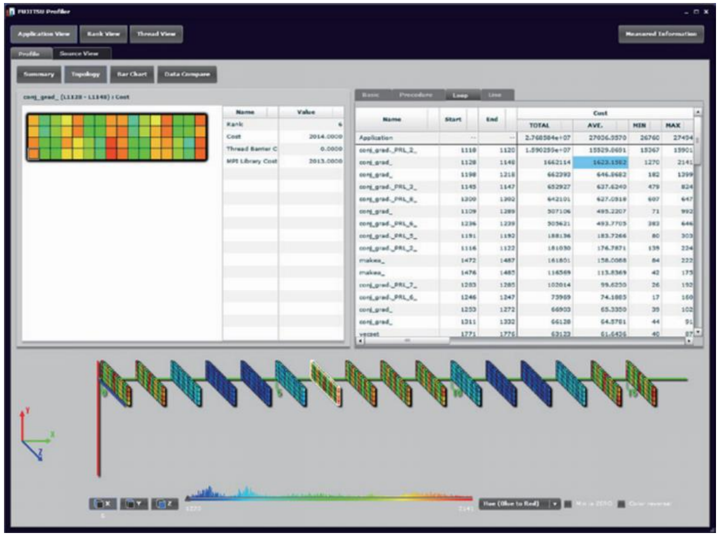
    Example: using the full system of the K computer (24 × 18 × 16 × 2 × 3 × 2)
    - Virtual 1D torus #PJM -L "node=82944“
    - Virtual 2D torus #PJM -L "node=576x144“
    - Virtual 3D torus #PJM -L "node=54x48x32“
  - A rank number reflects the logical coordinates of process
- Embedding a virtual torus into a physical rectangular region
  - A nearest neighbor node in the virtual torus space is guaranteed to be a nearest neighbor node in the physical 6D network
  - The task scheduler may add padding nodes and rotate the shape to increase the chance for allocation

Index

- Topology-aware task allocation
- Topology-aware optimization
- Tuned collective communication library
  - Low-level features of the network interface
  - Topology-aware algorithms (for long messages)

Manual Tuning with Profiling

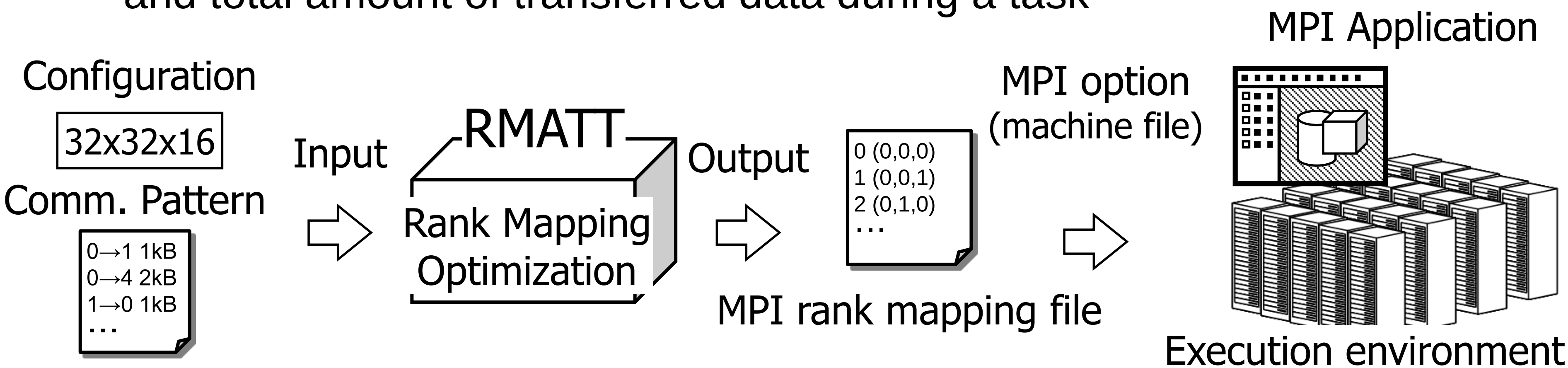
- Dynamic profiling
  - Enable profiling during the application's communication activity
  - The profiler periodically samples performance analysis (PA) counters
  - The profiling log is saved to storage after profiling
- PA counters of the Tofu interconnect
  - Each counter is a hardware 64-bit register
  - A set of PA counters is provided for each port of the router
    - Bytes transferred, busy cycles, idle cycles, packet buffer depleted cycles, etc.
- Visualization
  - Users find bottlenecks
- Manual performance tuning
  - MPI and task allocation options
  - Communication algorithms



Screen shot of the Fujitsu Profiler

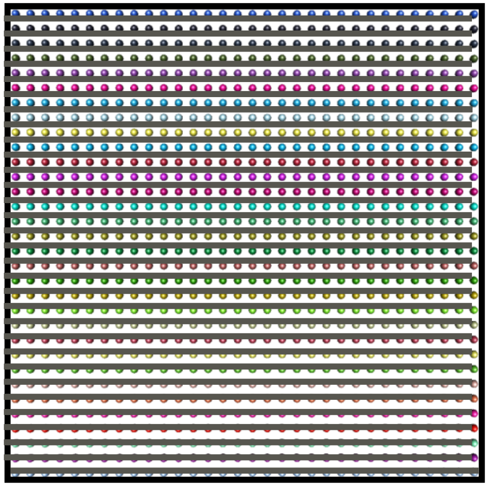
Automatic Tuning without Profiling

- Custom rank mapping order
  - A default rank mapping order often affects the communication performance in a multi-dimensional torus
  - One of the optimization candidates right after executing a vanilla code
- RMATT (Rank Mapping Automatic Tuning Tool)
  - Requires no profiling log but execution statistics
  - Calculates rank mapping order using the simulated annealing algorithm
  - Users input the shape of torus and a list of communication pattern
  - Each line of the list includes source and destination pair of processes and total amount of transferred data during a task

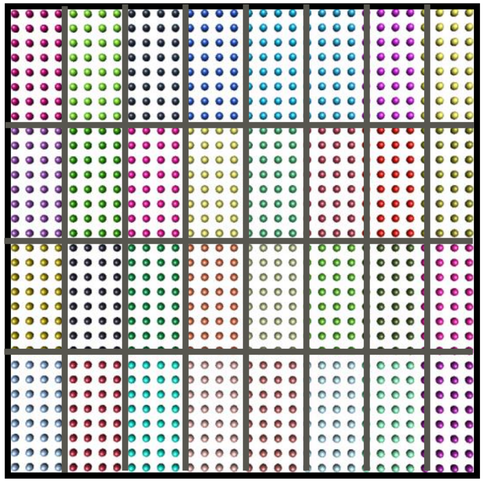


Evaluations of Improvement by RMATT

- NAS Parallel Benchmark (CG)
- Case 1: NPROCS=1024, CLASS=B, 2D Torus 32x32



Default（x-y order）



Rank map optimized by RMATT

	Default	RMATT
Execution time	1.33 sec	1.24 sec

7% improved
(includes calculation time)

- Case 2: NPROCS=8192, CLASS=D, 2D Torus 128x64

	Default	RMATT
Execution time	10.94 sec	9.98 sec

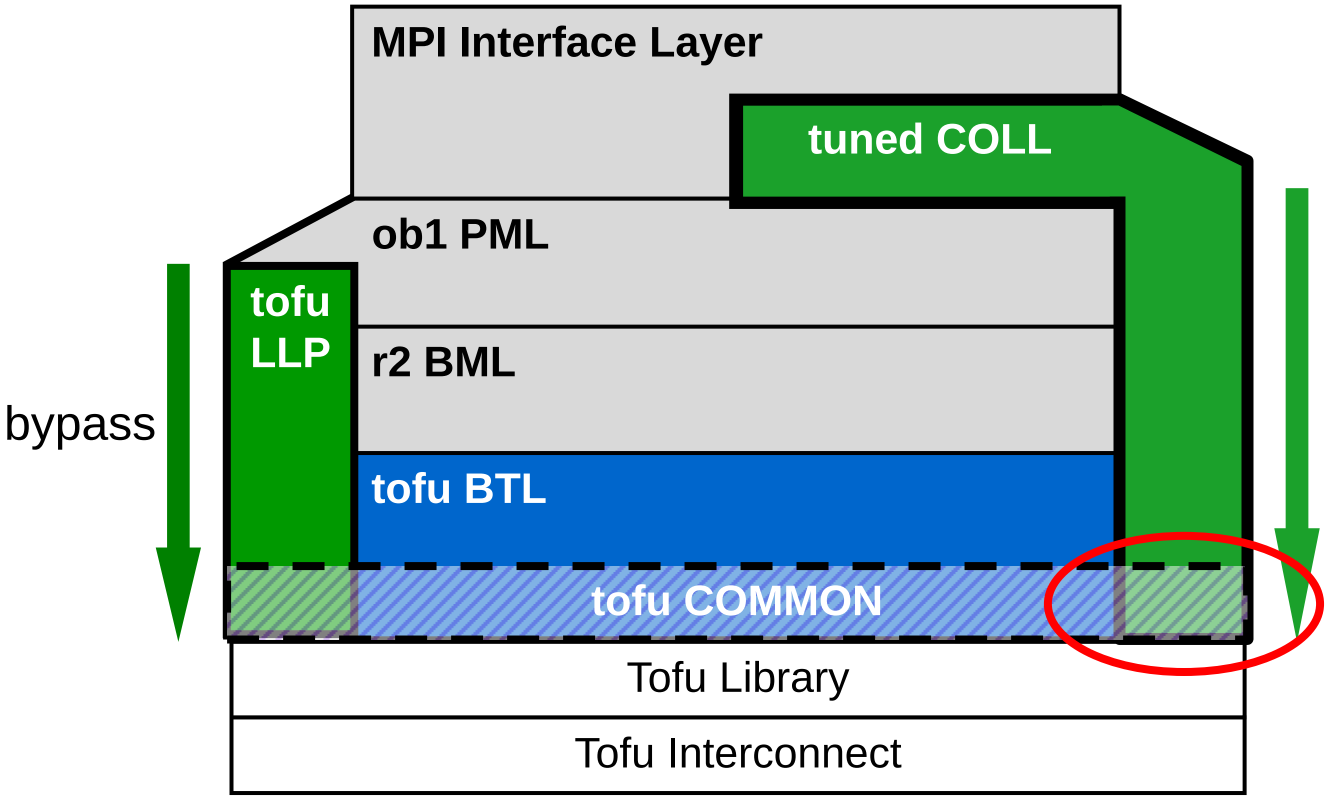
9% improved
(includes calculation time)

Index

- Topology-aware task allocation
- Topology-aware performance optimization
- Tuned collective communication library
  - Low-level features of the network interface
  - Topology-aware algorithms (for long messages)

Low-Level Network Interface

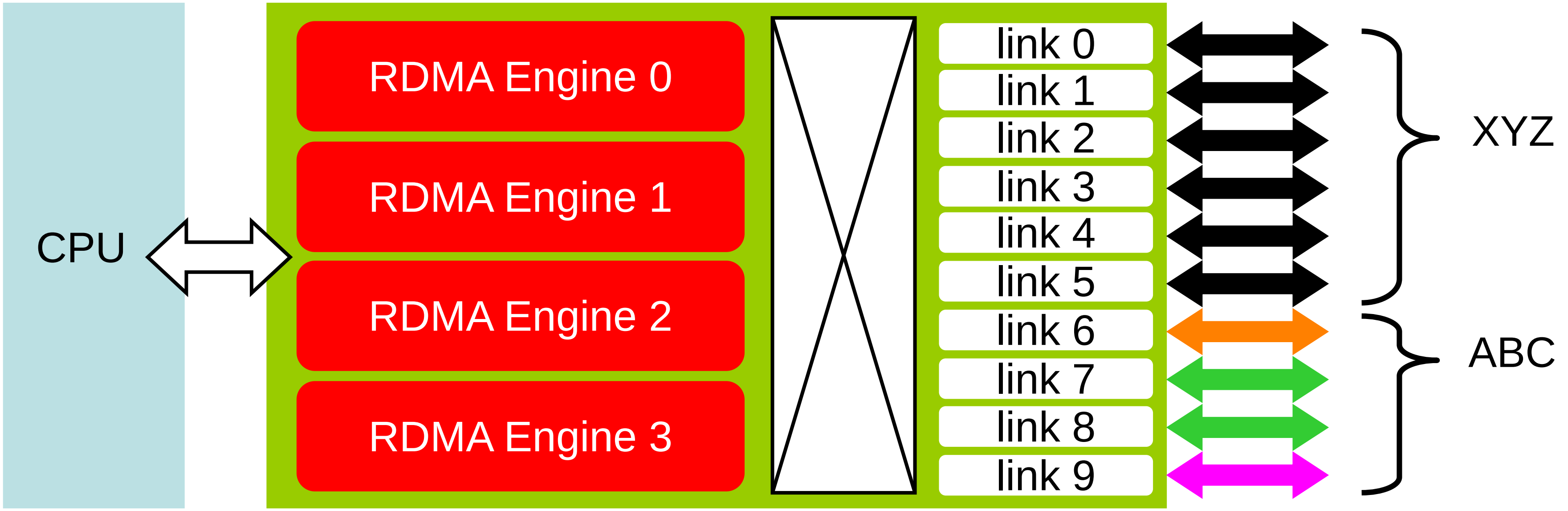
- Fujitsu's FJMPI is developed based on Open MPI
- The tuned collective communication library bypasses the Open MPI stack and uses the low-level network interface directly



Simultaneous Communication

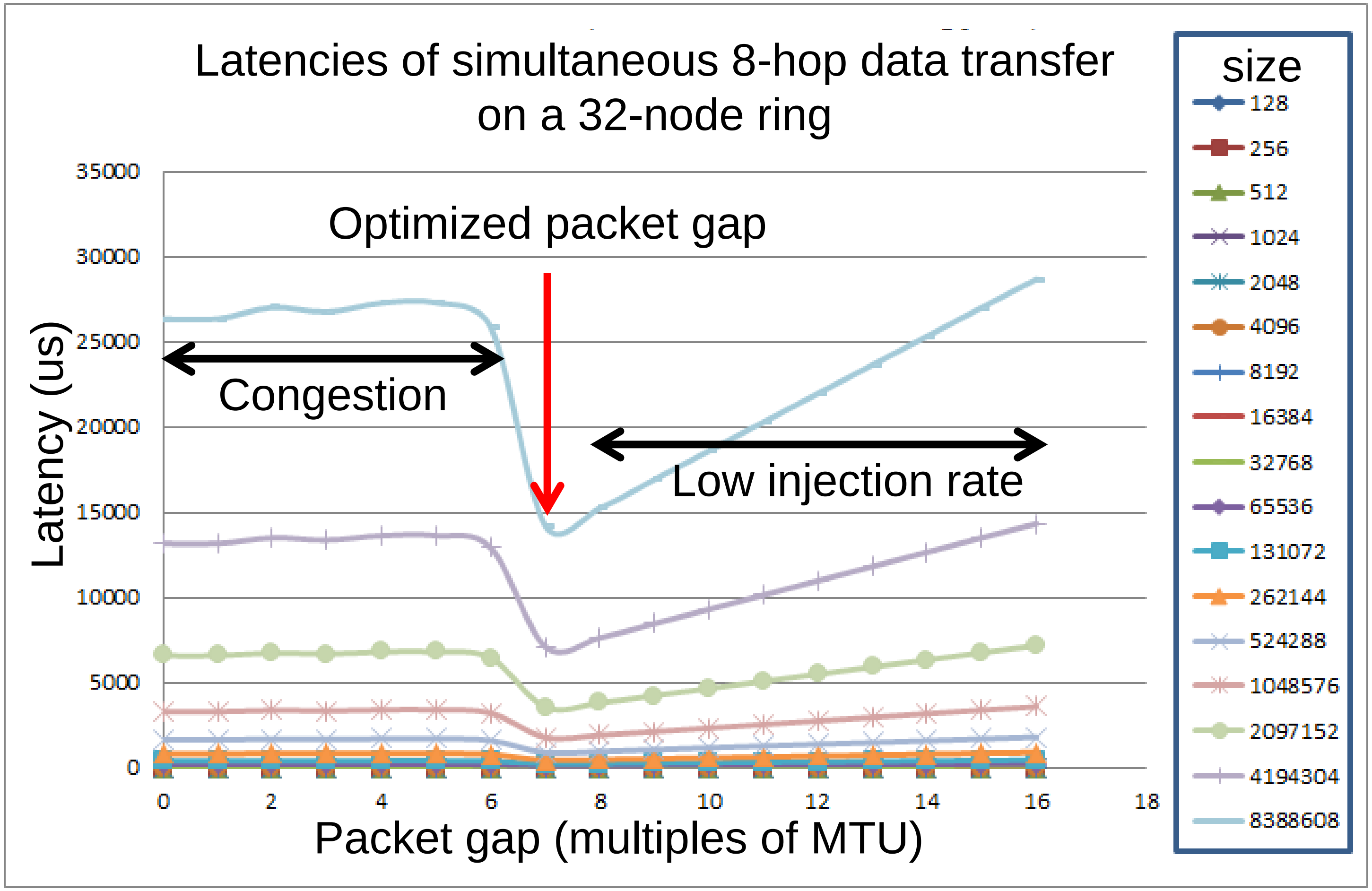
FUJITSU

- Four RDMA engines (Tofu network interfaces) per node
  - The peak injection bandwidth of each TNI is 5 GB/s for Tofu1 and 12.5 GB/s for Tofu2.
- The point-to-point messaging layer of the FJMPI uses four TNIs in a round-robin manner
- The tuned COLL identifies four TNIs to avoid a collision of the destination TNI



Injection Rate Control

- Contention depletes packet buffers and causes congestion
- Congestion can be avoided by reducing the injection rate



Index

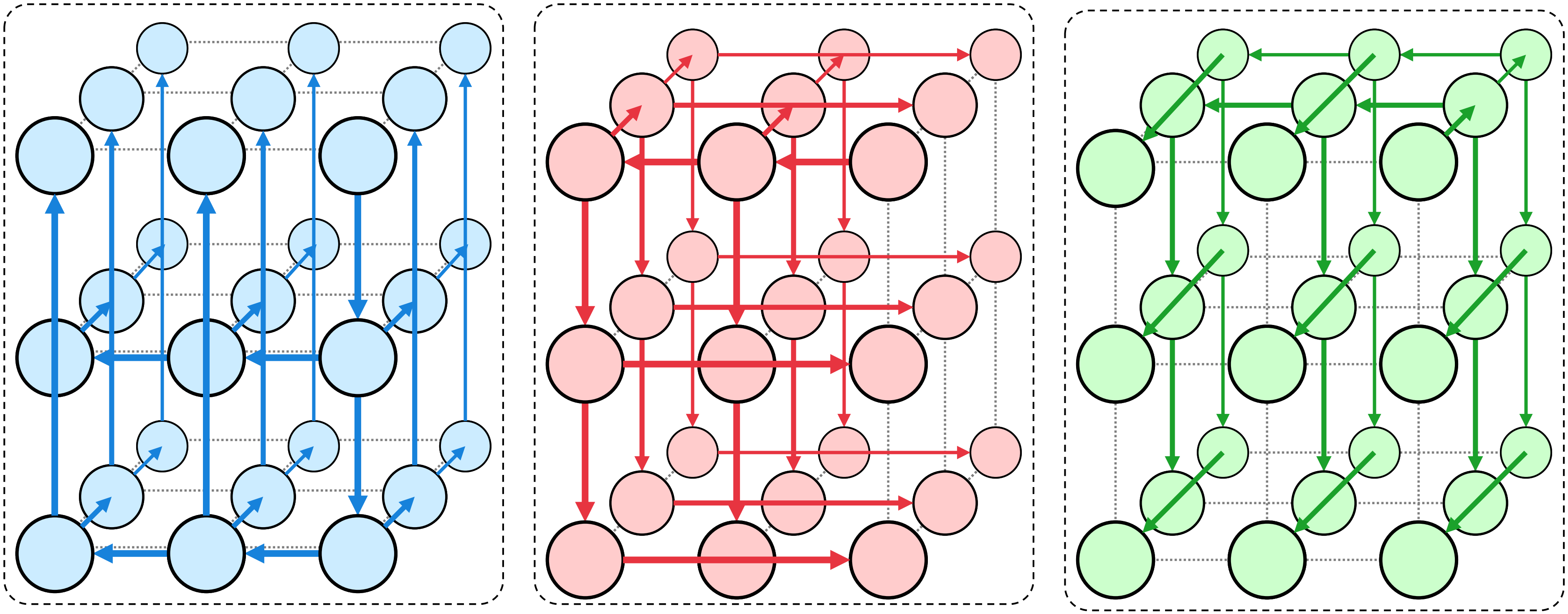
- Topology-aware task allocation
- Topology-aware performance optimization
- Tuned collective communication library
  - Low-level features of the network interface
  - Topology-aware algorithms (for long messages)

Overview

- Assumed environment
  - The shape of the communicator is a mesh or a torus
  - One process per node participates in inter-process communication
    - When there are multiple processes in a node, collective communication is fanned out through shared memory
- Optimization policies (for long messages only)
  - Use multiple network interfaces
  - Communicate with nearest neighbor nodes
  - Control the injection rate for communication with far nodes
- Algorithms implemented in the FJMPI
  - Triple trinary tree for broadcast and reduce
  - Three-phase quad rings for gather
  - Uniformly overlaid symmetrical pattern for all-to-all

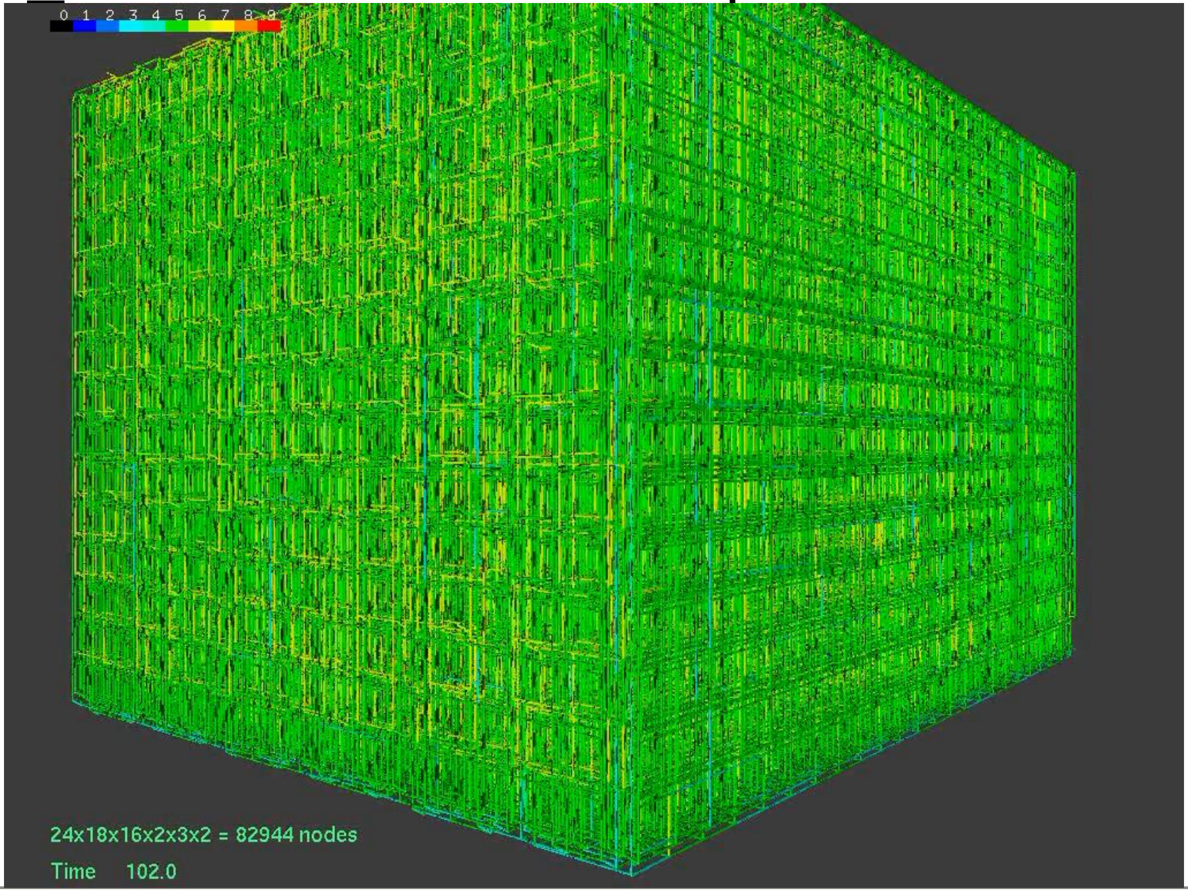
Triple Trinary Tree

- Broadcasts data by dividing into three parts and simultaneously propagating each part via a different path
- Each path is a spanning trinary tree, and the three trees share no directed edges
- By reversing the direction of all edges, data can be reduced



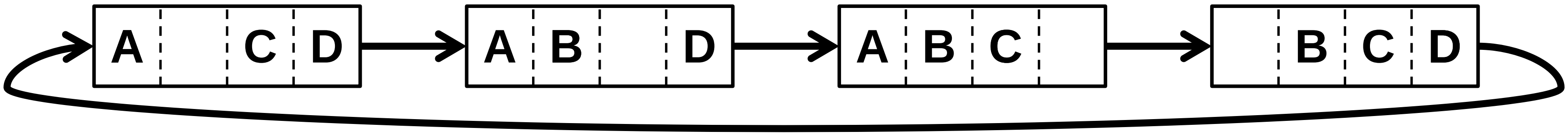
MPI_Allreduce

- Two phases
  - First phase – reduce data using triple trinary trees
  - Second phase – broadcast the reduced data using the reversed trees
- (video) MPI_Allreduce on the K computer

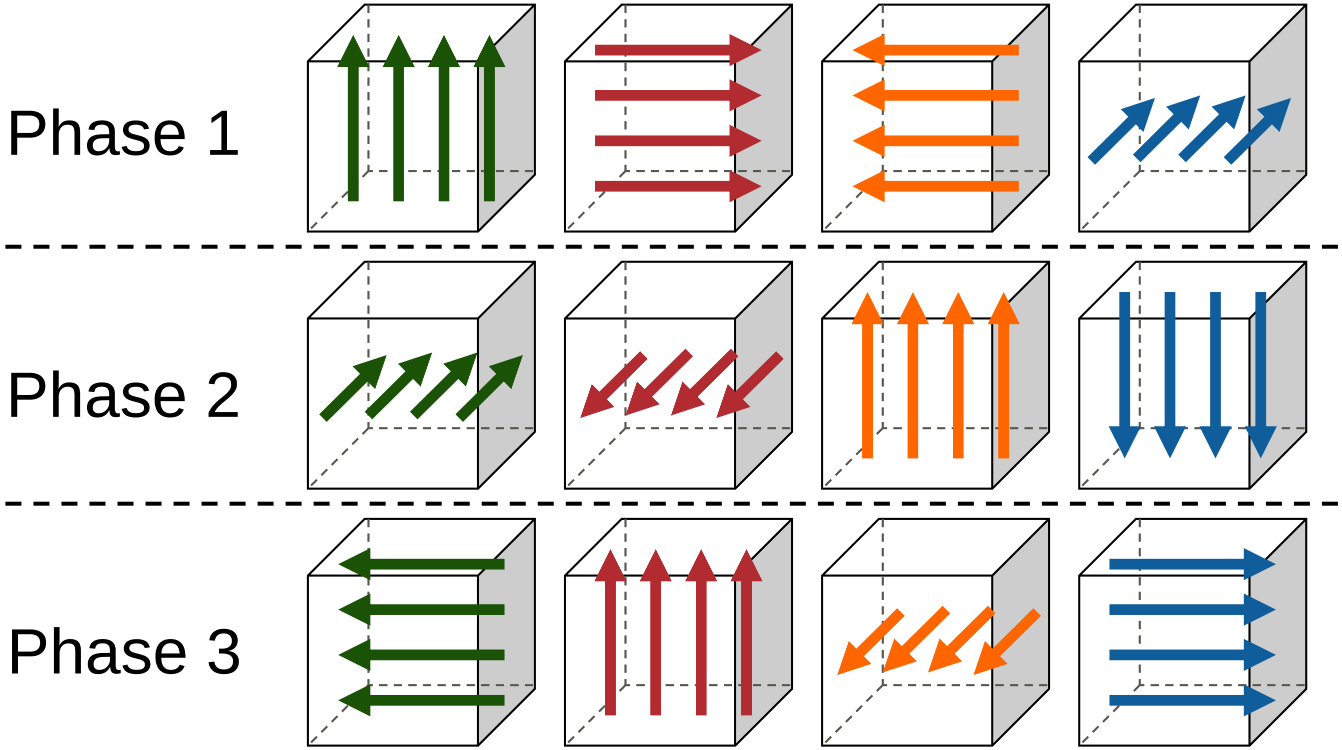


Three-Phase Quad Rings

- The ring all-gather algorithm transfers data cyclically

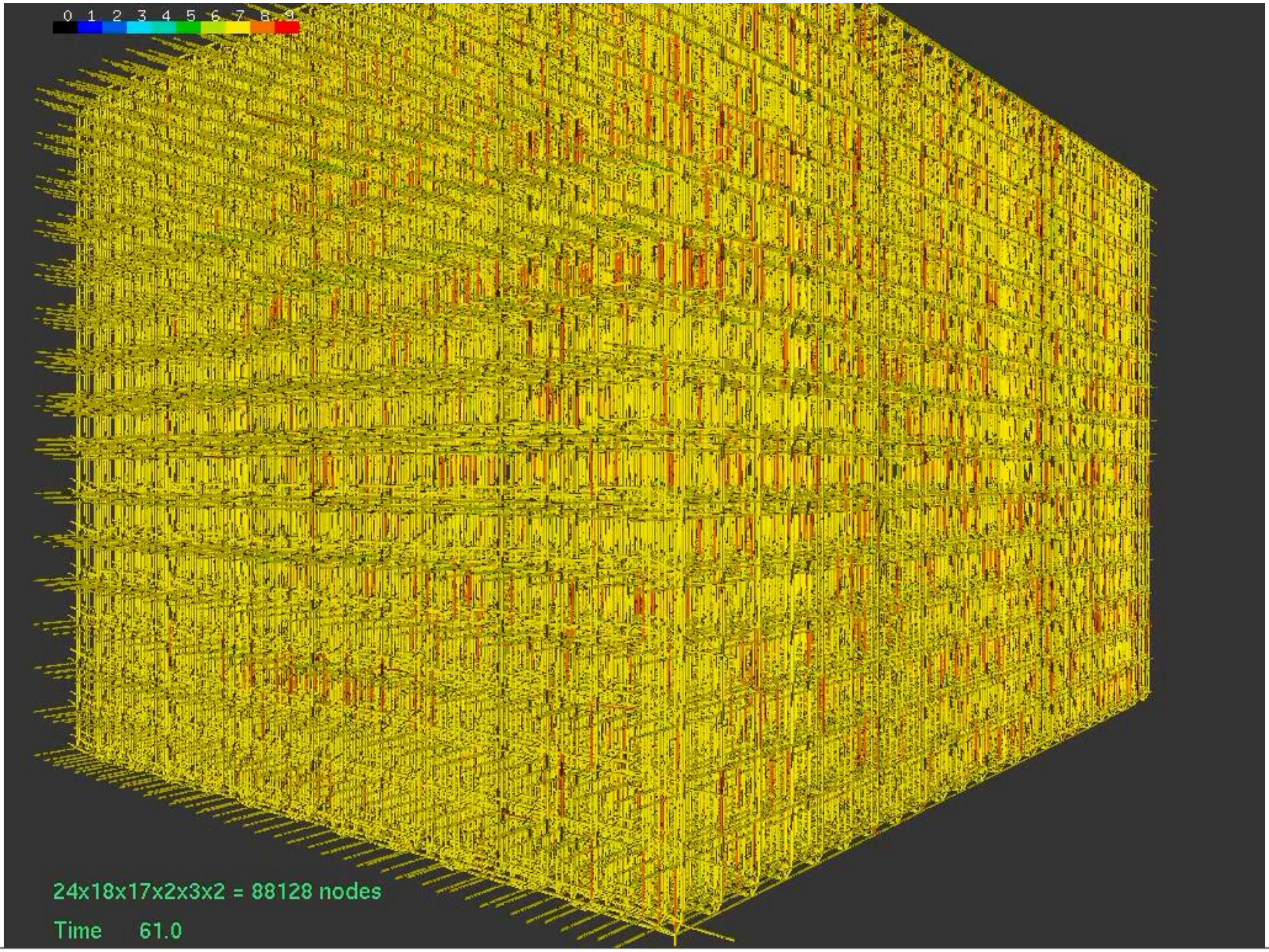


- Divides data into four parts, and simultaneously transfers each part along a different direction



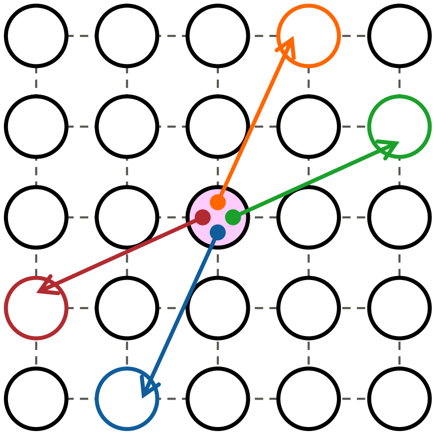
MPI_Allgather

- The three-phase quad ring algorithm
- (video) MPI_Allgather on the K computer

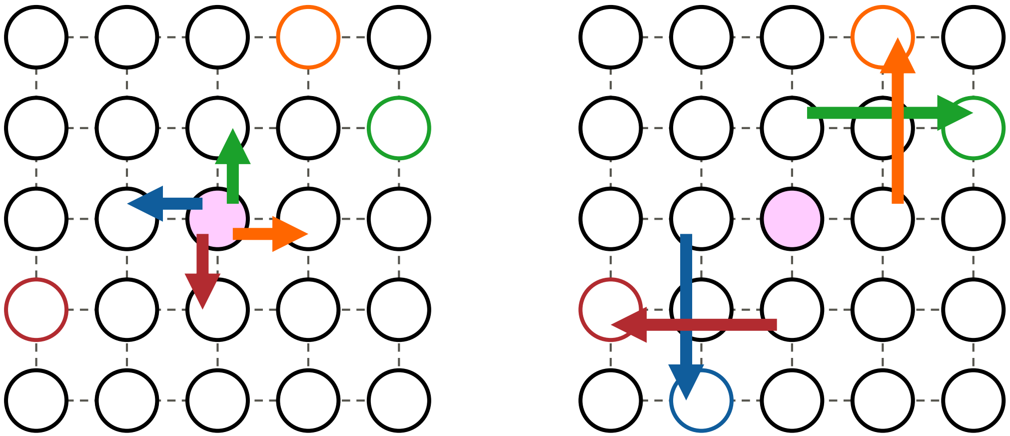


Uniformly Overlaid Symmetrical Pattern (1)

- A multi-phase all-to-all communication algorithm
  - In each phase, each process transfers data to multiple processes that have symmetrical relative coordinates

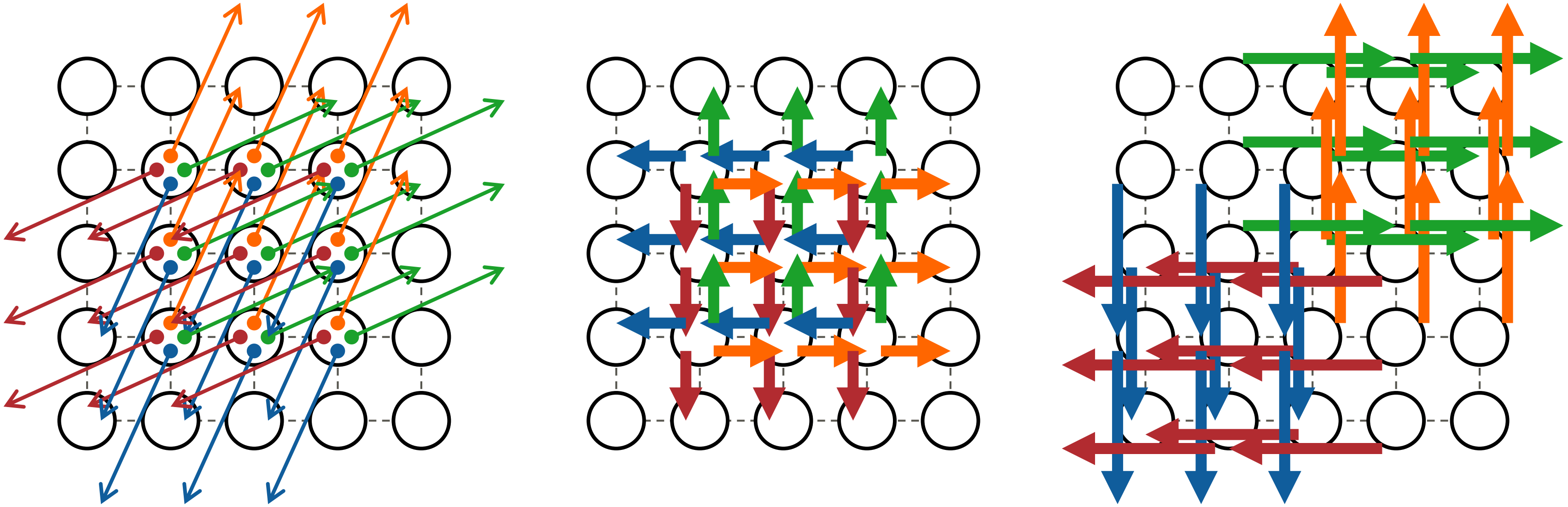


- Each phase is divided into sub-phases



Uniformly Overlaid Symmetrical Pattern (2)

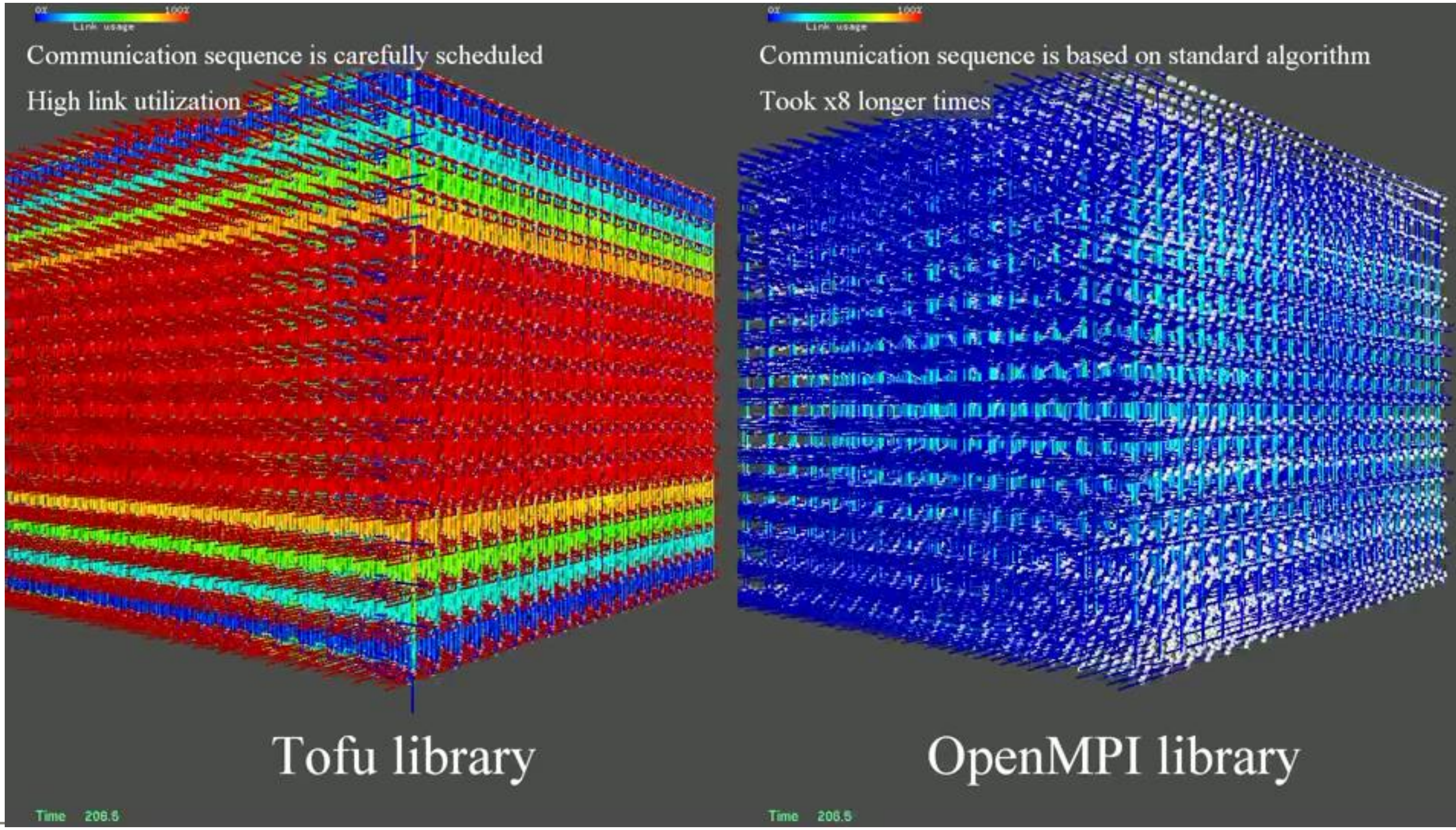
- For each phase, communication patterns of all processes are uniform



- For each sub-phase, the number of colliding transfers is the same as the hop count of a transfer
- Injection rate control for each sub-phase avoids congestion and increases effective throughput

MPI_Alltoall

- Uniformly overlaid symmetrical pattern algorithm
- (video) MPI_Alltoall on the K computer
  - Left: the uniformly overlaid symmetrical pattern algorithm
  - Right: default algorithm of the Open MPI



Summary

FUJITSU

- Topology awareness design of the Tofu series
- Task allocation
  - Virtual torus rank mapping
- Performance optimization
  - Tofu PA counters for manual tuning with the Fujitsu Profiler
  - Rank Mapping Automatic Tuning Tool (RMATT)
- Tuned collective communication library in the FJMPI
  - Utilizes low-level network features
    - Simultaneous communication
    - Injection rate control
  - Topology-aware algorithms for long messages
    - Triple trinary tree algorithm for broadcast and reduce
    - Three-phase quad rings algorithm for gather
    - Uniformly overlaid symmetric pattern algorithm for all-to-all

shaping tomorrow with you