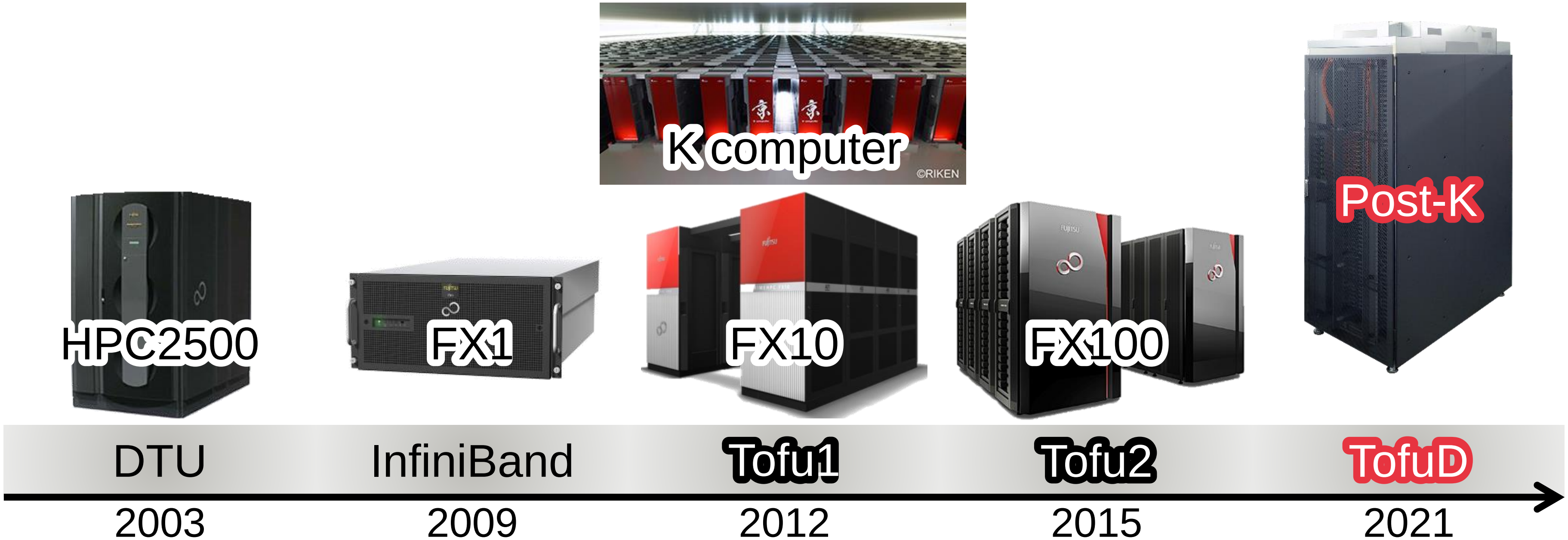
The Tofu Interconnect D

11 September 2018

Yuichiro Ajima, Takahiro Kawashima, Takayuki Okamoto, Naoyuki Shida, Kouichi Hirai, Toshiyuki Shimizu, Shinya Hiramoto, Yoshiro Ikeda, Takahide Yoshikawa, Kenji Uchida, Tomohiro Inoue

Fujitsu Limited

Introduction



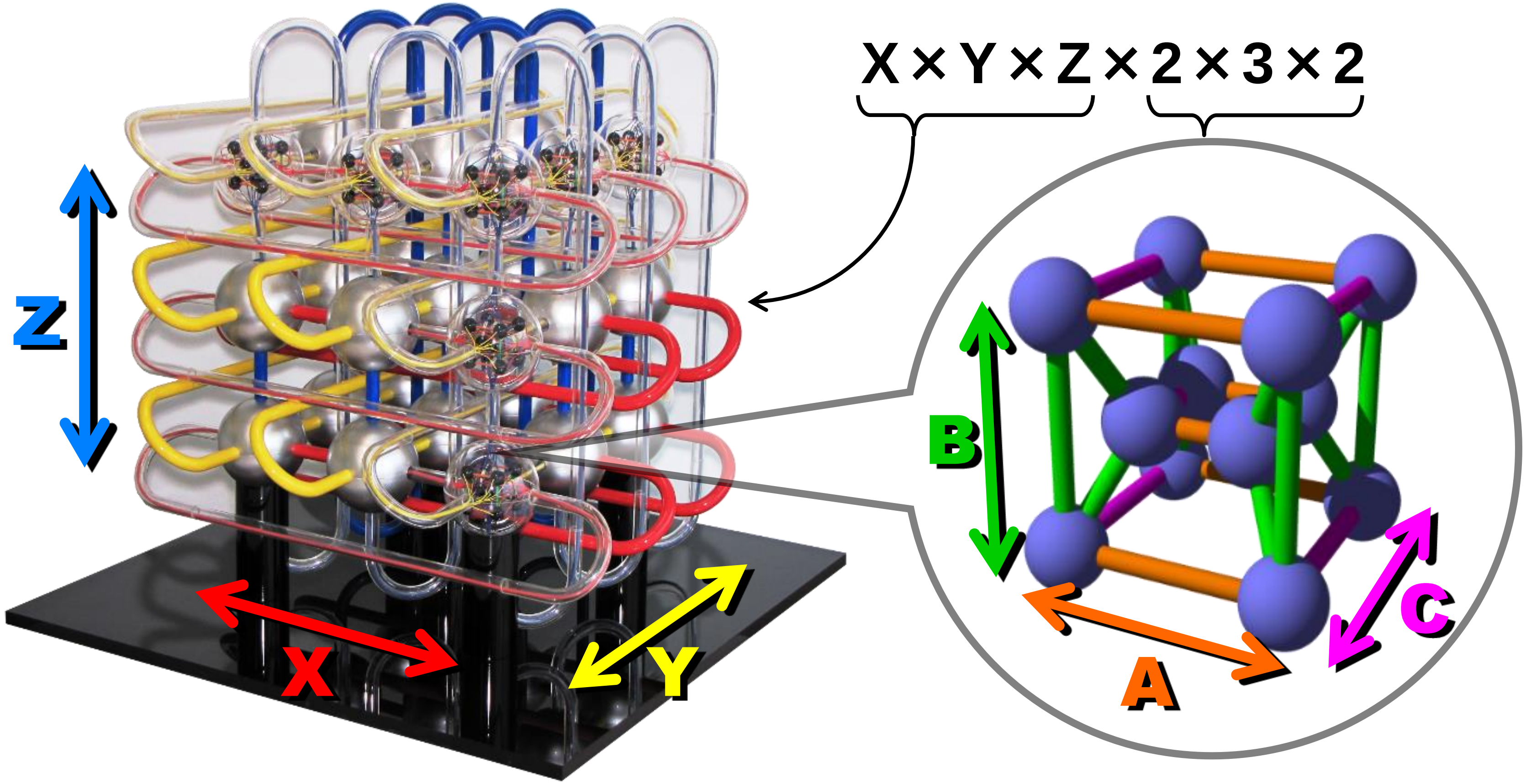
- The Tofu interconnect (Tofu1) for the K computer
  - Highly-scalable and fault-tolerant 6D mesh/torus network
- The Tofu interconnect 2 (Tofu2) for FX100 machines
- The Tofu Interconnect D (TofuD) for the post-K machine
  - High “density” of node: integrate more resources into a smaller node
  - Fault resilient of network: “dynamic” packet slicing for packet transfer

Features of the Tofu interconnect family

- 6D Mesh/Torus Network
- Virtual 3D-Torus Rank-mapping
- Implementations
- Communication Functions
- Tofu Barrier
- Networks of Recent World-class Systems

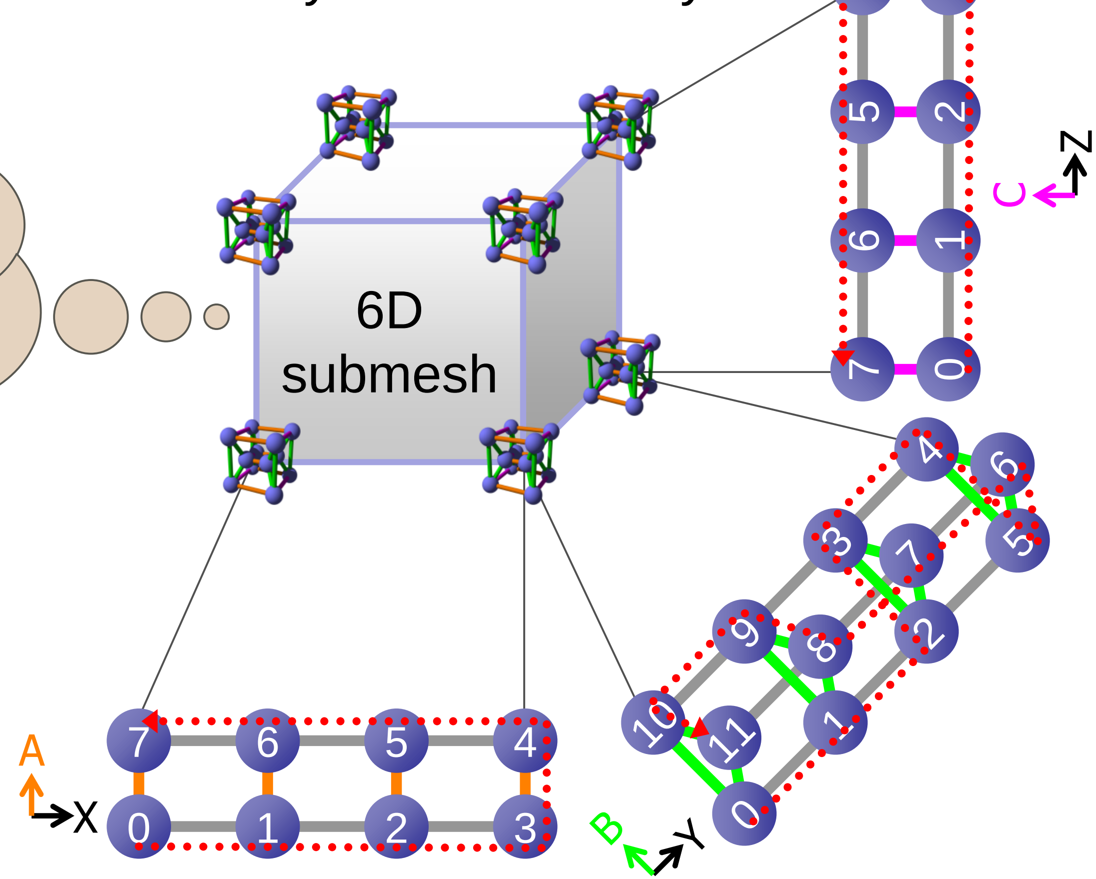
6D Mesh/Torus Network

- Six coordinate axes: X, Y, Z, A, B, C
  - X, Y, Z: the size varies according to the system configuration
  - A, B, C: the size is fixed to 2×3×2
- Tofu stands for “torus fusion”: (X, Y, Z)×(A, B, C)



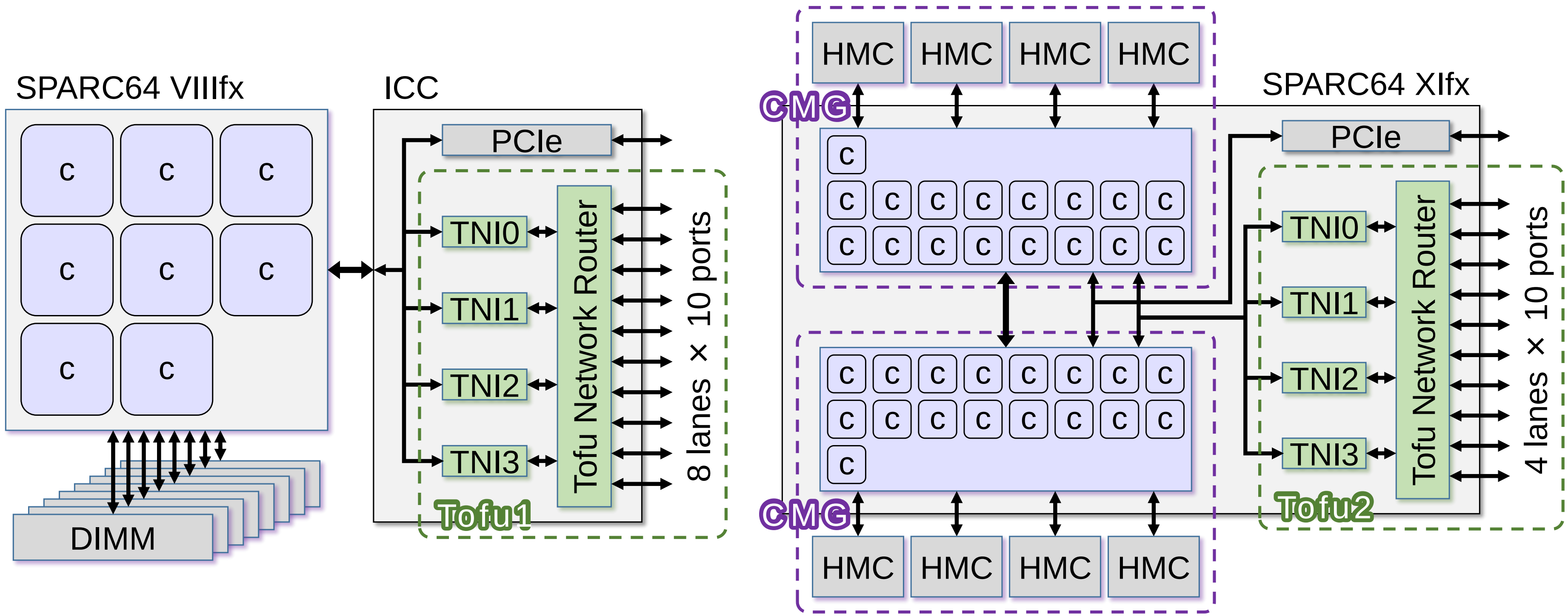
Virtual 3D-Torus Rank-mapping

- A rank-mapping option for topology-awareness
- A 3D torus rank can be mapped to a 6D submesh even if there is an unavailable node
  - This fault tolerance contributes to the system availability



Implementations

- Tofu1: implemented as an interconnect controller (ICC) chip
  - 4 Tofu network interfaces (TNIs) and 80 lanes of signals for the network
- Tofu2: integrated into a processor chip
  - The number of signal lanes for the network decreased to 40
  - Considering the balance with 128 signal lanes for memory

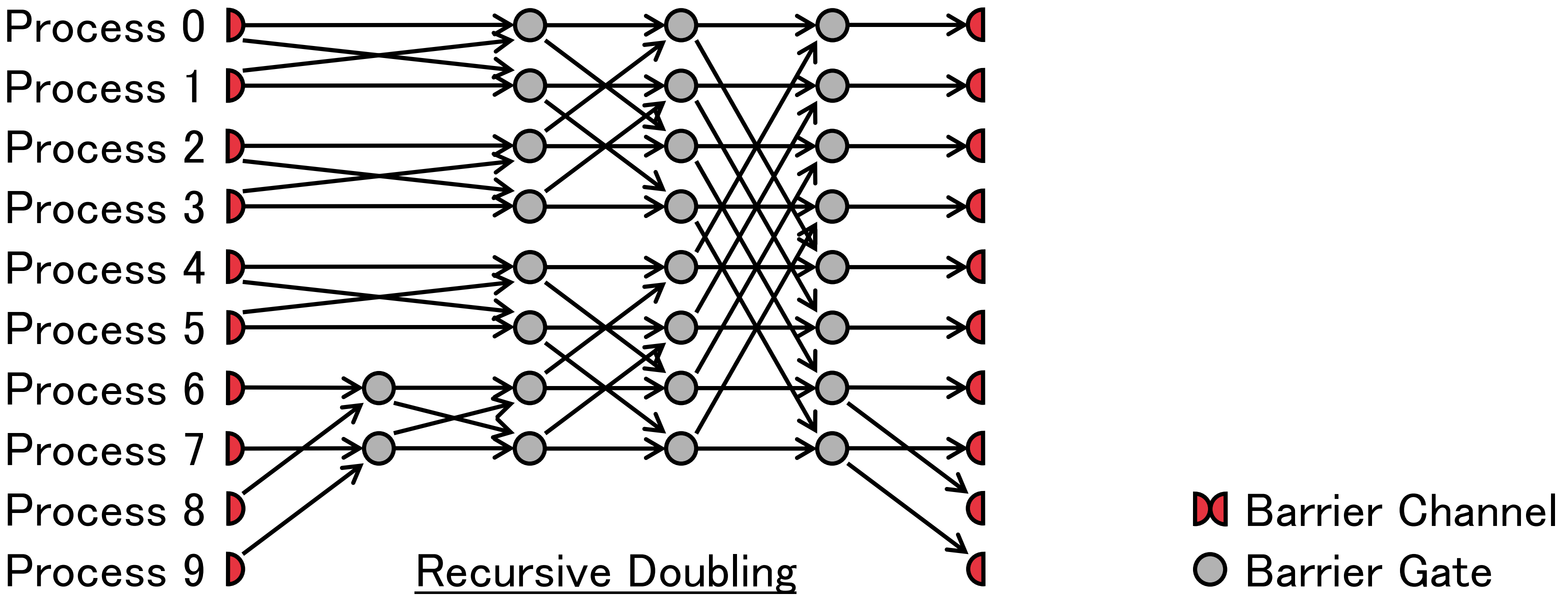


Communication Functions of TNI

- Remote direct memory access (RDMA)
  - Directly accesses process memory on remote node
  - RDMA Put transfers data to remote process memory
  - RDMA Get transfers data from remote process memory
  - RDMA Atomic modifies a shared variable in remote process memory
- Low latency features
  - Direct Descriptor: feeds communication commands from CPU registers
  - Cache Injection (since Tofu2): places received data into a CPU cache
- Tofu Barrier
  - Offload engine for collective communications such as synchronization

Tofu Barrier (1)

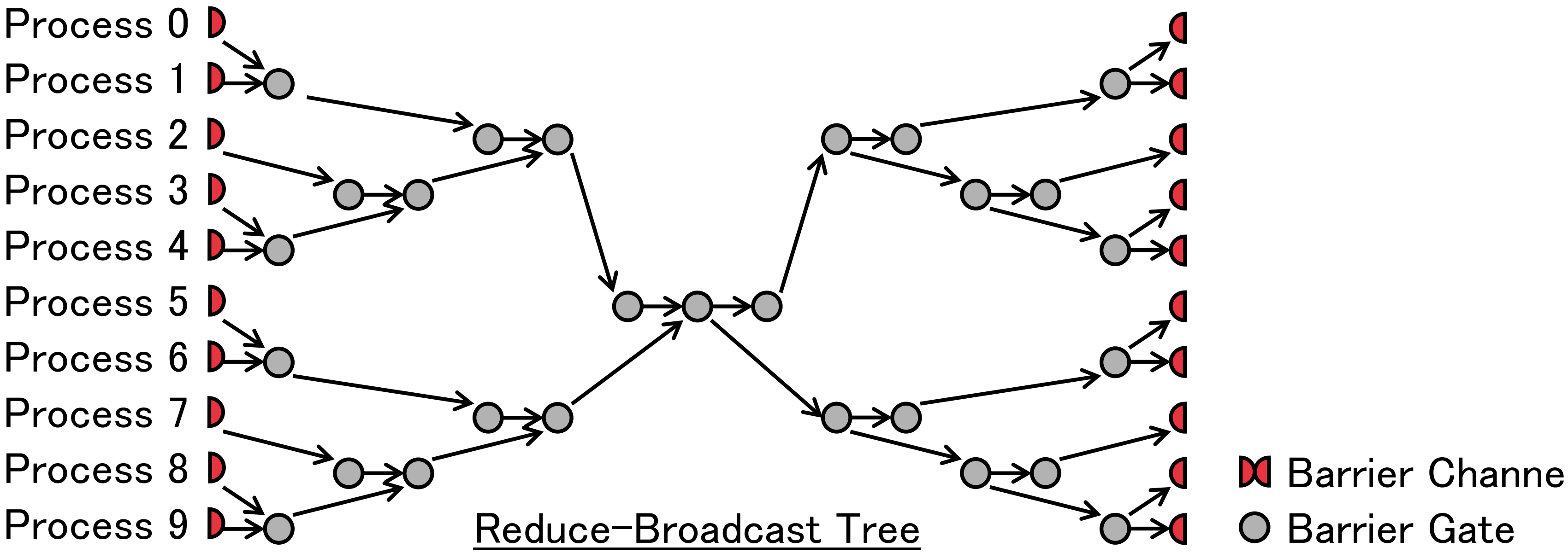
- Barrier gate (BG) is a hard-wired communication engine
  - Waits for two signals from other BGs and transmits two signals
- Barrier channel (BCH) is an interface of Tofu barrier
  - Each BCH is fixedly bound to a start-and-end point BG
- Tofu barrier can execute an arbitrary communication algorithm
  - Recursive-doubling algorithm uses log2(n) of BGs in each process



Tofu Barrier (2)

FUJITSU

- Reduce-broadcast algorithm uses a maximum of 5 BGs in each process



- In Tofu1 and Tofu2, only TNI #0 had a Tofu barrier
  - 8 BCHs and 64 BGs per node
  - Up to 8 communicators per node can use Tofu barrier simultaneously
- Intra-node synchronization is recommended to be performed using software to reduce consumption of BCH and BG

Networks of Recent World-class Systems

FUJITSU

System	Network	Total Injection Bandwidth (PB/s)		Bisection Bandwidth (TB/s)
Blue Gene/Q	Torus (5D)	1.97	40X	49
K Computer	Mesh/Torus (6D) Virtual Torus (3D)	1.66	36X	46 34
Sunway TaihuLight	Tapered Fat-Tree	0.51	7.3X	70
Piz Daint	Dragonfly	0.07	2.0X	36
Summit	Fat-Tree	0.12	1.0X	115
Oakforest-PACS	Fat-Tree	0.10	1.0X	102

- All systems have the same order of bisection bandwidth
  - No significant performance difference in global data exchange
- Torus networks have higher total injection bandwidth
  - Topology-aware communication such as nearest-neighbor data exchange results in higher performance

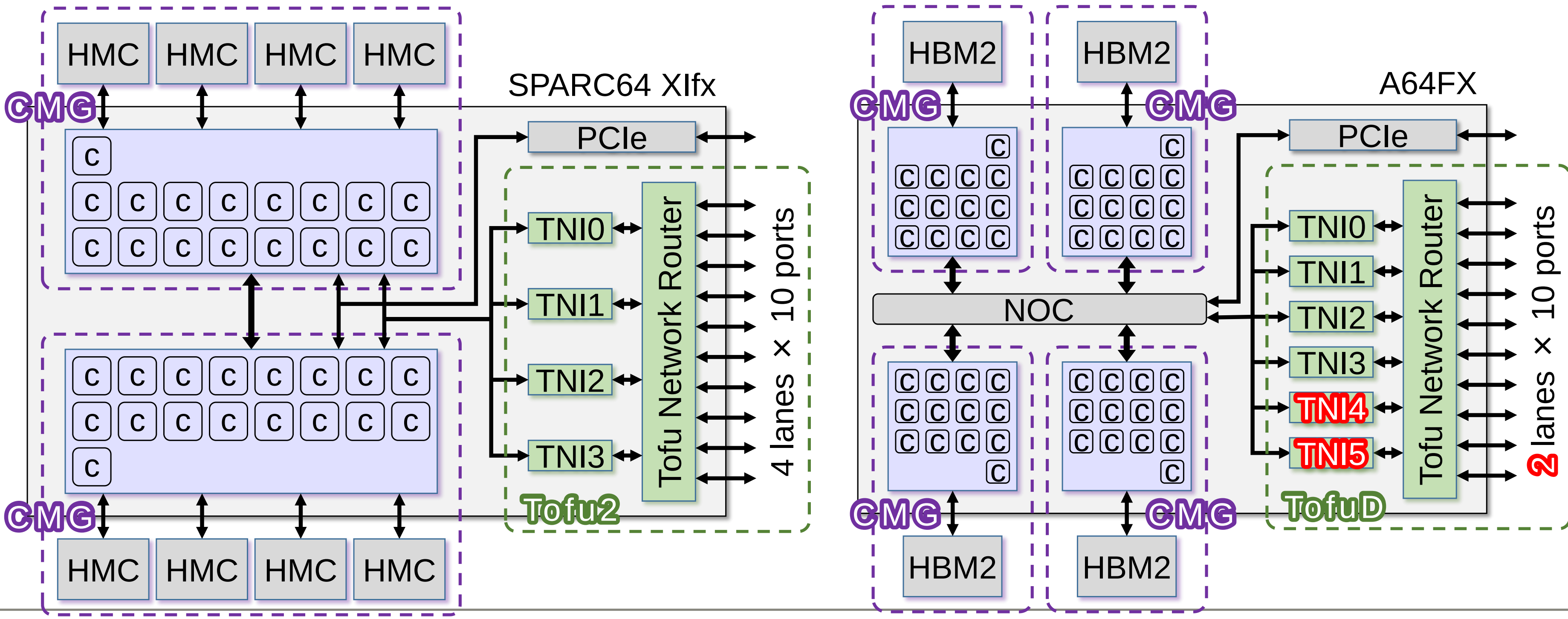
The Design of TofuD

- Higher-density Node Configuration
- Link Configuration and Injection Bandwidth
- Packaging
- Dynamic Packet Slicing
- Increased Tofu Barrier Resources

Higher-density Node Configuration

FUJITSU

- The CPU is smaller and the off-chip channels are halved
  - The number of 3D-stacked memories was halved from 8 to 4
  - Each Tofu link was reduced from 4 lanes to 2 lanes
- More resources are integrated into the CPU
  - The number of CPU Memory Groups (NUMA nodes) doubled from 2 to 4
  - The number of Tofu Network Interfaces increased from 4 to 6



Link Configuration and Injection Bandwidth

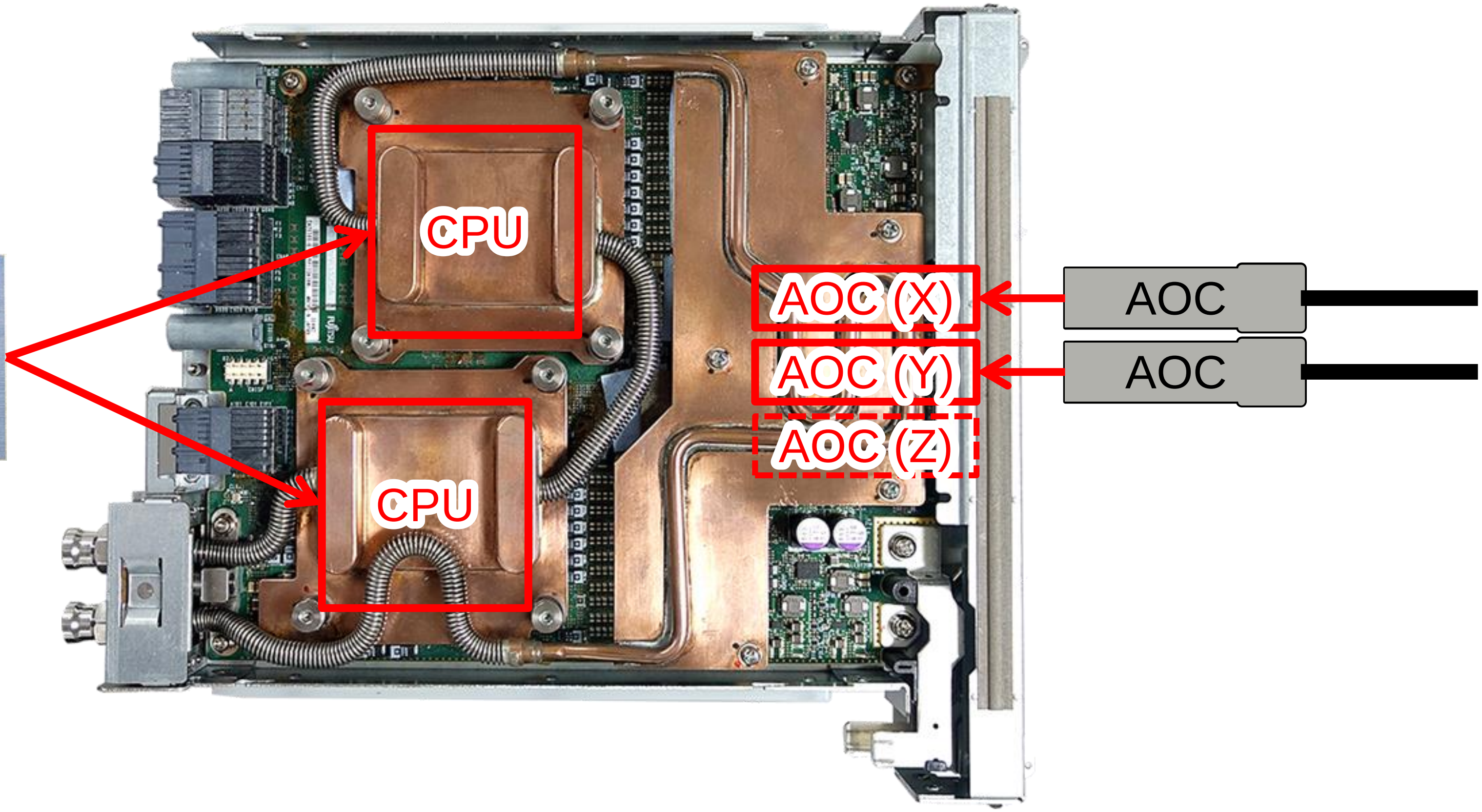
FUJITSU

	Tofu1	Tofu2	TofuD
Data rate (Gbps)	6.25	25.78125	28.05
Number of signal lanes per link	8	4	2
Link bandwidth (GB/s)	5.0	12.5	6.8
Number of TNIs per node	4	4	6
Injection bandwidth per node (GB/s)	**20**	**50**	**40.8**

- Data transfer rate increased from 25 Gbps to 28 Gbps
- Link bandwidth reduced from 12.5 GB/s to 6.8 GB/s
- TofuD simultaneously transmits in 6 directions
  - Increased from 4 directions in the case of Tofu1 and Tofu2
- Total injection bandwidth per node is 40.8 GB/s
  - Approximately, twice that of Tofu1 or 80% that of Tofu2

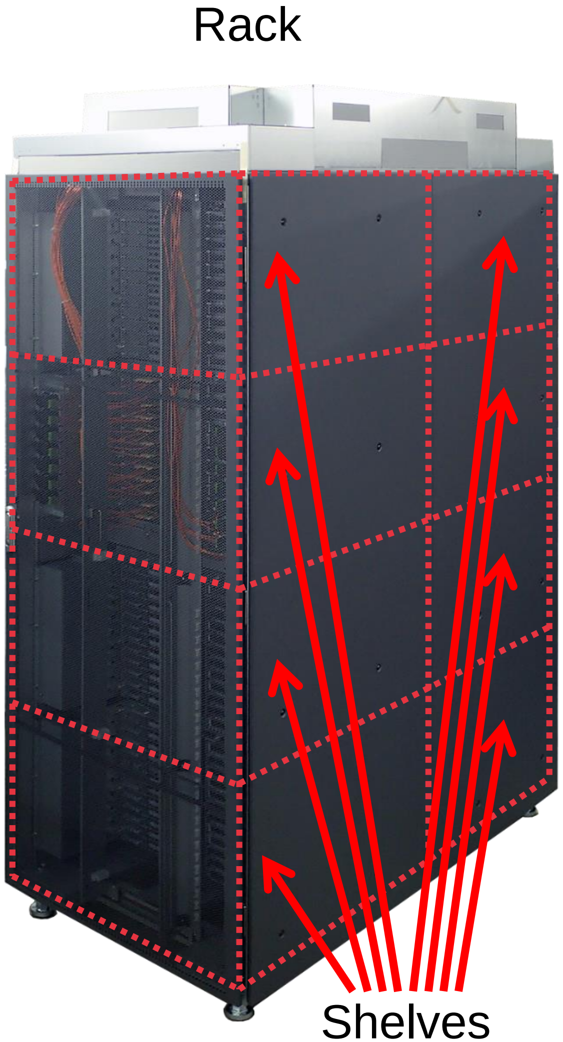
Packaging – CPU Memory Unit of Post-K

- Two CPUs connected with C-axis
  - X × Y × Z × A × B × C = 1 × 1 × 1 × 1 × 1 × 2
- Two or three active optical cable (AOC) cages on the board
  - Each cable bundles two lanes of signals from each of the two CPUs



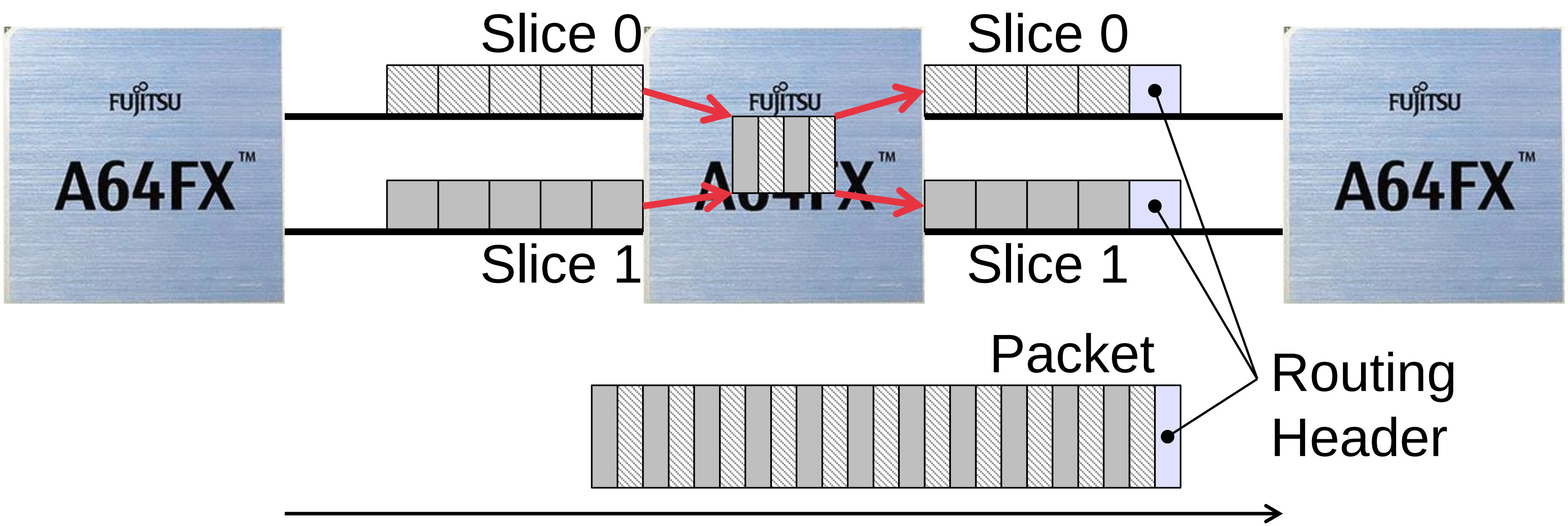
Packaging – Rack Structure of Post-K

- Rack
  - 8 shelves
  - 192 CMUs or 384 CPUs
- Shelf
  - 24 CMUs or 48 CPUs
  - X × Y × Z × A × B × C = 1 × 1 × 4 × 2 × 3 × 2
- Top or bottom half of rack
  - 4 shelves
  - X × Y × Z × A × B × C = 2 × 2 × 4 × 2 × 3 × 2



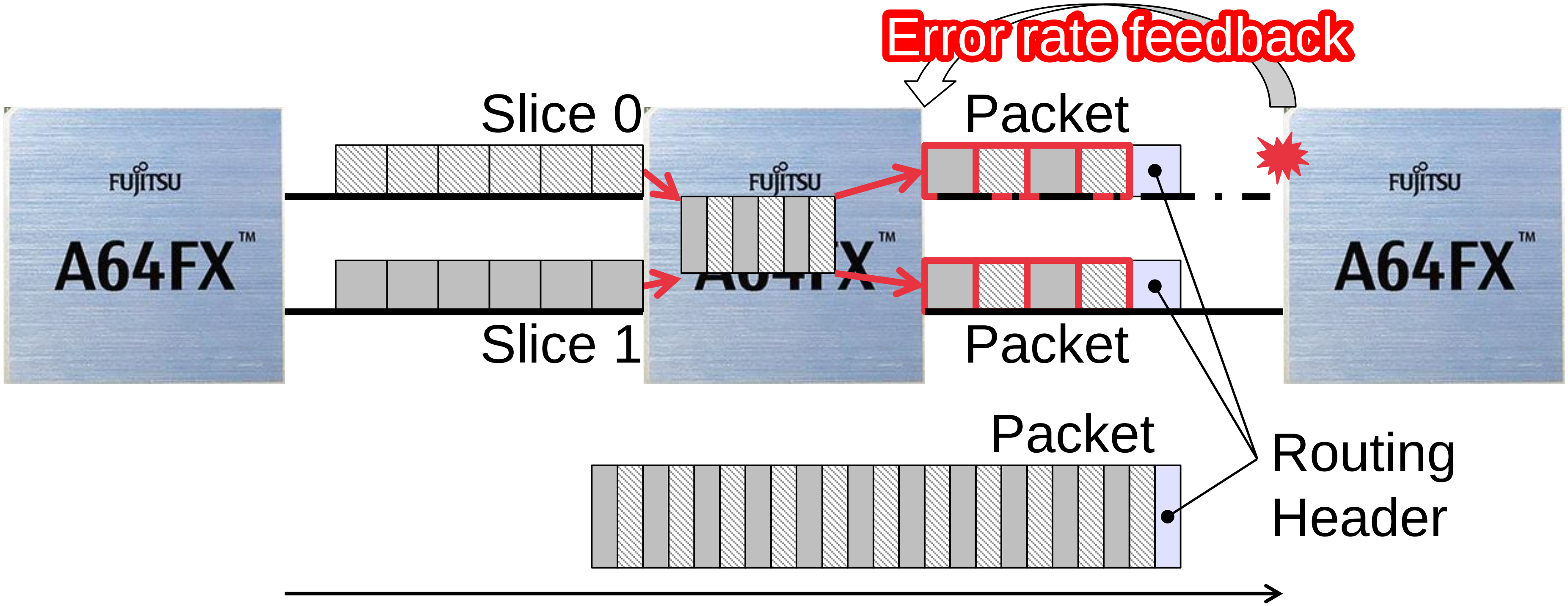
Dynamic Packet Slicing – Split Mode

- An upper layer in TofuD slices packets for each signal lane
  - Each signal lane of TofuD has an independent physical layer
  - In the ordinary multi-lane transmission, the physical layer has media-independent interface and hides the number of signal lanes
- For virtual cut-through packet transfer, the routing header is copied to both slices of the packet
- This normal operation mode is called split mode



Dynamic Packet Slicing – Duplicate Mode

- The upper layer duplicates packets when the error rate is high
- This fall-down mode is called duplicate mode
- The link can recover to the split mode
  - Each lane is never disconnected independently
  - The error rates of both lanes are continuously monitored and fed back



Increased Tofu Barrier Resources

		Tofu1	Tofu2	TofuD
Node	Number of BCHs	8	8	**96**
	Number of BGs	64	64	**288**
	Number of TNIs	4	4	6
	Number of CMGs	1	2	4
TNI	Number of BCHs	8	8	16
	Number of BGs	64	64	48

- The number of resources significantly increased
  - All 6 TNIs of TofuD have Tofu barrier
  - This change intended to support synchronization between CMGs
- The ratio of the BCHs to BGs was increased from 1:8 to 1:3
  - Assuming an increase in the usage of the reduce-broadcast tree algorithm

Performance Evaluations

- Evaluation Environment
- Put Latencies
- Latency Breakdown
- Put Throughputs
- Injection Rates
- Tofu Barrier

Evaluation Environment

- TofuD
  - Evaluated by hardware emulators using the production RTL codes
    - The simulation models were system-level and included multiple nodes
    - Simulated processors executed test programs
    - The test programs directly accessed the TofuD hardware
  - Results were measured from simulation waveforms

- Tofu1 and Tofu2
  - Evaluations used real machines
    - Real processors executed test programs
    - The test programs used low-level communication libraries
  - Results were measured using the processor’s cycle counter
  - Only latency breakdowns were obtained from simulation waveforms

Put Latencies

- 8B Put transfer between nodes on the same board
  - The low-latency features were used

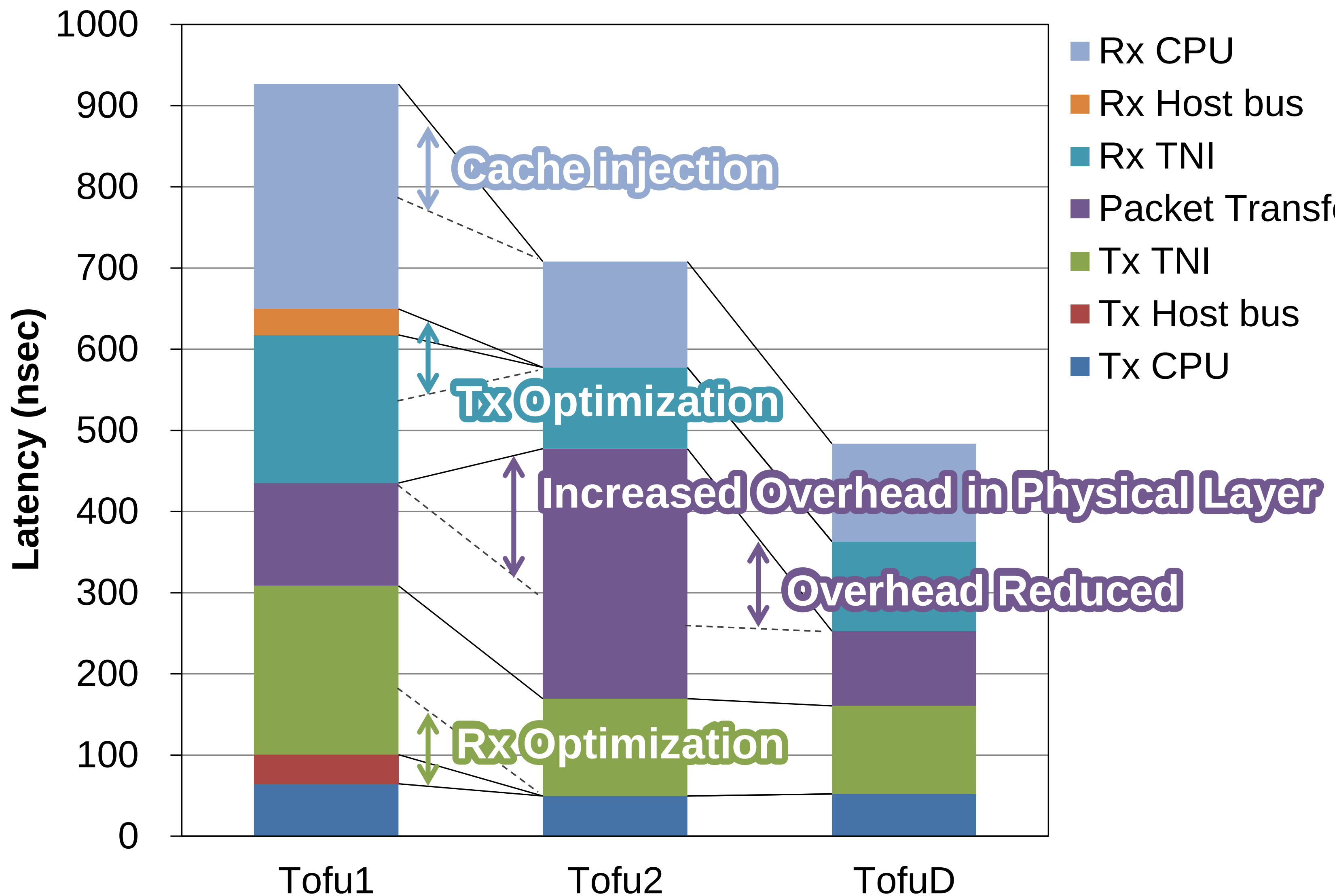
	Communication settings	Latency
Tofu1	Descriptor on main memory	1.15 μs
	Direct Descriptor	0.91 μs
Tofu2	Cache injection OFF	0.87 μs
	Cache injection ON	0.71 μs
TofuD	To/From far CMGs	0.54 μs
	To/From near CMGs	0.49 μs

0.20 μs

0.22 μs

- Tofu2 reduced the Put latency by 0.20 μs from that of Tofu1
  - The cache injection feature contributed to this reduction
- TofuD reduced the Put latency by 0.22 μs from that of Tofu2

Latency Breakdown



- The overhead increase in Tofu2 has been reduced

Put Throughputs

FUJITSU

- One-way Put transfer between nodes on the same board
  - Measured saturated throughput values at message sizes over 1 MiB

	Put throughput	Efficiency
Tofu1	4.76 GB/s	95 %
Tofu2	11.46 GB/s	92 %
TofuD	6.35 GB/s	93 %

- All measured efficiencies exceed 90%
  - The Tofu interconnect family has high bandwidth efficiency
  - The maximum packet size is large enough to encapsulate an IP packet

Injection Rates per Node

- Simultaneous Put transfers to multiple nearest-neighbor nodes
  - Tofu1 and Tofu2 used 4 TNIs, and TofuD used 6 TNIs

	Injection rate	Efficiency
Tofu1 (K)	15.0 GB/s	77 %
Tofu1 (FX10)	17.6 GB/s	88 %
Tofu2	45.8 GB/s	92 %
TofuD	38.1 GB/s	93 %

- The injection rate of TofuD was approximately 83% that of Tofu2
- The efficiencies of Tofu1 were lower than 90%
  - Because of a bottleneck in the bus that connects CPU and ICC
- The efficiencies of Tofu2 and TofuD exceeded 90 %
  - Integration into the processor chip removed the bottleneck

Tofu Barrier – Test Programs

FUJITSU

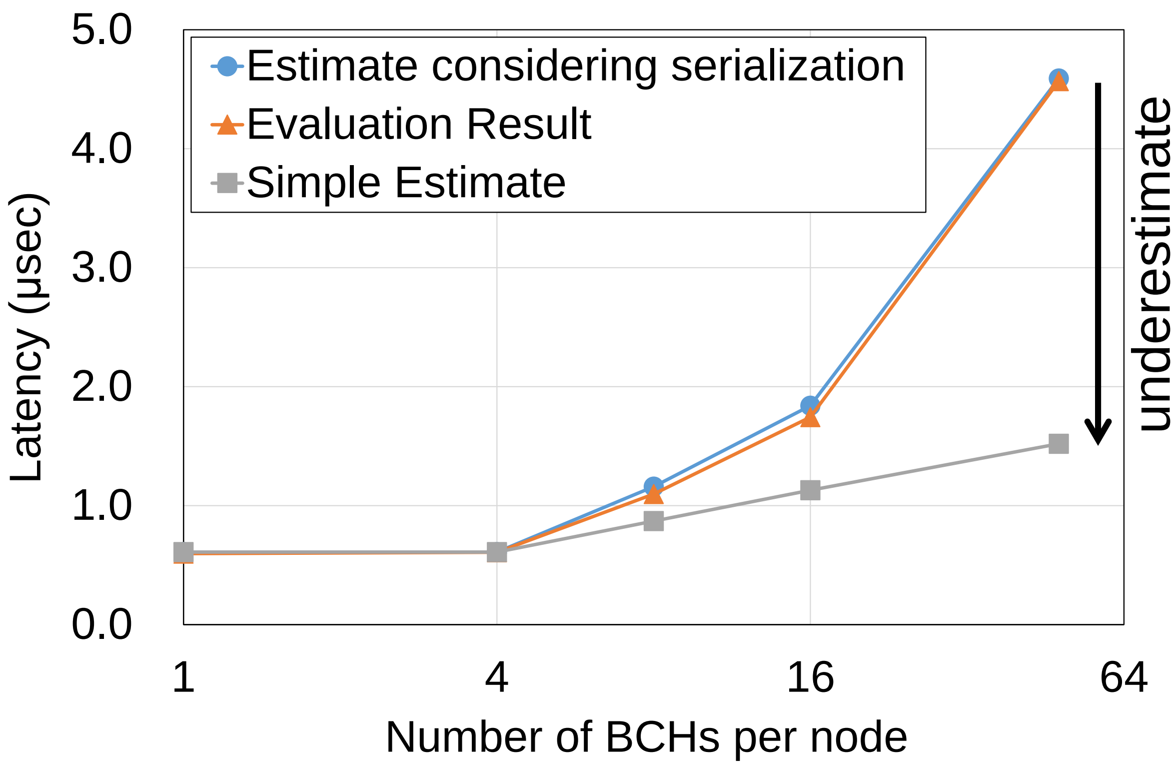
- The test program synchronized multiple BCHs in a node
  - Executed in one processor core to simplify the waveform analysis

Number of BCHs	1	4	8	16	48
Number of used TNIs	1	4	6	6	6
Max. number of BCHs per TNI	1	1	2	3	8
Max. number of BGs per TNI	2	2	5	9	24
Number of communication stages	2	2	4	6	9

- The test programs used the following algorithms;
  - The reduce-broadcast tree algorithm for intra-TNI synchronization
  - The recursive doubling algorithm for inter-TNI synchronization
- Simple estimations were also calculated
  - Accumulated logic circuit delays of BCH (0.48 μs) and BG (0.13 μs)
  - Considered only the number of communication stages

Tofu Barrier – Results

FUJITSU



- The simple estimate results were too low
  - Missing consideration of the serialization of BCHs/BGs processing
  - The modified estimates were consistent with the evaluation results
- BCHs need to be allocated in a round-robin manner to avoid sharing a TNI

Summary

- TofuD is developed for the post-K machine
- TofuD is designed to achieve high-density nodes and enhanced resilience with dynamic packet slicing
- The design of TofuD
  - Node and link configurations
  - CMU and rack packaging
  - Dynamic Packet Slicing
  - Increased Tofu Barrier Resources
- The evaluation results of TofuD
  - Latency was 0.49 μs, which was reduced by 0.22 μs from that of Tofu2
  - Throughput was 6.35 GB/s and the efficiency exceeded 90%
  - Injection rate was 38.1 GB/s, which was approximately 83% that of Tofu2
  - The evaluation results showed that it is necessary to allocate BCHs without sharing a TNI

shaping tomorrow with you