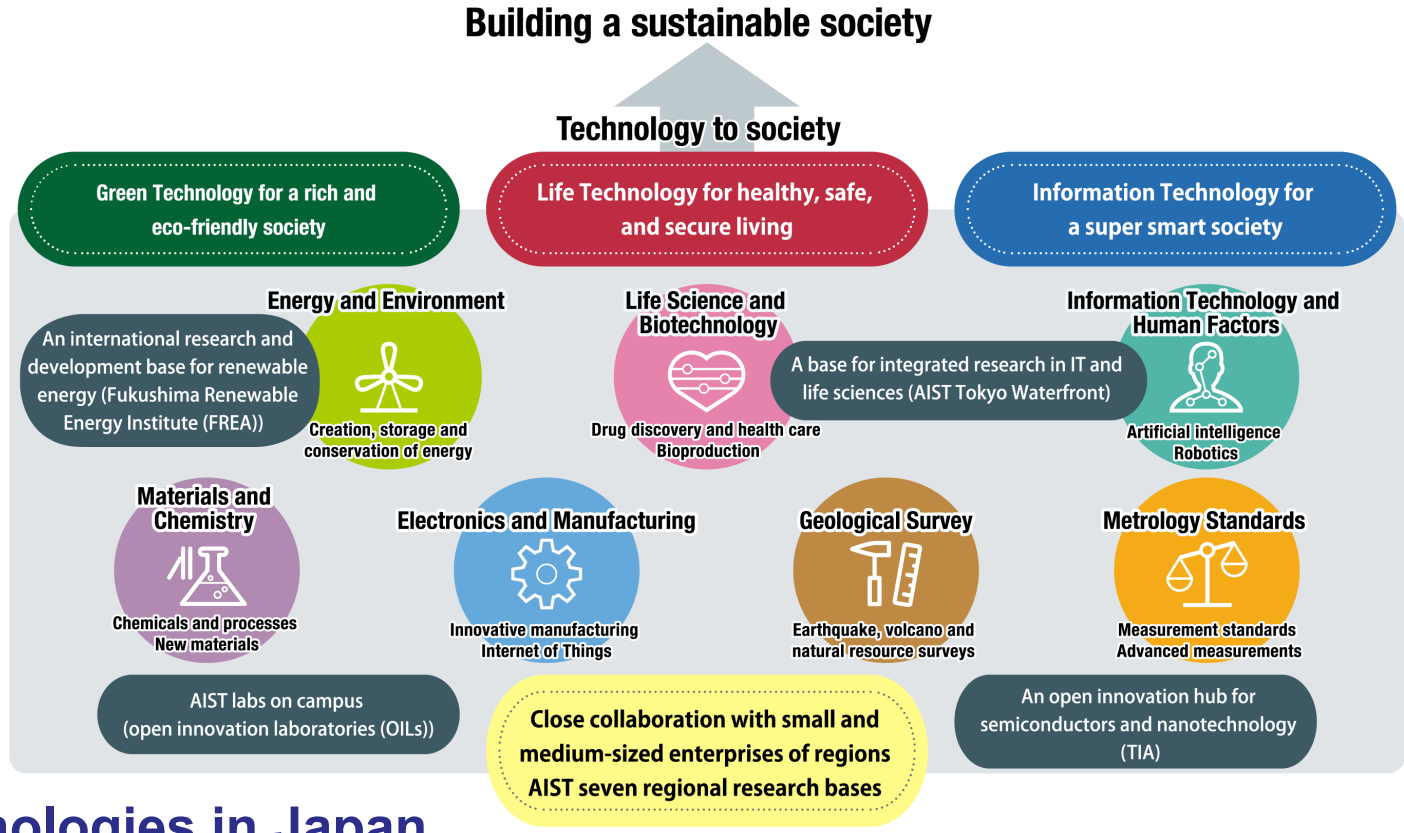
# The Latest MVAPICH2 Performance Results on ABCI

Shinichiro Takizawa

The National Institute of Advanced Industrial Science and Technology (AIST), Japan

# Introduction of AIST

- A research institute as a part of the Ministry of Economy, Trade and Industry (METI) of Japan
- Our mission
  - Integrate scientific and engineering knowledge to address industry and society needs
  - Bridge the gap between innovative technological seeds and commercialization




**We built ABCI for popularize AI technologies in Japan**

# ABCI: The World's First Large-Scale Open AI Infrastructure

東京大学 THE UNIVERSITY OF TOKYO AIST

Univ. Tokyo / Kashiwa II Campus



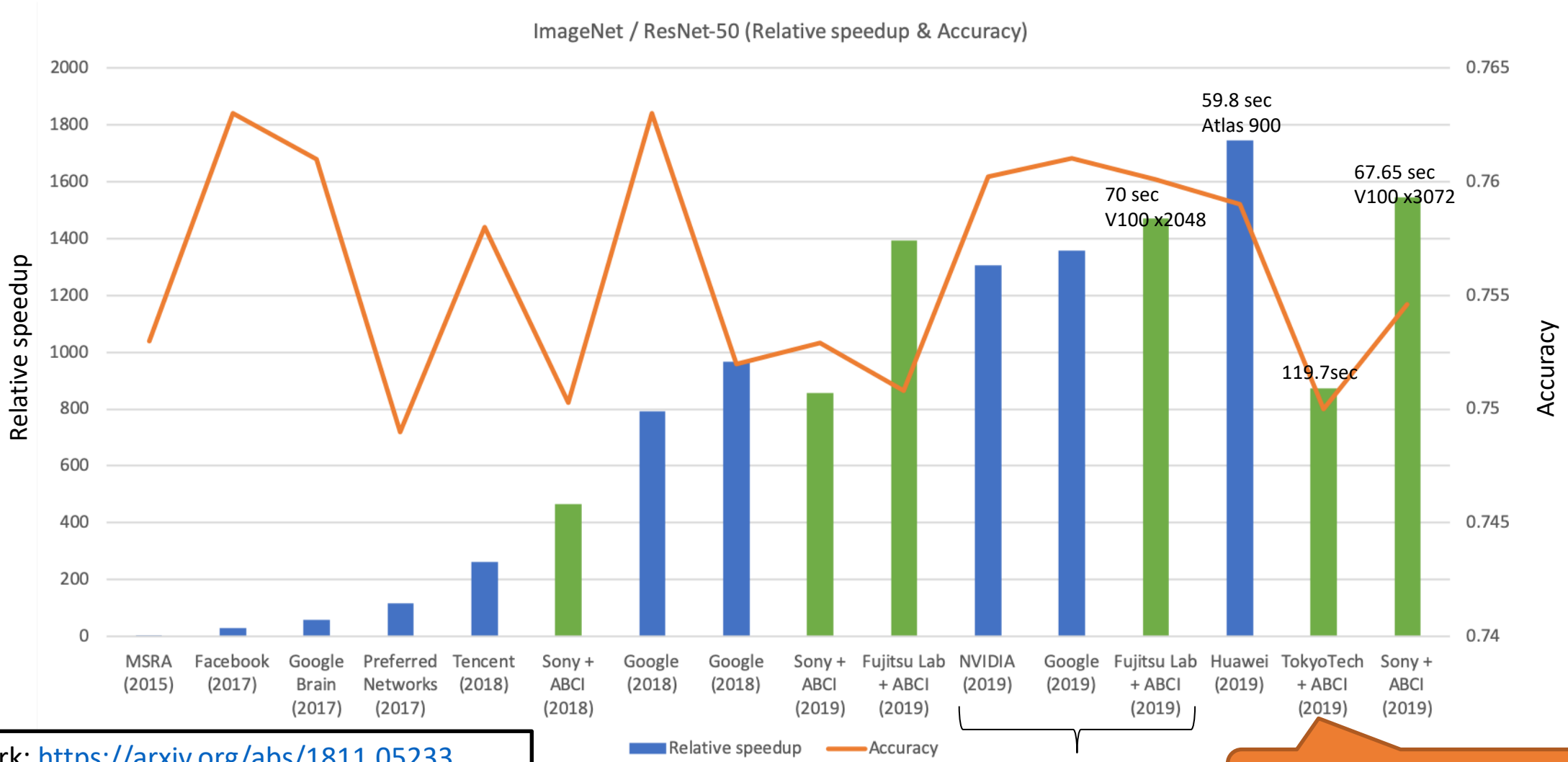
OPERATED SINCE AUGUST 1ST, 2018

## ABCI AI Bridging Cloud Infrastructure

- **Open, Public, and Dedicated** infrastructure for Al & Big Data Algorithms, Software, and Applications
- **Capacity computing system** for diverse AI applications, at the same time, capability computing system for grand challenge AI problems
- **Open Innovation Platform** to accelerate joint academic-industry R&D for AI

**Peak Performance:**
550 PFLOPS (FP16)
37 PFLOPS (FP64)
**Effective Performance: (as of Nov 2019)**
19.88 PFLOPS (#8 in TOP500)
14.423 GFLOPS/W (#6 in GREEN500)
508.85 TFLOPS (#4 in HPCG)
**Power Usage:** < 2.3 MW
**Average PUE:** < 1.1 (Estimated)

# ImageNet Training Performance on ABCI



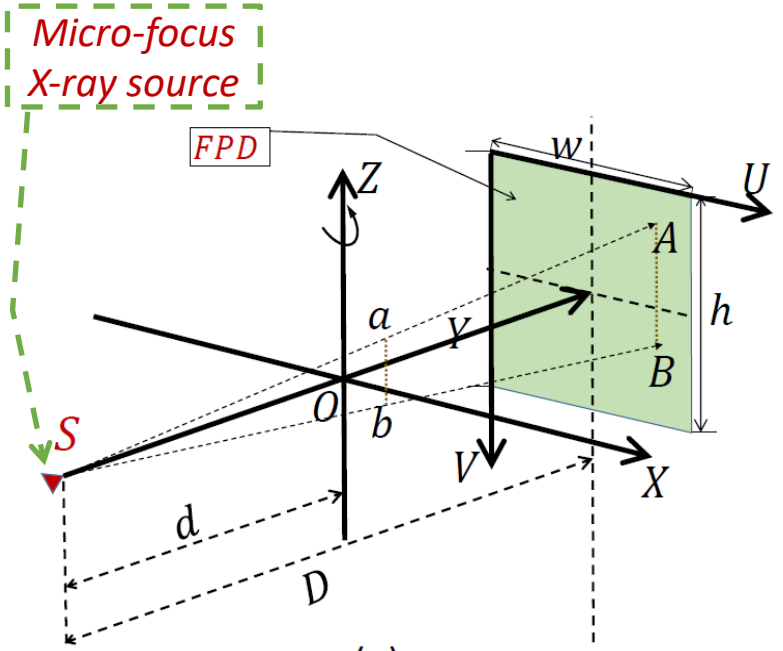

SONY's work: https://arxiv.org/abs/1811.05233
Fujitsu lab's work: https://arxiv.org/abs/1903.12650

They use K-FAC, not SGD. Training completes in 30 epochs.

# iFDK: CT Image Reconstruction Framework on ABCI

SC19 Denver, CO | hpc is now.

- A high-speed and high-resolution CT image reconstruction framework running on GPU clusters
  - Create 3D images from multiple 2D x-ray images
    - $4096^3$(256GB), $8192^3$(2TB) in a few minutes!
  - Demands for high-resolution CT image: non-invasive inspection, reverse engineering, etc.
- Our Achievements
  - New FDK algorithm whose computational cost is 1/6 against traditional algorithms
  - An efficient CUDA kernel that implement the algorithm
  - A parallel and distributed pipeline for high-resolution large volume image reconstruction
  - Good scalability on ABCI
    - 4K in 30 secs, 8K in 2 mins




(a)
CBCT geometry and trajectory

Thursday, 3:30pm - 4pm, @301-302-303
Peng Chen, Mohamed Wahib, Shinichiro Takizawa, Ryousei Takano, Satoshi Matsuoka.
iFDK: A Scalable Framework for Instant High-resolution Image Reconstruction

# ABCI DETAIL

## High-Performance Computing System

**550** PFlops(FP16), **37.2** PFlops(FP64)
**476** TiB Memory, **1.74** PB NVMe SSD

### Computing Nodes (w/ GPU) x 1088

| | |
|---|---|
| GPU | NVIDIA Tesla V100 SXM2 x 4 |
| CPU | Intel Xeon Gold 6148 (2.4GHz/20cores) x 2 |
| Memory | 384GiB |
| Local Storage | Intel SSD DC P4600 (NVMe) 1.6TB x 1 |
| Interconnect | InfiniBand EDR x 2 |

### Multi-platform Nodes (w/o GPU) x 10

- Intel Xeon Gold 6132 (2.6GHz/14cores) x 2
- 768GiB Memory, 3.8TB NVMe SSD, 1.5TB Intel Optane x2

**Interactive Nodes x 4**

**Management and Gateway Nodes x 15**

## Large-scale Storage System

### 1 PB User Home Directory (Lustre)

DDN SFA14KX (w/ SS9012 Enclosure x 10) x1
- 7.68TB SAS SSD x 185 for data
- 960GB SAS SSD x 13 for metadata

### 22 PB Group Shared Directory (GPFS)

DDN SFA14K (w/ SS8462 Enclosure x 10) x3
- 12TB 7.2Krpm NL-SAS HDD x 2400
- 3.84TB SAS SSD x 216

### 17 PB Object Storage (Scality RING)

HPE Apollo 4510 Gen10 x 24
- 12TB SATA HDD x 1440
- 3.2TB SSD x 24

## Interconnect (InfiniBand EDR)

- Mellanox CS7500 x 2
- Mellanox SB7890 x 229

## Service Network (10GbE)

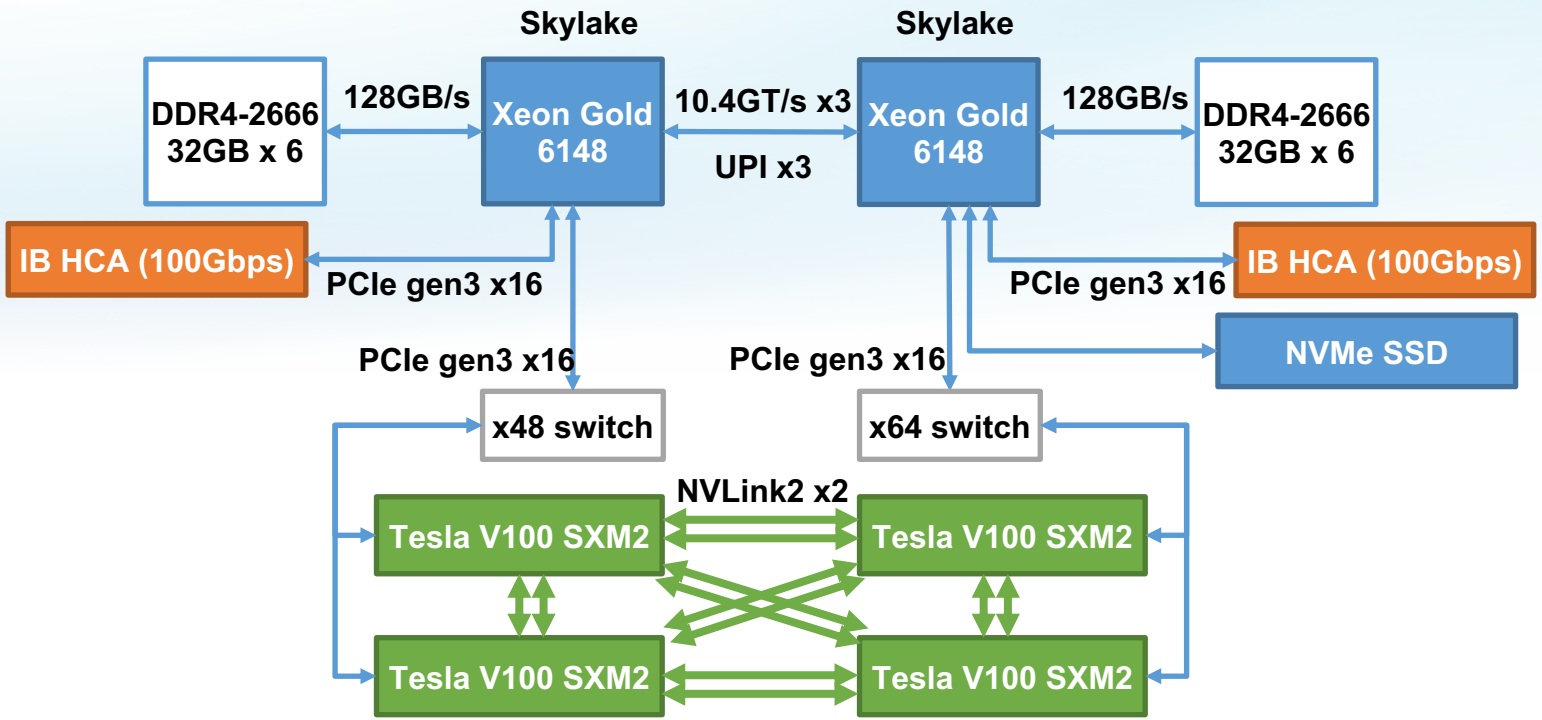
## Gateway and Firewall

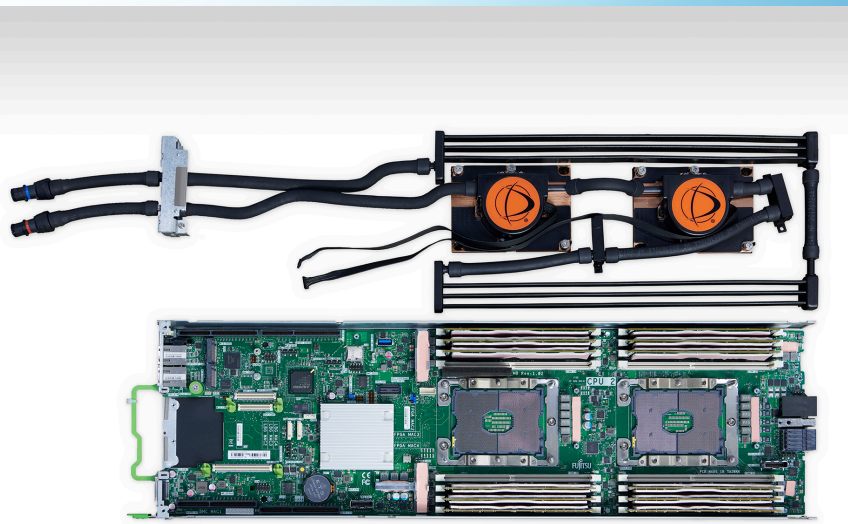
- Nexsus 3232C x2
- FortiGate 1500D x2
- FortiAnalyzer 400E x1

100Gbps SINET5

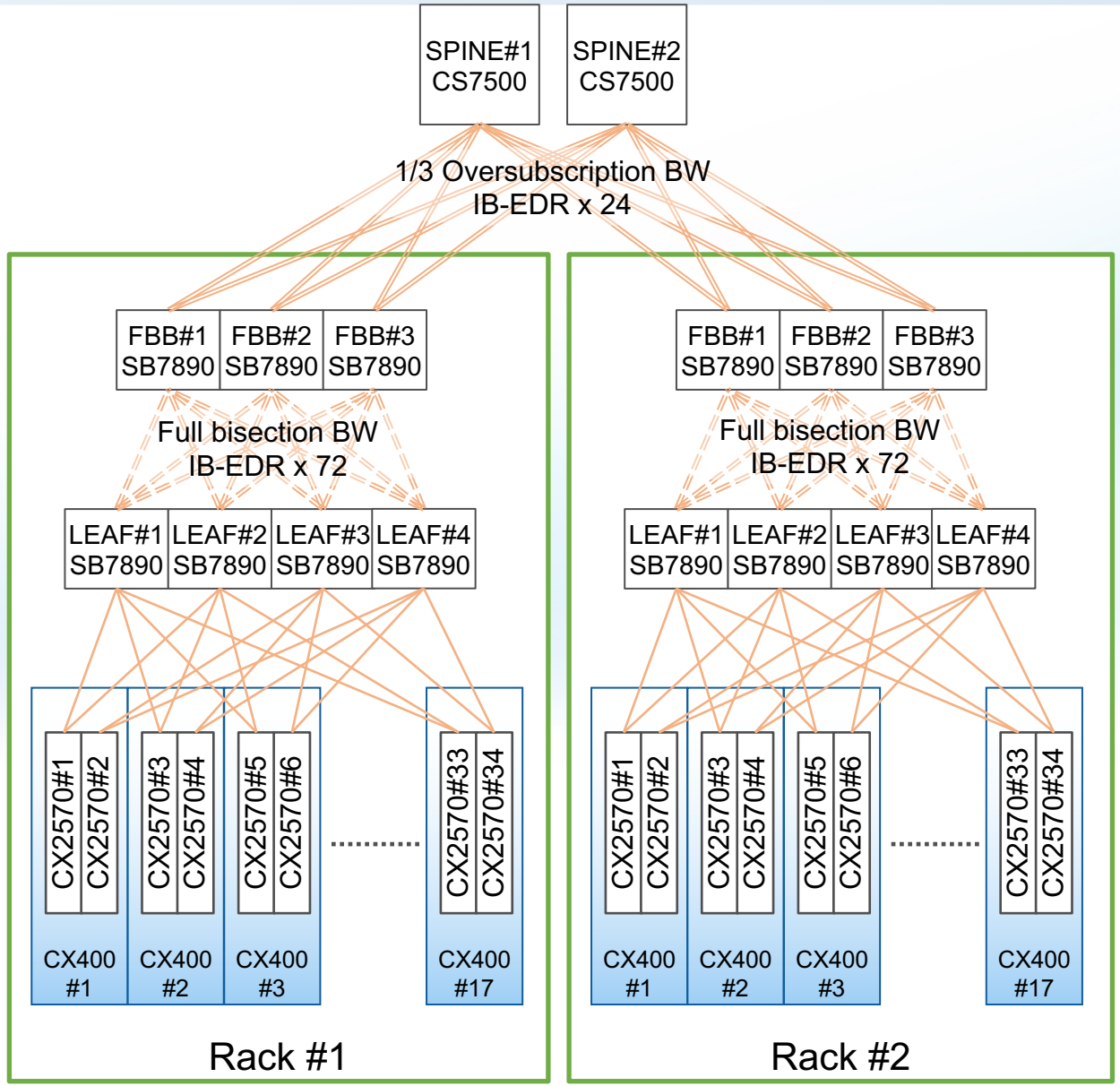
# ABCI Compute Node

| FUJITSU PRIMERGY Server (2 servers in 2U) | |
|---|---|
| CPU | Xeon Gold 6148 (27.5M Cache, 2.40 GHz, 20 Core) x2 |
| GPU | NVIDIA Tesla V100 (SXM2) x4 |
| Memory | 384GiB DDR4 2666MHz RDIMM |
| Local Storage | 1.6TB NVMe SSD (Intel SSD DC P4600 u.2) x1 |
| Interconnect | InfiniBand EDR x2 |



# ABCI Node Rack / Interconnect




- **Dense-packaged rack: 34 nodes, 136 Tesla V100**
  - Theoretical peak performance per rack : 1.16 PFlops (FP64), 17 PFlops (FP16) c.f. Google TPU 3.0 Pod (>100PFlops/8racks)
  - Power consumption per rack: 67.33 kW
- **Interconnect**
  - Fat-tree topology
  - Intra-rack: full bisection BW
  - Inter-rack : 1/3 over-subscription (2400/6800)
  - Without adoptive routing
  - Without Mellanox SHARP

x 32

InfiniBand EDR x4
InfiniBand EDR x6
InfiniBand EDR x1

# Reasons for the Interconnect Design

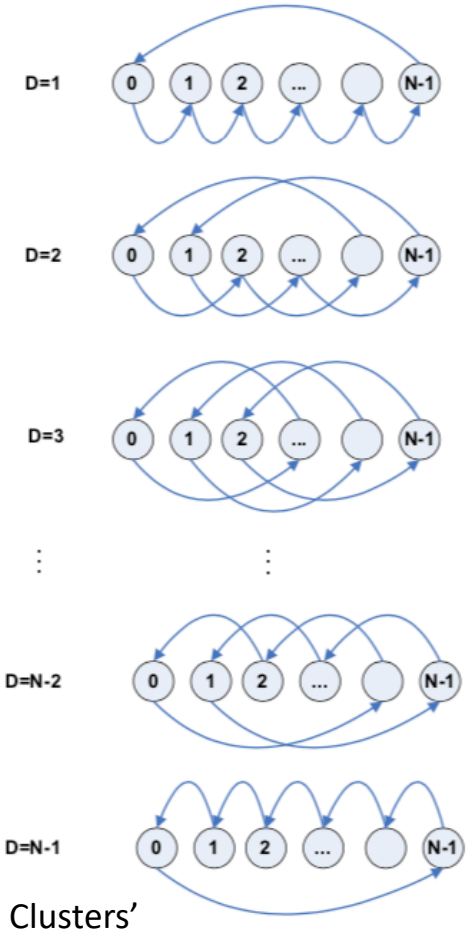
- 1/3 over-subscription network
  - A cost effective solution
    - Many DL training/inference apps do not use large number of GPUs/nodes
      => High bandwidth network is required by only a small set of nodes
    - 98% jobs on ABCI use single node, 1% jobs on ABCI use 2-34 nodes (Oct. 2019)
- Without adoptive routing
  - InfiniBand is shared by compute and IO communication, and the delivery vendor suggested not to use adaptive routing on such a configuration
- Without Mellanox SHARP
  - We use EDR, and EDR switches which do not support large size message in SHARP

# Performance Impact on MPI under 1/3 over-subscription without adoptive routing

Used **mpiGraph** to measure achievable bandwidth

| # Racks | Min MB/s | Max MB/s | Avg MB/s |
|---|---|---|---|
| 1 (34 Nodes) | 63.5% | 100.0% | 94.2% |
| 2 (68 Nodes) | 49.6% | 100.0% | 84.3% |
| 3 (102 Nodes) | 42.7% | 100.0% | 76.1% |
| 4 (136 Nodes) | 40.9% | 100.0% | 71.9% |

Comm. pattern




As degree of parallelism of DL jobs are not large, many ABCI users do not suffer from this bandwidth degradation.

Figure from 'Contention-free Routing for Shift-based Communication in MPI Applications on Large-scale Infiniband Clusters'

# ABCI Software Stack

| | |
|---|---|
| **Operating System** | RHEL / CentOS 7.6 |
| **Job Scheduler** | Univa Grid Engine 8.6.3 |
| **Container Engine** | Docker 17.12.0 (Users can use only supported container images)<br>Singularity 2.6.1 (Users can use any container images) |
| **MPI** | Intel MPI 2018.2.199<br>MVAPICH2 2.3, 2.3.2 / MVAPICH2-GDR 2.3, 2.3.1, 2.3.2<br>OpenMPI 1.10.7, 2.1.3, 2.1.5, 2.1.6, 3.0.3, 3.1.0, 3.1.2, 3.1.3 |
| **Development tools** | Intel Parallel Studio XE Cluster Edition 2017.8, 2018.2, 2018.3, 2019.3<br>PGI Professional Edition 17.10, 18.10, 19.3<br>NVIDIA CUDA SDK 8.0, 9.0, 9.1, 9.2, 10.0, 10.1<br>cuDNN 5.1, 6.0, 7.0, 7.1, 7.2, 7.3, 7.4, 7.5, 7.6<br>NCCL 1.3.5, 2.1, 2.2, 2.3, 2.4<br>Intel MKL 2017.8, 2018.2, 2018.3, 2019.3<br>GCC, Python, Ruby, R, OpenJDK, Go, Perl |
| **Deep Learning** | Caffe, Caffe2, TensorFlow, Theano, Torch, PyTorch, CNTK, MXnet, Chainer, Keras, etc.<br>*(Frameworks provided by NVIDIA GPU Cloud can also be deployed)* |
| **Big Data Processing** | Hadoop, Spark |

# Combining Software on ABCI (1/3)

- ABCI software are managed by Environment Modules
  - Environment Modules are configured so that necessary libs are loaded by considering version dependencies

| OpenMPI | No CUDA | CUDA 8.0 | CUDA9.0 | CUDA9.2 | CUDA10.0 |
|---|---|---|---|---|---|
| 3.1.3 | Yes | | | | |
| 2.1.6 | Yes | Yes | Yes | Yes | Yes |
| 2.1.5 | Yes | Yes | Yes | Yes | Yes |
| … | | | | | |

- E.g.) OpenMPI + CUDA

```
$ module load openmpi/2.1.6
```

➡ OpenMPI 2.1.6 w/o CUDA support is loaded

```
$ module load cuda/9.2/9.2.148.1
$ module load openmpi/2.1.6
```

➡ CUDA-aware OpenMPI 2.1.6 w/ CUDA 9.2 support is loaded

```
$ module load cuda/10.0/10.0.130.1
$ module load openmpi/2.1.6
```

➡ CUDA-aware OpenMPI 2.1.6 w/ CUDA 10.0 support is loaded

# Combining Software on ABCI (2/3)

- E.g.) MVAPICH2-GDR + CUDA

| MVAPICH2-GDR | CUDA9.0 | CUDA9.2 | CUDA10.0 |
|---|---|---|---|
| 2.3.2 | | Yes | Yes |
| 2.3 | Yes | Yes | |
| … | | | |

```
$ module load cuda/9.2/9.2.148.1
$ module load mvapich/mvapich2-gdr/2.3.2
```

➡ MVAPICH2-GDR 2.3.2 w/ CUDA 9.2 support is loaded

```
$ module load cuda/10.0/10.0.130.1
$ module load mvapich/mvapich2-gdr/2.3.2
```

➡ MVAPICH2-GDR 2.3.2 w/ CUDA 10.0 support is loaded

In case of unsupported version CUDA is loaded…

```
$ module load cuda/9.0/9.0.176.4
$ module load mvapich/mvapich2-gdr/2.3.2
WARNING: mvapich/mvapich2-gdr/2.3.2 cannot be loaded due to missing prereq.
HINT: at least one of the following modules must be loaded first: cuda/9.2 cuda/10.0
```

➡ Provide a hint to load supported version CUDA

# Combining Software on ABCI (3/3)

- E.g.) MVAPICH2 + CUDA
  - For the latest MVAPICH2, ABCI only supports no CUDA version
  - Use Spack to install CUDA-aware MVAPICH2

```
$ spack install cuda@9.0.176.4
$ spack install mvapich2@2.3 +cuda
```

Configuration for using ABCI-provided CUDA, instead of installing CUDA

Spack: packages.yaml

```
packages:
  cuda:
    paths:
      cuda@8.0.61.2%gcc@4.8.5:    /apps/cuda/8.0.61.2
      cuda@9.0.176.4%gcc@4.8.5:   /apps/cuda/9.0.176.4
      ...
      cuda@10.1.243%gcc@4.8.5:    /apps/cuda/10.1.243
    buildable: False
```

# MVAPICH2 PERFORMANCE ON ABCI

# Collective Comm. Performance on ABCI

- Measured performance of two collective comm. patterns
  - Allreduce: essential in distributed training of data parallel DL
  - Reduce: the most heavy comm. in our application (iFDK)
- Used OSU Micro-Benchmarks 5.6.1
  - NCCL Bench 2.0.0 is used for NCCL
  - Default parameters, 4 MPI proc/node

| Compiler | GCC 4.8.5 |
|---|---|
| CUDA | 10.0.130 |

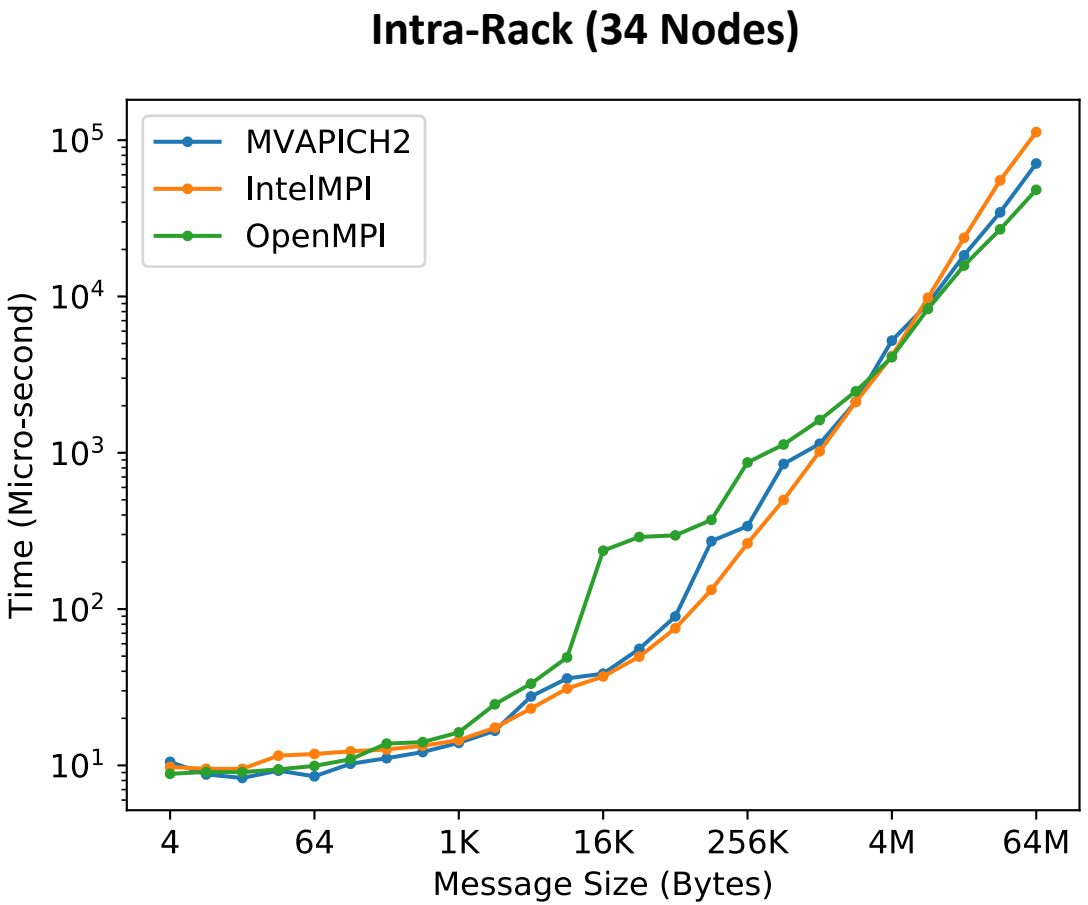
**Host to host Transfer**

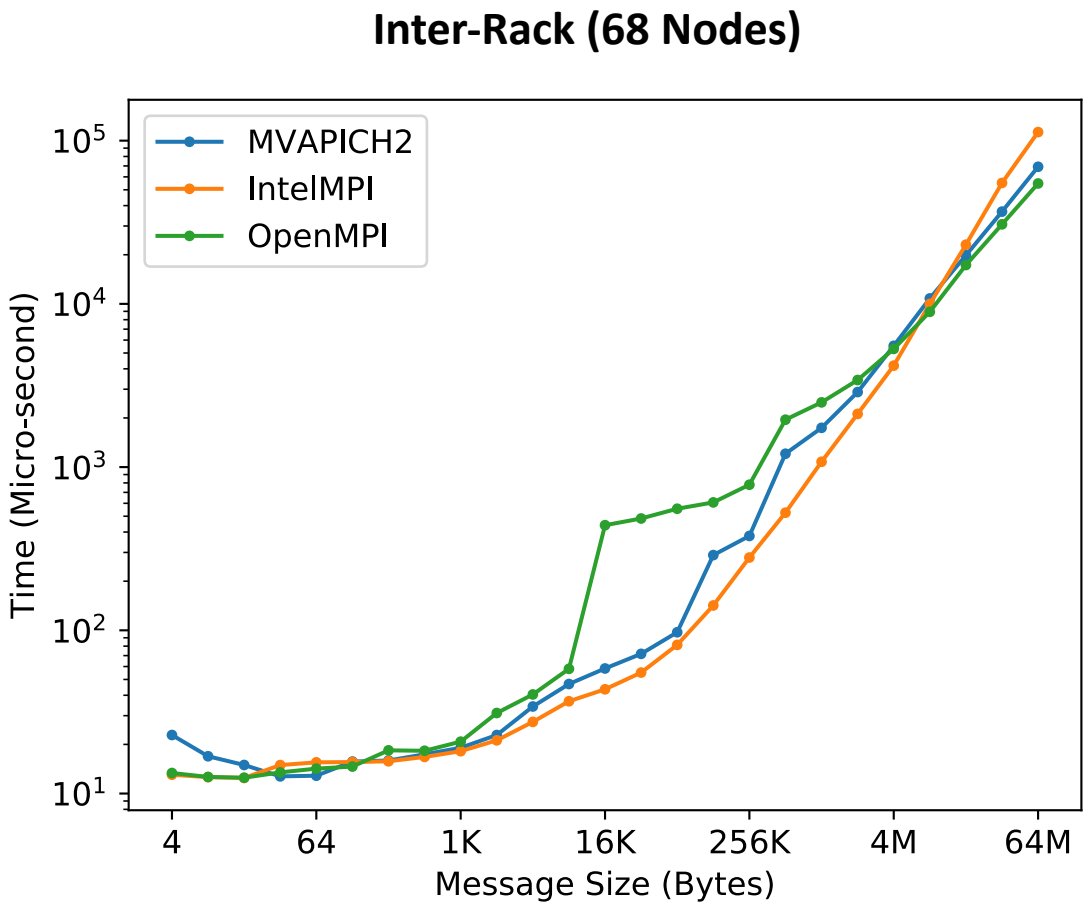
| MPI | Version |
|---|---|
| MVAPICH2 | MVAPICH2 2.3.1 |
| IntelMPI | IntelMPI 2018 Update 2 |
| OpenMPI | OpenMPI 3.1.3 |

**GPU to GPU Transfer**

| MPI, NCCL | Version |
|---|---|
| MVAPICH2 | MVAPICH2 2.3.1 |
| MVAPICH2-GDR | MVAPICH2-GDR 2.3.1, gdrcopy 1.2 |
| OpenMPI | OpenMPI 3.1.3 |
| NCCL | NCCL 2.4.7 |

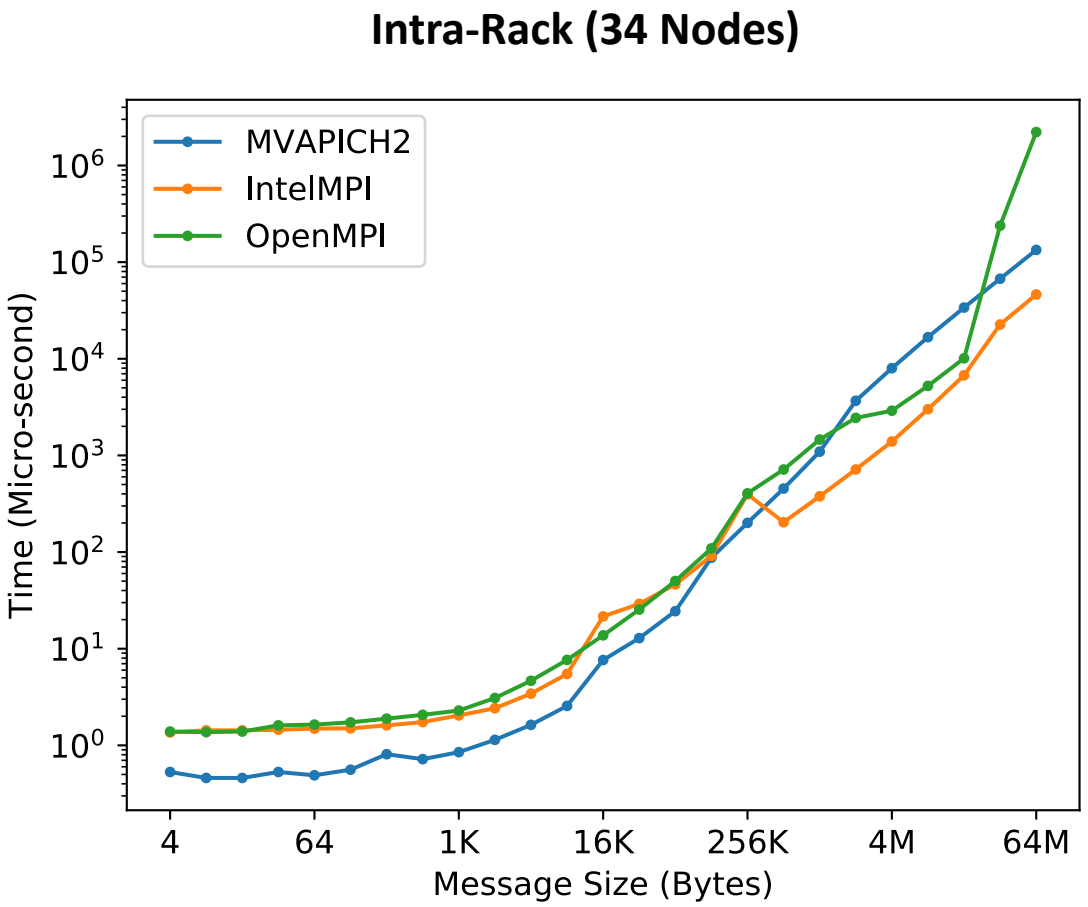
# H2H AllReduce

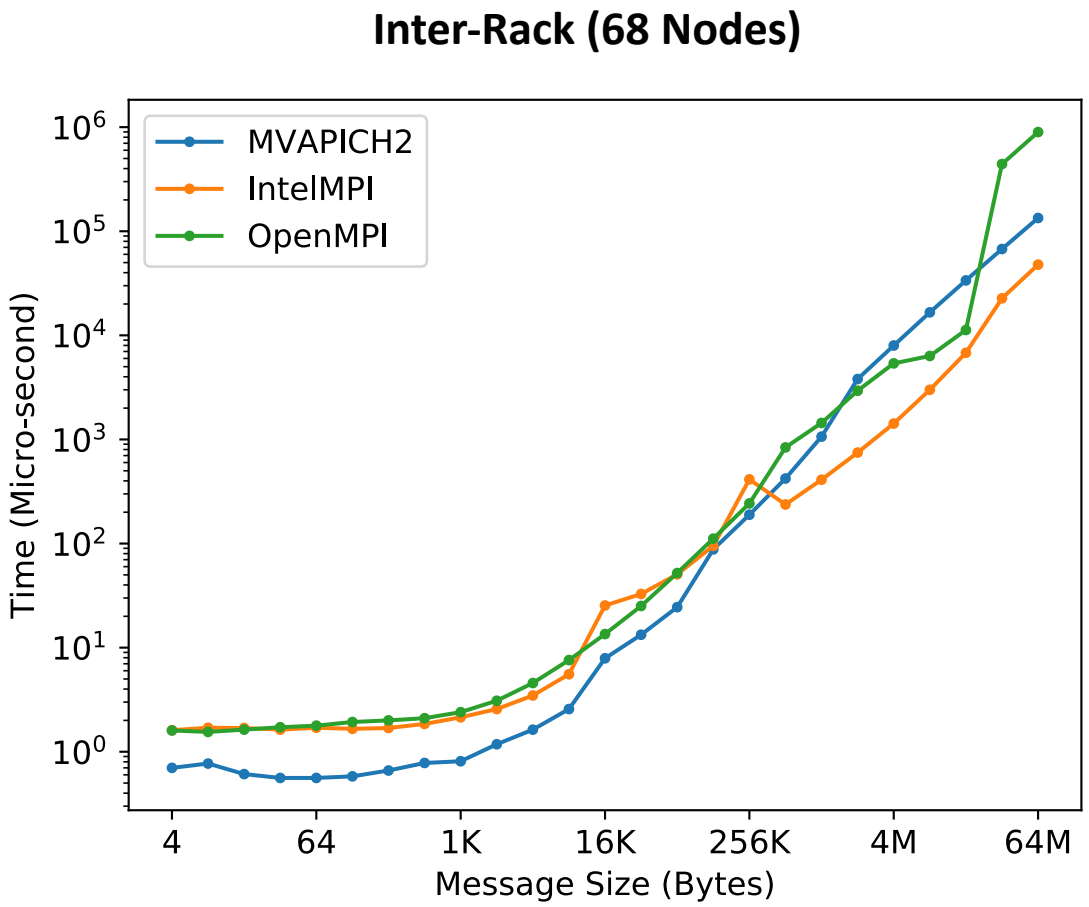



- MVAPICH2 is comparable to IntelMPI
- Performance do not suffer from 1/3 over-subscription inter-rack bandwidth

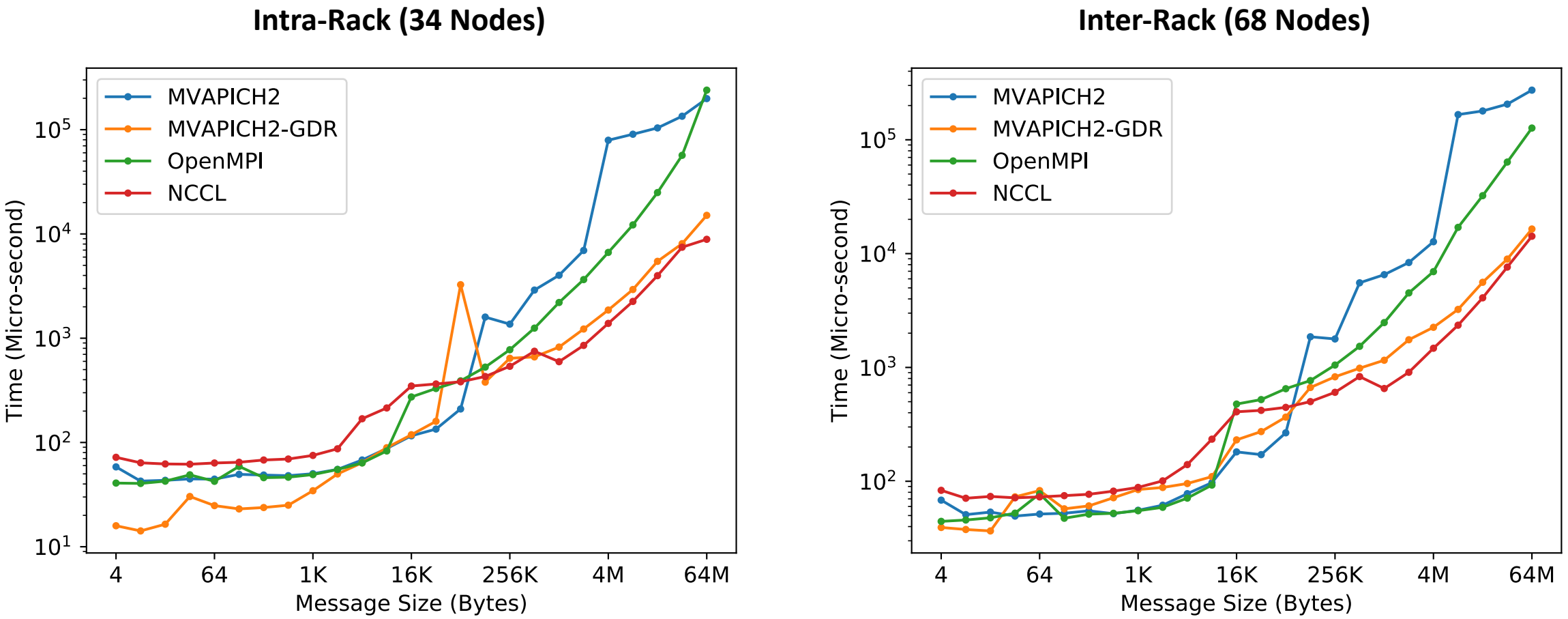
# H2H Reduce



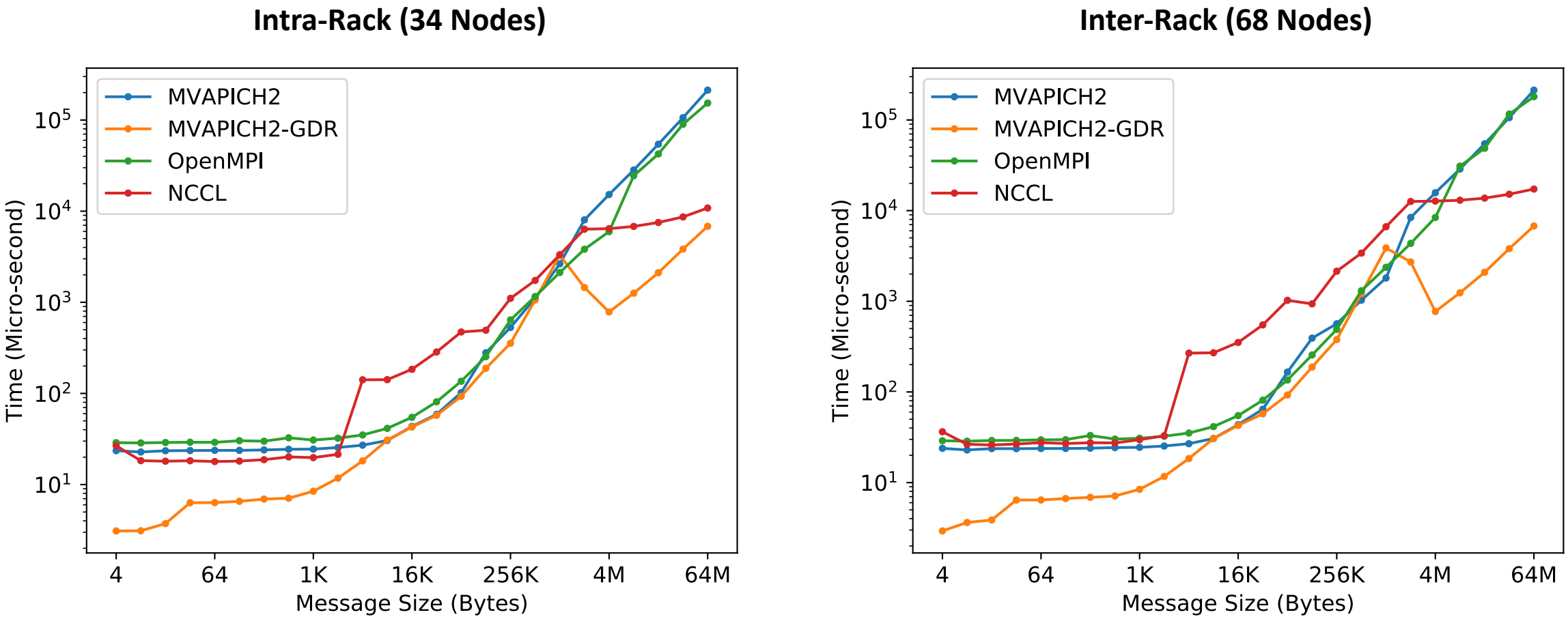

MVAPICH2 is better than others in small message, but IntelMPI outperforms MVAPICH in large message

# D2D AllReduce




MVAPICH2-GDR and NCCL outperforms others in large messages

# D2D Reduce




MVAPICH2-GDR achieves the best performance

# Summary


https://abci.ai/

- Introduce ABCI
  - Architecture, software stack and recent achievements
  - Detail network architecture
- Show MPI communication performance on ABCI
  - Effect of 1/3 inter-rack over-subscription bandwidth network
  - Two MPI Collectives: Allreduce, Reduce
- Future work
  - Encourage our DL users to use distributed training
  - Encourage our users to use MVAPICH!