Post-K: A Game Charing Supercomputer for Convergence of HPC and Big Data / AI.

Satoshi Matsuoka
Director, Riken Center for Computational Science / Professor, Tokyo Institute of Technology

ORAP Forum 20181106
Paris, France
Modern AI is enabled by Supercomputing

• 25 years of AI winter after failure of symbolic logic based methods (e.g., Prolog, ICOT) -> resurrection by DNN, basic algorithms in the 1980s but too expensive -> HPC made machines 10 million times faster in 30 years -> expensive training now possible

• Recent trends require more supercomputing power
  – Deeper, more complex networks (Capsule Networks)
  – Complex, multidimensional data (e.g., 3-D Hi-Res images)
  – Increasing training sets (incl. GANs)
  – Coupling with high-fidelity simulations
  – Etc.

*Fig. 2: Andrew Ng (Baidu) “What Data Scientists Should Know about Deep Learning”*
What is meant by Convergence of HPC & AI?

- Acceleration of Simulation (first principles methods) with AI (empirical method)
  - Interpolation & Extrapolation of long trajectory MD
  - Reducing parameter space on Pareto optimization of results
  - Adjusting convergence parameters for iterative methods etc.

- **AI replacing simulation**
  - When exact physical models are unclear, or excessively costly to compute

- **Acceleration of AI with HPC**
  - HPC Processing of training data - data cleansing
  -Acceleration of (Parallel) Training: Deeper networks, bigger training sets, complicated networks, high dimensional data…
  - Acceleration of Inference: above + real time streaming data
  - Various modern training algorithms: Reinforcement learning, GAN, Dilated Convolution, etc.
4 Layers of Parallelism in DNN Training

- **Hyper Parameter Search**
  - Searching optimal network configs & parameters
  - Parallel search, massive parallelism required

- **Data Parallelism**
  - Copy the network to compute nodes, feed different batch data, average => network reduction bound
  - TOFU: Extremely strong reduction, x6 EDR Infiniband

- **Model Parallelism (domain decomposition)**
  - Split and parallelize the layer calculations in propagation
  - Low latency required (bad for GPU) => strong latency tolerant cores + low latency TOFU network

- **Intra-Chip ILP, Vector and other low level Parallelism**
  - Parallelize the convolution operations etc.
  - SVE FP16+INT8 vectorization support + extremely high memory bandwidth w/HBM2

- Post-K could become world’s biggest & fastest platform for DNN training!
Deep Learning is “All about Scale”
Massive Parallelization is the key

- Data-parallel training with (Asynchronous) Stochastic Gradient Descent
  - Replicate network to all the nodes, feed different data, average the gradients periodically
  - Network All-Reduce Reduction in Megabytes~Gigabytes becomes the bottleneck at scale
  - NVIDIA: NVLink Hardware + NICL library (up to 8 GPUs on DGX-1, 16 on DGX-2 w/ NVL Switch)

Fig. 2: Andrew Ng (Baidu) “What Data Scientists Should Know about Deep Learning”

Fig. 3: Simplified DL workflow with ASGD per iteration:
1. Compute gradient
2. Exchange gradients via all-reduce; and
3. Update network parameters

November 7, 2018

Jens Domke
Predicting Statistics of Asynchronous SGD Parameters for a Large-Scale Distributed Deep Learning System on GPU Supercomputers

Background

• In large-scale Asynchronous Stochastic Gradient Descent (ASGD), mini-batch size and gradient staleness tend to be large and unpredictable, which increase the error of trained DNN

Proposal

• We propose a empirical performance model for an ASGD deep learning system SPRINT which considers probability distribution of mini-batch size and staleness

Objective function $E$

$$E(W(t)) = -\eta \sum_i \nabla E_i W(t+1)$$

DNN parameters space

Mini-batch size

- $N_{\text{Subbatch}}$: # of samples per one GPU iteration

Staleness

- $N_{\text{Minibatch}}$, $N_{\text{Staleness}}$

- Predicted
- Measured

Interconnect Performance as important as GPU Performance to accelerate DL

- ASGD DL system SPRINT (by DENSO IT Lab) and DL speedup prediction with performance model

\[ T_{Epoch} \approx \frac{N_{File} \times T_{GPU}}{N_{Node} \times N_{GPU} \times N_{Subbatch}} \]

- Data measured on T2 and KFC (both FDR) fitted to formulas
- Allreduce time (\( \in T_{GPU} \)) dep. on #nodes and #DL_parameters

\[ T_{Barrier} = (\alpha \log_2(N_{Node}) + \beta) \times N_{Param} \]

- Other approaches == similar improvements:
  - Cuda-Aware CNTK optimizes communication pipeline \( \rightarrow \) 15%—23% speedup (Banerjee et al. “Re-designing CNTK Deep Learning Framework on Modern GPU Enabled Clusters”)
  - Reduced precision (FP[16|8|1]) to minimize msg. size w/ no or minor accuracy loss

<table>
<thead>
<tr>
<th></th>
<th>( N_{Node} )</th>
<th>( N_{Subbatch} )</th>
<th>Average mini-batch size</th>
<th>Epoch time[s]</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>8</td>
<td>8</td>
<td>165.1</td>
<td>1779</td>
<td>-</td>
</tr>
<tr>
<td>FP16</td>
<td>7</td>
<td>22</td>
<td>170.1</td>
<td>1462</td>
<td>1.22</td>
</tr>
<tr>
<td>EDR IB</td>
<td>12</td>
<td>11</td>
<td>166.6</td>
<td>1245</td>
<td>1.43</td>
</tr>
<tr>
<td>FP16 + EDR IB</td>
<td>8</td>
<td>15</td>
<td>171.5</td>
<td>1128</td>
<td>1.58</td>
</tr>
</tbody>
</table>

Fig. 4: Oyama et al. “Predicting Statistics of Asynchronous SGD Parameters for a Large-Scale Distributed Deep Learning System on GPU Supercomputers”
Overview of TSUBAME3.0
BYTES-centric Architecture, Scalability to all 2160 GPUs, all nodes, the entire memory hierarchy

DDN Storage
(Lustre FS 15.9PB+Home 45TB)

Full Bisection Bandwidth
Intel Omni-Path Interconnect. 4 ports/node
Full Bisection / 432 Terabits/s bidirectional
~x2 BW of entire Internet backbone traffic

540 Compute Nodes SGI ICE XA + New Blade
Intel Xeon CPU x 2+NVIDIA Pascal GPUx4 (NV-Link)
256GB memory 2TB Intel NVMe SSD
47.2 AI-Petaflops, 12.1 Petaflops

Full Operations
Aug. 2017
TSUBAME3.0 Co-Designed SGI ICE-XA Blade (new)
- No exterior cable mess (power, NW, water)
- Plan to become a future HPE product
Liquid Cooled
“Hot Pluggable” ICE-XA Blade
Smaller than 1U server, no cables or pipes

Xeon x 2
PCIe Switch
> 20 TeraFlops
DFP
256GByte Memory

PCIe NVMe Drive Bay x 4

100Gbps x 4
≈ 400Gbps

Liquid Cooled NVMe
15 Compute Racks
4 DDN Storage Racks
3 Peripheral & SW racks
Total 22 Racks

144 GPUs & 72 CPUs/rack
Integrated
100/200Gbps Fabric Backplane
TSUBAME3: A Massively BYTES Centric Architecture for Converged BD/AI and HPC

- **Intra-node GPU via NVLink**: 20~40GB/s
- **Inter-node GPU via OmniPath**: 12.5GB/s fully switched
- **HBM2**: 64GB, 2.5TB/s
- **DDR4**: 256GB, 150GB/s
- **Intel Optane**: 1.5TB, 12GB/s (planned)
- **NVMe Flash**: 2TB, 3GB/s

16GB/s PCIe Fully Switched

Any "Big" Data in the system can be moved to anywhere via RDMA speeds minimum 12.5GBytes/s also with Stream Processing Scalable to all 2160 GPUs, not just 8

~4 Terabytes/node Hierarchical Memory for Big Data / AI (c.f. K-computer 16GB/node)

⇒ Over 2 Petabytes in TSUBAME3, Can be moved at 54 Terabyte/s or 1.7 Zetabytes / year
TSUBAME3: A MassivelyBYTES Centric Architecture for Converged BD/AI and HPC

- Intra-node GPU via NVLink: 20~40GB/s
- Inter-node GPU via OmniPath: 12.5GB/s fully switched
- HBM2 64GB 2.5TB/s
- DDR4 256GB 150GB/s
- Intel Optane 1.5TB 12GB/s (planned)
- NVMe Flash 2TB 3GB/s
- ~4 Terabytes/node Hierarchical Memory for Big Data / AI (c.f. K-computer 16GB/node)
- Over 2 Petabytes in TSUBAME3, Can be moved at 54 Terabyte/s or 1.7 Zetabytes/year

Any “Big” Data in the system can be moved to anywhere via RDMA speeds minimum 12.5GBytes/s also with Stream Processing Scalable to all 2160 GPUs, not just 8
TSUBAME3.0 became the first large production petaflops-scale supercomputer in the world to be #1 on the “Green500” power efficiency world ranking of supercomputers.

14.1 Gigaflops/W is more than x10 more efficient than PCs and Smartphones!
Drug Discovery Process w/ AI on TSUBAME3

[M. Sekijima, Tokyo Tech.]

Process of drug discovery

<table>
<thead>
<tr>
<th>Hit discovery</th>
<th>Optimization</th>
<th>Preclinical</th>
<th>Clinical trials</th>
<th>Approval</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5</td>
<td>1.0</td>
<td>6.5</td>
<td>1.5</td>
<td></td>
</tr>
</tbody>
</table>

Total time: 13.5 years¹
Total investment: 2.6 billion USD²

Aim to reduce 30% of cost by Computer-aided drug design

Method: docking-based virtual screening

Identify hit compounds from large (<10M) chemical library by simulating/evaluating protein-compound interaction

Problem: prediction accuracy

Decrease false positive through recognizing interaction pattern by machine learning

\[ \Delta G_{\text{bind}} = C_{\text{lipo-lipo}} \sum f(r_i) + C_{\text{bond-neutral-neutral}} \sum g(\Delta r) h(\Delta \alpha) + C_{\text{bonds-charge-charge}} \sum g(\Delta r) h(\Delta \alpha) + C_{\text{max-metal-atom}} + C_{\text{coth, H-atom}} + C_{\text{polar-phob}} + C_{\text{vDW, E-atom}} + \text{solvent terms} \]

Active compounds

Inactive compounds

**SIEVE-Score: method and evaluation**

- **Extract** interaction energy vector from docking (Glide was used)
  - hydrogen bonding, electrostatic and van der Waals energy between ligand and each residue of target protein
- **Train** random forest with known experiments,
- **Re-rank** by predicted probability of being “active”

**Evaluation: DUD-E dataset**

- 102 proteins, including cancer or hypertension targets
- 40,490 active, 1,414,640 inactive compounds in total
- 1,455,130 docked structures are learned
- 5-fold cross validation for each protein
- Metrics: enrichment factor 1%
  
The ratio of identified active compounds in top 1% compared to random prediction

---

The current status of AI & Big Data in Japan

We need the triage of advanced algorithms/infrastructure/data but we lack the cutting edge infrastructure dedicated to AI & Big Data (c.f. HPC)

Massive Rise in Computing Requirements (1 AI-PF/person?)

Over $1B Gov't AI investment over 10 years

AI & Data Infrastructures

Massive “Big” Data in Training

AI Venture Startups

Big Companies AI/BD R&D (also Science)

R&D ML Algorithms & SW

AI/BD Centers & Labs in National Labs & Universities

Use of Massive Scale Data now Wasted

Seeking Innovative Application of AI & Data

In HPC, Cloud continues to be insufficient for cutting edge research => dedicated SCs dominate & racing to Exascale

“Big” Data

IoT Communication, location & other data

Petabytes of Drive Recording Video

Web access and merchandice

FA&Robots

Use of Massive Scale Data now Wasted

Petabytes of Drive Recording Video

Web access and merchandice

FA&Robots

Massive “Big” Data in Training
ABCI: AI Bridging Cloud Infrastructure
“Architecting Ultra Efficient AI-HPC IDC”

- **Open, Public, and Dedicated** infrastructure for AI & Big Data Algorithms, Software, and Applications
- **Open Innovation Platform** to accelerate joint academic-industry R&D for AI, international collaborations are also welcome
- Top-level compute capability: **0.55 EFLOPS (DL), 37 PFLOPS (DP)**
- Top-level energy efficiency: lower PUE
- All design and implementations will be open-sourced
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

**Interactive Nodes x 4**

**Management and Gateway Nodes x 15**

**Interconnect (Infiniband EDR)**
- Mellanox CS7500 x 2
- Mellanox SB7890 x 229

Large-scale Storage System
- **22 PB GPFS**

**DDN SFA14K (w/ SS8462 Enclosure x 10) x 3**
- 12TB 7.2Krpm NL-SAS HDD x 2400
- 3.84TB SAS SSD x 216
- NSD Server x 12

**Protocol Nodes, etc x 8**

**Gateway and Firewall**
- Nexus 3232C x2
- FortGate 1500D x2
- FortAnalyzer 400E x1

**Service Network (10GbE)**

**100Gbps**

**SINET5**
<table>
<thead>
<tr>
<th>GPU:</th>
<th>CPU:</th>
<th>Compute Node (4GPUs, 2CPUs)</th>
<th>Node Chassis (2 Compute Nodes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.8 TFlops(FP64)</td>
<td>1.53 TFlops(FP64)</td>
<td>34.2 TFlops(FP64)</td>
<td>1.16 PFlops(FP64)</td>
</tr>
<tr>
<td>125 TFlops(FP16)</td>
<td>3.07 TFlops(FP32)</td>
<td>506 TFlops(FP16)</td>
<td>17.2 PFlops(FP16)</td>
</tr>
<tr>
<td>NVIDIA Tesla V100</td>
<td>Xeon Skylake-SP</td>
<td>68.5 PFlops(FP64)</td>
<td>0.55 EFlops(FP16)</td>
</tr>
</tbody>
</table>

- GPU: 7.8 TFlops(FP64), 125 TFlops(FP16)
- CPU: 1.53 TFlops(FP64), 3.07 TFlops(FP32)
- Compute Nodes: 1088
- GPUs: 4352
- 0.550 EFlops(FP16), 37.2 PFlops(FP64)
- 19.88 PFlops(Peak), Ranked #5 Top500 June 2018

- Chip: GPU, CPU
- Tesla V100
- Xeon Skylake-SP

- Node Chassis:
  - PRIMERGY CX2570 M4
  - PRIMERGY CX400 M4

- Rack (17 Chassis)
- System (32 Racks)

- 384 GiB MEM
- 200 Gbps NW BW
- 1.6TB NVMe SSD

- 3.72 TB/s MEM BW
- 131TB/s MEM BW
- Full Bisection BW within Rack
  - 70kW Max
  - 1/3 of Oversubscription BW
  - 2.3MW
**FUJITSU PRIMERGY Server (2 servers in 2U)**

<table>
<thead>
<tr>
<th>Component</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Xeon Gold 6148 (27.5M Cache, 2.40 GHz, 20 Core) x2</td>
</tr>
<tr>
<td>GPU</td>
<td>NVIDIA Tesla V100 (SXM2) x4</td>
</tr>
<tr>
<td>Memory</td>
<td>384GiB DDR4 2666MHz RDIMM</td>
</tr>
<tr>
<td>Local Storage</td>
<td>1.6TB NVMe SSD (Intel SSD DC P4600 u.2) x1</td>
</tr>
<tr>
<td>Interconnect</td>
<td>InfiniBand EDR x2</td>
</tr>
</tbody>
</table>

---

**Diagram**

- **CPU blade**
  - Xeon Gold 6148 (27.5M Cache, 2.40 GHz, 20 Core) x2
  - NVIDIA Tesla V100 (SXM2) x4
  - 384GiB DDR4 2666MHz RDIMM
  - 1.6TB NVMe SSD (Intel SSD DC P4600 u.2) x1
  - InfiniBand EDR x2

- **GPU blade**
  - Xeon Gold 6148 (27.5M Cache, 2.40 GHz, 20 Core) x2
  - NVIDIA Tesla V100 (SXM2) x4
  - 384GiB DDR4 2666MHz RDIMM
  - 1.6TB NVMe SSD (Intel SSD DC P4600 u.2) x1
  - InfiniBand EDR x2

**Interconnect**

- IB HCA (100Gbps)
- PCIe gen3 x16
- NVMe
- Skylake-SP
  - 128GB/s
  - 10.4GT/s x3
  - UPI x3
- Skylake-SP
  - 128GB/s
  - 10.4GT/s x3
  - UPI x3

**Memory**

- DDR4-2666
  - 32GB x 6

**NVLink2**

- x2
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)
- Power consumption per rack: 67.33 kW

Interconnect
- Fat-tree topology
- Intra-rack: full bisection BW
- Inter-rack: 1/3 over-subscription (2400/6800)
- Large-scale storage system: full bi-section BW
ABCI Procurement Benchmarks

- **Big Data Benchmarks**
  - (SPEC CPU Rate)
  - Graph 500
  - MinuteSort
  - Node Local Storage I/O
  - Parallel FS I/O

- **AI/ML Benchmarks**
  - Low precision GEMM
    - CNN Kernel, defines “AI-Flops”
  - Single Node CNN
    - AlexNet and GoogLeNet
    - ILSVRC2012 Dataset
  - Multi-Node Scalable CNN
    - Caffe+MPI
  - Large Memory CNN
    - Convnet on Chainer
  - RNN / LSTM
    - Neural Machine Translation on Torch

**No traditional HPC Simulation Benchmarks except SPEC CPU. Plan on “open-sourcing”**
Container is critical for AI R&D

- lang2program (referred in ACL2017)
  - [https://github.com/kelvinguu/lang2program](https://github.com/kelvinguu/lang2program)
  - Provided as a Dockerfile
  - Bunch of software needed to be run
    - Tensorflow, PostgresQL, Python Pip Packages, etc.
- It is not possible to run it on traditional large-scale HPC systems
  - Arbitrary/Voluntary installation of software => chaos
  - Docker => security reason

- We’re developing an easy-to-manage and flexible-to-use platform for deploying AI apps as “modules”
# AI Frameworks & Modules

- Host OS only provides the minimum set of software including:
  - Base drivers, libraries
- “Base” modules provide customized OS images including:
  - CUDA, cuDNN, NCCL, MPI, etc.
- “DL” modules provide deep learning frameworks and apps, which extend “Base” images.
- We mainly employ Singularity as the basis of our AI platform.

## Base
Base OS Images incl. CUDA, CuDNN, NCCL, MPI, Infiniband

## DL
Images for Deep Learning Frameworks based on HPCBase

<table>
<thead>
<tr>
<th>Framework</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caffe</td>
<td></td>
</tr>
<tr>
<td>Chainer</td>
<td></td>
</tr>
<tr>
<td>TensorFlow</td>
<td></td>
</tr>
<tr>
<td>MXNet</td>
<td></td>
</tr>
<tr>
<td>Torch</td>
<td></td>
</tr>
</tbody>
</table>

## Host OS
incl. Base Drivers, Libraries

<table>
<thead>
<tr>
<th>Driver</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU</td>
<td></td>
</tr>
<tr>
<td>Infiniband</td>
<td></td>
</tr>
<tr>
<td>GPFS</td>
<td></td>
</tr>
</tbody>
</table>
### Software

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operating System</strong></td>
<td>CentOS, RHEL</td>
</tr>
<tr>
<td><strong>Job Scheduler</strong></td>
<td>Univa Grid Engine</td>
</tr>
<tr>
<td><strong>Container Engine</strong></td>
<td>Docker, Singularity</td>
</tr>
<tr>
<td><strong>MPI</strong></td>
<td>OpenMPI, MVAPICH2, Intel MPI</td>
</tr>
<tr>
<td><strong>Deep Learning</strong></td>
<td>Caffe, Caffe2, TensorFlow, Theano, Torch, PyTorch, CNTK, MXnet, Chainer, Keras, etc.</td>
</tr>
<tr>
<td><strong>Big Data Processing</strong></td>
<td>Hadoop, Spark</td>
</tr>
</tbody>
</table>

### Container support

- Containers enable users to instantly try the state-of-the-art software developed in AI community
- ABCI supports two container technologies
  - **Docker**, having a large user community
  - **Singularity**, recently accepted HPC community
- ABCI provides various single-node/distributed deep learning framework container images optimized to achieve high performance on ABCI
## Service Types

<table>
<thead>
<tr>
<th>Service Type</th>
<th>Description</th>
<th>#Nodes (Min./Max.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spot</td>
<td>Batch job service</td>
<td>1 / 512</td>
</tr>
<tr>
<td>On-demand</td>
<td>Interactive job service</td>
<td>1 / 32</td>
</tr>
<tr>
<td>Reserved</td>
<td>Advanced reservation service</td>
<td>1 / 32</td>
</tr>
<tr>
<td>Group Storage</td>
<td>Shared storage service</td>
<td>N/A</td>
</tr>
</tbody>
</table>

※ABCI provides batch and interactive-type job execution service for maximizing throughput, advanced reservation service for dedicated use of nodes, IDE, and storage services.
Resource Types

※ Users can choose the most suitable computing instance from five resource types.

<table>
<thead>
<tr>
<th>Types</th>
<th>#CPU core Assign / Total</th>
<th>#GPU Assign / Total</th>
<th>Memory (GB) Assign / Total</th>
<th>Storage (TB) Assign / Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>F (Full Node)</td>
<td>40 / 40</td>
<td>4 / 4</td>
<td>360 / 384</td>
<td>1.4 / 1.6</td>
</tr>
<tr>
<td>G.large</td>
<td>20 / 40</td>
<td>4 / 4</td>
<td>240 / 384</td>
<td>0.7 / 1.6</td>
</tr>
<tr>
<td>G.small</td>
<td>5 / 40</td>
<td>1 / 4</td>
<td>60 / 384</td>
<td>0.175 / 1.6</td>
</tr>
<tr>
<td>C.large</td>
<td>20 / 40</td>
<td>0 / 4</td>
<td>120 / 384</td>
<td>0.7 / 1.6</td>
</tr>
<tr>
<td>C.small</td>
<td>5 / 40</td>
<td>0 / 4</td>
<td>30 / 384</td>
<td>0.175 / 1.6</td>
</tr>
</tbody>
</table>
ABCI Use Case: Distributed Deep Learning

- 64 Nodes (256 GPUs) in 1 hour
- DL Framework: ChainerMN v1.3.0
- Data Set: ImageNet-1K
- Model: ResNet-50
  - Batch size: 32 x 256
  - Learning Rate: starting 0.1 and x0.1 at 30, 60, 90 epoch w/ warm up scheduling
  - Momentum SGD (momentum=0.9)
  - Weight decay: 0.0001
  - Training Epoch: 100

Better

Training is finished within only one hour!
AI Datacenter
“Commoditizing supercomputer cooling technologies to Cloud (70kW/rack)”

- Single floor, cost effective building
- Hard concrete floor 2t/m² weight tolerance for racks and cooling pods
- Number of Racks
  - Initial: 90 (ABCI uses 41 racks)
  - Max: 144
- Power capacity: 3.25 MW
  - ABCI uses 2.3MW max
- Cooling capacity: 3.2MW
  - 70kW/rack: 60kW water + 10kW air
100% Free Cooling over the Entire Year

- Free cooling using a passive cooling tower
  - ○ • Low OPEX. No active chiller (mechanical refrigeration)
  - • • Generated water temperature is depended on the external weather (temp. and humidity)
- Hybrid water/air cooling with high-temperature cooling water (32℃)

https://www.sinko.co.jp/product/technical-column/09/
Hybrid Water/Air Cooling System

Fan Coil Unit (FCU)

Outdoor Facilities

Cooling Tower

CDU

Cooling Pod / Rack

Computing Node

Water 32°C

Water 40°C

Air Cooling

Air 35°C

Air 40°C

1st closed circuit

2nd closed circuit

Water 32°C

Water 40°C

Water Cooling

Water Block
Survived the extreme heat this summer!

Free cooling is possible during the first grand challenge on 22-26 July 2018.

Cooling Water (From ABCI): <40°C
Cooling Water (To ABCI): <32°C

Max./min. temperature in Kashiwa in July 2018
Cooling Pod

- Modularization of DC facilities: rack space, water/air cooling and power equipment
- No raised floor and skeleton frame built on concrete slab
- Effective air cooling by hot aisle containment
- Ease of maintenance
Inside Cooling Pod

Catwalk (for maintenance)

Hot aisle
<table>
<thead>
<tr>
<th>Institution</th>
<th>AAIC (AIST AI Cloud)</th>
<th>TSUBAME3.0</th>
<th>ABCI (AI Bridging Cloud)</th>
<th>Summit</th>
<th>TPU 3.0 Pod</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Nodes</td>
<td>50</td>
<td>540</td>
<td>1088</td>
<td>4608</td>
<td>NA</td>
</tr>
<tr>
<td>Throughput AI Processor</td>
<td>NVIDIA Tesla P100</td>
<td>NVIDIA Tesla P100</td>
<td>NVIDIA Tesla V100</td>
<td>NVIDIA Tesla V100</td>
<td>TPU 3.0</td>
</tr>
<tr>
<td>#GPUs</td>
<td>400</td>
<td>2160</td>
<td>4352</td>
<td>27648</td>
<td>NA</td>
</tr>
<tr>
<td>Peak FP64</td>
<td>2.2 PF</td>
<td>12.2 PF</td>
<td>37.2 PF</td>
<td>200 PF</td>
<td>NA</td>
</tr>
<tr>
<td>Peak DL</td>
<td>8.6 PF</td>
<td>47.2 PF</td>
<td>550 PF</td>
<td>3.3 EF</td>
<td>100 PF / Pod</td>
</tr>
<tr>
<td>TOP500*</td>
<td>287</td>
<td>19</td>
<td>5 (#1 Japan)</td>
<td>1</td>
<td>NA</td>
</tr>
<tr>
<td>GREEN500*</td>
<td>7</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>NA</td>
</tr>
<tr>
<td>Nodes / Rack</td>
<td>6</td>
<td>36</td>
<td>34</td>
<td>16</td>
<td>NA</td>
</tr>
<tr>
<td>GPUs / Rack</td>
<td>48</td>
<td>144</td>
<td>136</td>
<td>96</td>
<td>NA</td>
</tr>
<tr>
<td>kW / Rack</td>
<td>22 kW</td>
<td>64.8 kW</td>
<td>67.33 kW</td>
<td>45-55 kW (est.)</td>
<td>NA</td>
</tr>
<tr>
<td>AI-FLOPS / rack</td>
<td>0.9 PF</td>
<td>3.1 PF</td>
<td>17 PF</td>
<td>12 PF</td>
<td>12.5 PF (100 PF / 8 racks)</td>
</tr>
</tbody>
</table>

(As of June 2018)
## TOP500 List (June 2018)

<table>
<thead>
<tr>
<th>#</th>
<th>System</th>
<th>Architecture</th>
<th>Country</th>
<th>Rmax (TFlop/s)</th>
<th>Rpeak (TFlop/s)</th>
<th>Power (kW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Summit, ORNL</td>
<td>IBM, POWER9 + GPU(GV100)</td>
<td>USA</td>
<td>122,300.0</td>
<td>187,659.3</td>
<td>8,806</td>
</tr>
<tr>
<td>2</td>
<td>TaihuLight, NSCW</td>
<td>Sunway, SW26010</td>
<td>China</td>
<td>93,014.6</td>
<td>125,435.9</td>
<td>15,371</td>
</tr>
<tr>
<td>3</td>
<td>Sierra, LLNL</td>
<td>IBM POWER9 + GPU(GV100)</td>
<td>USA</td>
<td>71,610.0</td>
<td>119,193.6</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>Tianhe-2, NSCG</td>
<td>NUDT, CPU + Matrix-2000</td>
<td>China</td>
<td>61,444.5</td>
<td>100,678.7</td>
<td>18,482</td>
</tr>
<tr>
<td>5</td>
<td>ABCI, AIST</td>
<td>Fujitsu, CPU + GPU(V100)</td>
<td>Japan</td>
<td>19,880.0</td>
<td>32,576.6</td>
<td>1,649</td>
</tr>
<tr>
<td>6</td>
<td>Piz Daint, CSCS</td>
<td>Cray, CPU + GPU(P100)</td>
<td>Switzerland</td>
<td>19,590.0</td>
<td>25,326.3</td>
<td>2,272</td>
</tr>
<tr>
<td>7</td>
<td>Titan, ORNL</td>
<td>Cray, CPU + GPU(K20x)</td>
<td>USA</td>
<td>17,590.0</td>
<td>27,112.5</td>
<td>8,209</td>
</tr>
<tr>
<td>8</td>
<td>Sequoia, LLNL</td>
<td>IBM, BlueGene/Q</td>
<td>USA</td>
<td>17,173.2</td>
<td>20,132.7</td>
<td>7,890</td>
</tr>
<tr>
<td>9</td>
<td>Trinity, LANL/SNL</td>
<td>Cray, MIC(KNL)</td>
<td>USA</td>
<td>14,137.3</td>
<td>43,902.6</td>
<td>3,844</td>
</tr>
<tr>
<td>10</td>
<td>Cori, NERSC-LBNL</td>
<td>Cray, MIC(KNL)</td>
<td>USA</td>
<td>14,014.7</td>
<td>27,880.7</td>
<td>3,939</td>
</tr>
</tbody>
</table>
Comparing ABCI to Classical IDC
AI IDC CAPX/OPEX acceleration by > x100

Traditional Xeon IDC
~10KW/rack PUE 1.5~2
15~20 1U Xeon Servers
2 Tera AI-FLOPS(SFP) / server
30~40 Tera AI-FLOP / rack
Low cooling efficiency

ABCI “Open Source” IDC
~70KW/rack PUE 1.0x
~500 Tera AI-FLOPS(HFP) / server
~17 Peta AI-FLOPs / rack
Inexpensive, very high cooling efficiency (PUE~1.1)

Perf > x400~600
Power Eff > x200~300
TSUBAME Grand Challenge

Spring 2016
GTC-P (Bei Wang, Stephane Ethier, Bill Tang)
TSUBAME 2.5

Fall 2017
FRNN (Alexey Svyatkovskiy, Julian Kates-Harbeck, Bill Tang)
TSUBAME 3.0

Spring 2018
ChainerMN (Ryo Yokota, Kazuki Osawa, Yohei Tsuji, et al.)
TSUBAME 3.0
Fusion Recurrent Neural Network (FRNN)
Alexey Svyatkovskiy, Julian Kates-Harbeck, Bill Tang

Objective
Use machine learning to predict disruptions in ITER class fusion reactors

Methods
3 layer LSTM in Tensorflow
300 cells per layer

Contributions
Prediction 30ms before disruption is 96% which is much better than SVM and RF
ABCI-PRP: Grand Challenge Project

- Gerald Pao, Salk Institute; George Sugihara, SIO
- Creation of neuromorphic deep learning architectures by large-scale dynamic modeling of transparent fish brains on ABCI

ABCI is ideally suited to perform the convergent cross mapping (CCM) to interrogate the relationships between the ~120,000 neurons of the larval zebrafish brain during exposure to various stimuli.

Check Wassapon’s Poster (PP25)!

From Pao, 2018. Green are neurons, red are active neurons responding to hypoxic environment.
Training ImageNet in Minutes

Rio Yokota, Kazuki Osawa, Yuhei Tsuji, Yuichiro Ueno, Hiroki Naganuma, Shun Iwase, Kaku Linsho
Tokyo Institute of Technology

Facebook
Preferred Networks
UC Berkeley
Tencent
Tokyo Tech + AIST

<table>
<thead>
<tr>
<th>GPU/KNL</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>512</td>
<td>30 min</td>
</tr>
<tr>
<td>1024</td>
<td>15 min</td>
</tr>
<tr>
<td>2048</td>
<td>14 min</td>
</tr>
<tr>
<td>2048</td>
<td>6.6 min</td>
</tr>
<tr>
<td>4096</td>
<td>? min</td>
</tr>
</tbody>
</table>

Source: Ben-nun & Hoefler [1802.09941.pdf]
Second Order Methods

\[
\begin{align*}
\theta_{t+1} = \theta_t - \eta F^{-1} \nabla J \\
F^{-1} &\approx E(A^{-1} \otimes G^{-1}) \\
&\approx E(A)^{-1} \otimes E(G)^{-1}
\end{align*}
\]

Kronecker Factorization (K-FAC)
Parallel Scalability

Kronecker Factorization

Cross-over from model-parallel to data-parallel

Cross-over from data-parallel to model-parallel

Imagenet1k, ResNet50
Local batch size: 32
Apr 1 2018 Became Director of Riken-CCS: Science, of Computing, by Computing, and for Computing

Riken Center for Computational Science (R-CCS)
World Leading HPC Research, active collaborations w/Universities, national labs, & Industry

Sci. of Computing
Foundational research on computing in high performance for K, Post-K, and beyond towards the “Post-Moore” era, including future high performance architectures, new computing and programming models, system software, large scale systems modeling, big data analytics, and scalable artificial intelligence / machine learning

Sci. by Computing
Breakthrough Science & Technology using high performance computing capabilities of K, Post-K and beyond to address the issues of high public concern, in areas such as life sciences, climate & environment, disaster prediction & prevention, advanced manufacturing, applications of machine learning for Society 5.0.

Sci. for Computing
High Resolution, High Fidelity Analysis & Simulation

Mutual Synergy
New Materials & Electronic Devices e.g., Photonics, Neuromorphics, Quantum, Reconfigurable

Novel Future High Performance Computing Architectures & Algorithms
Japan Flagship 2020 “Post K” Supercomputer

CPU
- Many core, Xeon-Class ARM v8 cores + 512 bit SVE (scalable vector extensions)
- Multi-hundred petaflops peak total
- Power Knob feature

Memory
- 3-D stacked DRAM, Terabyte/s BW /chip

Interconnect
- TOFU-D CPU-integrated 6-D torus network
- I/O acceleration with massive SDs
- 30MW+ Power, liquid cooled
- Riken co-design with Fujitsu
- ? Million cores in system

Prime Minister Abe visiting K Computer 2013
1. Heritage of the K-Computer, HP in simulation via extensive Co-Design
   - High performance: up to x100 performance of K in real applications
   - Multitudes of Scientific Breakthroughs via Post-K application programs
   - Simultaneous high performance and ease-of-programming

2. New Technology Innovations of Post-K
   - High Performance, esp. via high memory BW
     Performance boost by “factors” c.f. mainstream CPUs in many HPC & Society5.0 apps
   - Very Green e.g. extreme power efficiency
     Ultra Power efficient design & various power control knobs
   - Arm Global Ecosystem & SVE contribution
     Top CPU in ARM Ecosystem of 21 billion chips/year, SVE co-design and world’s first implementation by Fujitsu
   - High Perf. on Society5.0 apps incl. AI
     Architectural features for high perf on Society 5.0 apps based on Big Data, AI/ML, CAE/EDA, Blockchain security, etc.

Global leadership not just in the machine & apps, but as cutting edge IT
Co-design for Post-K
(slides by Mitsuhisa Sato Team Leader of Architecture Development Team)
Deputy project leader, FLAGSHIP 2020 project
Deputy Director, RIKEN Center for Computational Science (R-CCS)

Analysis of applications to devise the most efficient solutions

Co-design from Apps to Architecture

- **Architectural Parameters to be determined**
  - #SIMD, SIMD length, #core, #NUMA node, O3 resources, specialized hardware
  - cache (size and bandwidth), memory technologies
  - Chip die-size, power consumption
  - Interconnect

- **We have selected a set of target applications**

- **Performance estimation tool**
  - Performance projection using Fujitsu FX100 execution profile to a set of arch. parameters.

- **Co-design Methodology (at early design phase)**
  1. Setting set of system parameters
  2. Tuning target applications under the system parameters
  3. Evaluating execution time using prediction tools
  4. Identifying hardware bottlenecks and changing the set of system parameters

Target applications representatives of almost all our applications in terms of computational methods and communication patterns in order to design architectural features.

<table>
<thead>
<tr>
<th>Target Application</th>
<th>Program</th>
<th>Brief description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. GENESIS</td>
<td>MD for proteins</td>
<td></td>
</tr>
<tr>
<td>2. Genomon</td>
<td>Genome processing (Genome alignment)</td>
<td></td>
</tr>
<tr>
<td>3. GAMERA</td>
<td>Earthquake simulator (FEM in unstructured &amp; structured grid)</td>
<td></td>
</tr>
<tr>
<td>4. NICAM+LETK</td>
<td>Weather prediction system using Big data (structured grid stencil &amp; ensemble Kalman filter)</td>
<td></td>
</tr>
<tr>
<td>5. NTChem</td>
<td>molecular electronic (structure calculation)</td>
<td></td>
</tr>
<tr>
<td>6. FFB</td>
<td>Large Eddy Simulation (unstructured grid)</td>
<td></td>
</tr>
<tr>
<td>7. RSDFT</td>
<td>an ab-initio program (density functional theory)</td>
<td></td>
</tr>
<tr>
<td>8. Adventure</td>
<td>Computational Mechanics System for Large Scale Analysis and Design (unstructured grid)</td>
<td></td>
</tr>
<tr>
<td>9. CCS-QCD</td>
<td>Lattice QCD simulation (structured grid Monte Carlo)</td>
<td></td>
</tr>
</tbody>
</table>
Genesis MD: proteins in a cell environment

Protein simulation before K

- Simulation of a protein in isolation
  Folding simulation of Villin, a small protein with 36 amino acids

Protein simulation with K

- All atom simulation of a cell interior
- Cytoplasm of Mycoplasma genitalium
NICAM: Global Climate Simulation

- Global cloud resolving model with 0.87 km-mesh which allows resolution of cumulus clouds
- Month-long forecasts of Madden-Julian oscillations in the tropics is realized.

Co-design of Apps for Architecture

- **Tools for performance tuning**
  - Performance estimation tool
    - Performance projection using Fujitsu FX100 execution profile
    - Gives “target” performance
  - **Post-K processor simulator**
    - Based on gem5, O3, cycle-level simulation
    - Very slow, so limited to kernel-level evaluation

- **Co-design of apps**
  1. Estimate “target” performance using performance estimation tool
  2. Extract kernel code for simulator
  3. Measure exec time using simulator
  4. Feed-back to code optimization
  5. Feed-back to compiler
ARM HPC ECOSYSTEM
(slides courtesy Prof. Simon McIntosh-Smith, U-Bristol)

- “Isambard” Cavium TX2 HPC Cluster
- Various Portings and Benchmarking
  - Practically all x86 codes work “out-of-the-box”
  - Compiler dependency more crucial c.f. ISA
  - Performance competitive due to most applications being memory BW dependent, and Cavium BW 33% superior

"Isambard", a new Tier 2 HPC service from GW4. Named in honour of Isambard Kingdom Brunel

Isambard system specification (red = new info):
- Cray “Scout” system – XC50 series
- Aries interconnect
- 10,000+ Armv8 cores
- Cavium ThunderX2 processors
- 2x 32core @ >2GHz per node
- Cray software tools
- Technology comparison:
  - x86, Xeon Phi, Pascal GPUs
  - Phase 1 installed March 2017
  - The Arm part arrives early 2018

The Isambard project’s focus will be on the top 10 most heavily used codes on Archer in 2017:
- VASP, CASTEP, GROMACS, CP2K, UM, HYDRA, NAMD, Oasis, SBLI, NEMO
- Note: 8 of these 10 codes are written in FORTRAN
- Additional important codes for project partners:
  - OpenFOAM, OpenIFS, WRF, CASINO, LAMMPS, ...
- We want to collaborate wherever possible!
- Accelerate the adoption of Arm in HPC
Post K Processor is…

- an Many-Core ARM CPU…
  - 48 compute cores + 2 or 4 assistant (OS) cores
  - Brand new core design
  - Near Xeon-Class Integer performance core
  - ARM V8 --- 64bit ARM ecosystem
  - Tofu 3 + PCIe 3 external connection

- …but also a GPU-like processor
  - SVE 512 bit vector extensions (ARM & Fujitsu)
  - Integer (1, 2, 4, 8 bytes) + Float (16, 32, 64 bytes)
  - Cache + scratchpad local memory (sector cache)
  - Multi-stack 3D memory – Massive Mem BW (Bytes/DPF ~0.4)
    - Streaming memory access, strided access, scatter/gather etc.
    - Intra-chip barrier synch. and other memory enhancing features

GPU-like High performance in HPC, AI/Big Data, Auto Driving…
A64FX Chip Overview

**Architecture Features**

- Armv8.2-A (AArch64 only)
- SVE 512-bit wide SIMD
- 48 computing cores + 4 assistant cores*
  *All the cores are identical
- HBM2 32GiB
- Tofu 6D Mesh/Torus 28Gbps x 2 lanes x 10 ports
- PCIe Gen3 16 lanes

**7nm FinFET**

- 8,786M transistors
- 594 package signal pins

**Peak Performance (Efficiency)**

- >2.7TFLOPS (>90%@DGEMM)
- Memory B/W 1024GB/s (>80%@Stream Triad)
A64FX Core Pipeline

A64FX enhances and inherits superior features of SPARC64

- Inherits superscalar, out-of-order, branch prediction, etc.
- Enhances SIMD and predicate operations
  - 2x 512-bit wide SIMD FMA + Predicate Operation + 4x ALU (shared w/ 2x AGEN)
  - 2x 512-bit wide SIMD load or 512-bit wide SIMD store
Many-Core Architecture

- A64FX consists of four CMGs (Core Memory Group)
  - A CMG consists of 13 cores, an L2 cache and a memory controller
    - One out of 13 cores is an assistant core which handles daemon, I/O, etc.
  - Four CMGs keep cache coherency by ccNUMA with on-chip directory
  - X-bar connection in a CMG maximizes high efficiency for throughput of the L2 cache
  - Process binding in a CMG allows linear scalability up to 48 cores

- On-chip-network with a wide ring bus secures I/O performance
High Bandwidth

- Extremely high bandwidth in caches and memory
  - A64FX has out-of-order mechanisms in cores, caches and memory controllers. It maximizes the capability of each layer’s bandwidth.
### Post-K A64fx A0 (ES) performance

<table>
<thead>
<tr>
<th></th>
<th>Performance / CPU</th>
<th>Machine Performance (HPC)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Peak TF (DFP)</td>
<td>Peak Mem. BW</td>
</tr>
<tr>
<td>Post-K A64fx (A0 Eng. Sample)</td>
<td>2.764/3.072</td>
<td>1024GB/s</td>
</tr>
<tr>
<td>Intel KNL</td>
<td>3.0464</td>
<td>600GB/s</td>
</tr>
<tr>
<td>Intel Skylake</td>
<td>1.6128</td>
<td>127.8GB/s</td>
</tr>
<tr>
<td>NVIDIA V100 (DGX-2)</td>
<td>7.8</td>
<td>900 GB/s</td>
</tr>
</tbody>
</table>
A64FX boosts performance up by microarchitectural enhancements, 512-bit wide SIMD, HBM2 and process technology

- > 2.5x faster in HPC/AI benchmarks than SPARC64 XIfx (Fujitsu’s previous HPC CPU)
- The results are based on the Fujitsu compiler optimized for our microarchitecture and SVE

Baseline FX100 (SPARC 64 fxIIX) But how fast is that c.f. Xeon?
NAS Parallel Benchmark of FX100

[Slide by Ikuo Miyoshi, Fujitsu, SSKen2015]

OpenMP版を用いてノードあたり演算性能を評価

FX10に対するノードあたり性能向上比

Note: Haswell:
1 node = 2 chips
32 threads&cores
FX100: 1 node = 1 chip
32 threads&cores

使用コード: NAS Parallel Benchmarks Ver. 3.3.1 OpenMP版クラスC
# Fiber (Post-K) MiniApp on FX100

**[Slide by Ikuo Miyoshi, Fujitsu, SSKen2015]**

## 評価条件

<table>
<thead>
<tr>
<th>アプリ名</th>
<th>問題サイズ</th>
<th>評価区間</th>
<th>スレッド数 × プロセス数 (左からFX10、FX100、Haswell)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCS QCD</td>
<td>32 × 32 × 32 × 32</td>
<td>BiCGStab</td>
<td>16t × 2p, 32t × 1p, 16t × 2p</td>
</tr>
<tr>
<td>NICAM-DC-MINI</td>
<td>gl05rl00z80pe10</td>
<td>Dynamics</td>
<td>3t × 10p</td>
</tr>
<tr>
<td>FFB-MINI</td>
<td>1,048,576要素</td>
<td>MAIN_LOOP</td>
<td>1t × 32p, 8t × 4p, 1t × 32p</td>
</tr>
<tr>
<td>FFVC-MINI</td>
<td>256 × 256 × 256</td>
<td>Total</td>
<td>4t × 8p, 16t × 2p</td>
</tr>
<tr>
<td>NTChem-MINI</td>
<td>taxol</td>
<td>RIMP2_Driver</td>
<td>1t × 32p, 16t × 2p, 2t × 16p</td>
</tr>
</tbody>
</table>

## 測定結果

FX100は、FX10比平均3.3倍、Haswell比平均1.4倍のノードあたり性能

Note: Haswell: 1 node = 2 chips, 32 threads&cores
FX100: 1 node = 1 chip, 32 threads&cores

注) FFVCには開発版コンパイラ、NTChemには開発版数学ライブラリを使用。QCDではセクタキャッシュ利用、FFBではループリローリングのコード変更を実施
Power Management (Cont.)

- "Power knob" for power optimization
- A64FX provides power management function called "Power Knob"
  - Applications can change hardware configurations for power optimization
  - Power knobs and Energy monitor/analyzer will help users to optimize power consumption of their applications

<A64FX Power Knob Diagram>

Decode width: 2

EX pipeline usage: EXA only

Frequency reduction

FL pipeline usage: FLA only

HBM2 B/W adjustment (units of 10%)
Fujitsu Mission Critical Technologies

- Large systems require extensive RAS capability of CPU and interconnect
- A64FX has a mainframe class RAS for integrity and stability. It contributes to very low CPU failure rate and high system stability
  - ECC or duplication for all caches
  - Parity check for execution units
  - Hardware instruction retry
  - Hardware lane recovery for Tofu links
  - ~128,400 error checkers in total

<table>
<thead>
<tr>
<th>Units</th>
<th>Error Detection and Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache (Tag)</td>
<td>ECC, Duplicate &amp; Parity</td>
</tr>
<tr>
<td>Cache (Data)</td>
<td>ECC, Parity</td>
</tr>
<tr>
<td>Register</td>
<td>ECC (INT), Parity(Others)</td>
</tr>
<tr>
<td>Execution Unit</td>
<td>Parity, Residue</td>
</tr>
<tr>
<td>Core</td>
<td>Hardware Instruction Retry</td>
</tr>
<tr>
<td>Tofu</td>
<td>Hardware Lane Recovery</td>
</tr>
</tbody>
</table>

Green: 1 bit error Correctable
Yellow: 1 bit error Detectable
Gray : 1 bit error harmless
Post-K Chassis, PCB (w/DLC), and A64fx CPU Package

A0 Chip Booted in June
Undergoing Tests
B0 underway

CPU Package

CMU

W 800mm
D1400mm
H2000mm
384 nodes

Copyright 2018 FUJITSU LIMITED
Higher-density Node Configuration

- 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
Injection Rates per Node

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

<table>
<thead>
<tr>
<th></th>
<th>Injection rate</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tofu1 (K)</td>
<td>15.0 GB/s</td>
<td>77 %</td>
</tr>
<tr>
<td>Tofu1 (FX10)</td>
<td>17.6 GB/s</td>
<td>88 %</td>
</tr>
<tr>
<td>Tofu2</td>
<td>45.8 GB/s</td>
<td>92 %</td>
</tr>
<tr>
<td>TofuD</td>
<td>38.1 GB/s</td>
<td>93 %</td>
</tr>
</tbody>
</table>

- 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
An Overview of Post-K Hardware

- **Compute Node, Compute + I/O Node** connected by 6D mesh/torus Interconnect
- **3-level hierarchical storage system**
  - **1st Layer**
    - Cache for global file system
    - Temporary file systems
      - Local file system for compute node
      - Shared file system for a job
  - **2nd Layer**
    - Lustre-based global file system
  - **3rd Layer**
    - Storage for archive
  - **>100,000 nodes**
  - **x100PF, >100PB/s mem BW**
  - **Approaching 10 million CPU cores**
Massive Scale Deep Learning on Post-K

Post-K Processor
◆ High perf FP16&Int8
◆ High mem BW for convolution
◆ Built-in scalable Tofu network

High Performance DNN Convolution

Unprecedented DL scalability
High Performance and Ultra-Scalable Network for massive scaling model & data parallelism

TOFU Network w/high injection BW for fast reduction

Unprecedented Scalability of Data/
Low Precision ALU + High Memory Bandwidth + Advanced Combining of Convolution Algorithms (FFT+Winograd+GEMM)
Selecting the Optimal Convolution Kernel

- **NEW! Micro Batching:** Tokyo Tech. and ETH [Oyama, Tan, Hoefler & Matsuoka, IEEE Cluster 2019]
  - Use the “micro-batch” technique to select the best convolution kernel
    - Direct, GEMM, FFT, Winograd
    - Optimize both speed and memory size
  - On high-end GPUs, in many cases Winograd or FFT chosen over GEMM
    - They are faster but use more memory
  - Currently implemented as cuDNN wrapper, applicable to all frameworks
  - For Post-K, (1) Winograd/FFT are selected more often, and (2) performance will be similar to GPUs in such cases

Evaluation: WR using Dynamic Programming

- μ-cuDNN achieved 2.33x speedup on forward convolution of AlexNet conv2

Evaluation: WD using Integer LP

- SELECTED Parameters:
  - Workspace size of 64 MiB, mini-batch size of 256
  - Numbers on each rectangles represent micro-batch sizes

- For Post-K, (1) Winograd/FFT are selected more often, and (2) performance will be similar to GPUs in such cases
Large Scale simulation and AI coming together
[Ichimura et. al. Univ. of Tokyo, IEEE/ACM SC17 Best Poster]

130 billion freedom earthquake of entire Tokyo on K-Computer (ACM Gordon Bell Prize Finalist, SC16,17 Best Poster)

AI Trained by Simulation to generate candidate soft soil structure

Soft Soil $<$100m

Earthquake

Too Many Instances

Candidate Underground Structure 1

Candidate Underground Structure 2
Cutting Edge Research AI Infrastructures in Japan
Accelerating BD/AI with HPC (w/accompanying BYTES)
(and my effort to design & build them)

**In Production**

- **Aug. 2017**
  - TSUBAME3.0 (Tokyo Tech./HPE)
  - 47.2 AI-PF (65.8 AI-PF w/Tsubame2.5)

- **Mar. 2017**
  - AIST AI Cloud
    - (AIST-AIRC/NEC)
    - 8.2 AI-PF

- **Aug 2018**
  - ABCI (AIST-AIRC)
    - 550 AI-Petaflops
    - AI-ExaFlop era
    - 2020 Post-K Multi AI-Exaflops
    - order of magnitude over ABCI

**In Preparation**

- Aug 2018
  - ABCI (AIST-AIRC)
    - 550 AI-Petaflops
    - X4~6?

R&D Investments into world leading AI/BD HW & SW & Algorithms and their co-design for cutting edge Infrastructure absolutely necessary (just as is with Japan Post-K and US ECP in HPC)

Oct. 2015

**TSUBAME-KFC/DL**

(Tokyo Tech./NEC)

1.4 AI-PF(Petaflops)