DL4Fugaku: AI frameworks on Fugaku
BOS: Challenges and opportunities with running AI workloads on HPC systems
11th JLESC Workshop, September 9th, 2020

Kento Sato
High Performance Big Data Research Team
RIKEN Center for Computational Science (RIKEN R-CCS)

※ Some of software introduced in the slides is under development. Experimental results will be changed in future in the course of tuning
Supercomputer Fugaku & Deep learning

- Large-scale deep learning is emerging as an essential machine learning approach for many research challenges such as image recognition, segmentation and natural language processing.
- Fast and scalable large-scale deep learning enables us to train neural networks with more training data in shorter time.
- Fugaku/A64FX is expected to achieve high performance DNN training/inference.
- GPU has become a popular platform for executing DL, but we revisit the idea of running DL on CPUs in large-scale environments.

A64FX: Summary

- Arm SVE, high performance and high efficiency
  - DP performance: 2.7+ TFLOPS, >90%@DGEMM
  - Memory BW: 1024 GB/s, >80%@STREAM Triad

→ High performance FP16/INT8
→ High bandwidth memory (1024 GB/sec)
→ Scalable TofuD interconnect

To make use of Fugaku/A64FX performance, tuning AI software stack is indispensable.

Source: Toshiyuki Shimizu, Post-K Supercomputer with Fujitsu’s Original CPU, A64FX Powered by Arm ISA
Objective: Fast and scalable deep learning on Fugaku/A64FX
- Conduct porting, performance analysis and tuning
- Deploy large-scale deep learning environment
- Enhance the usability for production use in Fugaku

MOU for RIKEN/Fujitsu collaboration on AI framework development in Fugaku

RIKEN R-CCS internal teams are working together
- Under collaboration with Industry & academia
- Porting, tracing DL, performance analysis, tuning, merge to upstream

RIKEN R-CCS

<table>
<thead>
<tr>
<th>Operation team</th>
<th>Application tuning development unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research teams</td>
<td>High performance AI system research team</td>
</tr>
<tr>
<td></td>
<td>High performance big data research team</td>
</tr>
<tr>
<td></td>
<td>Large-scale parallel numerical computing technology research team</td>
</tr>
</tbody>
</table>

Collaborators

- AIST
- ARM
- Cybozu
- Fujitsu Laboratories
- Fujitsu limited
- Linaro
- Tokyo Tech

(alphabetical order)

Nov. 25th, 2019

Right: Naoki Shinjo, Head of Unit. Platform Development Unit, Fujitsu Limited
Left: Satoshi Matsuoka, R-CCS Director
Survey on DL framework usage in Japan

- **Period**
  - Oct., 2019 to Nov., 2019

- **Target organizations and users**
  - RIKEN R-CCS
  - RIKEN AIP
  - Users from HPCI Strategic Program
  - Users of ABCI at AIST
  → Potential Fugaku users who use DL frameworks answered this questionnaire

Popular DL frameworks are either TensorFlow, PyTorch or Chainer
→ We plan to support these three frameworks on Fugaku

※ “Other” users develop and use their own DL frameworks
Porting and Tuning approach

● Deep learning software stack
  ● Deep learning frameworks are relying on low-level numerical libraries optimized for specific hardware
    ● cuDNN for NVIDIA GPU, OneDNN for Intel CPU, ??? for A64FX

● Approach
  ● We decided to tune OneDNN for Fugaku’s A64FX CPUs (OneDNN_aarch64) instead of full scratch development

● Current status
  ● Most of porting and tuning are finished
  ● The source codes are in a github repository
    ● https://github.com/fujitsu/dnnl_aarch64
  ● We also contribute to upstream of OneDNN repo

Frameworks
DL frameworks
(TensorFlow, PyTorch, Chainer etc.)

Low-level libraries
 cuDNN  OneDNN  OneDNN_aarch64

Hardware
 NVIDIA GPU  Intel CPU  A64FX

Intel Math Kernel Library for Deep Neural Networks (Intel MKL-DNN)
⇒ Deep Neural Network Library (DNNL)
⇒ oneAPI Deep Neural Network Library (oneDNN)

Slide courtesy of Jin Takahashi, Fujitsu laboratory ltd. with translation and modifications
Perfomrance Evaluation: ResNet-50 on A64FX (A single node)

- **Environment**
  - HW: A64FX (2.2GHz, 48 cores, HBM2 32GB)
  - SW: Fujitsu compiler (fcc), Fujitsu numerical libraries (SSL-II)

<table>
<thead>
<tr>
<th>PyTorch v1.5.0</th>
<th>TensorFlow v2.1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Training (FP32)</strong></td>
<td><strong>Training (FP32)</strong></td>
</tr>
<tr>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
</tr>
<tr>
<td>OneDNN_aarch64 tcmalloc</td>
<td>OneDNN_aarch64 tcmalloc</td>
</tr>
<tr>
<td>✔️ ✔️ ✔️</td>
<td>✔️ ✔️ ✔️</td>
</tr>
<tr>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
</tr>
<tr>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
</tr>
<tr>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
</tr>
<tr>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
</tr>
<tr>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
</tr>
<tr>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
</tr>
<tr>
<td><img src="image" alt="Graph" /></td>
<td><img src="image" alt="Graph" /></td>
</tr>
</tbody>
</table>

*3: Batch Size = 512 x 4
*4: Batch Size = 61 x 4
*5: Batch Size = 128 x 4


Slide courtesy of Jin Takahashi, Fujitsu laboratory ltd. with translation and modifications
Performance Evaluation: ResNet-50 on A64FX (Multi-node)

- **Environment**
  - **HW:** A64FX (2.2GHz, 48 cores, HBM 32GB), TofuD interconnect
  - **SW:** Fujitsu compiler (fcc), Fujitsu numerical libraries (SSL-II), Horovod

Now, we are benchmarking in larger scale in other NNs

More results will be open soon

Slide courtesy of Jin Takahashi, Fujitsu laboratory ltd. with translation and modifications