Hardware Optimization in Distributed Deep Learning

Hiroshi James KAWAGUCHI
Kobe University
2018/2/26

KISS Lab members

Kobe University
Integrated
Silicon and
Software architecture lab
(KISS)

Masahiko YOSHIMOTO

Shintaro IZUMI

Yuna TAMURA

Hiroshi KAWAGUCHI

Research fields

Research fields: hardware architecture and circuits

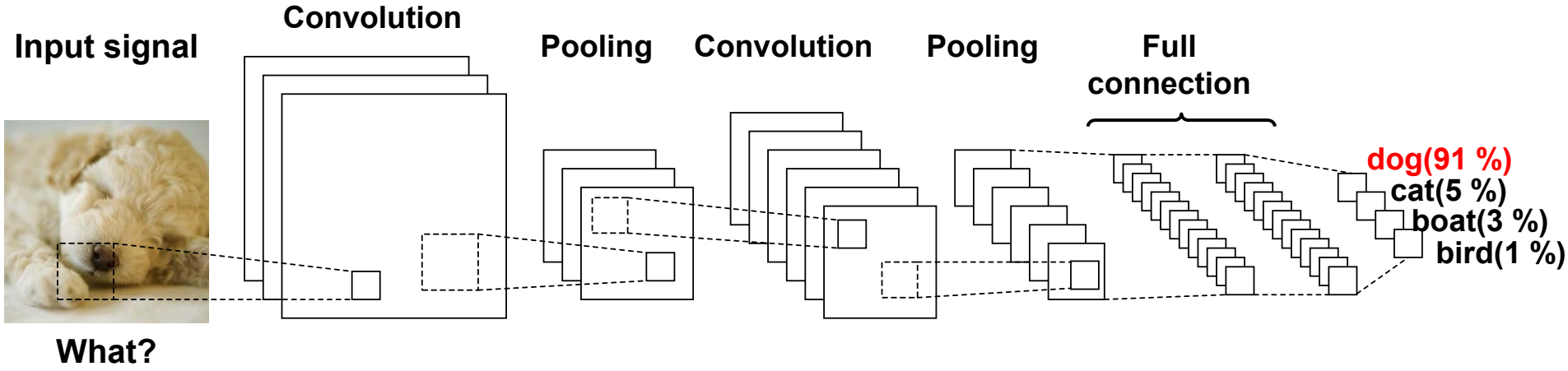
- **Signal processor architecture**
  - **→ Deep learning hardware**
- **Memory circuits (SRAM, MRAM, FeRAM)**
  - **→ Low-power image memory**

Publications:

- **70+ journal papers, 150+ conference papers**
- **Google Scholar Citation**
  (citations: 6915, h-index: 32)

Why deep neural networks?

- **Deep neural network (DNN)**
  - DNN has many layers of convolutional neural network (CNN), which imitates part of the human visual cortex in the cerebrum.



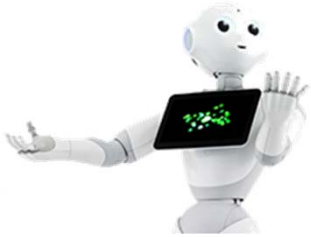
- **Application areas**

DNN has general-purpose characteristics and abilities.

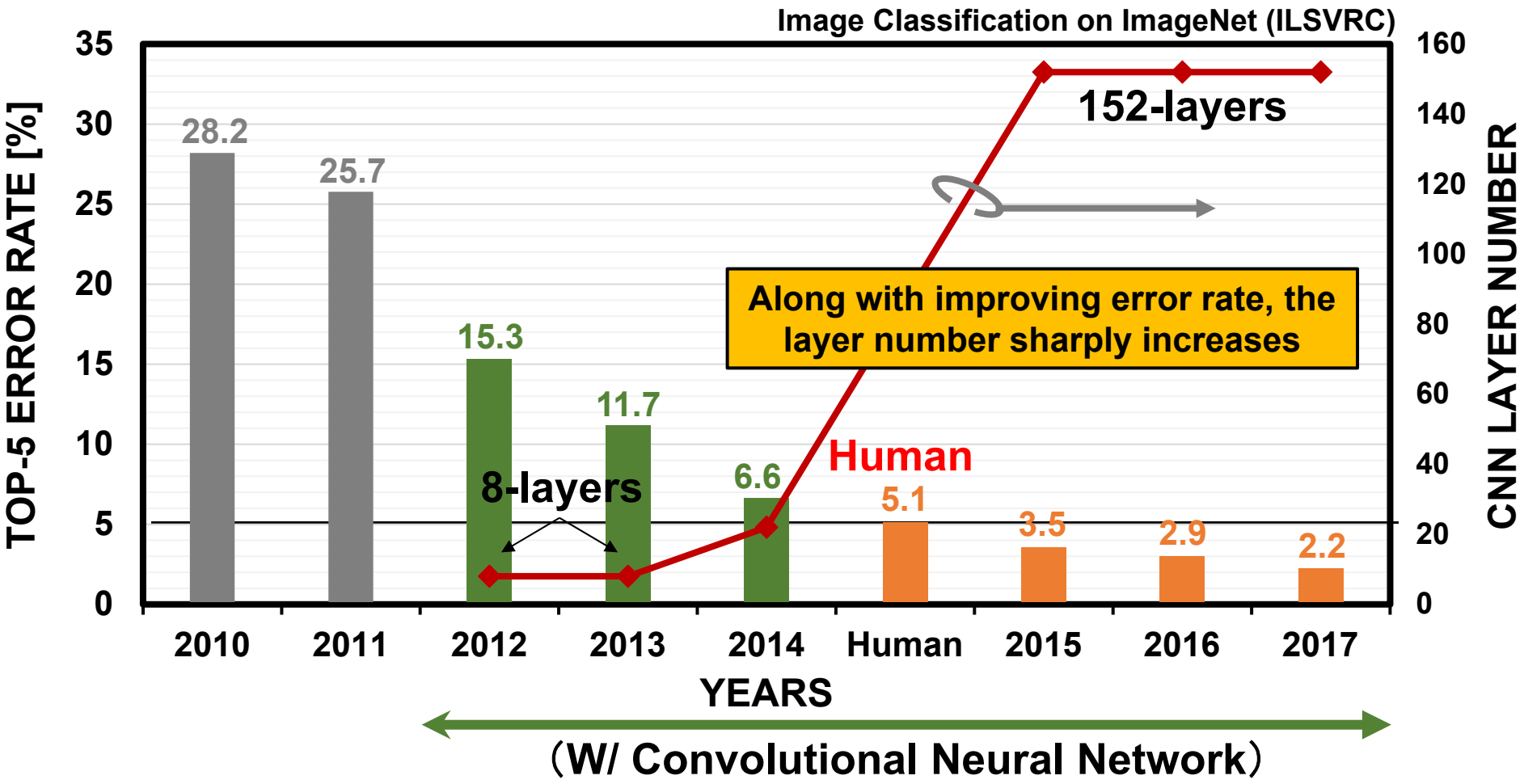
Automotive

Medicine, biology

Robotics



Correlation between accuracy and deepness



Precision of image recognition continue to improve with a deeper and larger-scale neural network models.

Therefore, DNN requests a 'deeper and larger model'.

ISSCC 2017, J. Cong, et. Al. "Neural Network on FPGAs"

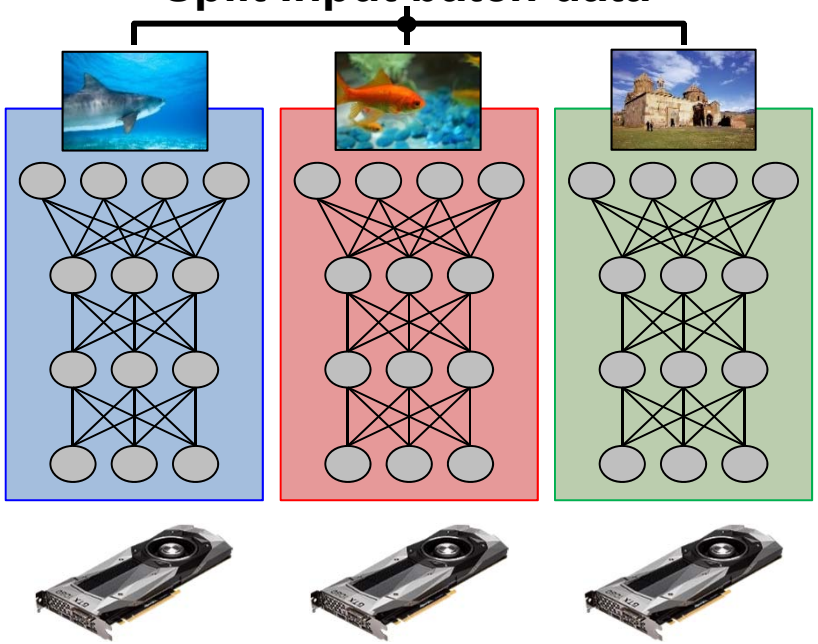
Parallel network training models

- **Deeper network takes longer training time.**
  - **5–6 days** to train AlexNet w/ ImageNet, on x2 NVIDIA GTX580
  - **3 weeks** to train ResNet-200 w/ ImageNet, on x8 GPGPU

- **Data parallelism**

  has divided dimensions of data.

  Split Input batch data

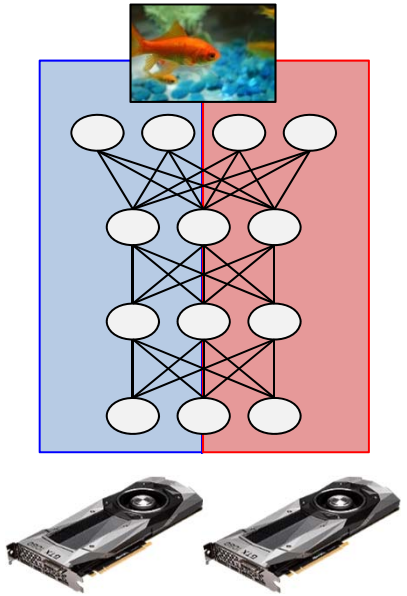


  Each worker trains on the same network but with a different data example.

- **Model parallelism**

  has divided dimensions of a model .

  Input batch data



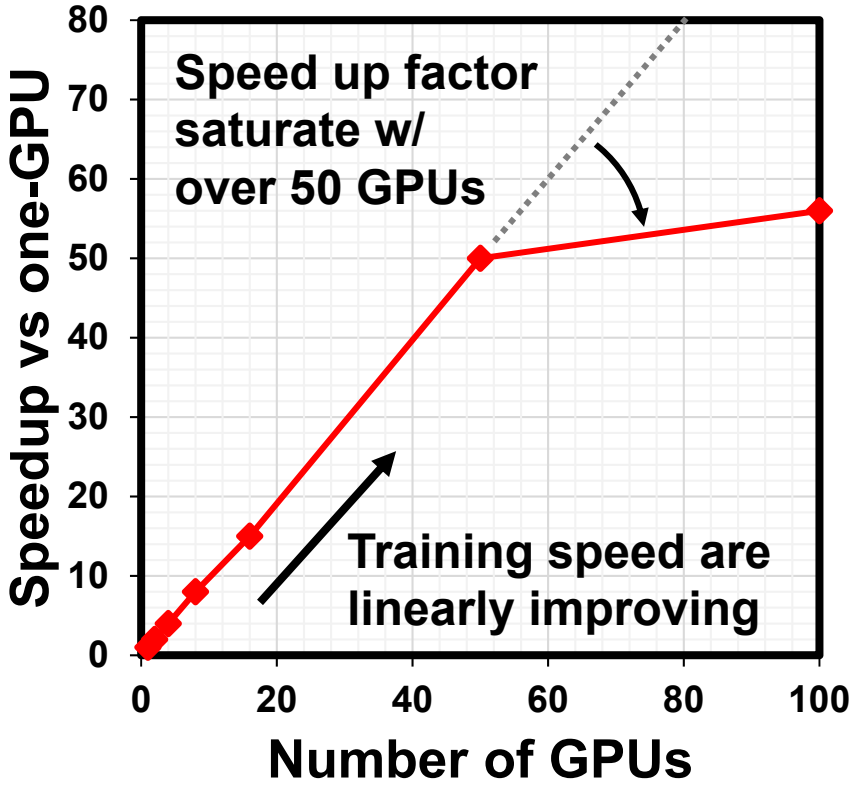
  Each worker trains a different part of the model (network).

CVPR 2016, K. He, et.al., June 2016. *ECCV 2016, K. He, et.al., July 2016.*

Scalability challenge in parallelism

- To improve the parallelization scalability is one of challenges in large-scale distributed neural networks.

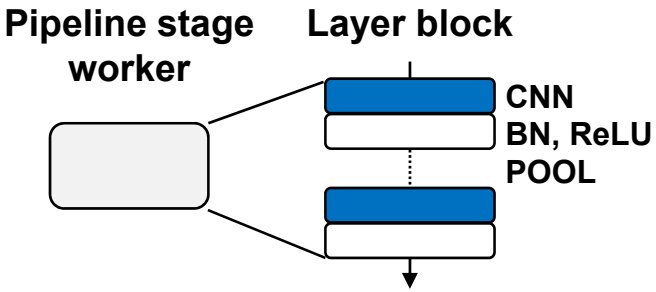
- Accelerate training time is the key for future DNN application.
- External data communication between GPUs and servers are increased for weight unification with the number of GPUs.
- Speed up factor is saturated with large-number of GPUs, that is the scalability limitation in distributed deep learning.



Data from Google Research Blog
https://research.googleblog.com/2016/04/announcing-tensorflow-08-now-with.html

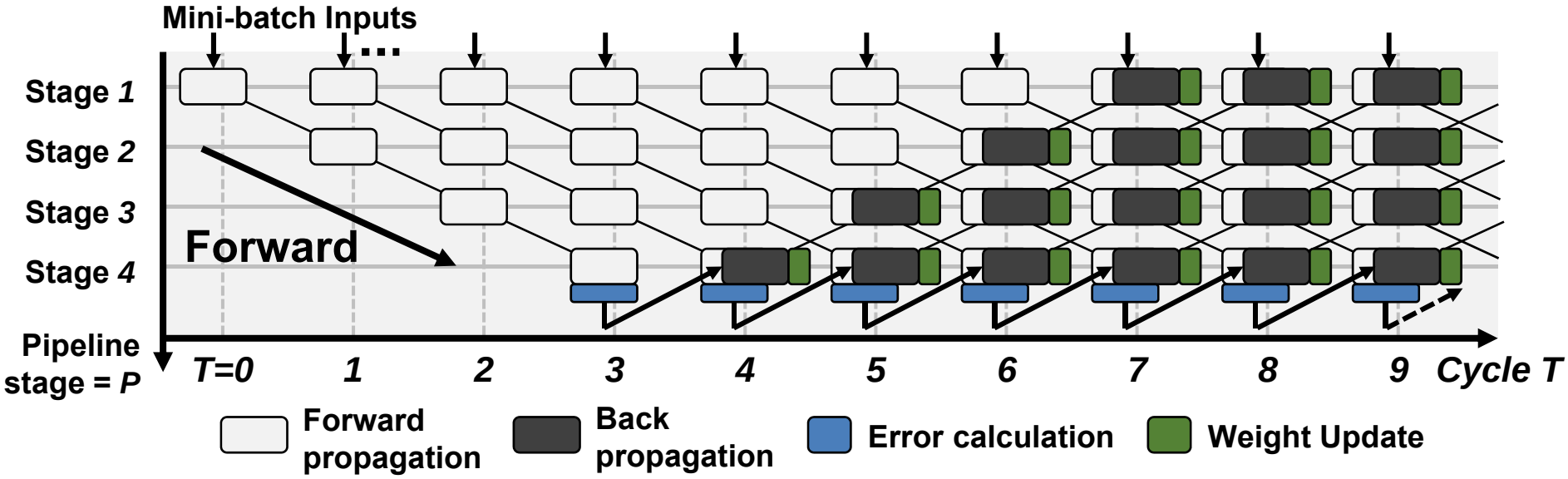
Concept of layer-block-wise pipeline

- **Pipeline stage and layer-block**



  - Each pipeline stage with multiple layers has a worker for both forward and back propagations
  - A worker keeps a single weight matrix corresponding to its own layer network

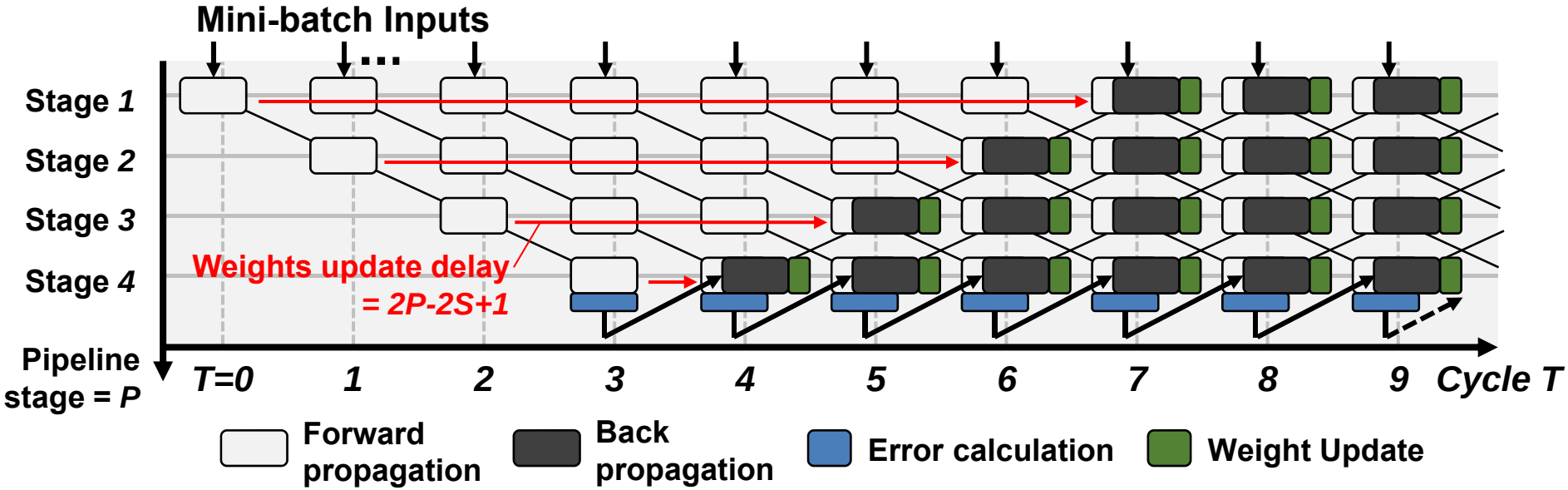
- **Conceptual *4*-stage pipeline**



  - The number of pipeline divisions depends on the DNN models
  - A worker keeps a single weight matrix

Process flow of layer-block-wise pipeline

- Pipeline have only one weight corresponding to the network model.
- Thus, the weights are updated with a latency of **2P−2S+1** due to the single weight parameter corresponding to the model.

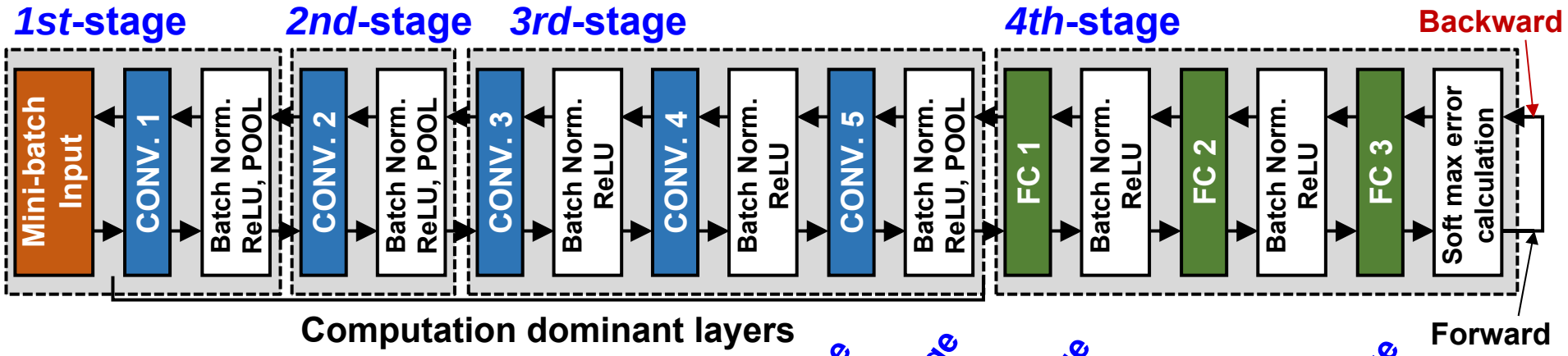


P = number of layer blocks, *S* = current pipeline stage, *T* = current cycle

It can be said that the layer-block-wise pipeline has a concept of approximate computing instead of the naive SGD

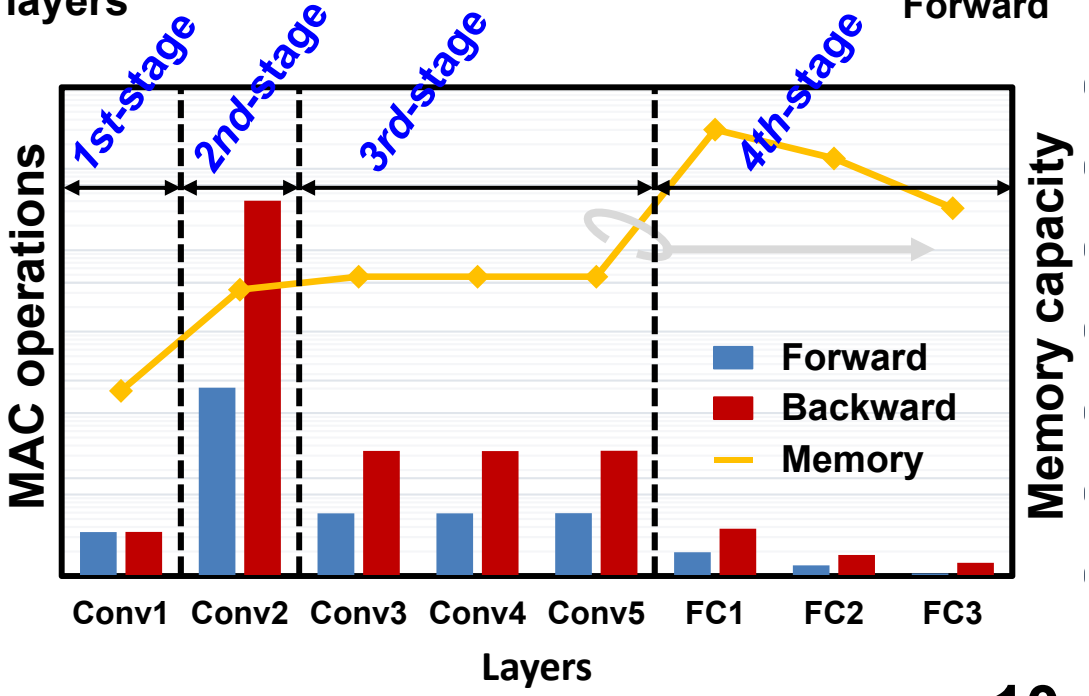
Partitioning for layer-block-wise pipeline (4)

(a) VGG-F network w/ *4*-stage pipeline



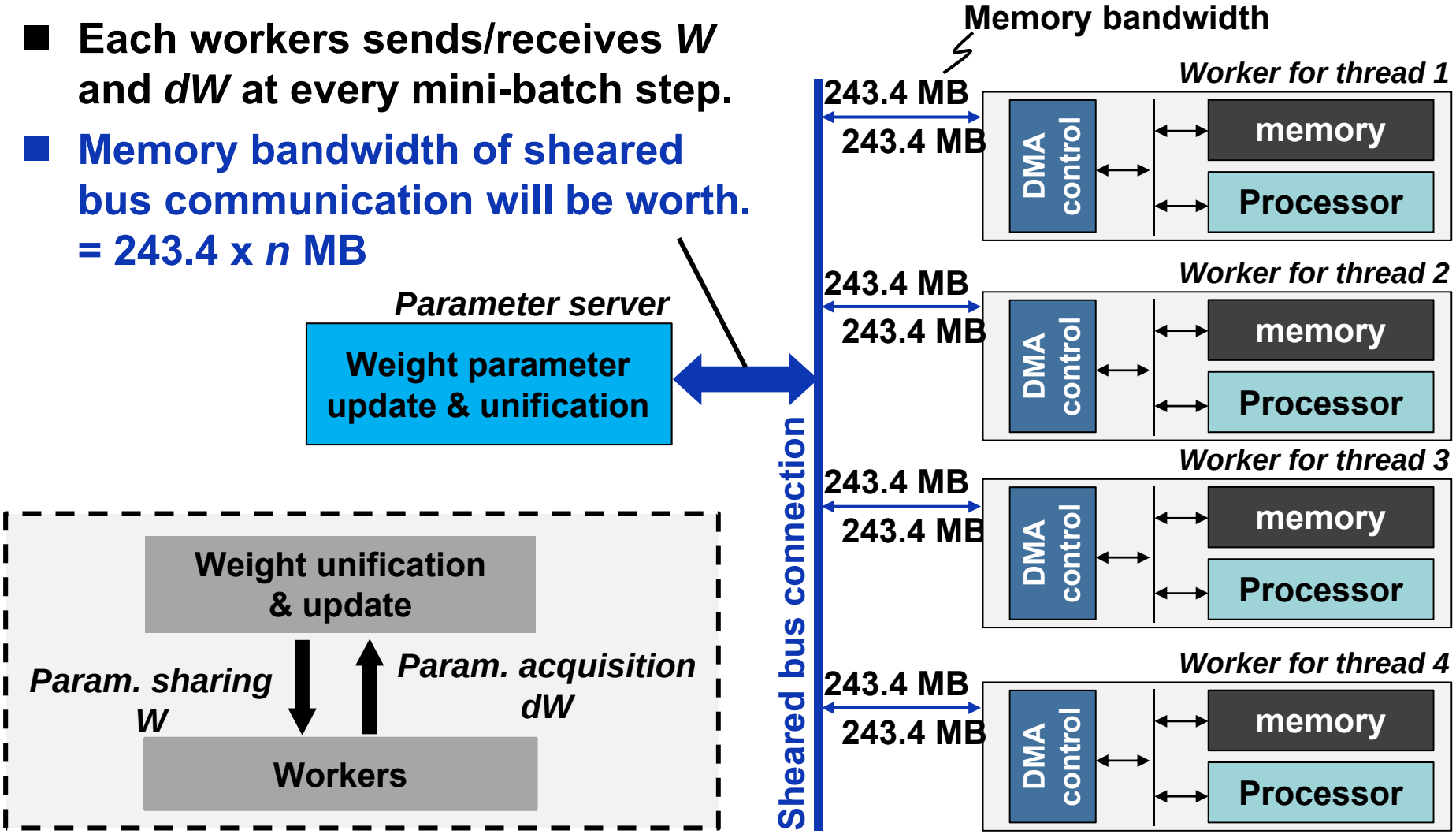
(b) VGG-F computation

- When the four-stages pipeline computation have wide variation
- How to align a computations or memory size remains as an issue to be tackled for a more effective pipeline.



Hardware model evaluation

- *4*-th multithreaded w/ sheared bus connection
  - Each workers sends/receives *W* and *dW* at every mini-batch step.
  - Memory bandwidth of sheared bus communication will be worth. = 243.4 x *n* MB



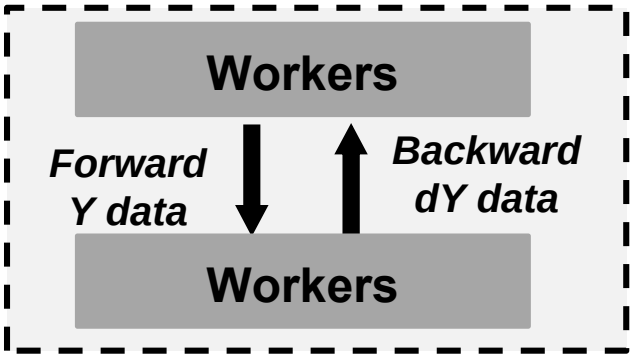
(a) Multithreaded data flow

(b) Multithreaded architecture model

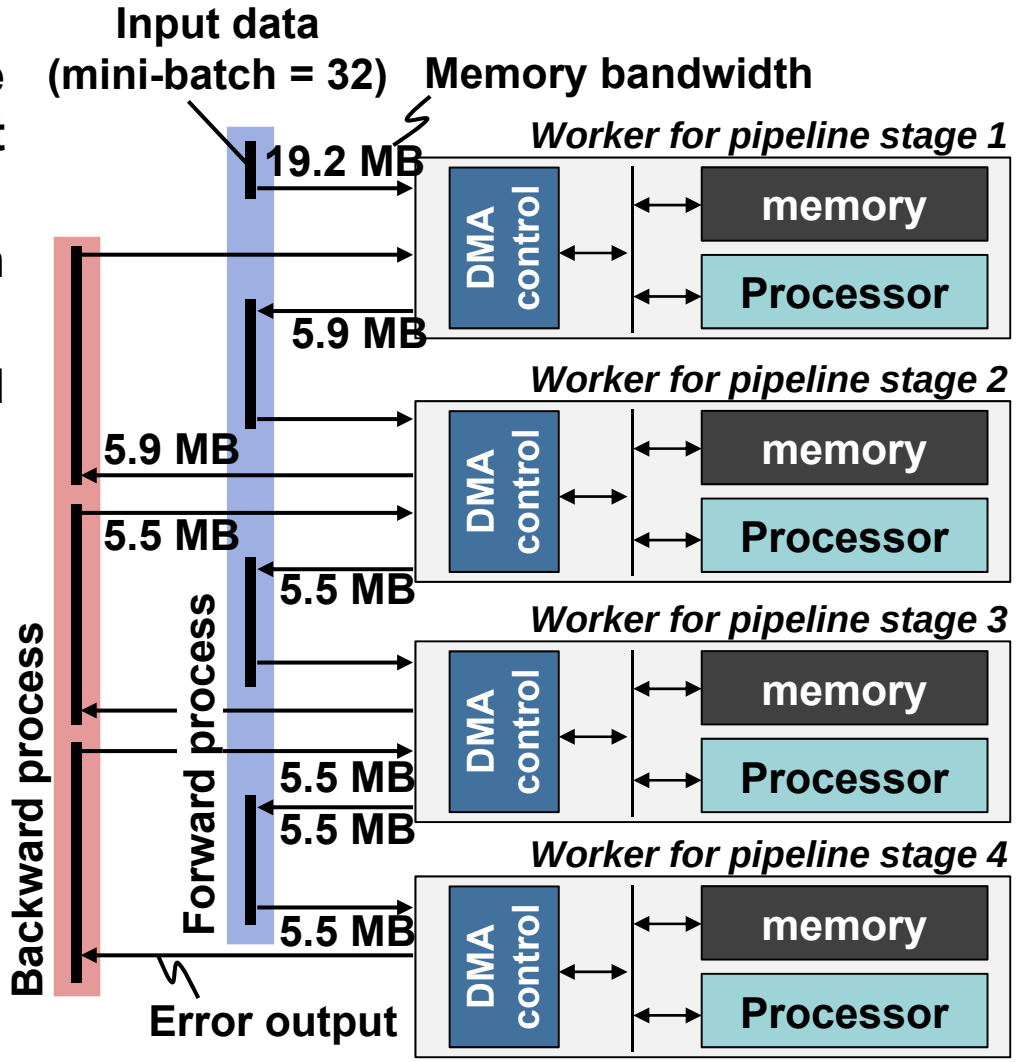
Hardware model evaluation

- ***4*-stage pipeline w/ segmented bus structure**

- Proposed pipeline communicate only between two adjacent workers.
- External data communication depends on a transfer data size.
- Small transfer amount is helpful to shorten computational time.



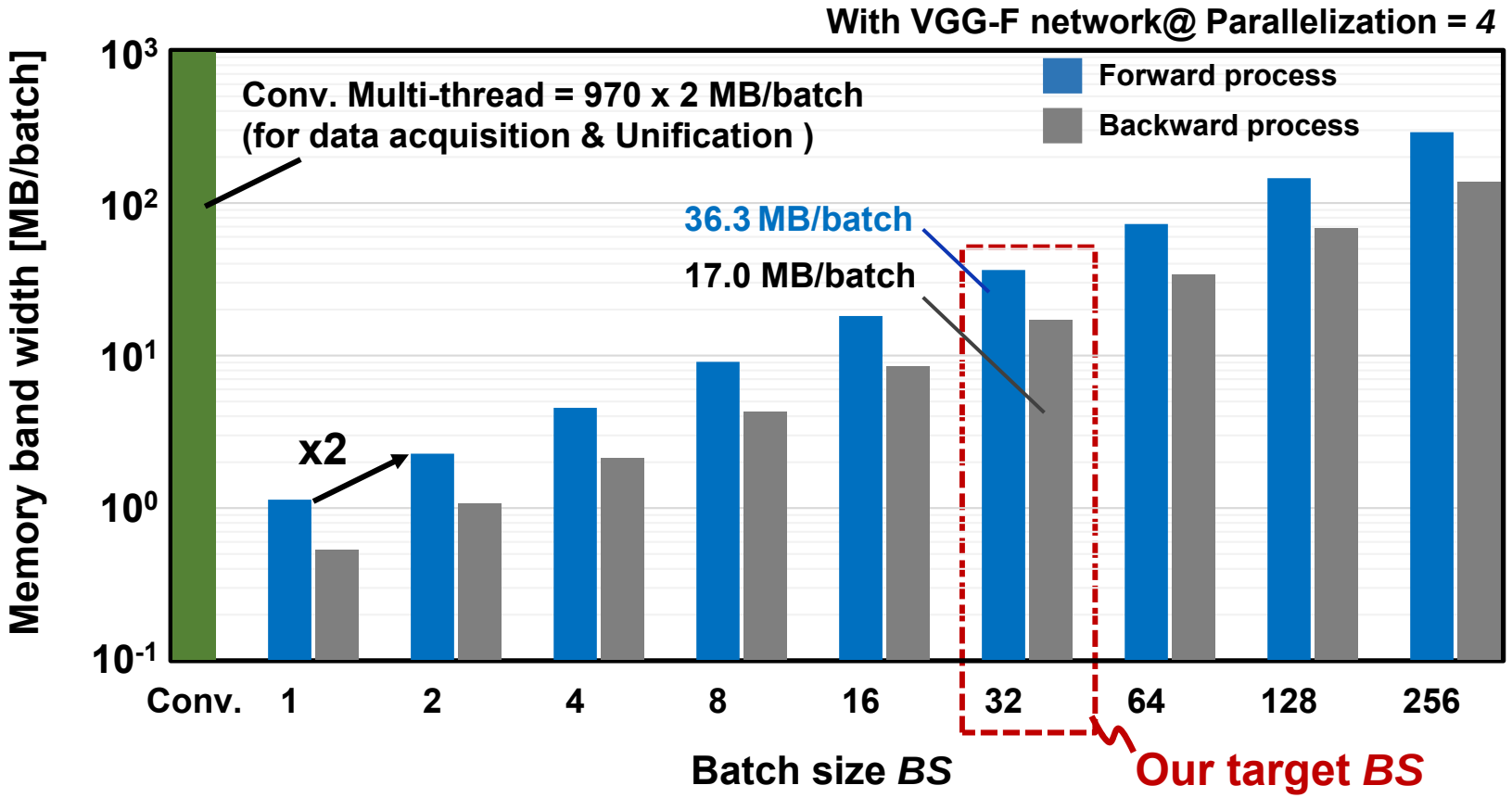
(a) Pipelined data flow



(b) Pipelined architecture model

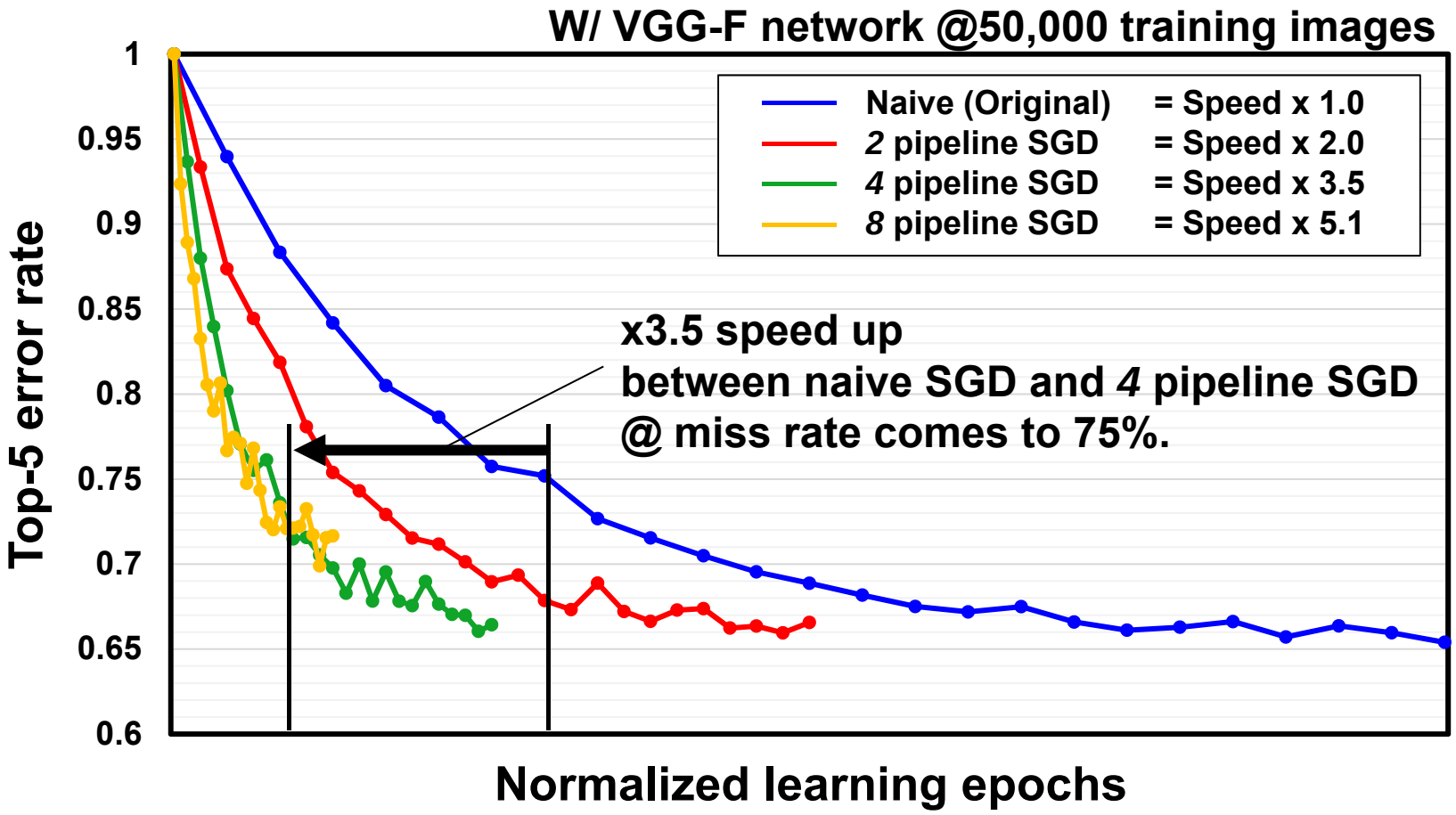
Data transfer amount comparison

- The memory bandwidth evaluation with different mini batch size in layer-block-wise pipeline @ Parallelization = *4*



The proposed pipeline reduce the memory bandwidth by 97.25% (this value will be $\frac{1}{36.4}$) compared with 4-multithread, at *BS* = *32*.

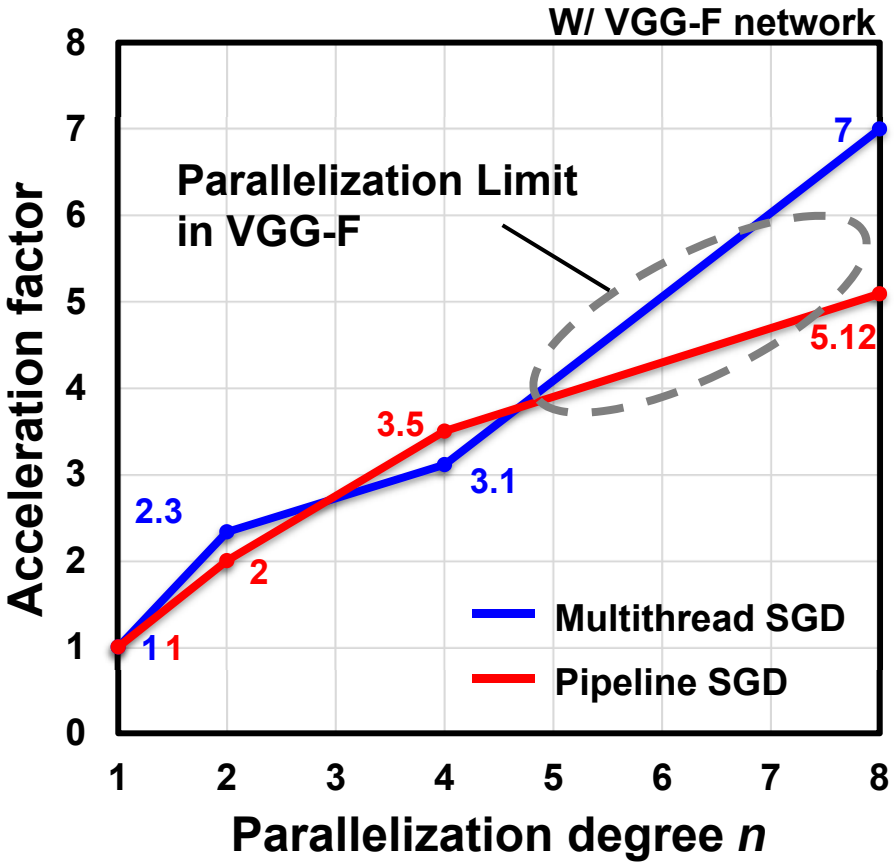
Training convergences evaluation



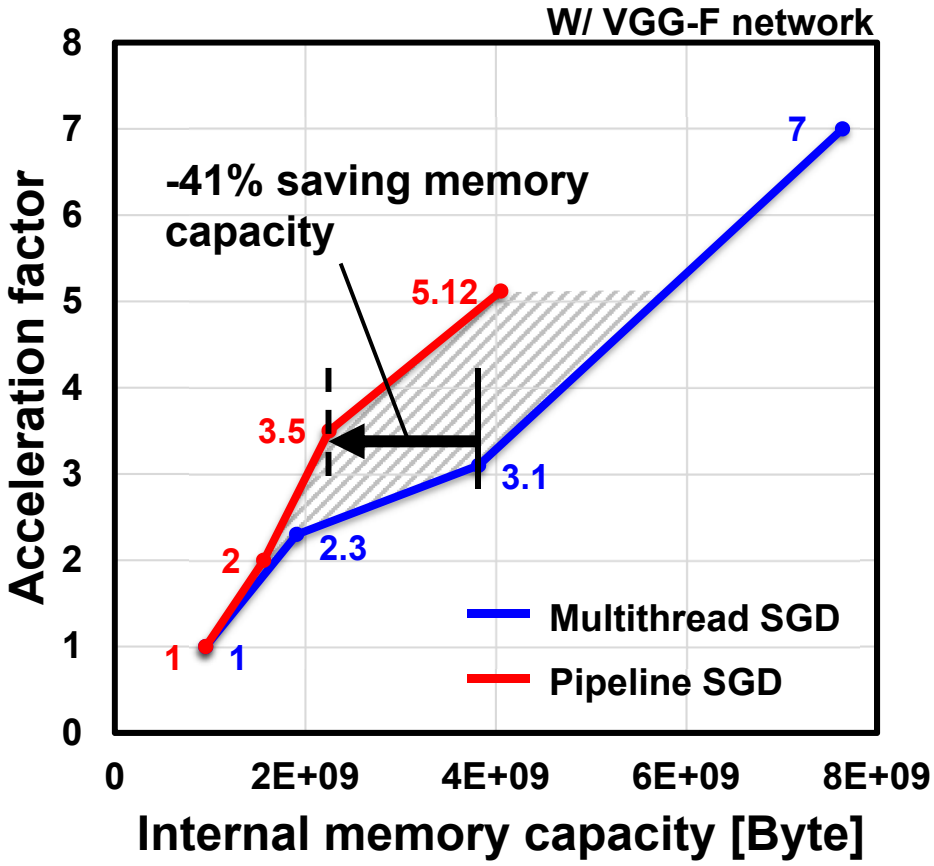
Layer-block-wise pipeline is 2.0 times faster in the *2*-stage pipeline, and is 3.5 times faster in the *4*-stage pipeline.

Convergence speed vs hardware cost

- Convergence speed comparison



- Acceleration factors w/ varied memory



The layer-block-wise pipeline has 41% less memory when parallelization degree = 4, with better acceleration performance per memory capacity: 2.25 GB for pipeline and 3.82 GB for multithread.

Summary

- The layer-block-wise pipeline with segmented I/O bus architecture is proposed.
- The proposed scheme suppresses transfer memory bandwidth and memory capacity with maintaining scalability of parallelism.
  - The proposed pipeline reduces the memory bandwidth by 97.25% (this value will be 1/36.4) compared with *4*-multithread, at *BS* = *32*.
  - The memory capacity of the *4*-stage pipeline is reduced by 41% in comparison with multithreading when a batch size is *32* in VGG-F.

Cooperation with UGA

- **In the field of hardware design**
  - **Hardware system for machine learning**
  - **Low-power high-speed signal processor**
  - **Memory circuits (SRAM, MRAM, FeRAM)**