

# Large-Message Size Allreduce at Wire Speed For Distributed Deep Learning

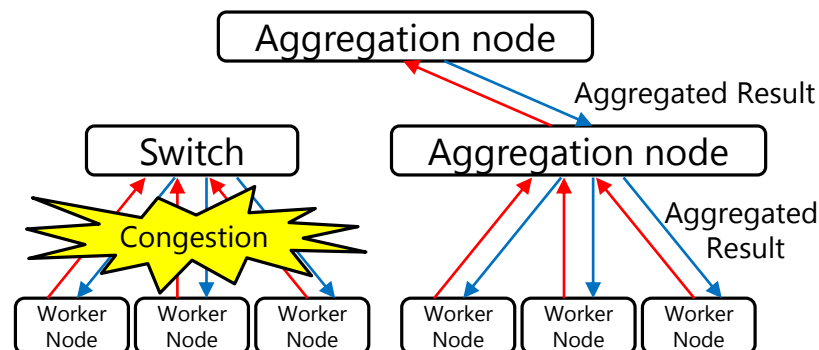
Tanaka Kenji<sup>1</sup>, Yuki Arikawa<sup>1</sup>, Kenji Kawai<sup>1</sup>, Junichi Kato<sup>1</sup>, Tsuyoshi Ito<sup>1</sup>, Huy Cu Ngo<sup>1</sup>,  
Kazutaka Morita<sup>2</sup>, Fumiaki Miura<sup>2</sup>, Takeshi Sakamoto<sup>1</sup>, Satoshi Shigematsu<sup>1</sup>

1. NTT Device Technology Labs, NTT Corporation, Kanagawa, Japan, 2. NTT Software Innovation Center, NTT Corporation, Kanagawa, Japan  
{kenji.tanaka.hr, yuki.arikawa.zs, kazutak.morita.fp, takeshi.sakamoto.wa}@hco.ntt.co.jp



## Summary

### Data Parallel Distributed DNN is Network Intensive



The message size of gradients in ResNet-50 is 100 MB<sup>[3]</sup>.

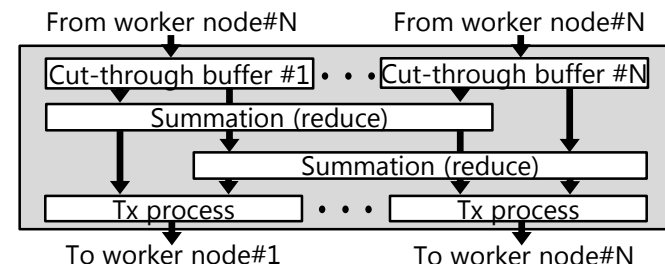
### Efficient Networking is a Key to scale Distributed DNN

**Achievement: Wire-Speed Allreduce for Large-Message**  
Allreduce is almost at wire-speed (96%) with message sizes of 64KB or more.  
Latency can be reduced by 65% at the same size as ResNet-50's gradients<sup>[7]</sup>.

### Allreduce-Specific Dataflow Accelerator

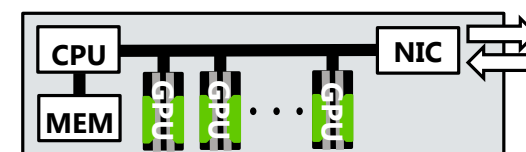
#### ■ Aggregation node

- ✓ Implemented Hardware (FPGA)
- ✓ Cut-Through Buffering
- ✓ Parallel processing



#### ■ Worker node

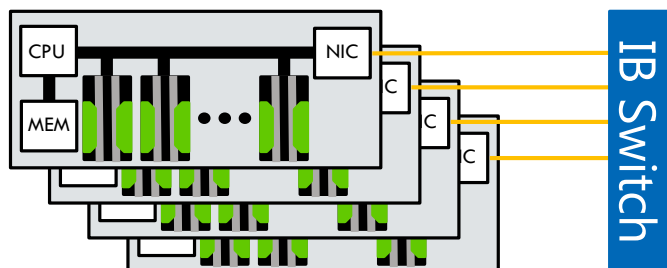
- ✓ Pipeline data transfer
- ✓ Optimizing the number and size of buffers



## Introduction

### ■ Acceleration for Distributed Deep Learning

Distributed Deep Learning with synchronous data-parallelism approach show good performance<sup>[1, 2]</sup>.



### ■ Allreduce for large-message size

Since the message size of gradients reaches  $10^5 - 10^9$  Byte (ResNet-50's gradients: 97.5 MB)<sup>[3]</sup>, increasing the throughput and reducing the latency of Allreduce for large message are key issues.

## Related Work

### ■ Intra-node Allreduce

#### Ring-Allreduce Algorithm<sup>[4]</sup>

e.g. NVIDIA Collective Communications Library<sup>[5]</sup>

**Pro:** Bandwidth congestion does not occur  
**Con:** Operation time  $\propto$  number of transfers

It is better to use NCCL for intra-node and alternative collective-communication methods for inter-node<sup>[6]</sup>.

### ■ Inter-node Allreduce

#### Hierarchical Allreduce Algorithm & Aggregation node

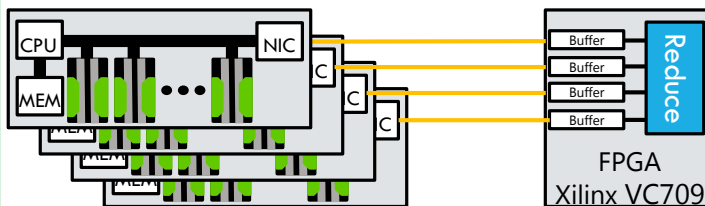
e.g. Scalable Hierarchical Aggregation Protocol<sup>[7]</sup>

**Pro:** Allreduce while data is being transferred

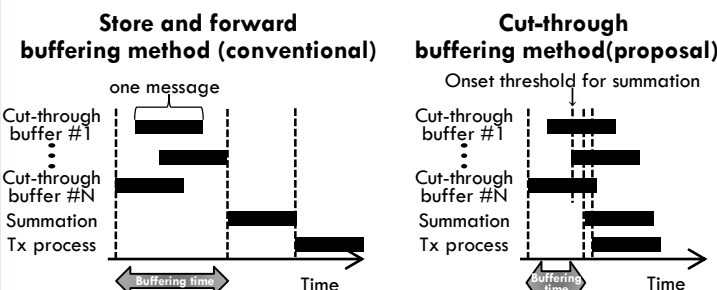
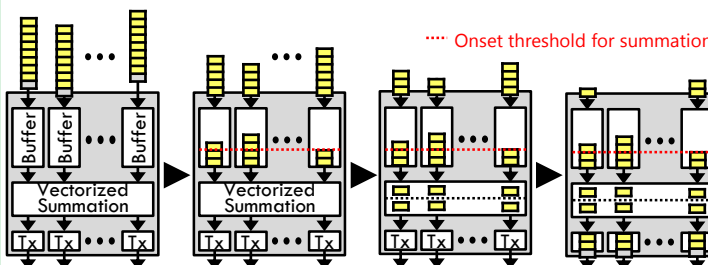
**Con:** Allreduce at wire-speed is difficult due to data store latency by storing and forwarding buffering method<sup>[8, 9]</sup>.

## Architecture and Mechanism

To achieve Allreduce at wire speed, we proposed a dataflow architecture.



### ■ Allreduce-Specific Dataflow Accelerator



The vectorized summation is started when head of input frames exceeds the threshold, which is set in cut-through buffering.

### ■ Gradients Casting Technique

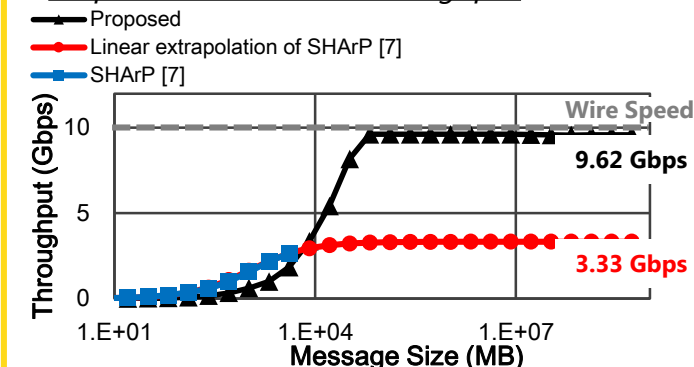
Reducing gradient-casting latency between GPUs and NIC

- Controlling the number and size of Rx and Tx buffers
- Performing pipeline data-transfer asynchronously

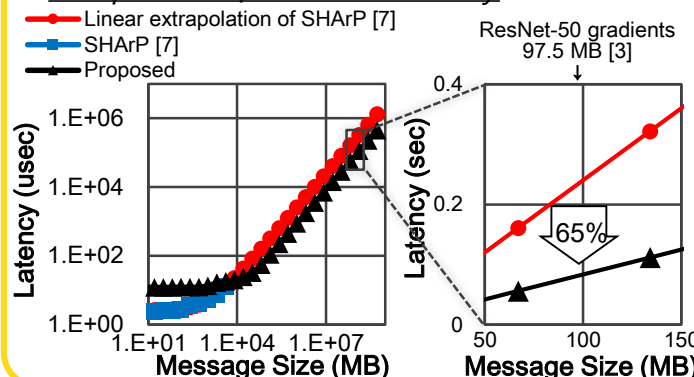
## Experimental Result

■ Accelerator Processing Speed: 13.6 Gbps@1 KB

■ Proposed Architecture Throughput:



■ Comparison of Allreduce Latency



## Conclusions & Future Work

Experimental results reveal

- The proposed architecture performs Allreduce at 96% of wire speed for a large-message.
- The latency of Allreduce is reduced by 65% as compared with SHARP when applied for ResNet-50.

We will build a large-scale distributed DL infrastructure using proposed low-latency Allreduce-specific dataflow architecture in the near future.

## Evaluation environment

### **Propose:**

(4 worker node)

CPU: Xeon E5-2603 with 64 GB RAM

GPU: Nvidia Tesla K20

NIC: Intel 82599 10 Gigabit Ethernet Controller

(1 aggregation node)

FPGA: Xilinx VC709.

### **SHArP [7]:**

(128 worker node)

CPU: Intel Two 14-core Broadwell with 256 GB RAM

HCA: Mellanox ConnectX-4 HCA 100 Gigabit InfiniBand.

(8 aggregation node)

Switch: Mellanox SwitchIB-2.