IBM Research AI

Efficient Communication Library for Large-Scale Deep Learning

Mar 26, 2018

Minsik Cho (minsikcho@us.ibm.com)

Deep Learning changing Our Life

Automotive/transportation

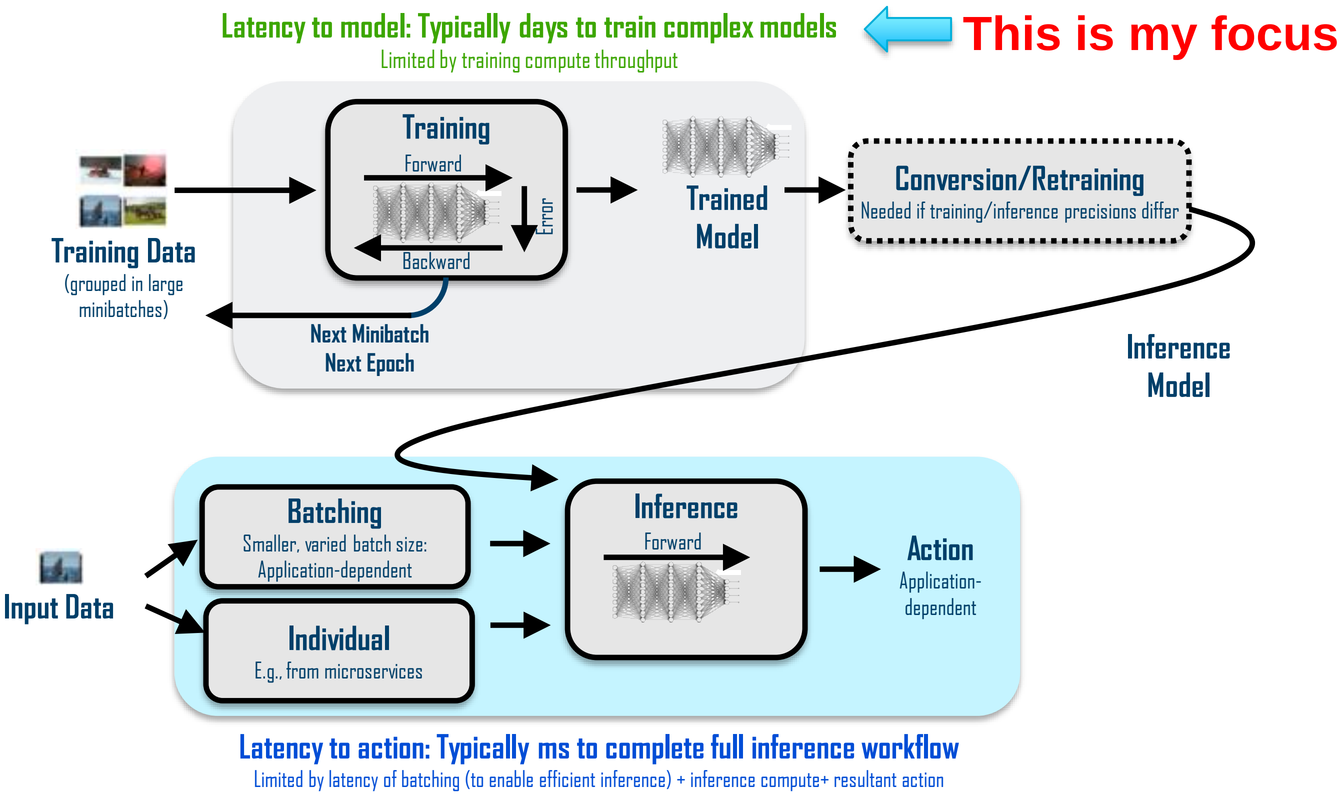
Security/public safety

Medicine and Biology

Media and Entertainment

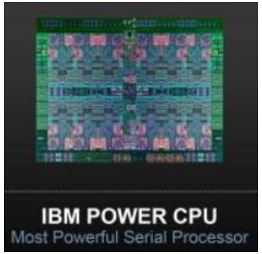
Consumer Web, Mobile, Retail

Deep Learning Workflow



Advance in Computation for Deep Learning

[P. Goldsborough]

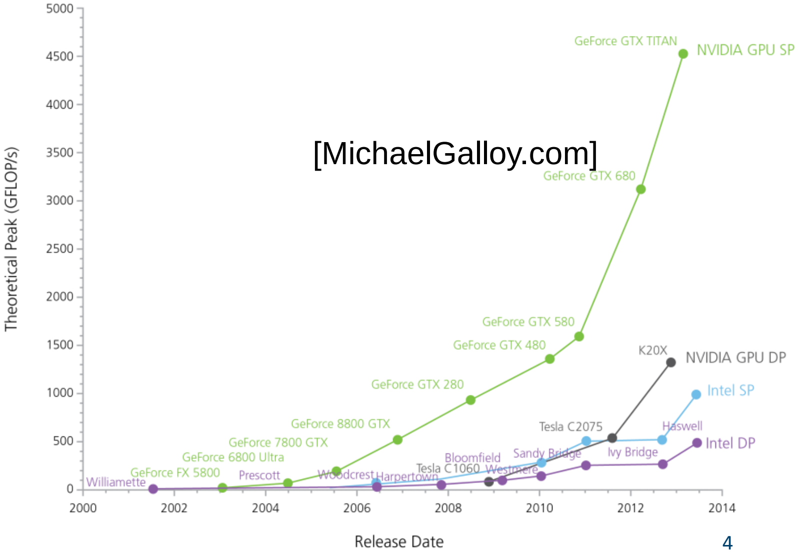


CPU

GPU

ASIC/FPGA

- 10-100 TFLOPS
- Very good scaling for last 15 years

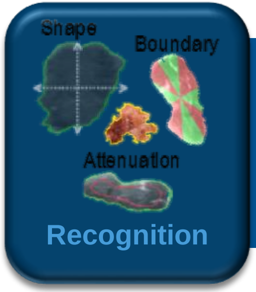


Motivation: Ok, ever-fast computation. Is this enough?

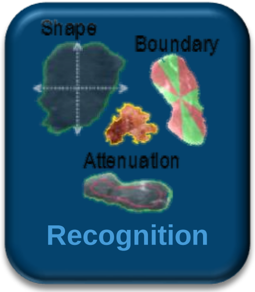
- ImageNet1K : 1.2M images, 1K classes, Resnet101
  - Batch-size = 32 (limited by GPU memory)
  - Iteration time = 300ms
  - #iterations per epoch = 38000
  - Total training time for 100 epoch = 13.2 days
- ImageNet22K : 7.5M images, 22K classes , Resnet101
  - Total training time for 100 epoch = 35.2 days
- **No, it is NOT**
  - **1.2M samples are still at toy scale**
  - **Computation scaling is not fast enough**
    - **the data explosion/model complexity**
    - **Innovation will take too long, or even stop at some point**
  - **I cannot wait for days to get my model trained!**

Faster Training Time with Distributed Deep Learning

IBM



9 Days



4 Hours | 4 Hours | 4 Hours | 4 Hours | 4 Hours | 4 Hours | 4 Hours | 4 Hours

What will you do?
Iterate more and create more accurate models?
Create more models?
Both?

4 Hours | 4 Hours | 4 Hours | 4 Hours

100x

Learning runs with Power 9*

54x

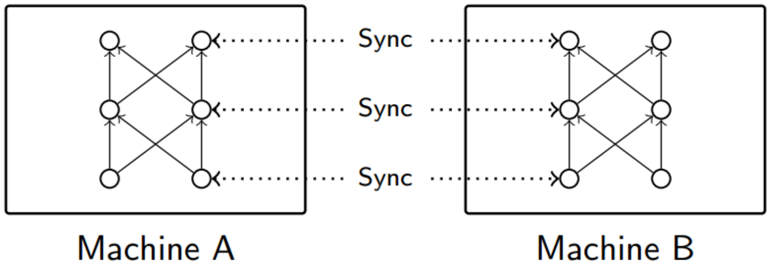
Learning runs with Power 8

IBMPowerAI

Distributed Deep Learning

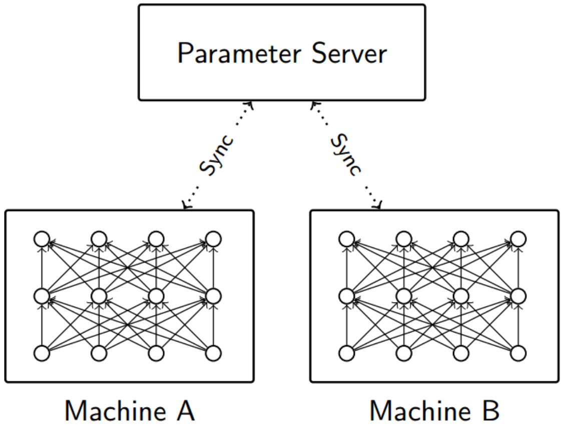
[P. Goldsborough]

Anything

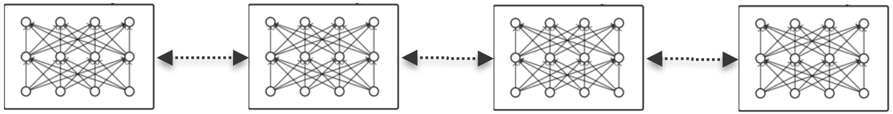


Model parallelism
(complex partitioning)

Gradient/weight (10MB-1GB)

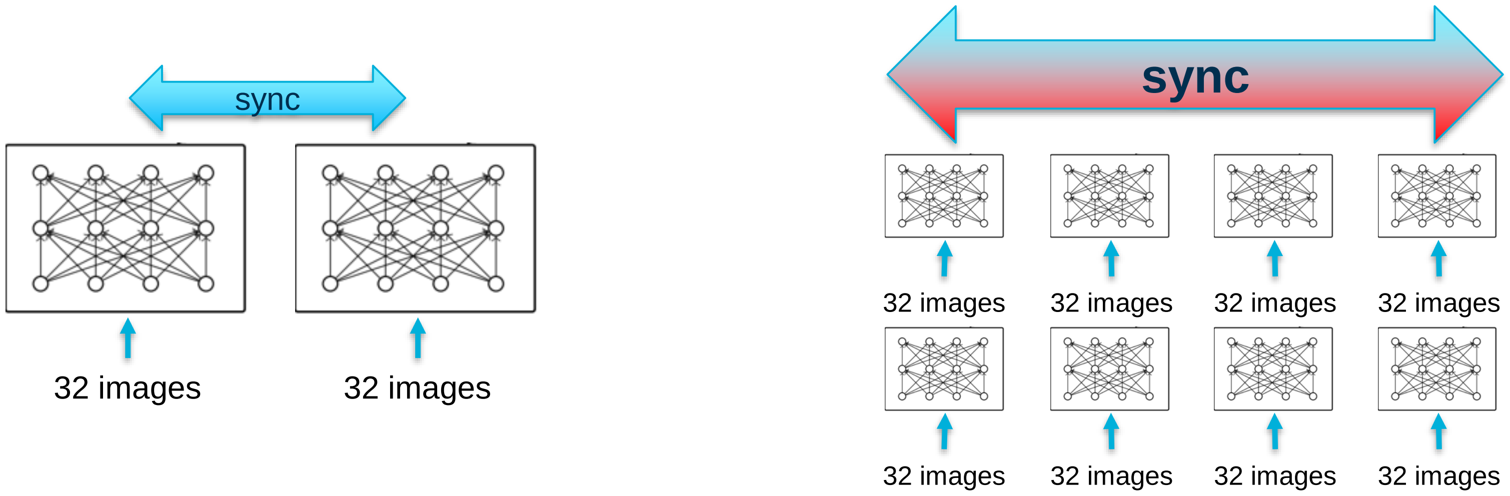


Data parallelism : Parm-Server



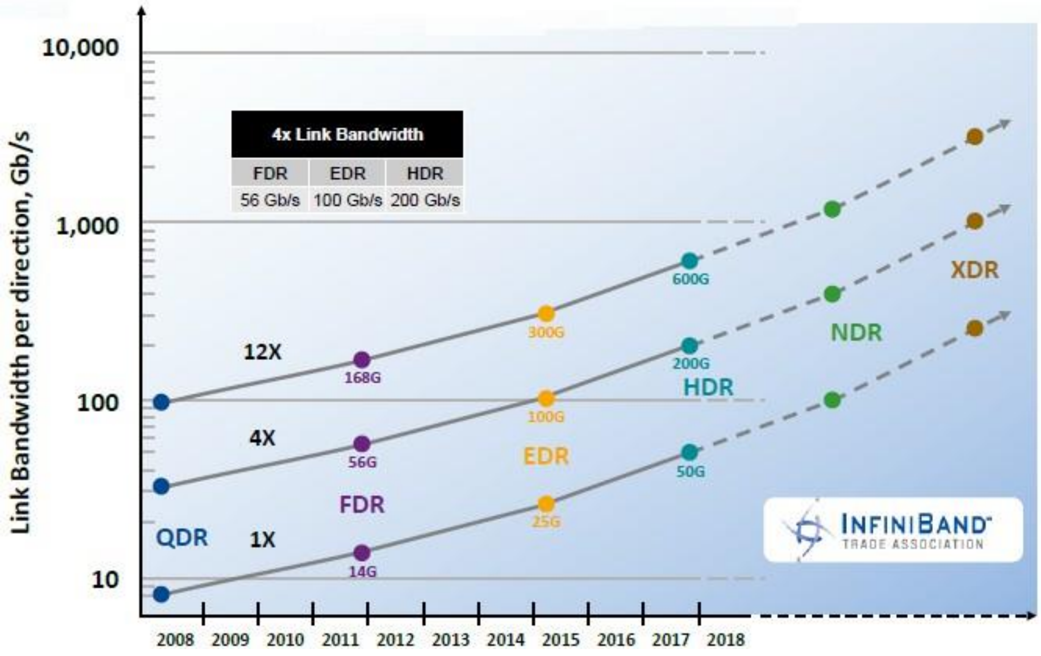
Data parallelism : Allreduce

Communication : Overhead



- In weak-scaling
  - Computation cost remains constant
  - Communication cost increases with more learners/GPUs
- Computation /Communication is the key for large-scale deep learning
  - Increase Computation
  - Faster Communication

Advance in Communication for Deep Learning



- Still scaling, but not fast enough
  - Computation is still ahead
  - Data perhaps grows much faster

Designing Large-scale Deep Learning

Communication
Gradient count
Network BW
Faster algorithm

Good balance

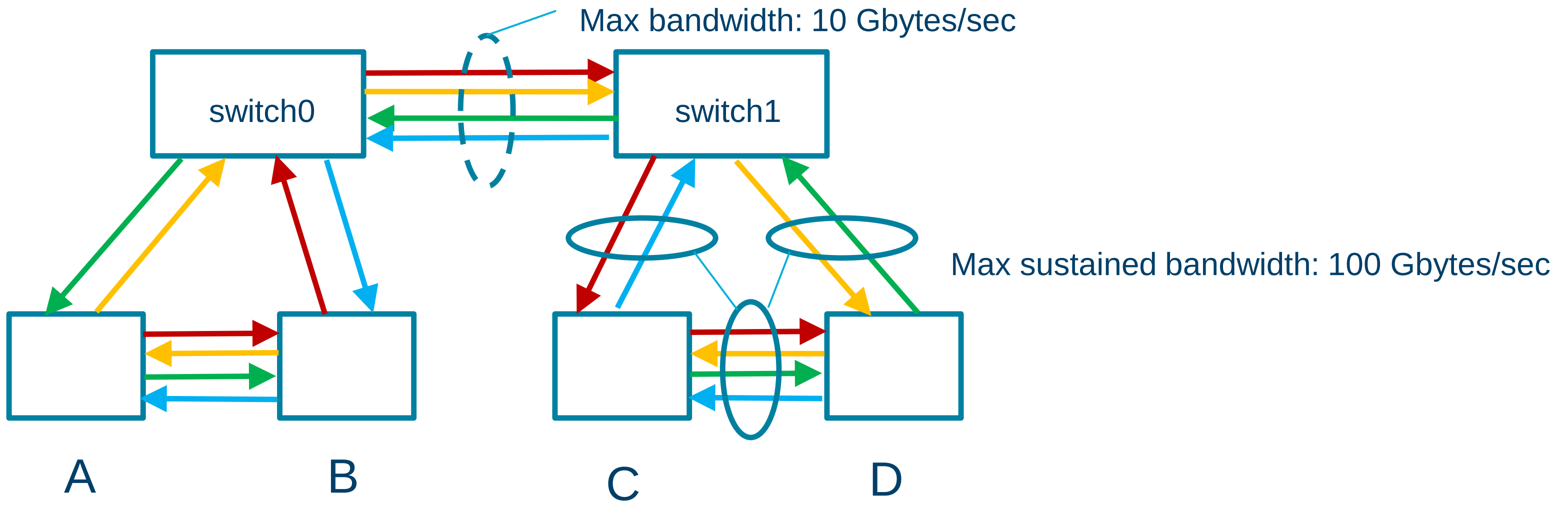
Computation
Model depth
GPU throughput
Faster algorithm
Mini-batch size

- Model/Application
  - Deeper/wider model to increase compute time
  - Smaller gradient count to reduce communication time
- System
  - Balance network and computing resources
  - Select mini-batch size to adjust the ratio
    - Larger mini-batch size to lower the ratio
    - Too big mini-batch size can hurt convergence and accuracy
  - Network-topology aware communication

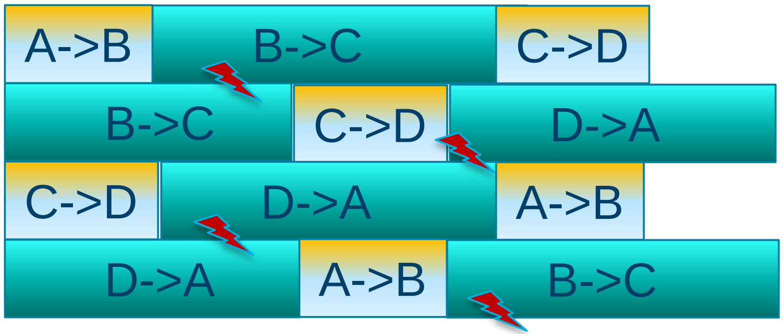
IBM PowerAI DDL (Distributed Deep Learning Library)

- Collaborative communication library for Distributed Deep Learning
  - MPI-like interface for easy-integration
  - Enables deep learning software to scale to 100s severs with CPU/GPUs
  - Works across variety of system sizes
  - Works with variety of network types, switch topologies
- DDL orchestrates the data communication
  - Plan efficient communication pattern on a hierarchal network environment
  - Actual point-point data transfer via NCCL or MPI
- Currently integrated into
  - Supported: Caffe, Tensorflow, Chainer, Torch
  - Ongoing : Caffe2, PyTorch, Keras (TF-backend)
- Currently US patent-pending

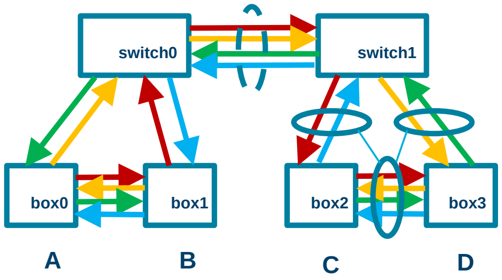
DDL : Topology-aware communication

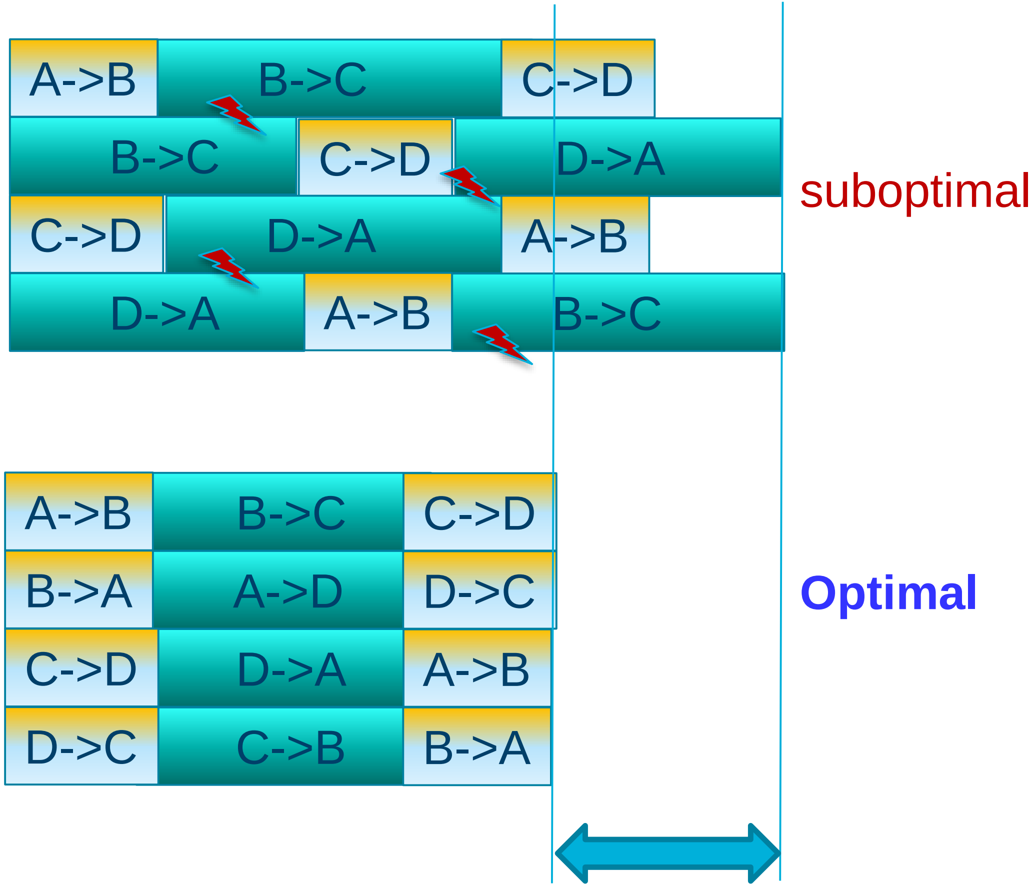


- Example
  - A, B, C, D broadcast to all others
  - Suffers from congestion
  - Suffers from lower BW



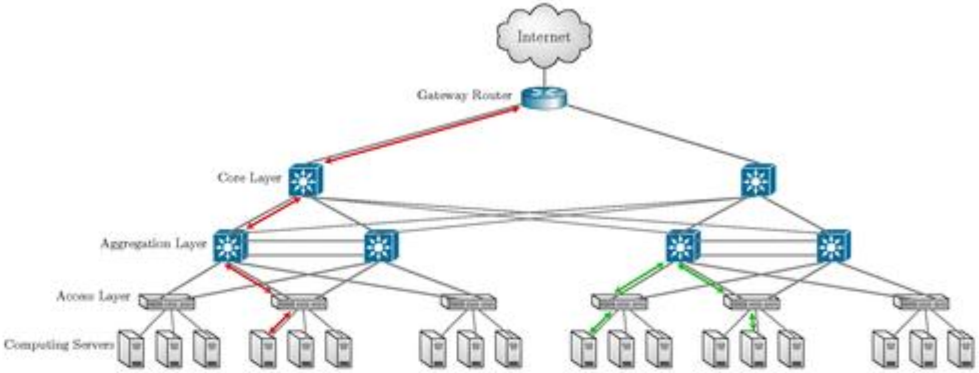
DDL : Topology-aware communication





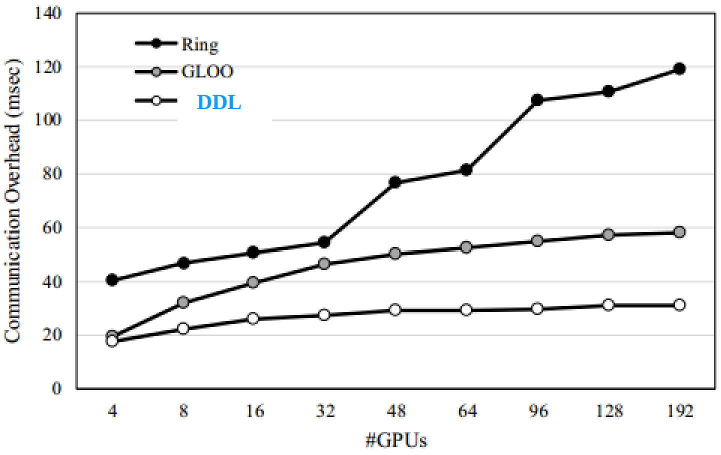
- It's a mapping problem
  - System-specific network
  - Application-specific traffic
- DDL does differently
  - To minimize bus contention
  - To minimize crossing lower BW

DDL : Problem Definition and Solution

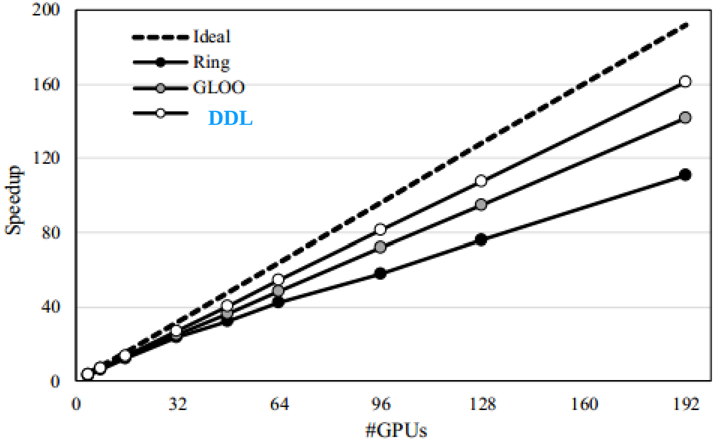


- Assumption
  - network topology with various bandwidths
- Problem Definition
  - min-cost multi-commodity flow problem
  - NP-hard problem but can be solved easily if graph size is small (ie 4 vertices)
- DDL solves a typical case/topology offline
  - if the cluster/cloud has provide such topology, it performs very well

DDL : How well it performs on Caffe2



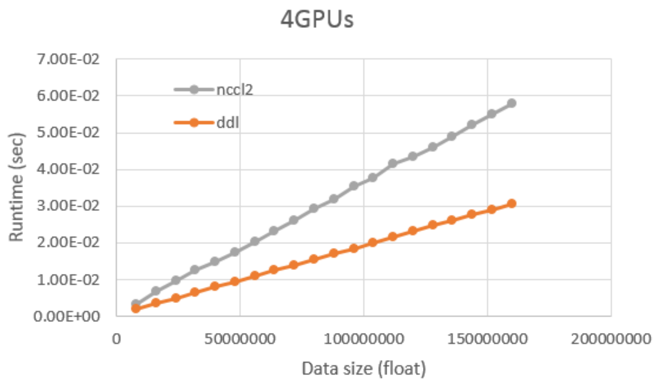
(a) Communication Overhead

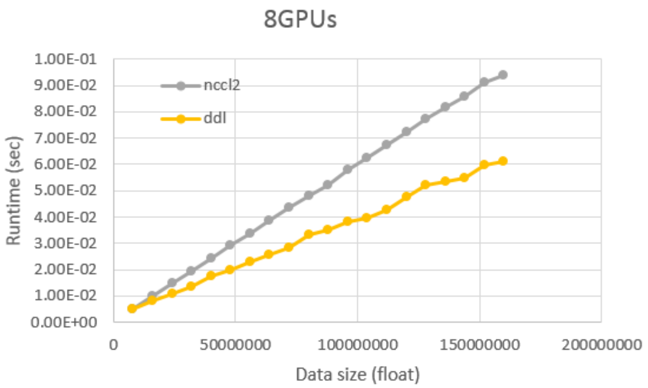


(b) Scaling Efficiency

- 48 IBM S822LC with PPC64LE RHEL
  - 3 racks and 16 hosts on each, connected though 10GB/s IB
  - Each host has 4 P100-SXM2 with CUDA8, CUDNN5
- Comparing algorithms on Resnet50 + Imagenet1K (preloaded to RAMDisk) mbs=32
  - MPI_Allreduce
  - Ring (all-reduce from Baidu in Feb 2017)
  - GLOO (from Facebook) : NCCL+ib_verb

Comparison with NCCL 2.1.x Allreduce (POWER)



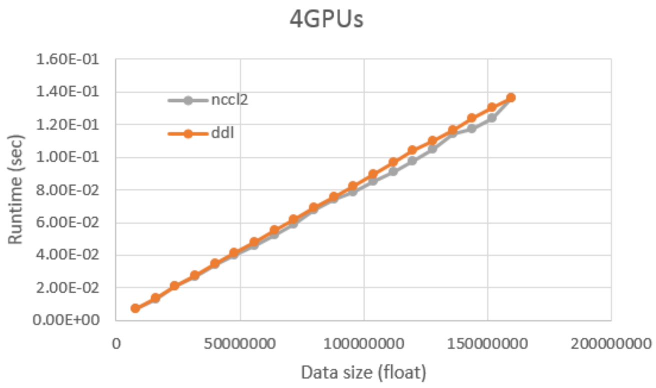


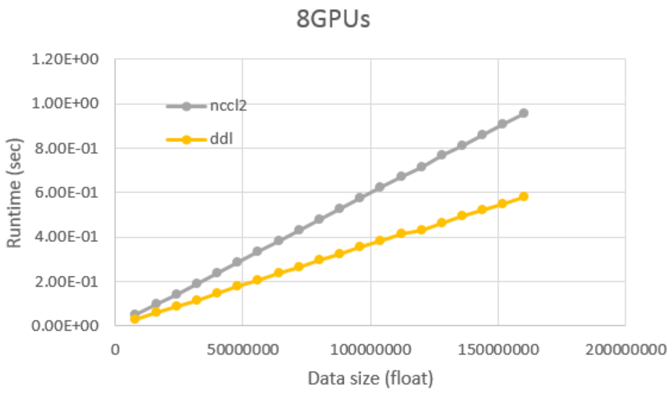
Exploiting in-system topology

Exploiting in/cross-system topology

- IBM P9 Newell Systems (NVLink) with V100s
- 100Gbps InfiniBand

Comparison with NCCL 2.1.x Allreduce (X86)





NO in-system topology

Exploiting cross-system topology

- X86 Systems (PCIe) with P100s
- 10Gbps Ethernet

Conclusion

- DDL is a topology-aware communication library in PowerAI
- DDL delivers the industry-best performance with
  - Network hierarchy
  - Multi-tier bandwidth
- DDL is suitable for common distributed training on cloud environment

BACKUP