Toward Training a Large 3D Cosmological CNN with Hybrid Parallelization

Yosuke Oyama[1,2,⋆], Naoya Maruyama[2], Nikoli Dryden[3,2], Peter Harrington[4], Jan Balewski[4], Satoshi Matsuoka[5,1], Marc Snir[3], Peter Nugent[4], and Brian Van Essen[2]

[1]Tokyo Institute of Technology, [2]Lawrence Livermore National Laboratory, [3]University of Illinois at Urbana-Champaign, [4]Lawrence Berkeley National Laboratory, [5]Riken Center for Computational Science

⋆oyama.y.aa@m.titech.ac.jp

Overview

We present a case study of large-scale training of a 3D convolutional neural network model for cosmological analyses of dark matter distributions. This work extends existing work [1] for predicting cosmological parameters using CNNs for better prediction accuracy and performance by exploiting finer-grained parallelism in distributed convolutions. We show significant improvements using the latest complex cosmological dataset in both strong scaling and weak scaling, achieving 1.42 PFlop/s (111x of speedup over one computing node) on a single training task with a mini-batch size of 128 by using 512 Tesla V100 GPUs. Our framework enables to train a huge CNN whose input size is 4×512^3 that was previously unfeasible due to its memory pressure.

Background: Hybrid-parallel training

- Hybrid-parallelism has advantages over data/model-parallelism
  - More amount of parallelism: The mini-batch size (N) $\times$ layer size (W^n, where n is the number of layer dimensions)
  - Less influence on resulting inference accuracy by keeping the mini-batch size small

Table 1. Comparison of parallel strategies for training a CNN.

	Data	Model	Hybrid
What to parallelize	Samples	Layers	Samples & Layers
Available parallelism	$O(N)$	$O(W^n)$	$O(NW^n)$
GPU memory pressure	✗	✓	✓
Influence on accuracy	✗	✓+	✓
Weak-scaling	✓	N/A	✓
Strong-scaling	✓+	✓	✓+

CosmoFlow

- The CosmoFlow dataset [1, 2] is composed of a set of 3D dark matter distributions along with their cosmological parameters
  - We use a CNN composed of seven 3D convolutional layers and three fully-connected layers
  - A single network cannot be trained with the original input size due to its memory pressure

Table 2. Summary of the CosmoFlow dataset and the network architecture.

Input width (W)	128	256	512
# of input channels (C)	**4**	**4**	**4**
# of samples	65,728	8,216	1,027
Dataset size [TiB]	1.00	1.00	1.00
# of conv. ops. [GFlops/sample]	55.55	443.8	3550
(Forward)	18.52	147.9	1183
Memory [GiB/sample]	0.824	6.59	**52.7**
# of parameters [10^6]	9.44	9.44	9.44

Proposal: LBANN + Distconv for 3D CNNs

Distconv [3]: A hybrid-parallel CNN training implementation

- Distconv distributes the computation of convolutional layers to a set of GPUs
  1. Perform convolution to the center part of an input tensor
  2. Start a halo exchange among GPUs in the same sample group in an asynchronous stream
     - Repeat the one-dimensional halo exchange three times to perform the three-dimensional halo exchange
  3. Convolution is performed to the halo region in the halo stream

Two different I/O and communication-avoiding data readers

- Use the Conduit [4] data exchange library as an I/O backend ("Conduit")
  - preload the entire dataset into CPU memory before training starts
  - employ an MPI-based data exchange to shuffle the data to the process that requires it
- Read the dataset from the global file system or node-local SSDs ("Direct I/O")

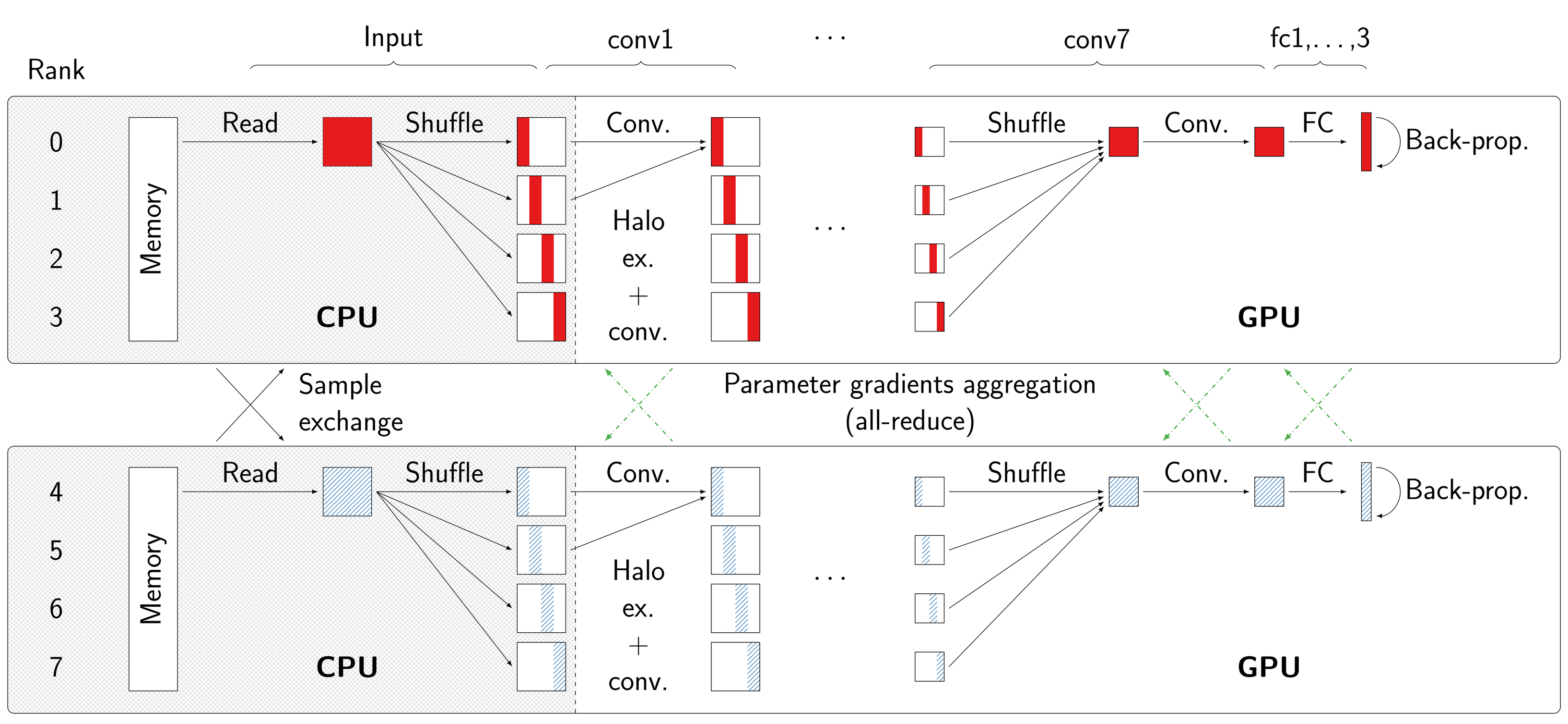
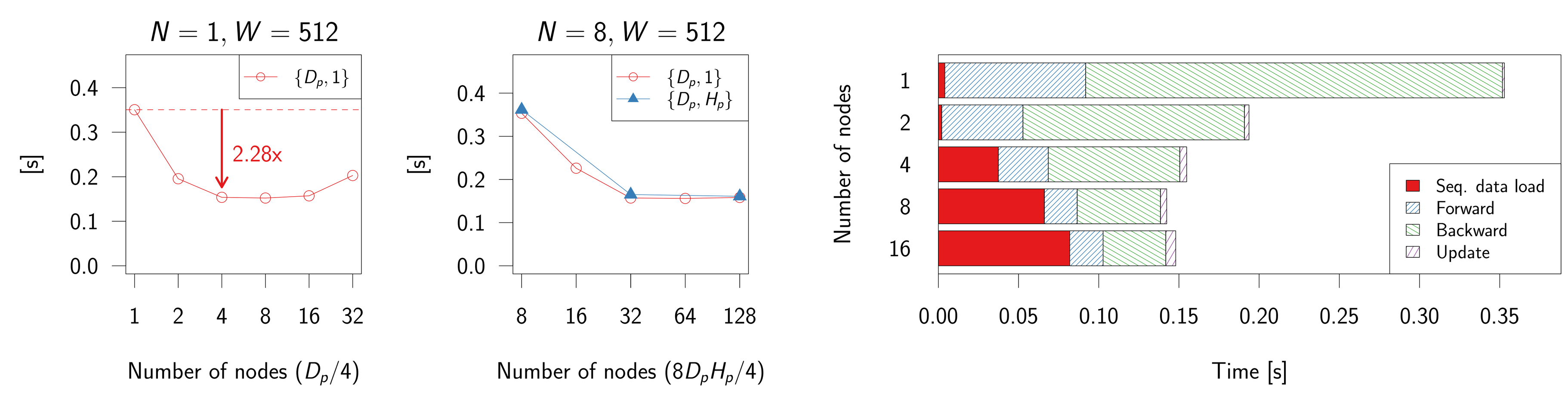


Figure 1. Overview of hybrid-parallel LBANN training.

Strong scaling

- 2.28x of speedup on 4 nodes (16 GPUs) over one node with a mini-batch size of 1
  - The main bottleneck is I/O of excessive data size (1 GiB/sample) via PCIe and inter-node data shuffle



(a) Strong scaling with two different mini-batch sizes (b) Time breakdown with $N = 1$, $W = 512$

Figure 2. Strong scaling of the CosmoFlow network. $\{D_p, H_p\}$ represents the depth and the height dimensions are distributed among D_p and H_p process groups respectively.

Weak scaling

- 119x and 120x of speedup on 128 nodes from 1 node with the $W = 128$ cubes and $W = 256$ cubes datasets respectively
- 111x of speedup over 1 node with $W = 512$ by exploiting hybrid-parallelism even if layer-wise communication is introduced
  - 1.19x of speedup by increasing the number of nodes from 64 to 128 with $N = 64$

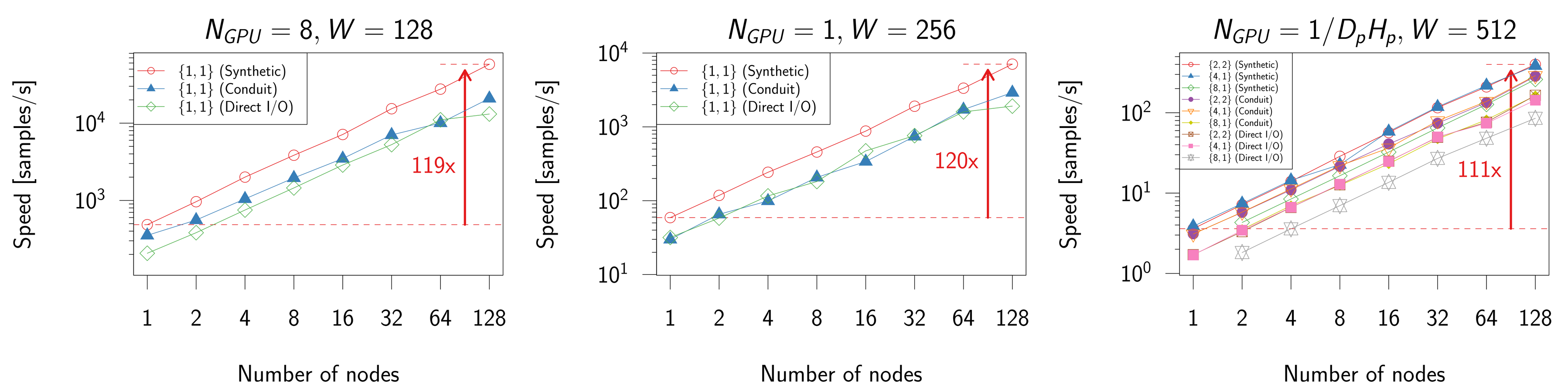


Figure 3. Weak scaling of the CosmoFlow network. N_{GPU} represents the number of data samples per GPU.

Detailed analysis of the training timeline

- By distributing $N = 64$ from 64 nodes to 128 nodes,
  - the computation efficiency per sample is degraded as the batch size per GPU is halved
  - due to this computational inefficiency, part of all-reduce cannot be hidden in the main stream
- The overhead introduced by Distconv ("DC", blue) is nearly negligible than the computational kernels ("Main", red)

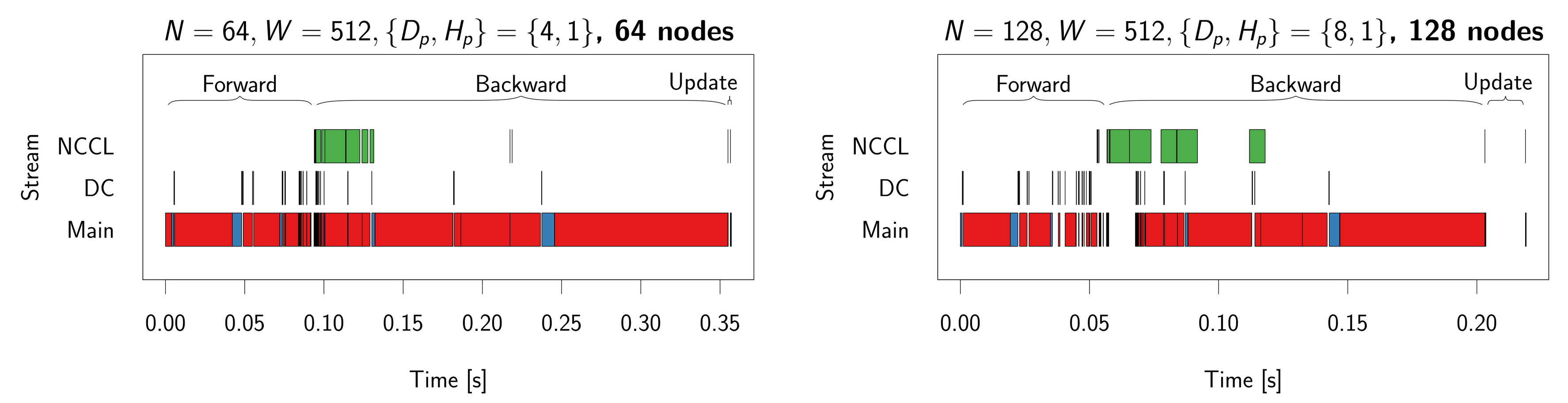


Figure 4. The GPU timeline of a single training iteration.

Acknowledgement

This research was supported by JSPS KAKENHI Grant Number JP18J22858, Japan. This research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy's Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation's exascale computing imperative. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231. LLNL-POST-776501.

References

[1] A. Mathuriya, D. Bard, P. Mendygral, L. Meadows, J. Arnemann, L. Shao, S. He, T. Kärnä, D. Moise, S. J. Pennycook, K. Maschhoff, J. Sewall, N. Kumar, S. Ho, M. F. Ringenburg, Prabhat, and V. Lee, "Cosmoflow: Using deep learning to learn the universe at scale," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis*, ser. SC '18. Piscataway, NJ, USA: IEEE Press, 2018, pp. 65:1–65:11. [Online]. Available: http://dl.acm.org/citation.cfm?id=3291656.3291743

[2] National Energy Research Scientific Computing Center, "Cosmoflow datasets," 2019. [Online]. Available: https://portal.nersc.gov/project/m3363

[3] N. Dryden, N. Maruyama, T. Benson, T. Moon, M. Snir, and B. Van Essen, "Improving strong-scaling of cnn training by exploiting finer-grained parallelism," in *Proceedings of the International Parallel and Distributed Processing Symposium*, ser. IPDPS '19, 2019.

[4] Lawrence Livermore National Laboratory, "Conduit," 2019. [Online]. Available: https://github.com/LLNL/conduit