Efficient Sample Exchange for Large-Scale Distributed Deep Learning with Local Sampling

Truong Thao Nguyen, Yusuke Tanimura
National Institute of Advanced Industrial Science and Technology (AIST),Tokyo, Japan

Research Goal (What?)

Training Deep Learning models on a large-scale computer system is becoming a commonplace. Driven by the increase in complexity and size of models (**trillions of parameters**) and dataset size (**billions of data points**), training models are becoming longer and more costly. In this work, we target **speeding up the training phase** of Large-Scale Deep Learning on the GPU-cluster **while maintaining accuracy**.

Research Approaches (How?)

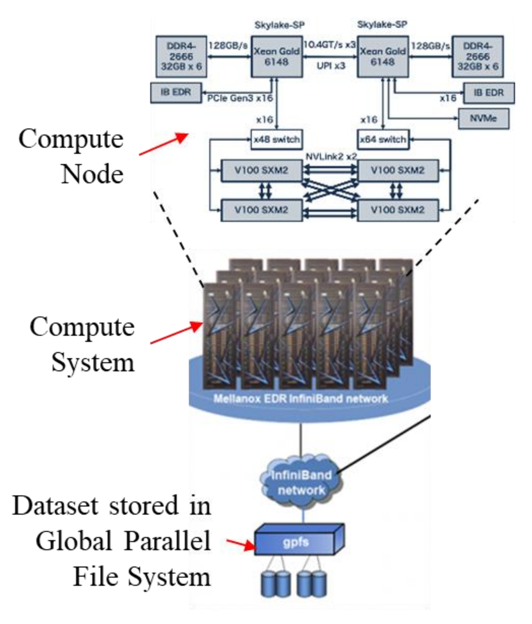
Approaches	Training Stack				Training Phases				
	Training Algorithm	Parallel Strategy	Communication Algorithm	Computer System	IO	FW	BW	GE	WU
Synchronous/Async SGD	●	●	○	○	-	-	-	✓	✓
2^{nd} order method	●	○	○	○	-	-	-	✓	✓
Data Prefetching/Sampling [1]	●	○	○	●	✓	-	-	-	-
Model prunning	●	○	○	○	-	✓	✓	✓	✓
Hiding training samples [5]	●	○	○	○	✓	✓	✓	✓	✓
Data vs. model parallelism [2]	○	●	○	○	✓	✓	✓	✓	-
Centralized vs.decentralized	○	●	●	○	✓	-	-	✓	-
Allreduce Algorithm [3]	○	○	●	●	-	-	-	✓	-
Data Compression [4]	○	●	●	○	-	-	-	✓	-

This work

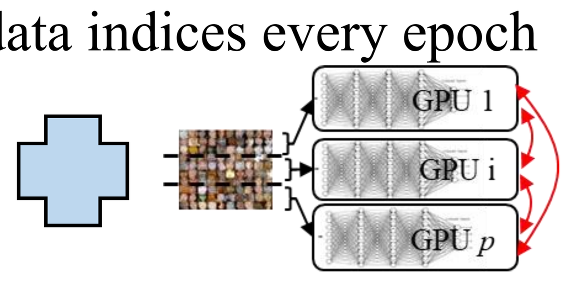
Background

1. Data Parallelism becomes practical

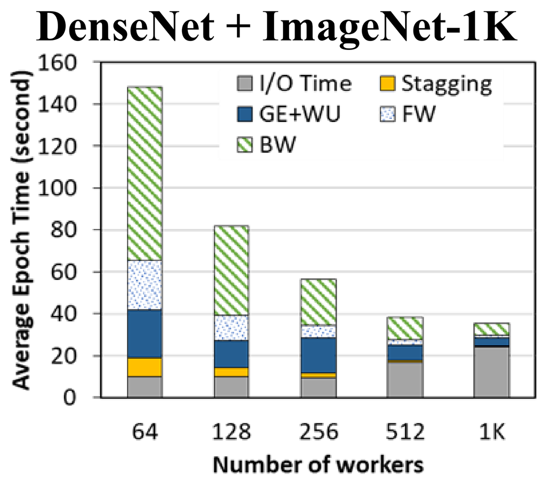
- Selects *b* samples ***randomly*** each iteration
- By shuffling the data indices every epoch



Supercomputer/HPC System



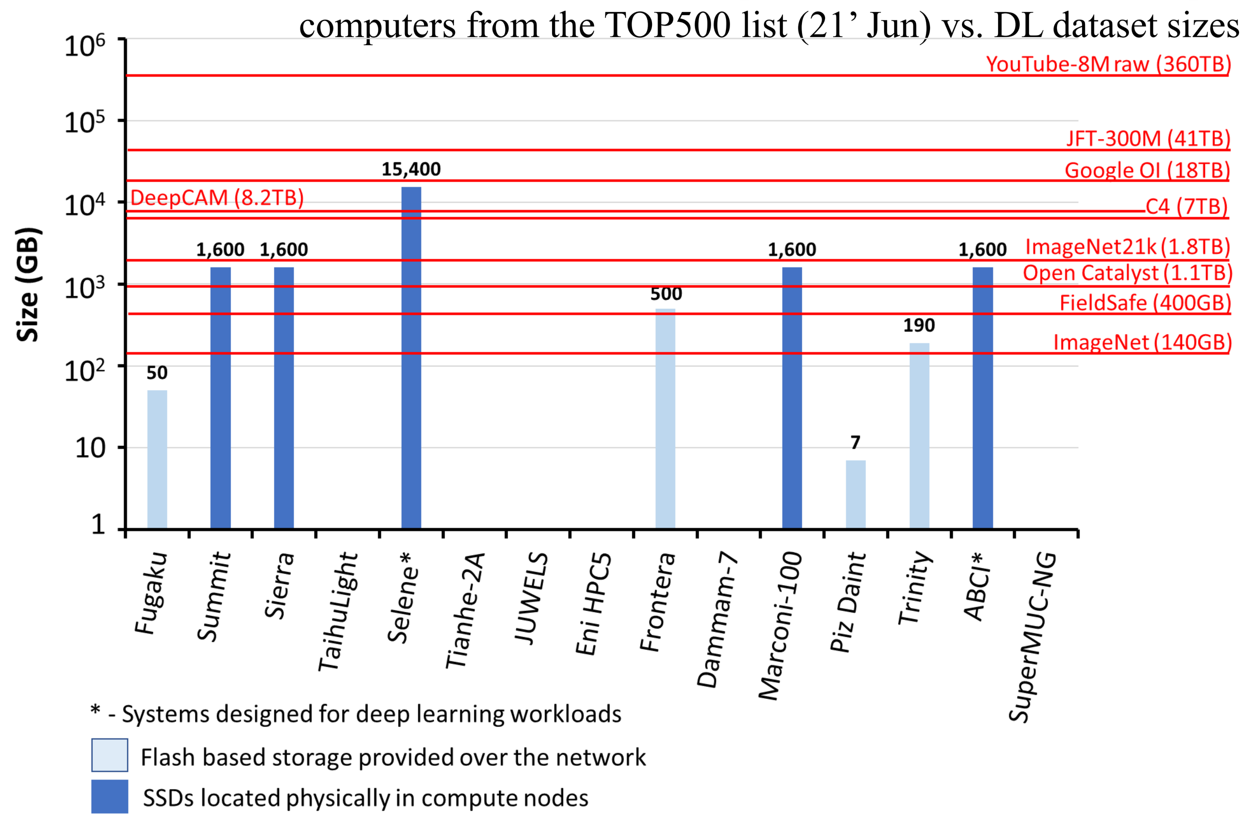
DenseNet + ImageNet-1K



2. I/O for large-scale training becomes bottleneck

- **Global shuffling**
  - Each worker can access all samples
  - Using Global File System:
    - I/O cost is sensitive to the network
  - Using Local Storage, e.g., SSDs
    - 70% runtime reduction
    - Replication of dataset to local SSDs
    - Only if the entire dataset fits
- **Local shuffling if dataset is too large**
  - Split dataset among workers
  - Sampling samples from local dataset
  - Accuracy reduction

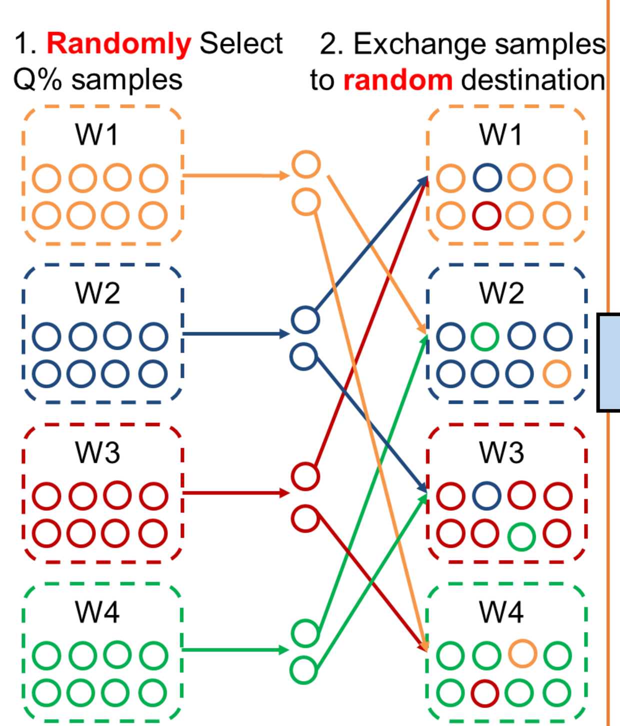
Dedicated node local storage on the fifteen fastest super-computers from the TOP500 list (21' Jun) vs. DL dataset sizes



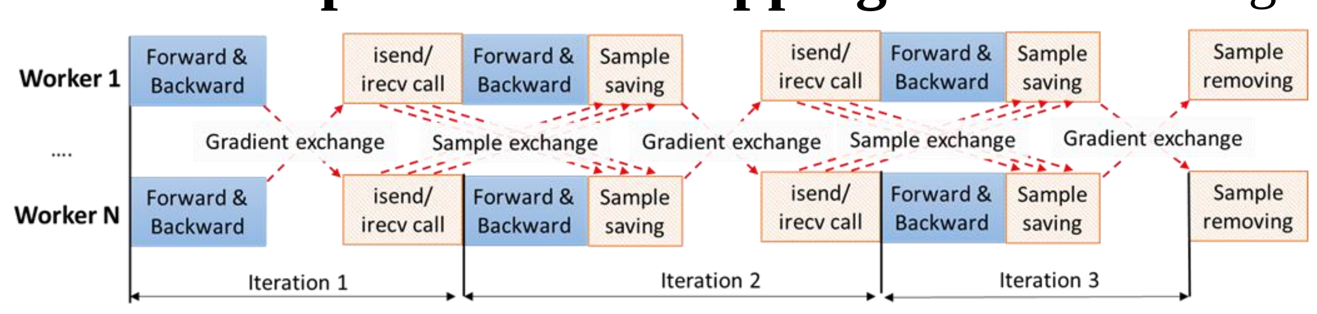
Partial Local Shuffling (PLS) [1]

1. Random exchanging samples

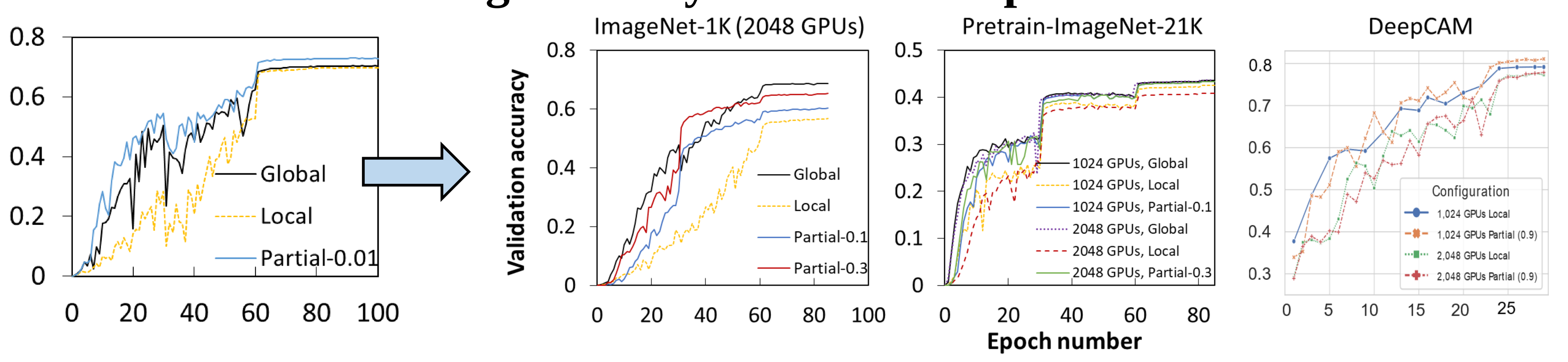
- Split dataset among workers **once**
- Sampling samples from local dataset
- **Workers exchange a Q% of local samples every epoch**
- Q = 0% is local shuffling, Q = 100% is global shuffling



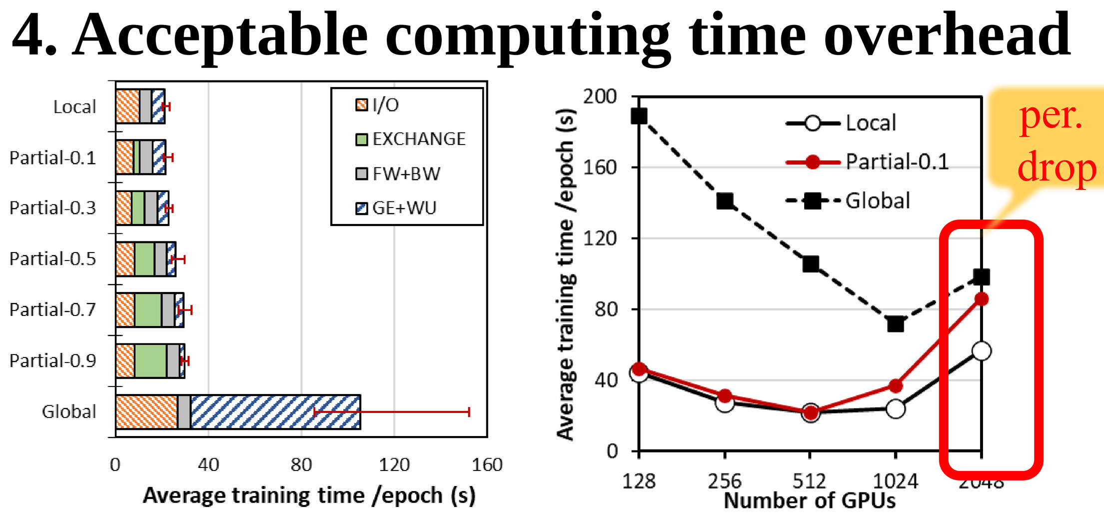
2. IO & Computation overlapping w/ non-blocking MPI



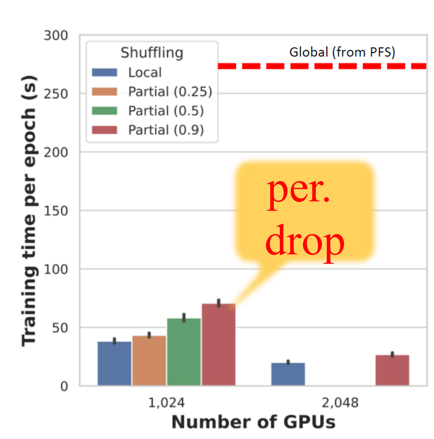
3. Maintain the training accuracy while stores up to 0.03% dataset



4. Acceptable computing time overhead



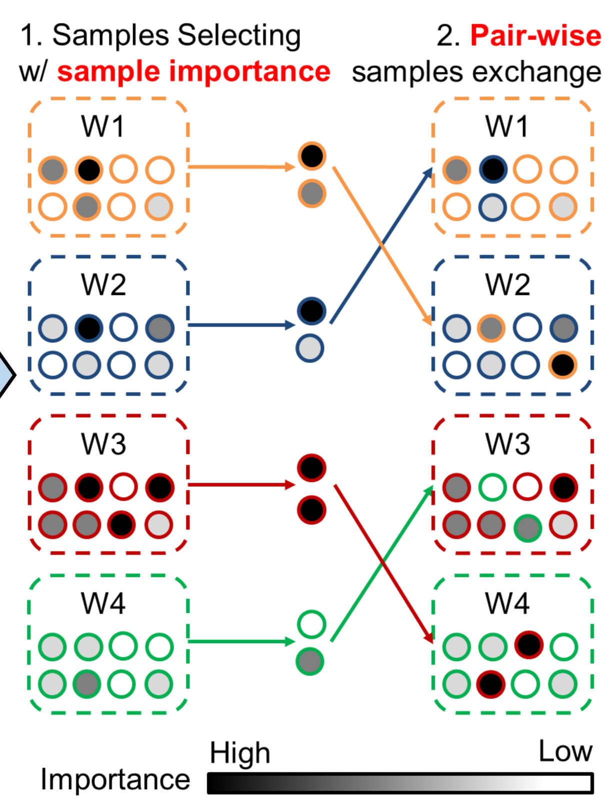
Imagenet-1K, Resnet50 on ABCI



DeepCAM on ABCI

Global shuffling infeasible, estimated from PFS performance

Proposed Exchange Scheme (this work)



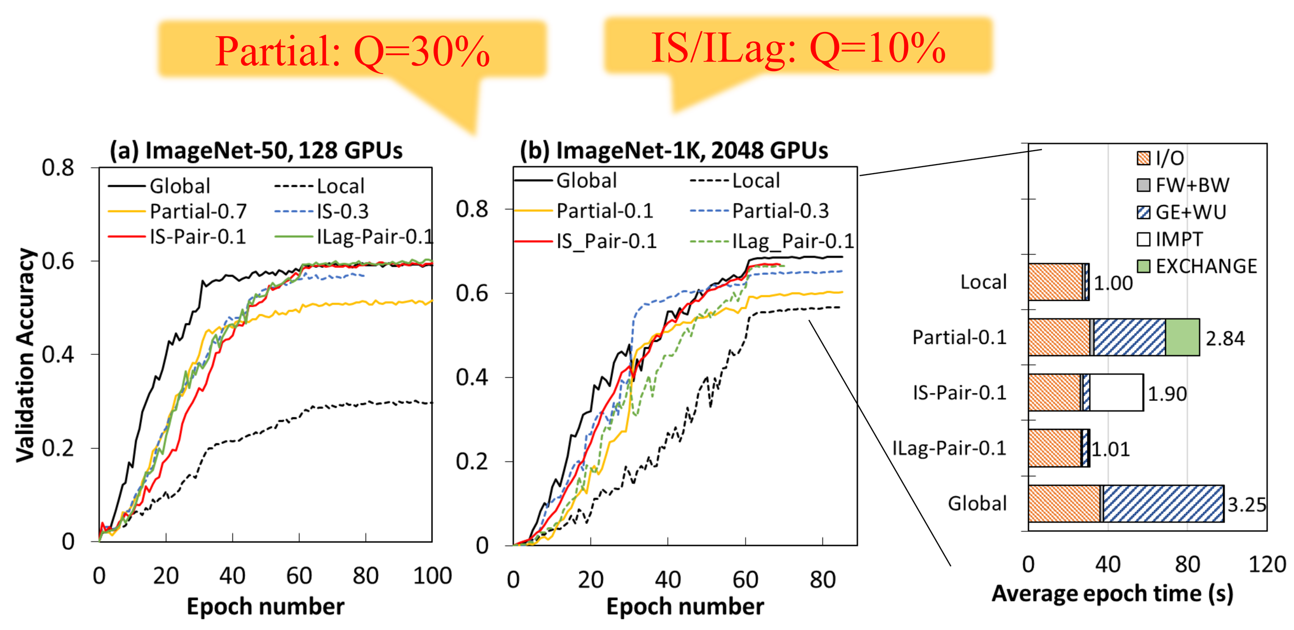
1. Sample selecting

- $p_i = \frac{I_i^t}{\sum I_i^t}$ for sample *i*
- I_i^t : importance at epoch t of sample *i*
  - loss in epoch *t*-1

2. Pair-wise manner

Workers with most important samples – less important samples

3. IS achieve the same accuracy while exchanging a smaller number of samples



4. Better performance

- Pair-wise scheme reduces Exchange overhead
- Lagging loss reduces the computation overhead

Conclusion

- Local sampling + sample exchanging reduces I/O time while maintain accuracy
- Exchanging using importance of samples improves performance

References

[1] Truong Thao Nguyen, et. al. "*Why Globally Re-shuffle? Revisiting Data Shuffling in Large Scale Deep Learning*", IEEE IPDPS 2022.
[2] Kahira, Albert Njoroge, Truong Thao Nguyen, et. al. "*An oracle for guiding large-scale model/hybrid parallel training of CNN.*", IEEE HPDC2021.
[3] Truong Thao Nguyen, et. al. "*Efficient MPI-AllReduce for large-scale deep learning on GPU-clusters.*" CCPE 2021.
[4] Truong Thao Nguyen, et. al. "*An allreduce algorithm and network co-design for large-scale training of distributed deep learning*", CCGRID2021.
[5] Truong Thao Nguyen, et. al. "*KAKURENBO: Adaptively Hiding Samples in Deep Neural Network Training*", Neurips2023.