High Performance IO for Large Scale Deep Learning

Truong Thao Nguyen

National Institute of Advanced Industrial Science and Technology (AIST), Japan

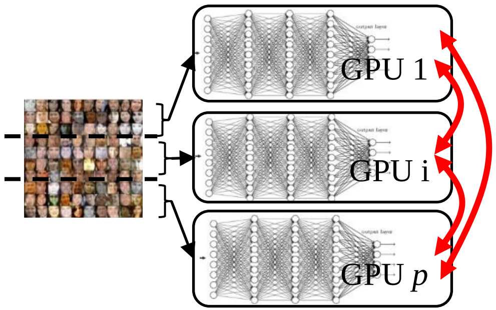
National Institute of Advanced Industrial Science and Technology
AIST

In collaboration with *Mohamed Wahib (AIST), Balazs (Bali) Gerofi, Emil Vatai, Alex Drozd, Jens Domke (RIKEN), Francois Trahay (Telecom SudParis), Jianwei Liao (South China Univ.)*

ABCI グランドチャレンジ成果報告会, January 2022

Motivation

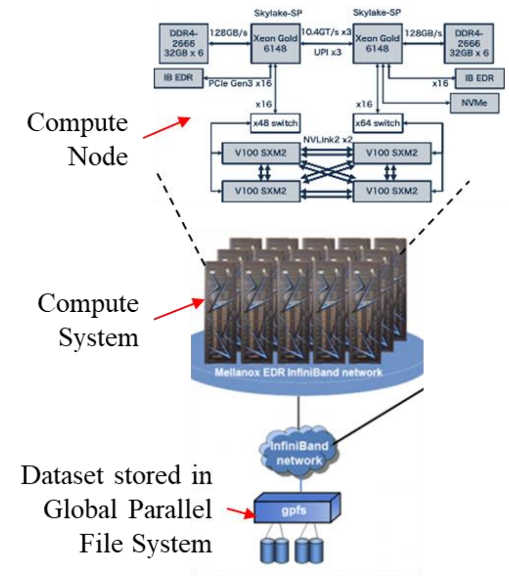
- Explosion of Deep Learning (DL)
  - Effectiveness in a variety of applications
- Long training time limits the development of new applications
  - Training GPT-3 model (<u>355 years</u> on a V100 GPU, <u>cost $4.6M)</u>
- Parallel training on HPC systems, e.g., ABCI
  - Input samples are accessed in a random fashion
  - Enormous pressure on the I/O subsystem



➔ Our target: **large-scale** training, e.g., <u>100s-1000s of GPUs</u>

IO for Large-scale Distributed Deep Learning

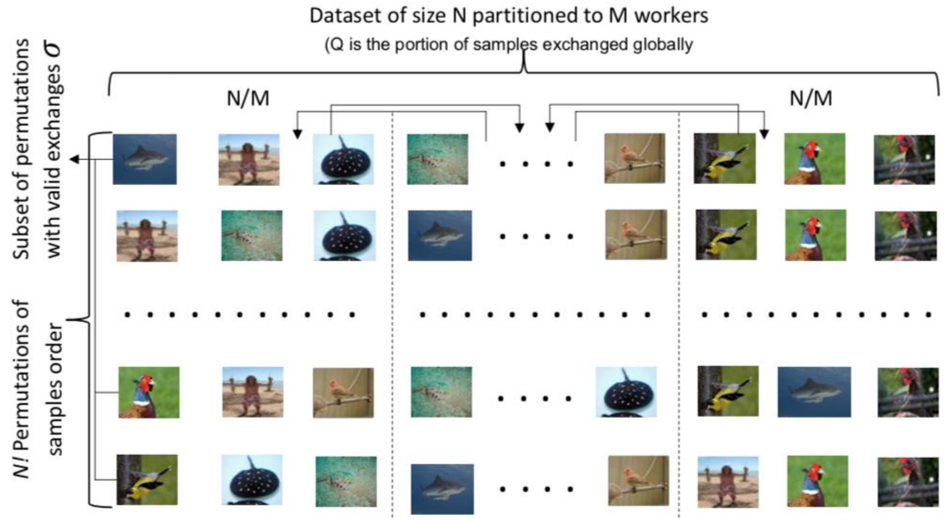
- Global shuffling
  - Using Global File System:
    - I/O time is sensitive to the network status
  - Using Local Storage, e.g., SSD
    - 70% runtime reduction
    - Replication of input to local SSDs
    - *Only if the entire data set fits*
- Local shuffling if dataset is too large
  - Split data set among workers
  - Sample the data locally
  - *Effect on convergence is not well understood/studied*



Supercomputer/HPC System

1. What is the impact of local sampling on accuracy in a spit dataset scenario?
2. Can the access pattern be localized without impacting training accuracy?
3. How to exploit such access strategy to reduce the training time?

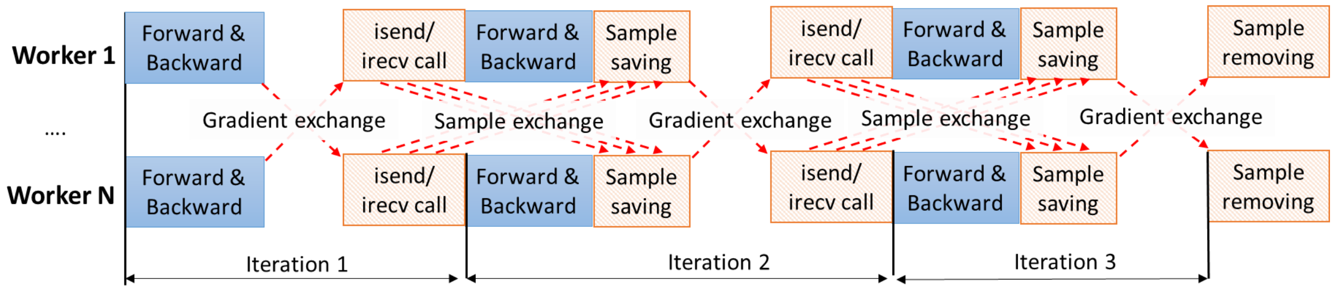
Data Shuffling in Distributed SGD



- **N is the number of samples, M is the number of workers (e.g., GPUs, MPI ranks, etc.)**
- **Three shuffling policies:**
  - Global shuffling: each worker can access all samples
  - Local shuffling: each worker accesses only its local portion of the data set (N/M samples)
  - ***Proposal: Partial Local Shuffling (PLS)***: a Q (0 < Q < 1) portion of the local samples is exchanged with random other nodes
    - Q = 0 is local, Q = 1 is global
    - We are interested in characterizing how small Q we can get away with without impacting convergence rate
    - Tradeoff among local storage capacity, performance (i.e., runtime), impact on training accuracy

PLS: Design and Implementation

- **Samples exchange (IO & Computation overlapping)**
  - Use non-blocking MPI calls (i.e., MPI_Isend/recv())



- **Easy implementation in Pytorch**

Training code with global shuffling

```
train_dataset = ImageFolder(train_dir, transformations)
train_sampler = DistributedSampler(train_dataset, size, rank)
train_loader = DataLoader(train_dataset, batch_size=b, train_sampler)
```

Training code with (Partial) Local Shuffling

```
train_dataset = PLS.ImageFolder(train_dir, class_file, transformations)
train_sampler = DistributedSampler(train_dataset, size, rank=rank)
train_loader = DataLoader(train_dataset, batch_size=b, train_sampler)
scheduler = PLS.Scheduler(train_dataset, batch_size=b, fraction=Q)
...
train(epoch):
    scheduler.scheduling(epoch)
    ...... # Training loop here
    send_req, recv_req = scheduler.communicate() # Non-blocking exchange
    scheduler.synchronize(send_req, recv_req) # Wait to finish exchange
    scheduler.clean_local_storage() # Remove exchanged samples on the storage
```

Evaluation

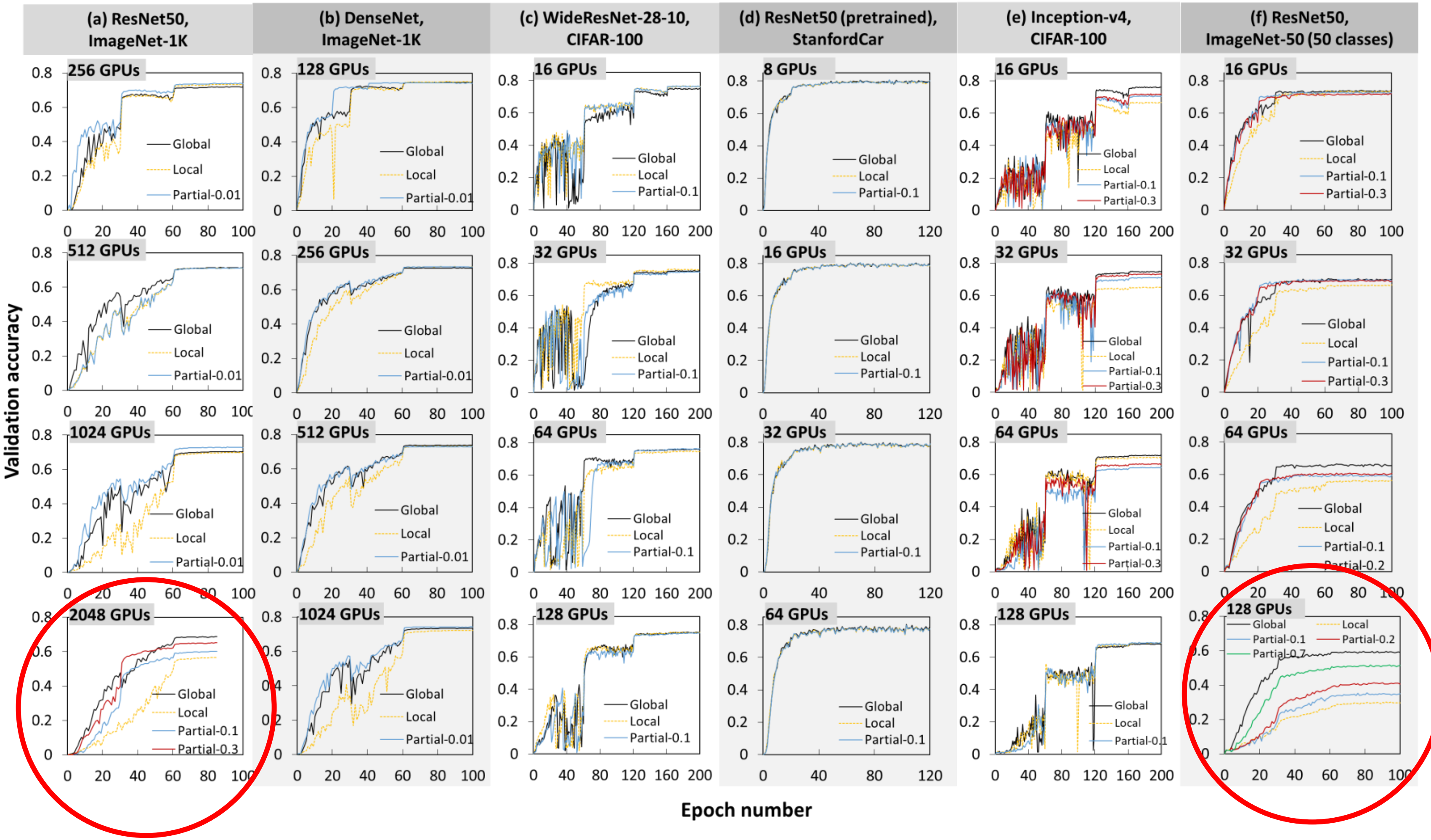
- **Evaluation Platform**: ABCI
  - 1,088 compute nodes (CN)
  - 2 Intel Xeon Gold + 4 NVIDIA V100 per node
  - Infiniband EDR
  - 1.6 TB SSD local per CN
- **Models and Data Sets**

TABLE I: Datasets and Models Used in Experiments (*)Trained on a subset of the original dataset. (**) Use pre-trained model.

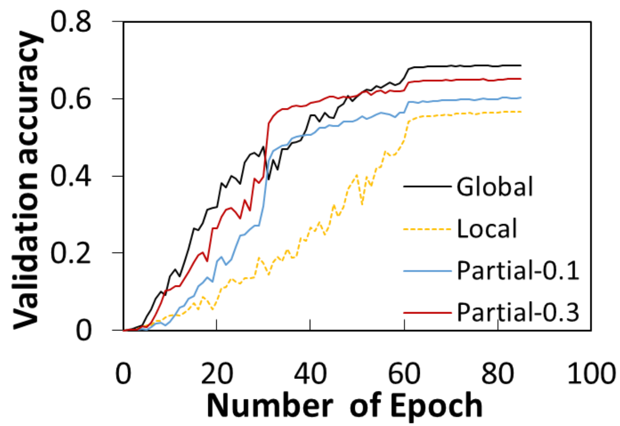
Model	Dataset	#Samples	Size
Resnet50 [26]	ImageNet-1K [9]	1.2M	~ 140GB
Densenet161 [27]			
Resnet50 [26]	ImageNet-50(*) [9]	~65K	~ 2GB
WideResNet-28-10 [28]	CIFAR-100 [29]	50K	~160 MB
Inceptionv4 [30]			
Resnet50 (**) [26]	Standford Cars [31]	8144	~ 934 MB
Resnet50 [26]	ImageNet-21K(*) [9]	~ 9.3M	~1.1 TB
DeepCAM [26]	DeepCAM [1]	~ 122K	~8.2 TB

- **Shuffling policies:**
  - Global shuffling
  - Local shuffling
  - Partial local shuffling (e.g. partial-0.1: exchange 10% of samples)

Local Shuffling Sufficient (when scale is small)



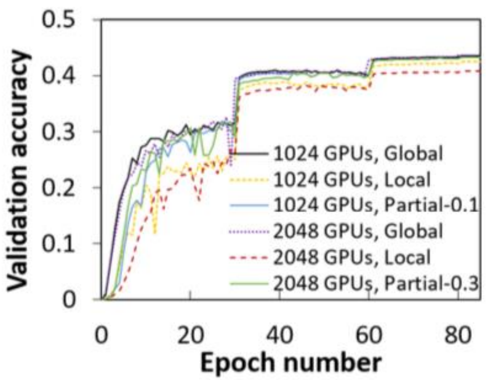
Partial Local Shuffling Improves Accuracy



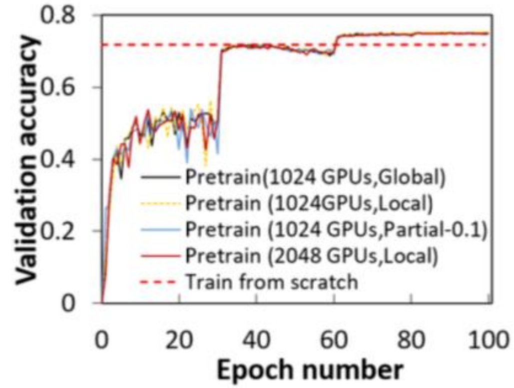
Resnet50 with Imagenet-1K, 2048 GPUs

- **Local shuffling accuracy decreases as the number of workers scales**
  - For 2048 workers, each worker only trains on approximately 600 samples
- **Partial-0.1 local shuffling slightly increases the accuracy and partial-0.3 local shuffling achieves close to the same accuracy as global shuffling**
  - Each of the 2048 workers store only ~**0.06%** of the data set locally
  - *Feasible even without local storage?*

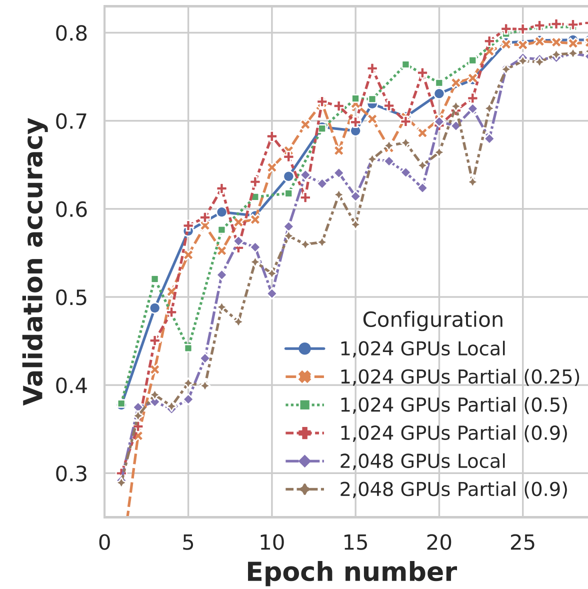
ImageNet-21K and DeepCAM on ABCI



(a) Upstream training.



(b) Downstream training (256 GPUs).



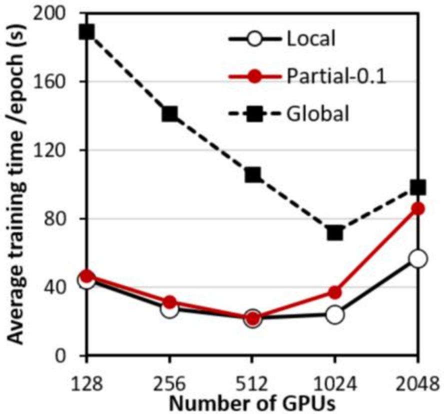
Resnet50 with Imagenet-21K validation accuracy

DeepCAM validation accuracy

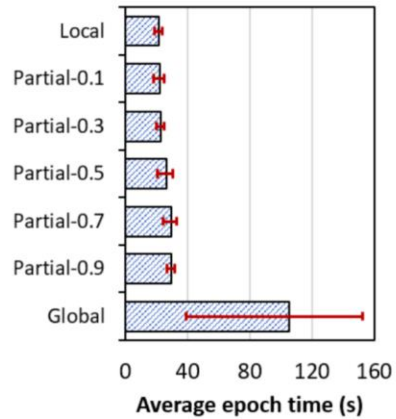
- Upstream training:
  - Significant gap between local vs. global
  - Partial-0.3 provides same accuracy as global
- Downstream:
  - Almost no difference among different configurations

- **On 1K GPUs:** ~2% improvement on accuracy with 0.9 partial
- **On 2K GPUs:** Almost no effect
  - Relatively small number of samples?

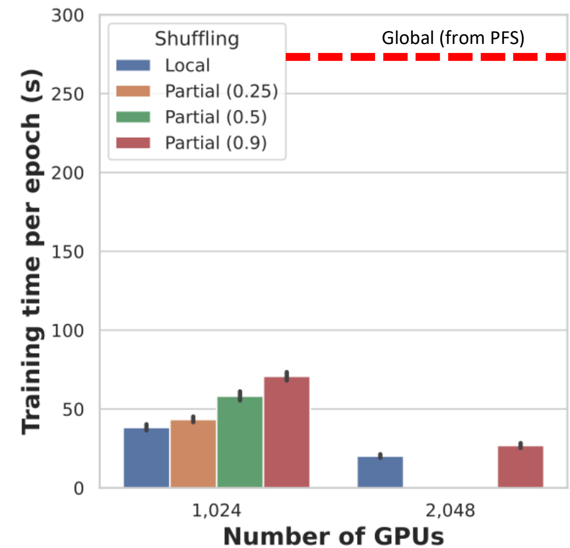
Performance and Scalability of Partial Shuffling



(a) Scalability evaluation



(b) Impact of the exchange rate on performance (512 GPUs)



Resnet50 with Imagenet-1K

DeepCAM

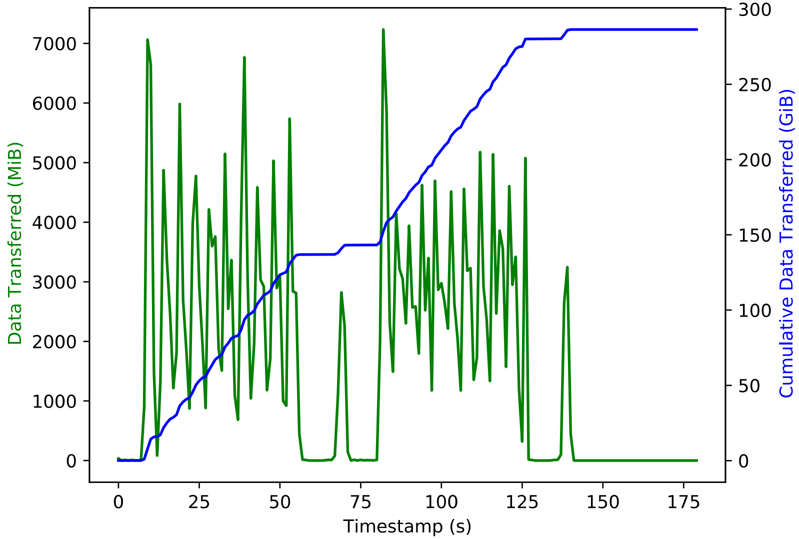
- **Resnet50 on ImageNet1K:**
  - Good scalability for up to 1K GPUs
  - Performance drop on 2K
  - Still significant improvement compared to global out of PFS

- **DeepCAM:**
  - Visible cost of partial scheme compared to local only due to large sample size (~60MB)
  - Global shuffling infeasible, estimated from PFS performance

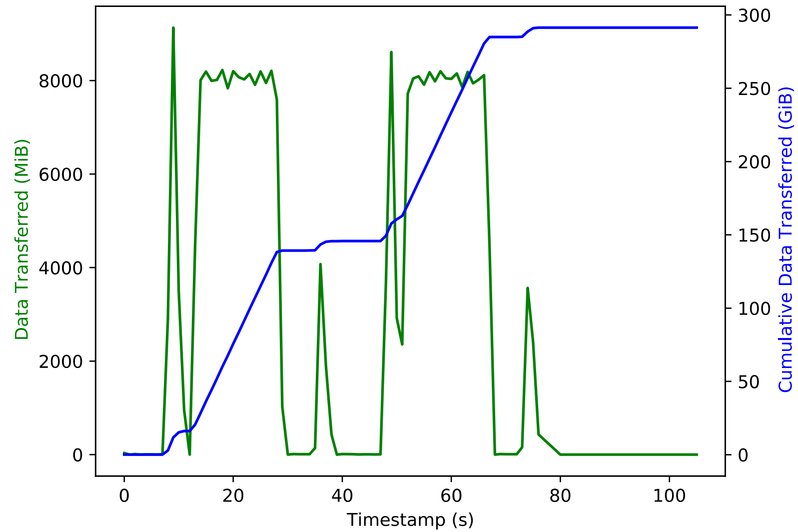
THANK YOU

Backup Slides

I/O cost of global shuffling on Parallel File System vs. SSDs (ABCI)



Parallel filesystem



SSD

- ResNet-50 on ImageNet-1k running on 64 nodes (256 GPUs)
- Darshan profile of two training epochs
- SSDs' sustained aggregate BW much higher than PFS leading to 70% runtime reduction