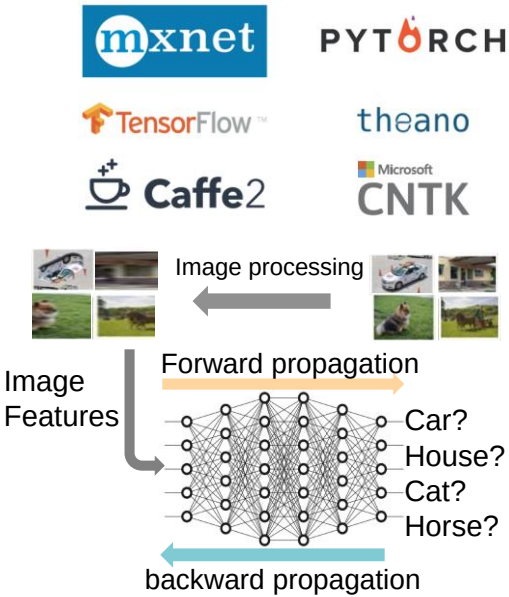
Accelerating Deep Learning Training

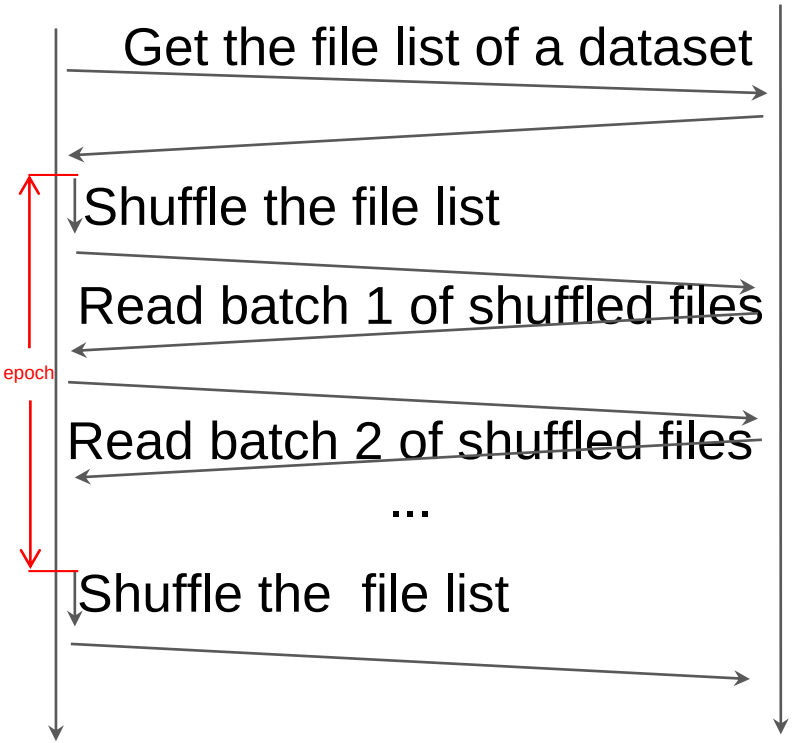
Qiong Luo
luo@cse.ust.hk

Joint work with Lipeng Wang and SenseTime Corp.

File Accesses in Deep Learning Training

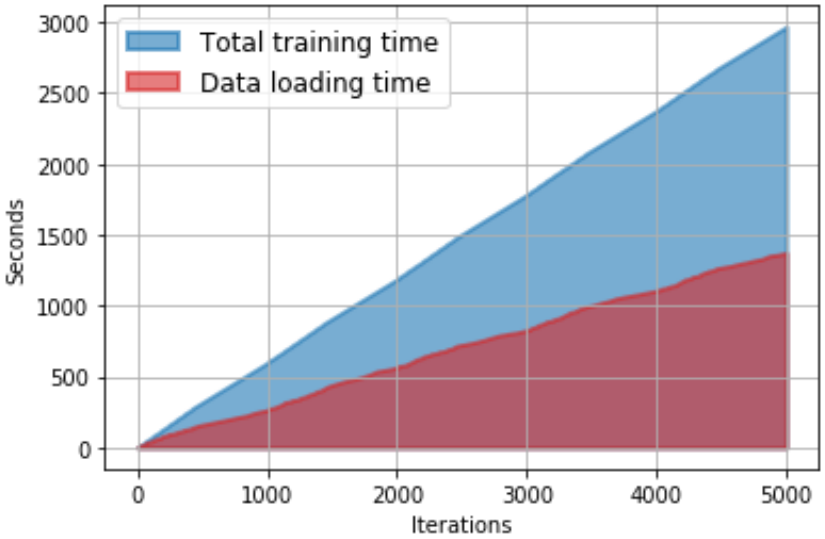


Training Framework

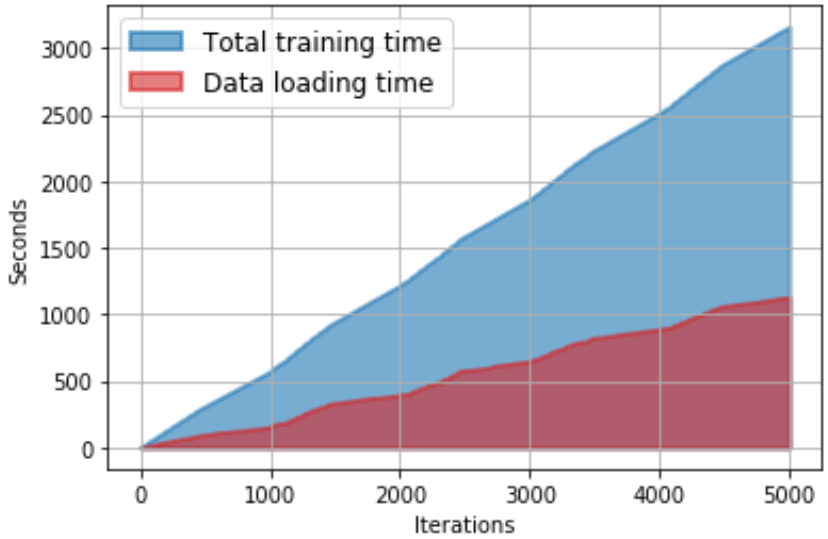


Distributed Storage

IO Time vs. Total Training time

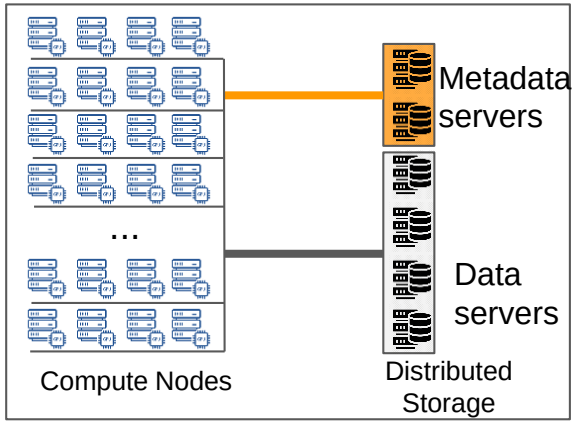


Data loading takes around 1/2 time of the total training of AlexNet on ImagetNet of 1.28 million images

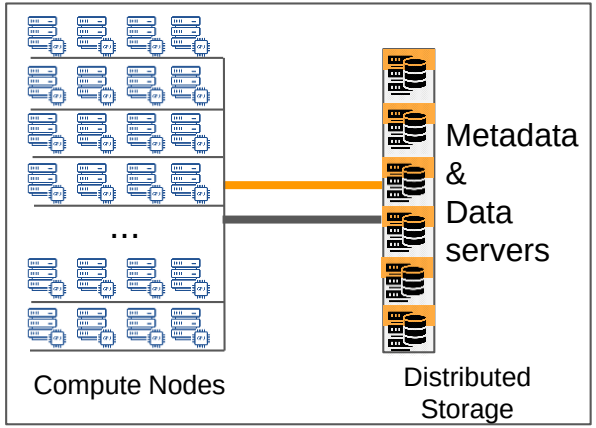


Data loading takes around 1/3 time of the total training of ResNet-50 on ImagetNet of 1.28 million images

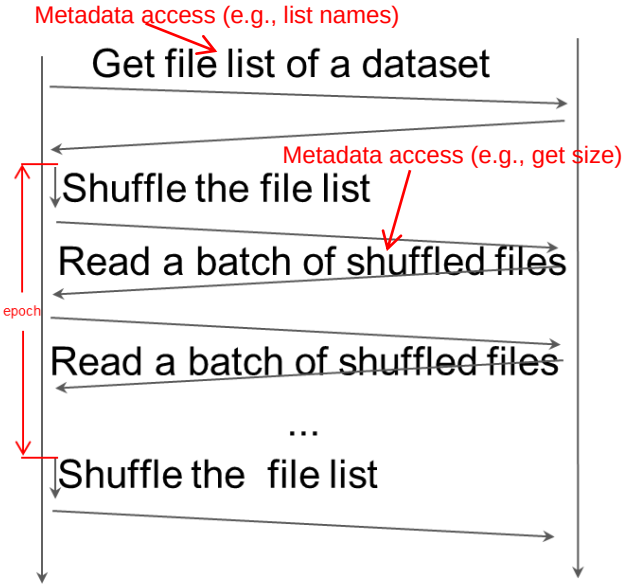
Problems of Existing Systems – Metadata Bottleneck



Separated metadata servers and data servers (e.g., Lustre, GFS)

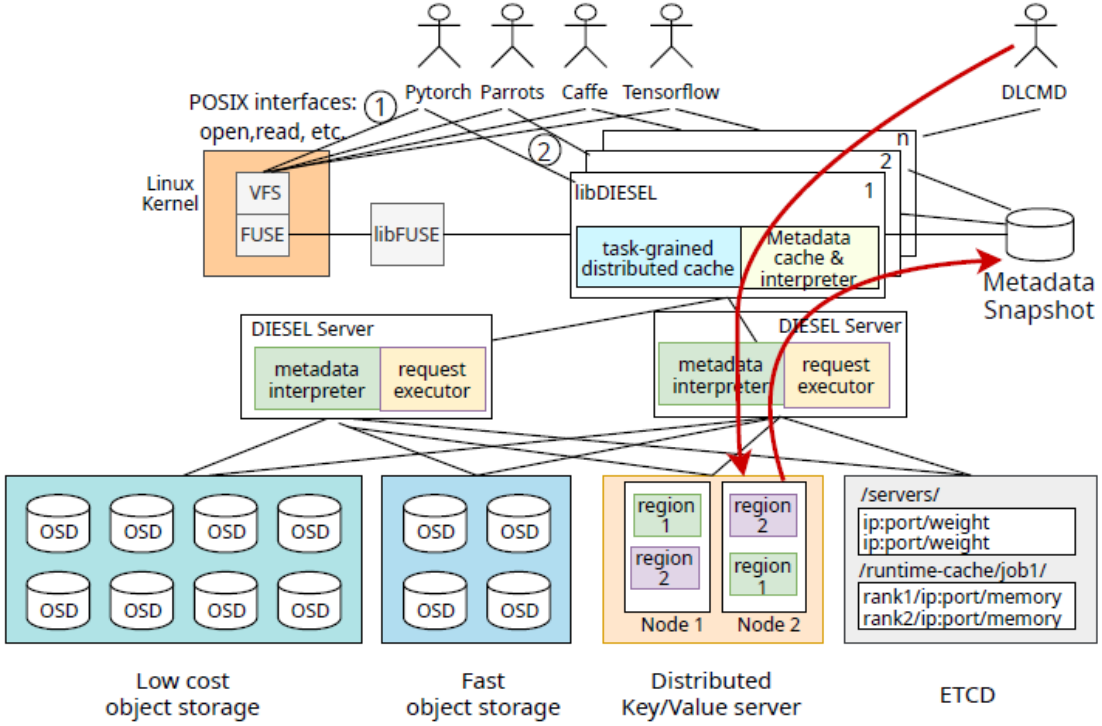


Distribute metadata and data on all storage servers (e.g., Ceph, GlusterFS)



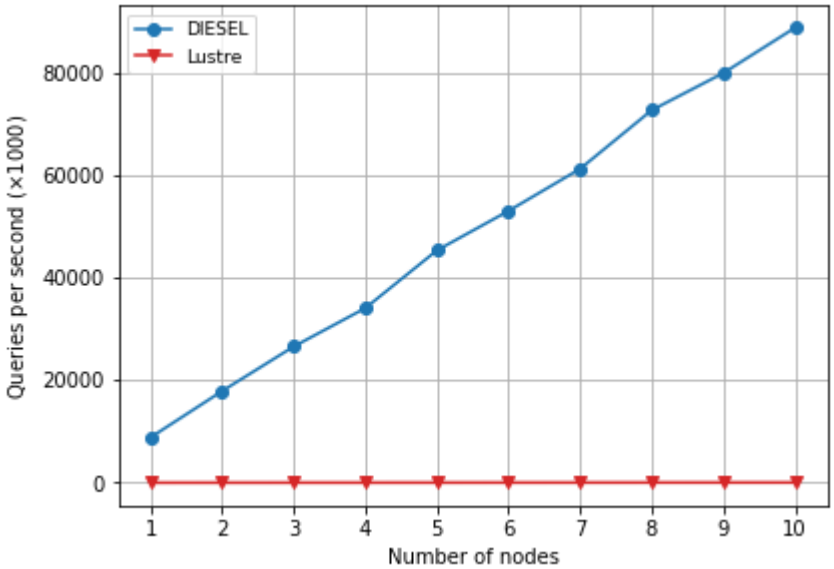
- Dataset contains lots of small files (e.g., ImageNet has 1.28 million files with average size 110KB)
- Existing storage systems have poor scalability on metadata access

Our Solution 1: Metadata Snapshot



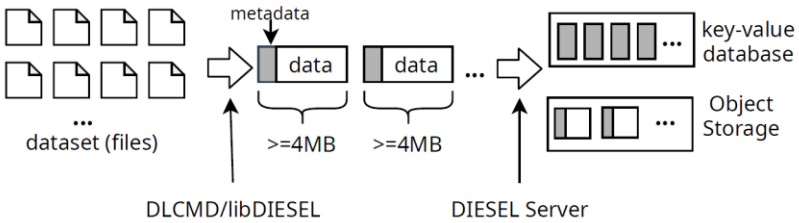
Since dataset is not changed during training, we can download the metadata snapshot of the training dataset to compute nodes. Therefore, metadata access are served locally. The metadata bottleneck on storage servers is removed.

Evaluation Results of Metadata Snapshot

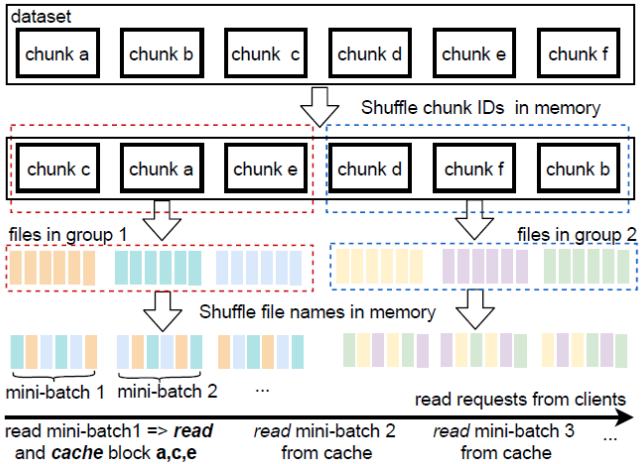


Our system achieves linear scalability on metadata access.

Our Solution 2: Chunk-Based File Organization

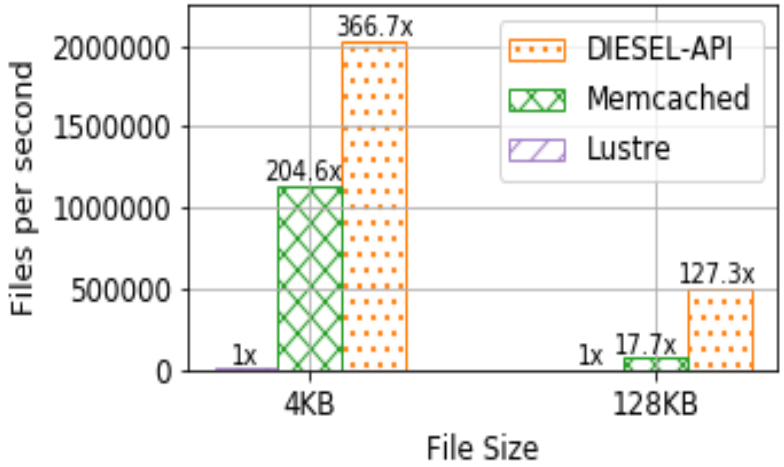


Merge small files to large data chunks



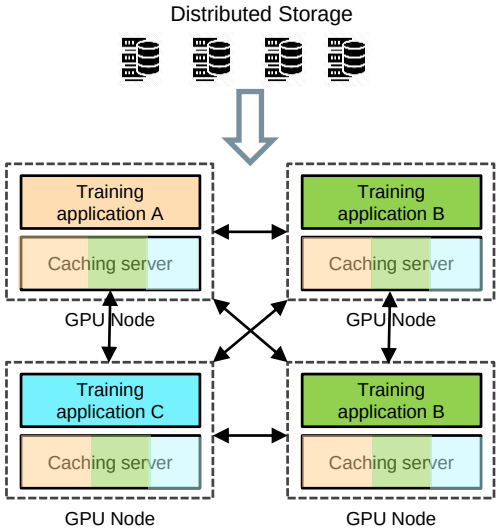
Chunk-wise shuffle converts small file reads to large chunk reads to utilize the maximum read bandwidth of underlying storage system.

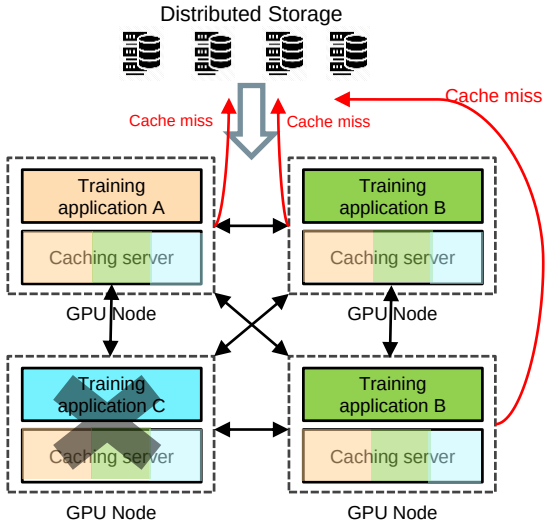
Evaluation Results of Chunk-wise Shuffle Method

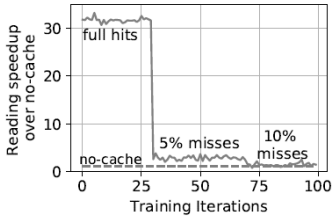


With Chunk-wise shuffle our system is hundreds of times faster than the Lustre file system.

Problems of Existing Systems – Global caching

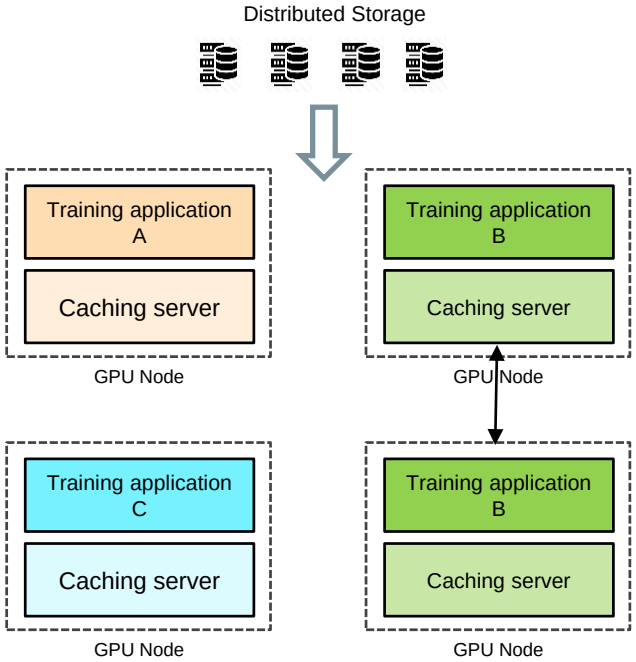






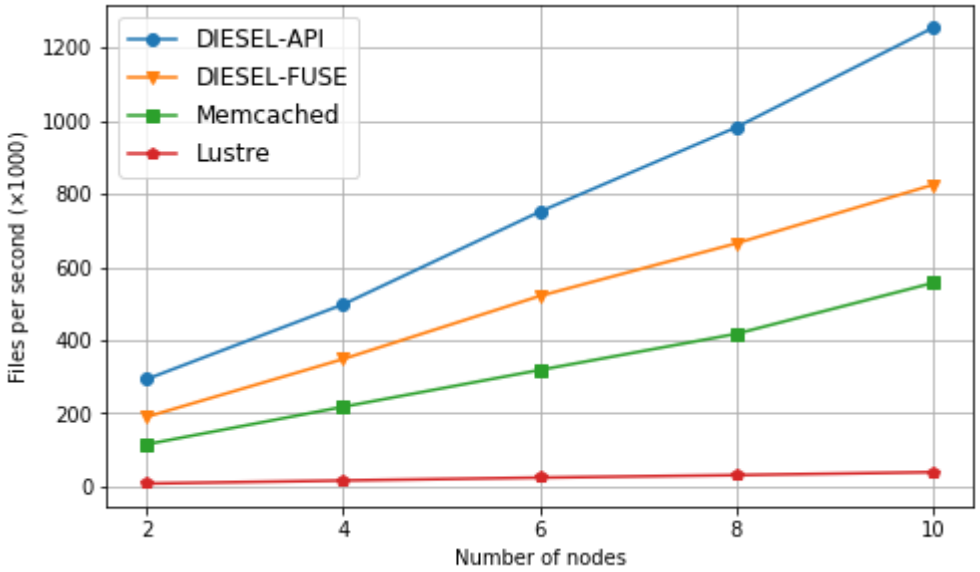
- Reading speed drops a lot with cache misses
- Large dataset cannot be fully cached in the global caching system

Our Solution 3: Task-Grained Distributed Cache



The training dataset is cached along with the training task. This way node failures are contained within each training task.

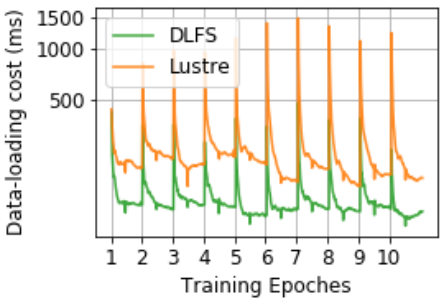
Evaluation Results on Distributed Cache



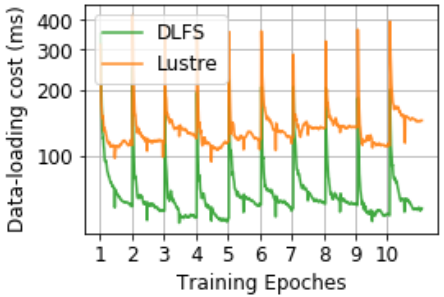
The task-grained distributed cache outperforms existing caching and storage systems.

IO Time

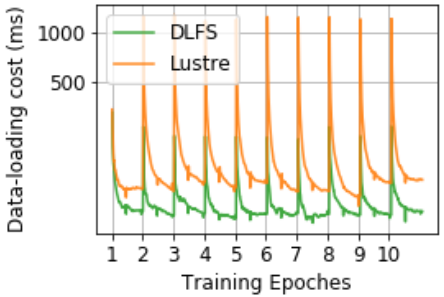
Our system halves the data loading time of an existing storage system:



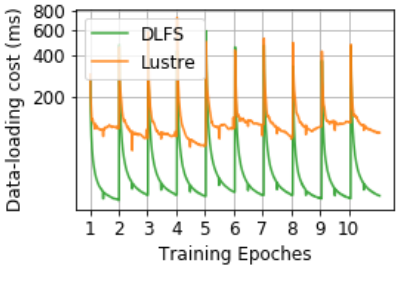
AlexNet on ImageNet



ResNet-18 on ImageNet

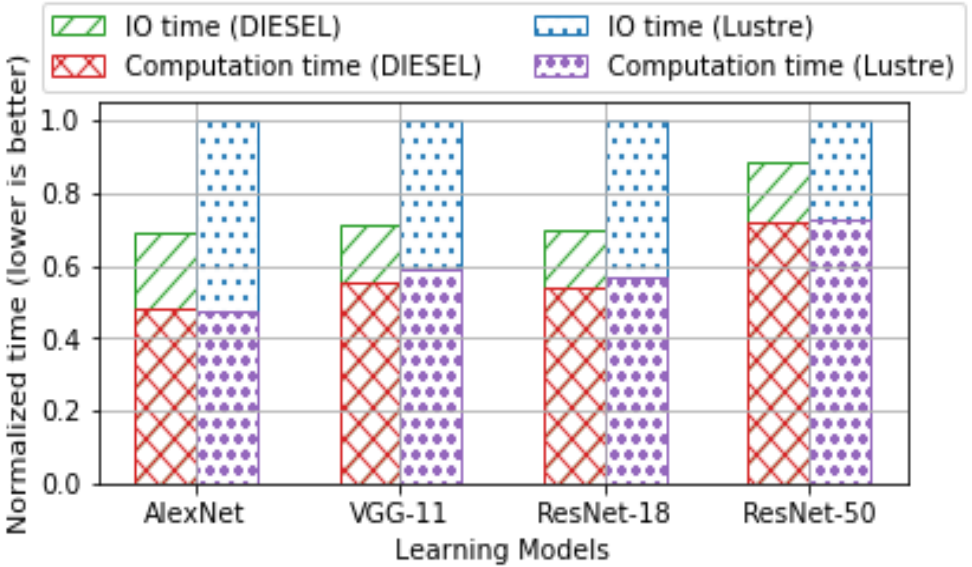


ResNet-50 on ImageNet



VGG-11 on ImageNet

End-to-End Training Time



Our system reduced the IO time by 51-58% and the total time by 15-27%.
The reduction in training time is about 8-9 hours: from 37-66 hours to 29-57 hours.

Summary

We design and implement a custom storage and caching system for deep learning training.

- To remove the bottleneck on metadata access, we create metadata snapshots for datasets
- To improve repeated shuffled accesses to small files, we introduce chunk-based storage and apply chunk-wise shuffle
- To reduce the impact of node failures, we develop task-grained distributed caches to contains node failure within each training task

Simple Ideas, Efficient Implementations, Ease of Use

Q & A

Qiong Luo
Accelerating Deep Learning Training
luo@cse.ust.hk

Meeting ID: 911 1031 3912
Password: RTF2020