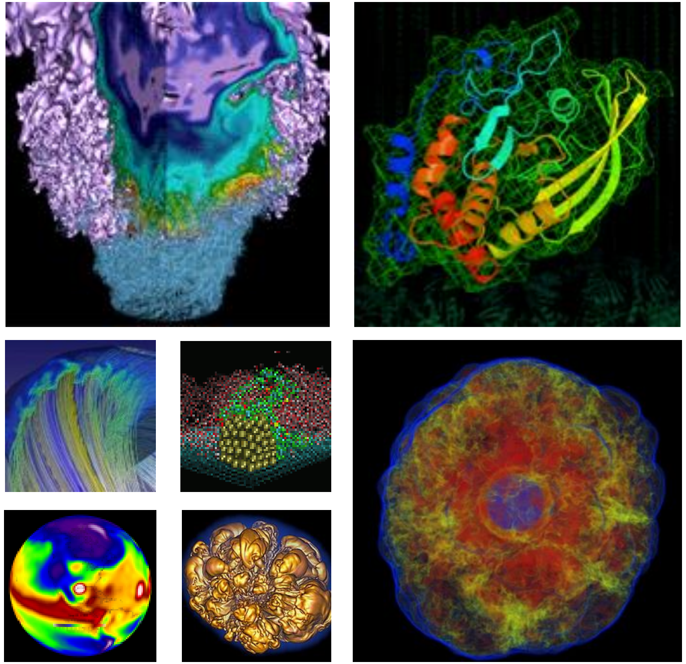
Initial Characterization of I/O in Large-Scale Deep Learning Applications



November 12, 2018

Fahim Chowdhury, Jialin Liu, Quincey Koziol, Thorsten Kurth, Steven Farrell, Suren Byna, Prabhat, and Weikuan Yu

Outline

- Objectives
- DL Benchmarks at NERSC
- Profiling Approaches
- Experimental Results
- Future Work

Outline

- **Objectives**
- DL Benchmarks at NERSC
- Profiling Approaches
- Experimental Results
- Future Work

Objectives

- Deep Learning (DL) applications demand large-scale computing facilities.
- DL applications require efficient I/O support in the data processing pipeline to accelerate the training phase.
- The goals of this project are
  - Exploring I/O patterns invoked through multiple DL applications running on HPC systems
  - Addressing possible bottlenecks caused by I/O in the training phase
  - Developing optimization strategies to overcome the possible I/O bottlenecks

Objectives

- Deep Learning (DL) applications demand large-scale computing facilities.
- DL applications require efficient I/O support in the data processing pipeline to accelerate the training phase.
- The goals of this project are
  - **Exploring I/O patterns invoked through multiple DL applications running on HPC systems**
  - Addressing possible bottlenecks caused by I/O in the training phase
  - Developing optimization strategies to overcome the possible I/O bottlenecks

Outline

- Objectives
- **DL Benchmarks at NERSC**
- Profiling Approaches
- Experimental Results
- Future Work

HEPCNNB Overview

- **High Energy Physics Deep Learning Convolutional Neural Network Benchmark (HEPCNNB)**
  - Runs on distributed TensorFlow using Horovod
  - Can generate particle events that can be described by standard model physics and particle events with R-parity violating Supersymmetry
  - Uses a 496 GB dataset of 2048 HDF5 files representing particle collisions generated by a fast Monte-Carlo generator named Delphes at CERN

CDB Overview

- **Climate Data Benchmark (CDB)**
  - Runs on distributed TensorFlow using Horovod
  - Can act as an image recognition model to detect patterns for extreme weather
  - Uses a 3.5 TB dataset of 62738 HDF5 images representing climate data
  - Leverages TensorFlow Dataset API and python's multiprocessing package for input pipelining

Outline

- Objectives
- DL Benchmarks at NERSC
- **Profiling Approaches**
- Experimental Results
- Future Work

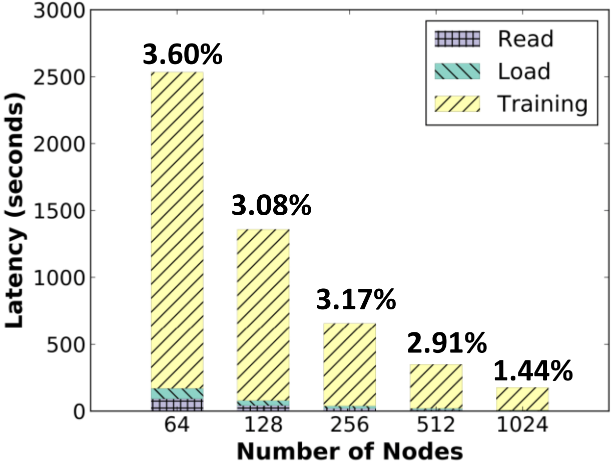
Profiling Approaches

- Develop *TimeLogger* tool based on python to profile application layer
- Determine the total latency from merged interval list for each training component
- Explore TensorFlow Runtime Tracing Metadata Visualization (TRTMV) tool developed at Google and extract I/O specific metadata
- Working on integration of runtime metadata from application and framework layer
- Work available in: https://github.com/NERSC/DL-Parallel-IO

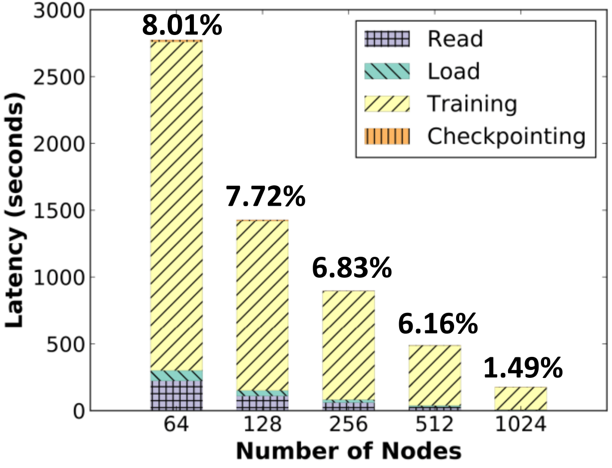
Outline

- Objectives
- DL Benchmarks at NERSC
- Profiling Approaches
- **Experimental Results**
- Future Work

HEPCNNB Latency Breakdown



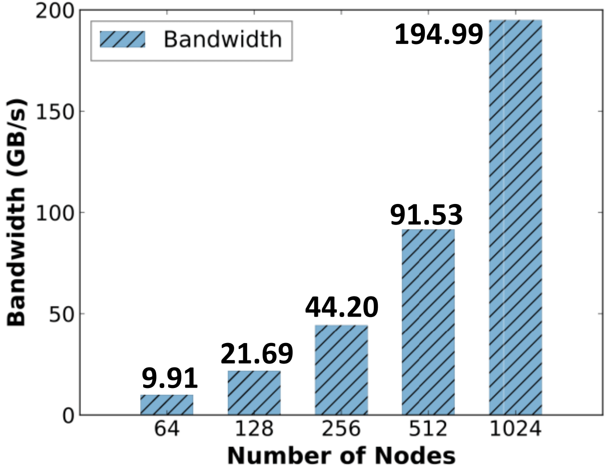
Local Shuffle



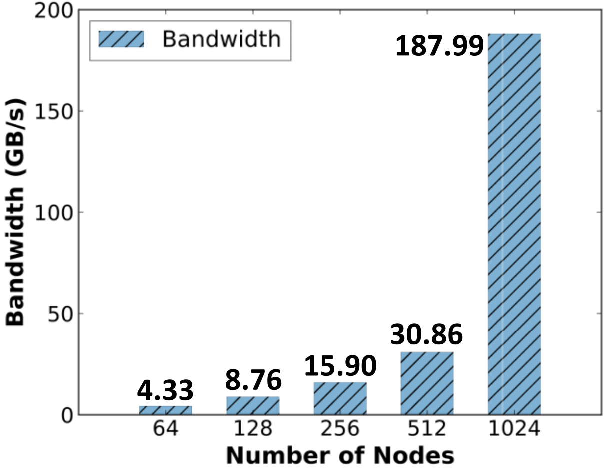
Global Shuffle

- I/O takes more time when Global Shuffling is introduced
- Global Shuffling affects I/O even for small dataset and only 5 epochs training
- I/O bottleneck can become more severe with increasing epochs

HEPCNNB Read Bandwidth



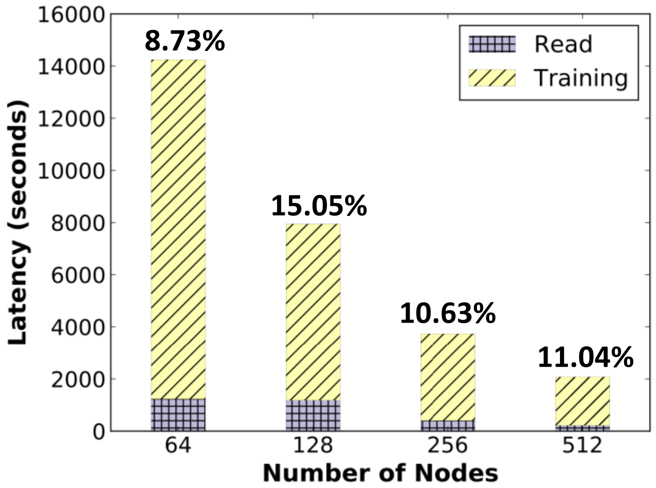
Local Shuffle

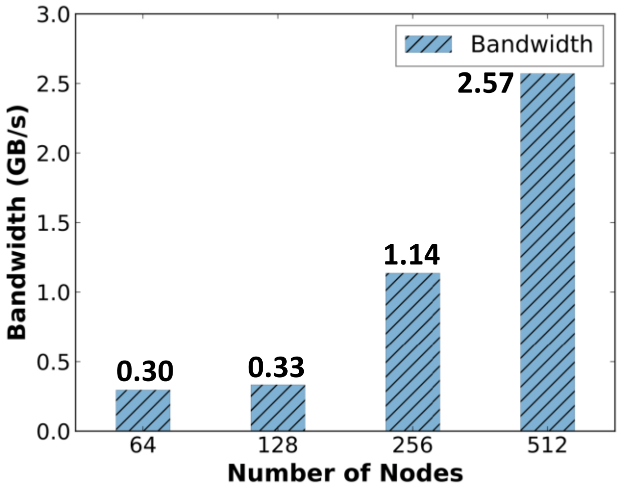


Global Shuffle

- I/O takes more time when Global Shuffling is introduced
- Global Shuffling affects I/O even for small dataset and only 5 epochs training
- I/O bottleneck can become more severe with increasing epochs

CDB Latency and Read Bandwidth





- The percentage of I/O in the training process is more when dataset is larger
- The I/O percentage increases with the number of nodes
- Training benefits more from the scaling than I/O

Outline

- Objectives
- DL Benchmarks at NERSC
- Profiling Approaches
- Experimental Results
- **Future Work**

Future Work

- To integrate *TRTMV* results with *TimeLogger* data for better profiling of highly parallelized I/O pipeline
- To explore the I/O patterns and determine possible I/O bottlenecks in distributed TensorFlow
- To develop an optimized cross-framework I/O strategy to overcome the possible I/O bottlenecks

Thank You