Argonne NATIONAL LABORATORY

ILLINOIS INSTITUTE OF TECHNOLOGY

Deep Learning I/O: Benchmark and Profiling

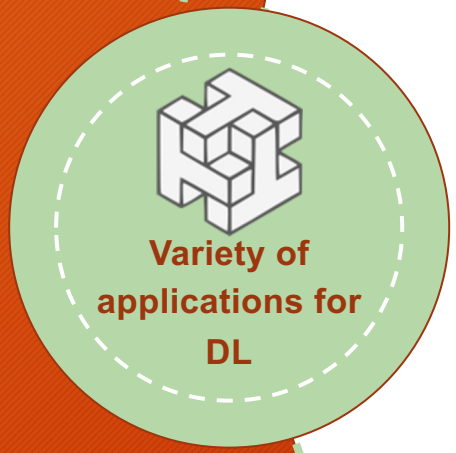
Hariharan Devarajan (hariharandev1@llnl.gov)

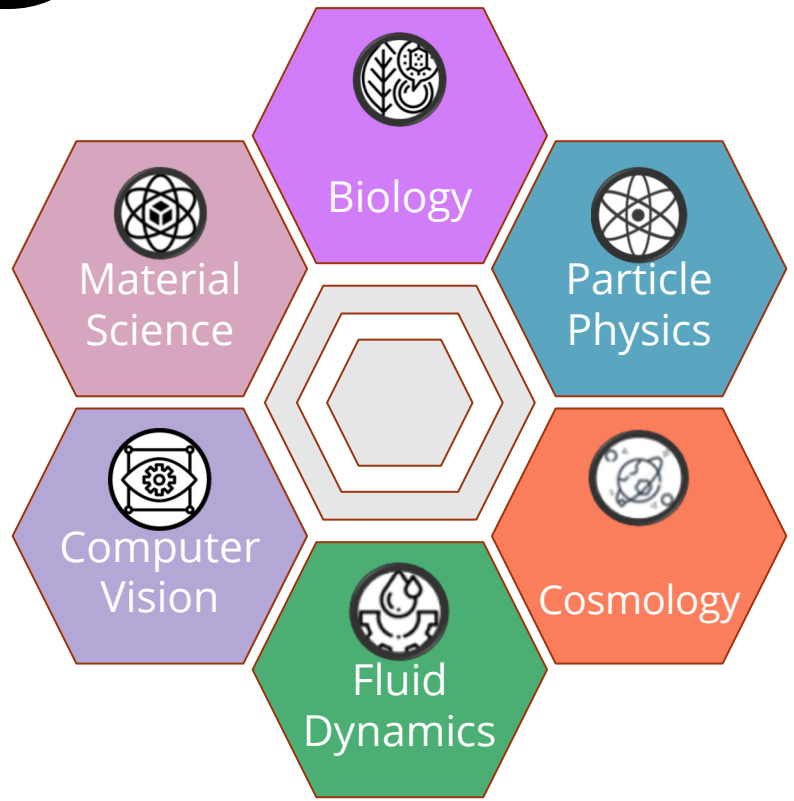
Lawrence Livermore National Laboratory

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-PRES-861305).

Deep Learning to accelerate science

- Diverse scientific domains utilize deep learning to solve scientific problems.
  - accelerate time-to-results.
  - extract patterns of large data.
- Data is the core of all deep learning methods.





Understanding the I/O behavior of scientific DL applications is crucial for scientific discovery.

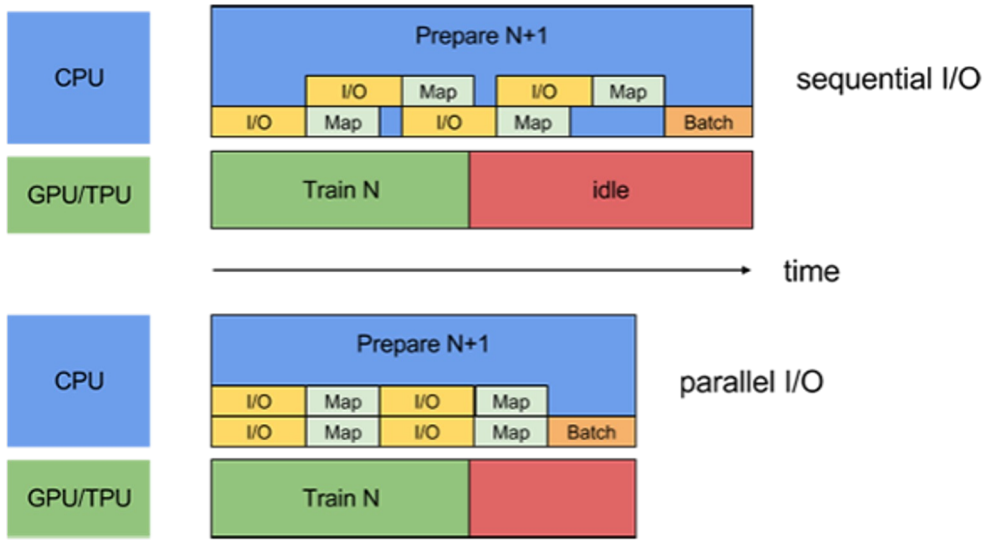
Data Management in DL frameworks

Popular DL frameworks provide methods for tuning the input pipeline.

Major phases for data management

- Data Ingestion
  - different data sources
  - data representations
- Data Processing
  - Parallelism and pipeline through worker assignment.
- Data Communication
  - RDMA,
  - GPU-Direct, and
  - NVLink

Input pipeline in TensorFlow with optimization example.



https://towardsdatascience.com/building-efficient-data-pipelines-using-tensorflow-8f647f03b4ce

Traditional Vs Deep Learning Stack

I/O tasks	Data Phase	Process Management	Language
• Coordinated Vs Independent	• Bulk Synchronous Vs Asynchronous background I/O	• MPI Vs. - Fork - Spawn - Threading - DDP - Horovod	• C/C++/Fortran Vs. Python/C/C++

Observation

I/O research for DL applications require an **understanding** of data access behavior of modern scientific applications.

There is a need of an I/O **benchmark and profiler** representing I/O for deep learning workloads

Hypothesis

DLIO Benchmark

A Data-Centric Benchmark for Scientific DL Applications

Code

https://github.com/argonne-lcf/dlio_benchmark.git

Code

https://github.com/hariharan-devarajan/dlio-profiler.git

A Dataflow Tracer for large-scale Deep Learning workloads

DLIO Profiler

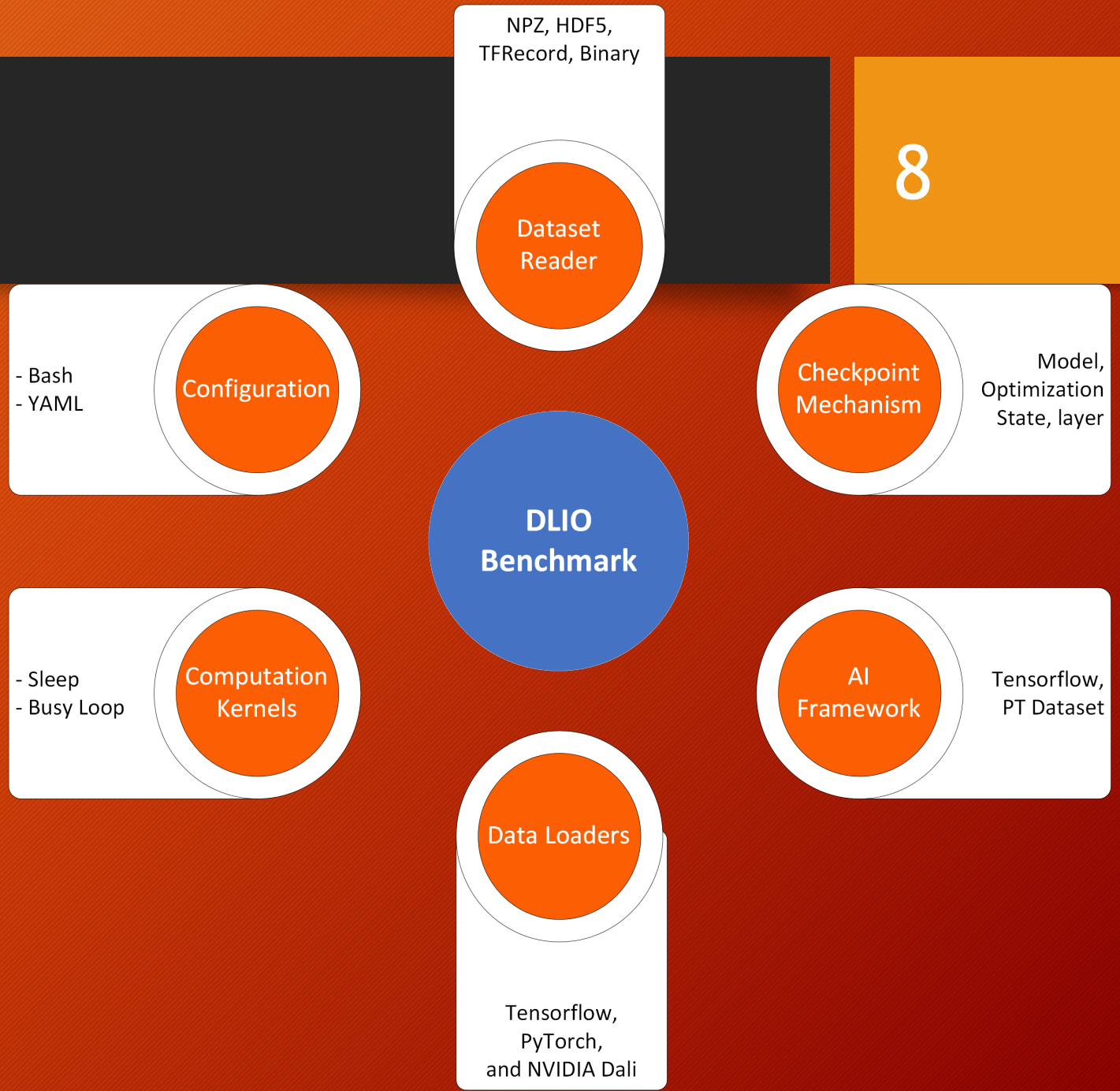
DLIO Benchmark

A Data-Centric Benchmark for Scientific DL Applications

High-Level Design

- Component Design
- Reader allows different data formats
- Checkpoint allows different checkpointing implementations
- Data loaders implement different input pipeline.

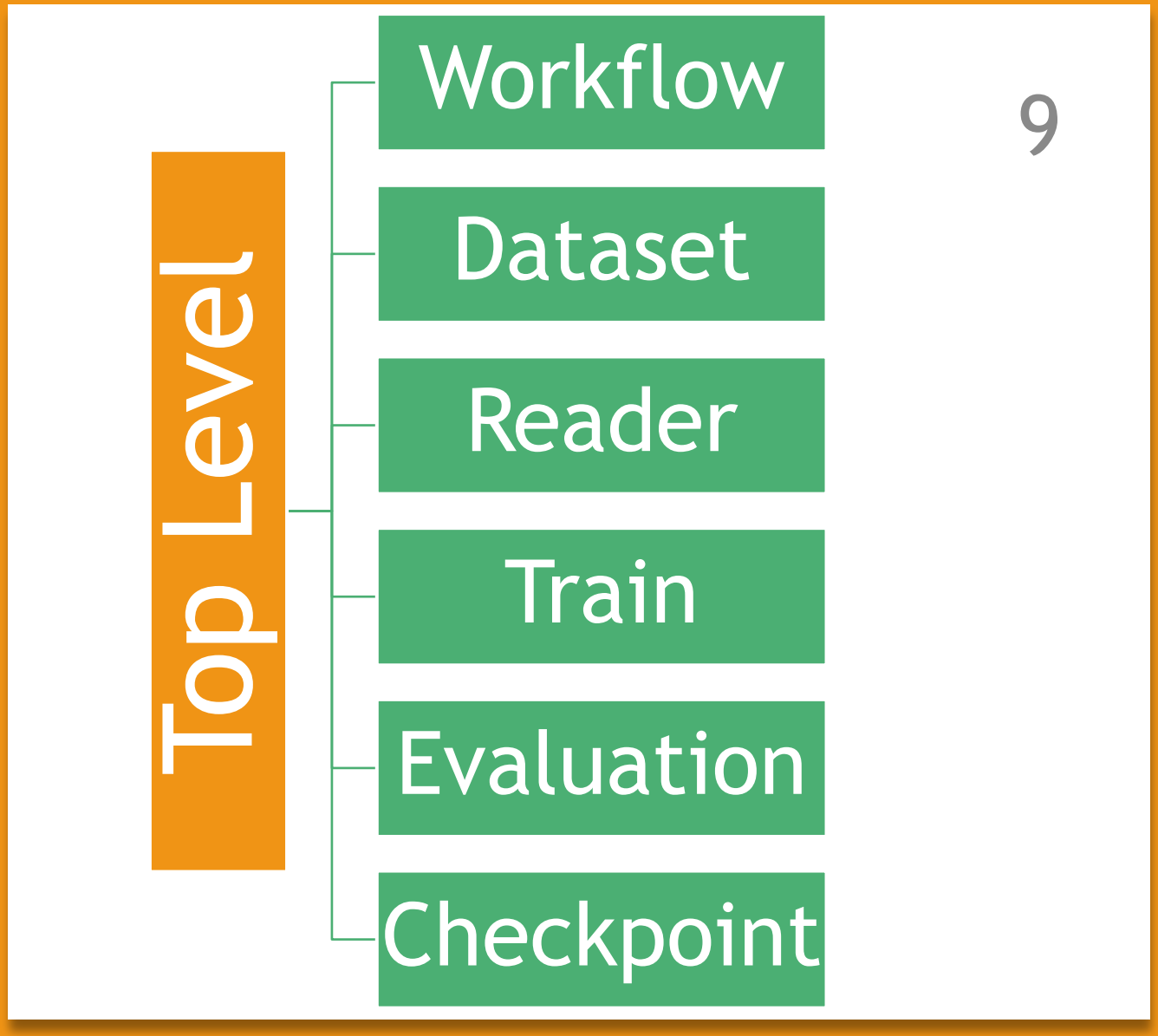
All Components support plugin design to add custom implementations.



Configurations for DLIO Benchmark

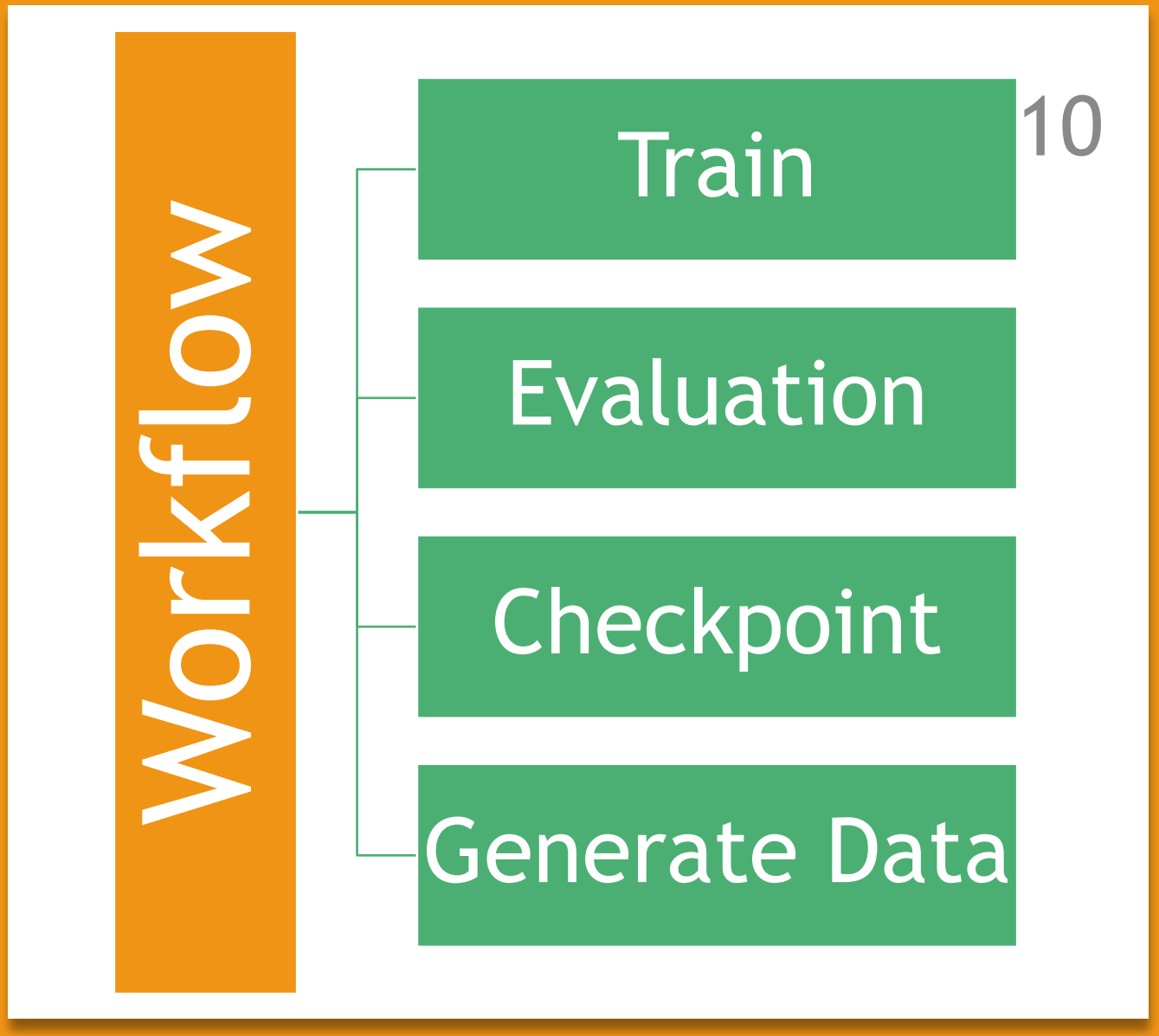
Top Level

SCAN ME



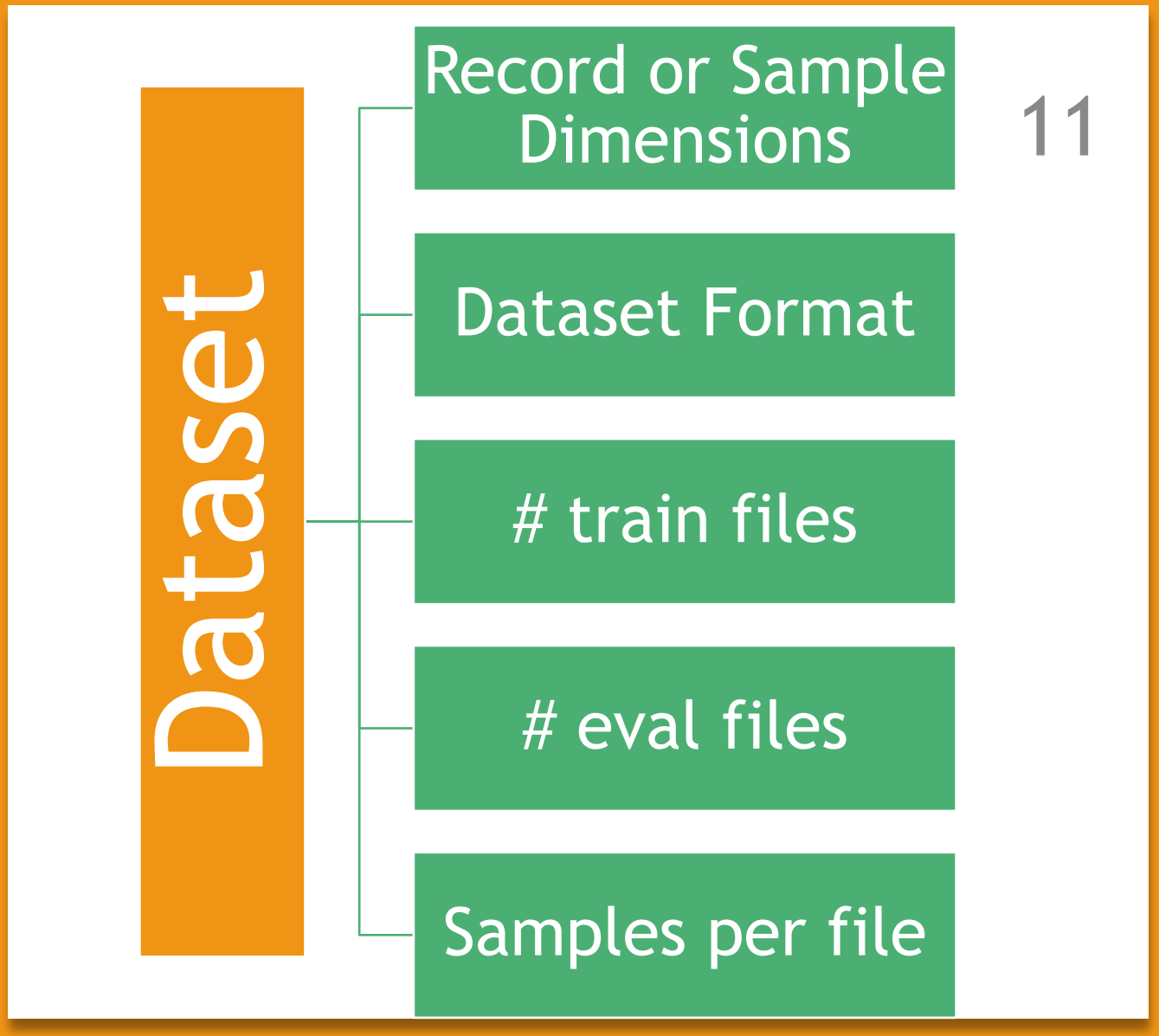
Configurations for DLIO Benchmark

Workflow



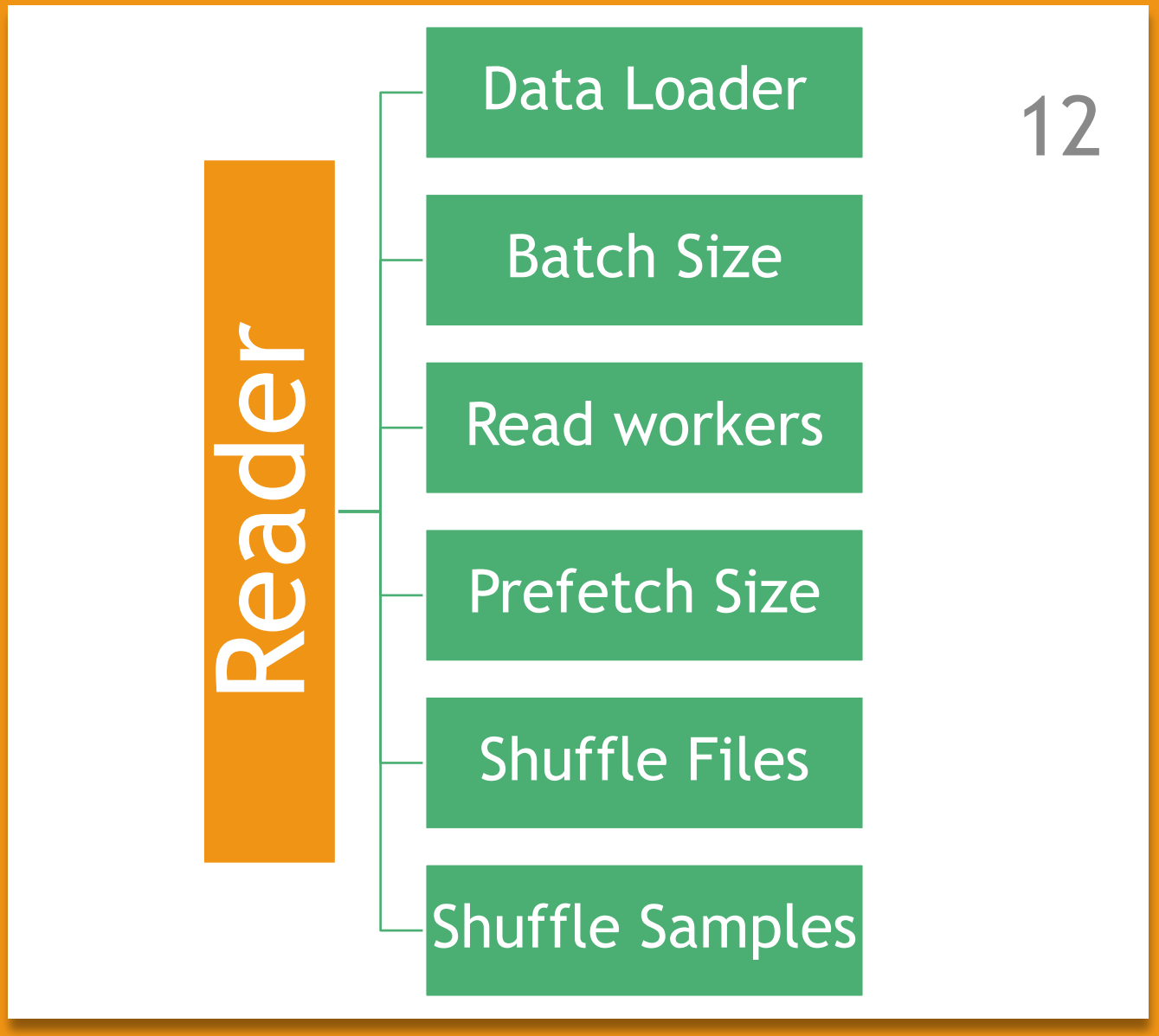
Configurations for DLIO Benchmark

Dataset



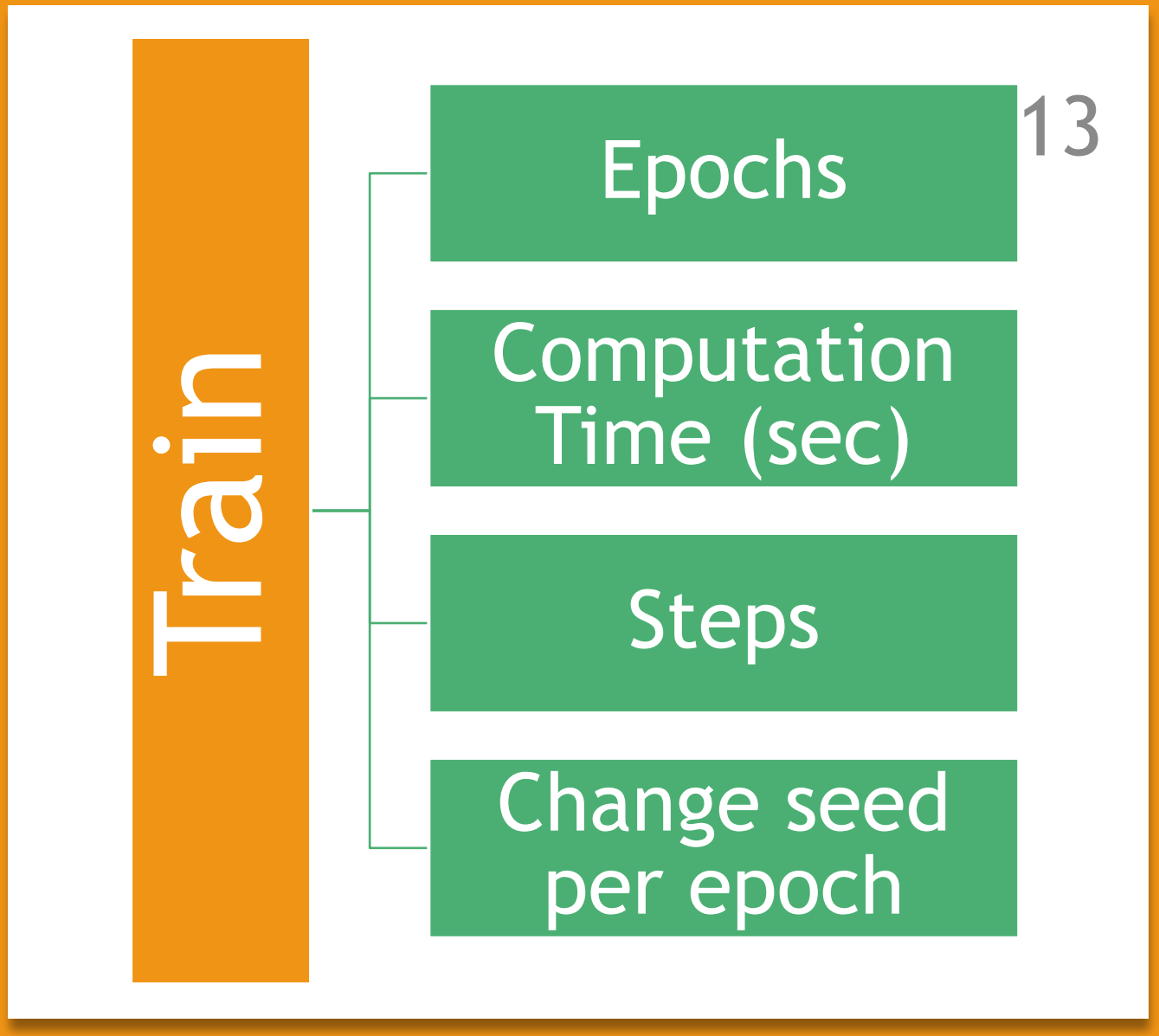
Configurations for DLIO Benchmark

Reader



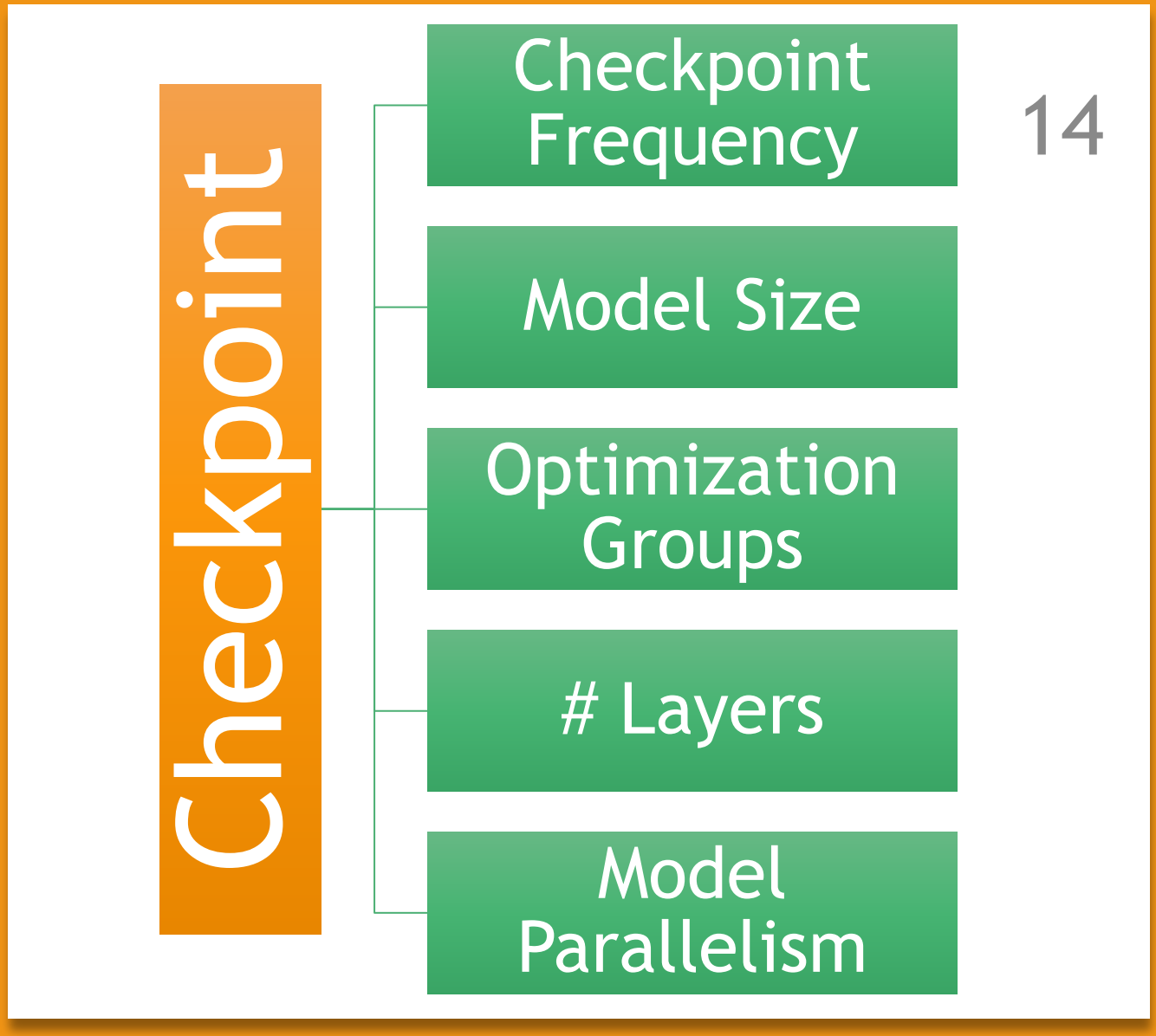
Configurations for DLIO Benchmark

Train



Configurations for DLIO Benchmark

Checkpoint



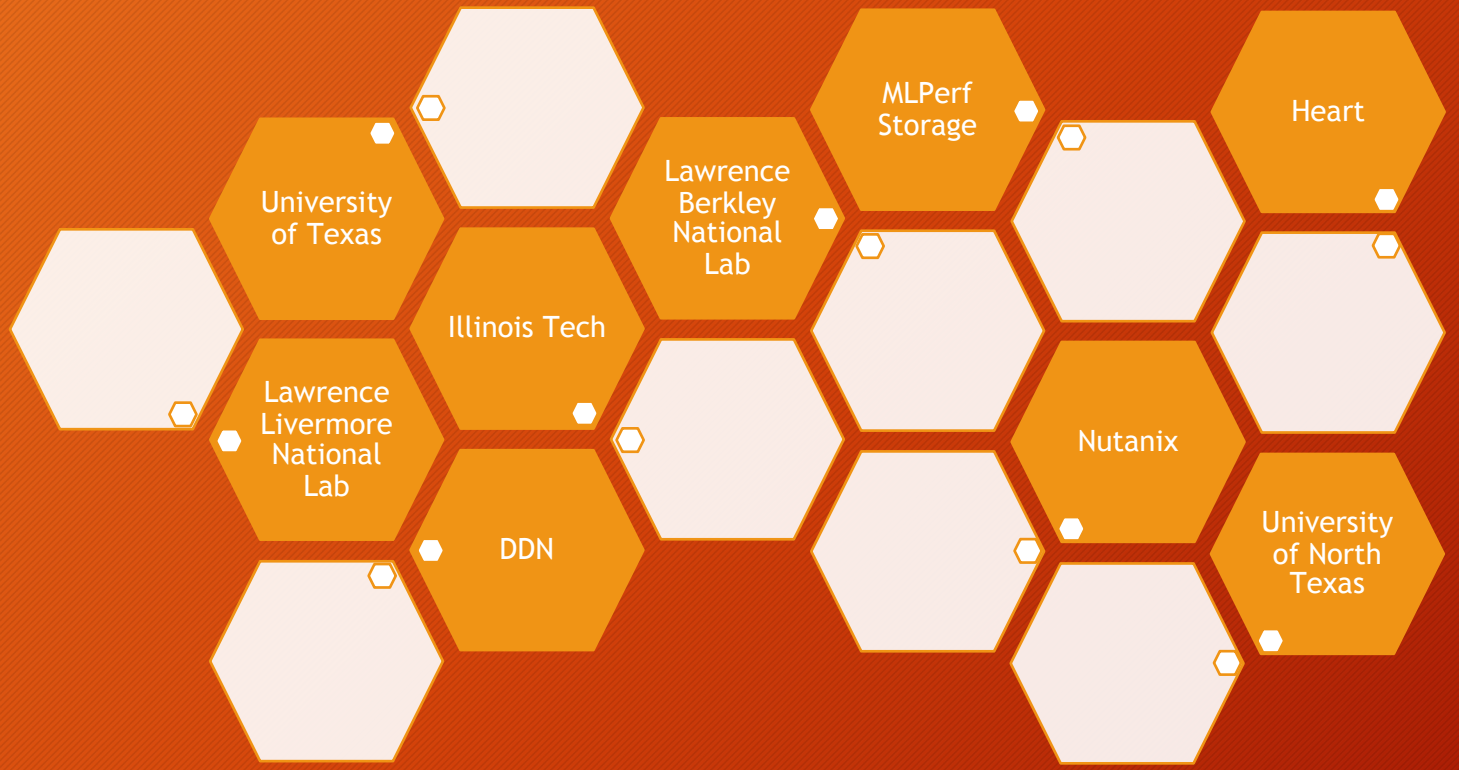
Supported Workloads

1. Unet3D
2. Cosmoflow
3. Resnet50
4. BERT

Experimental

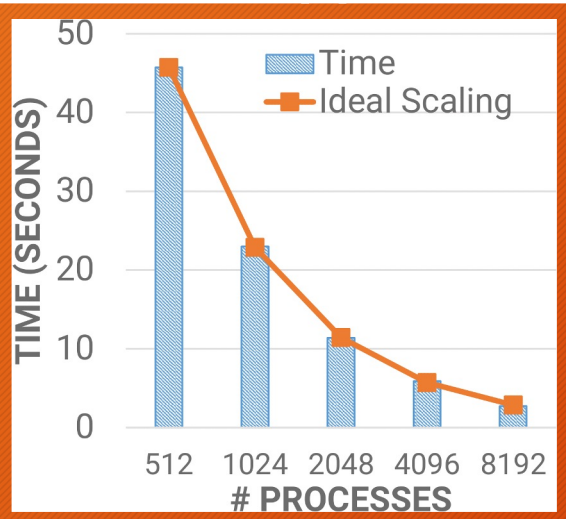
5. DLRM
6. Megatron Deepspeed

Current known adopters

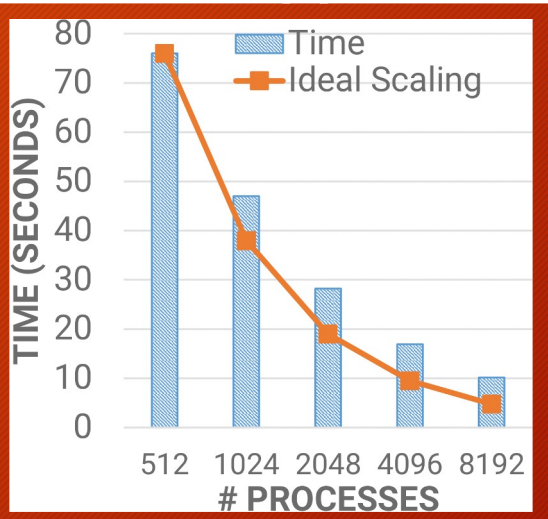


DLIO Benchmark Scaling

Multi-file application



Shared File applications



Observation

- Splitting data across multiple files achieves better I/O performance than using one large shared file.
- Additionally, only TFRecord have I/O optimization for input pipeline which is not present in scientific data formats such as HDF5.

Code

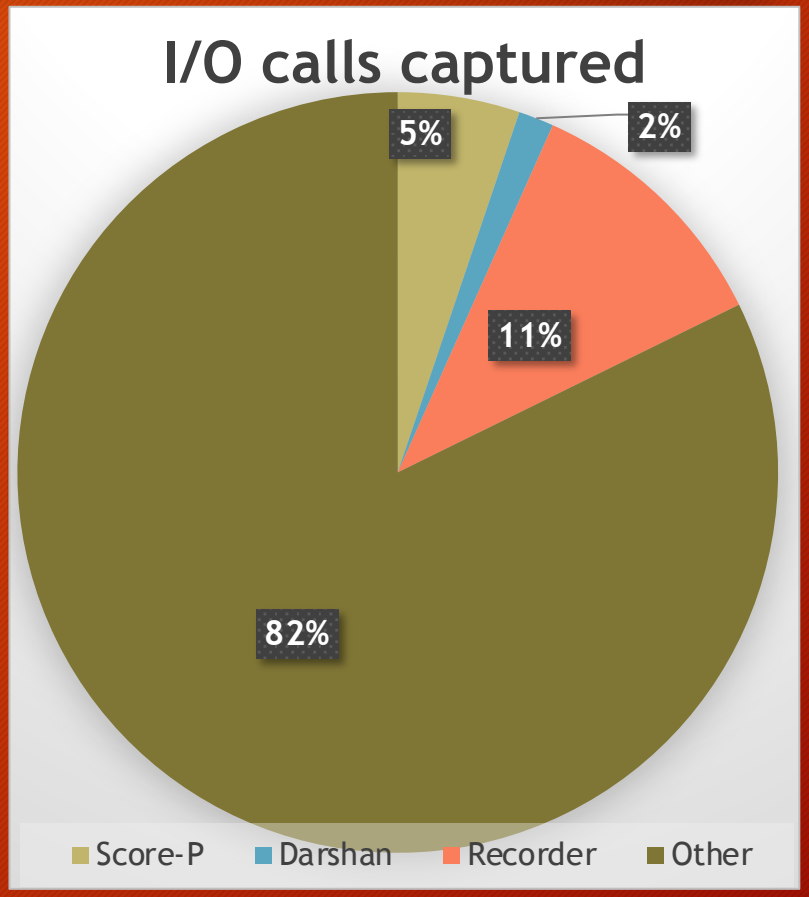
https://github.com/hariharan-devarajan/dlio-profiler.git

DLIO Profiler

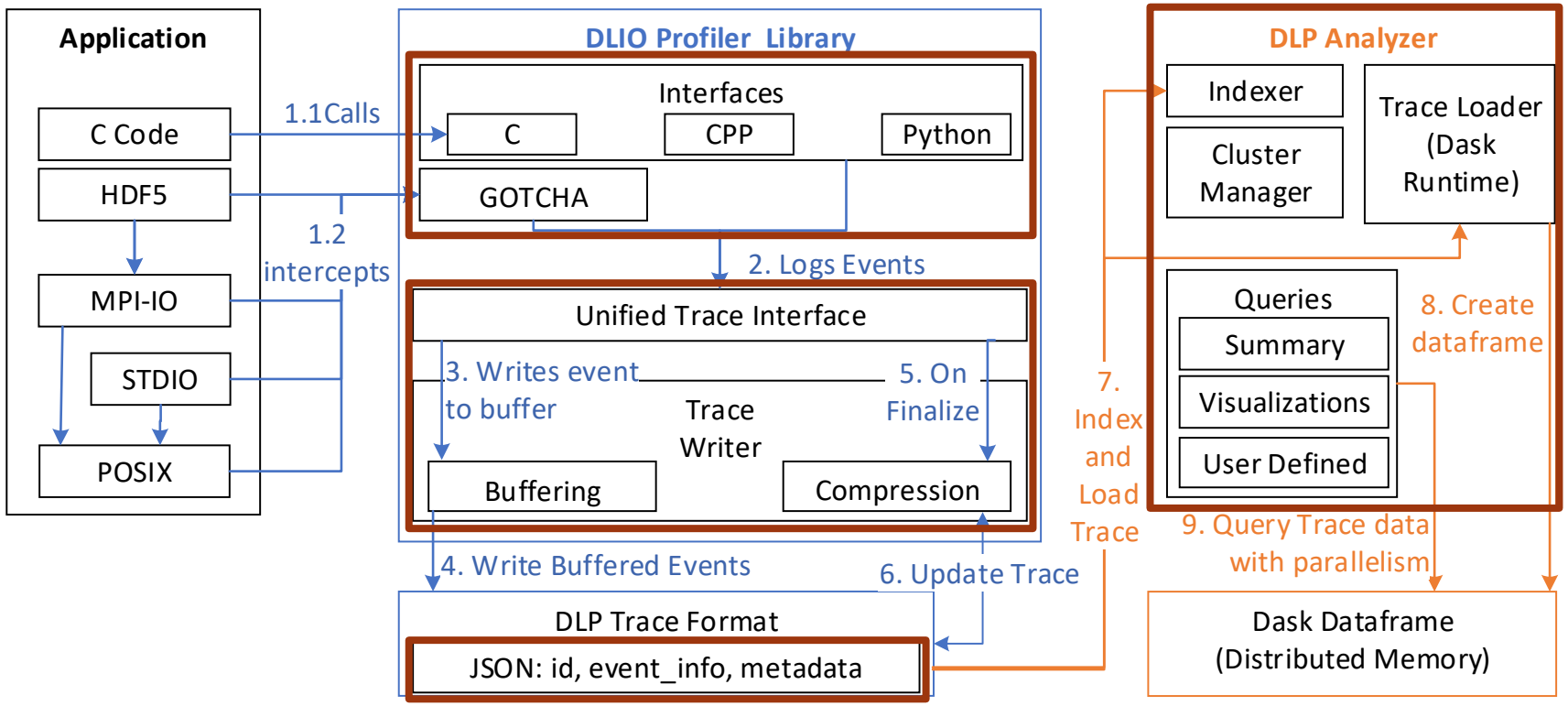
A Data-Centric Benchmark for Scientific DL Applications

Challenges of Current tools

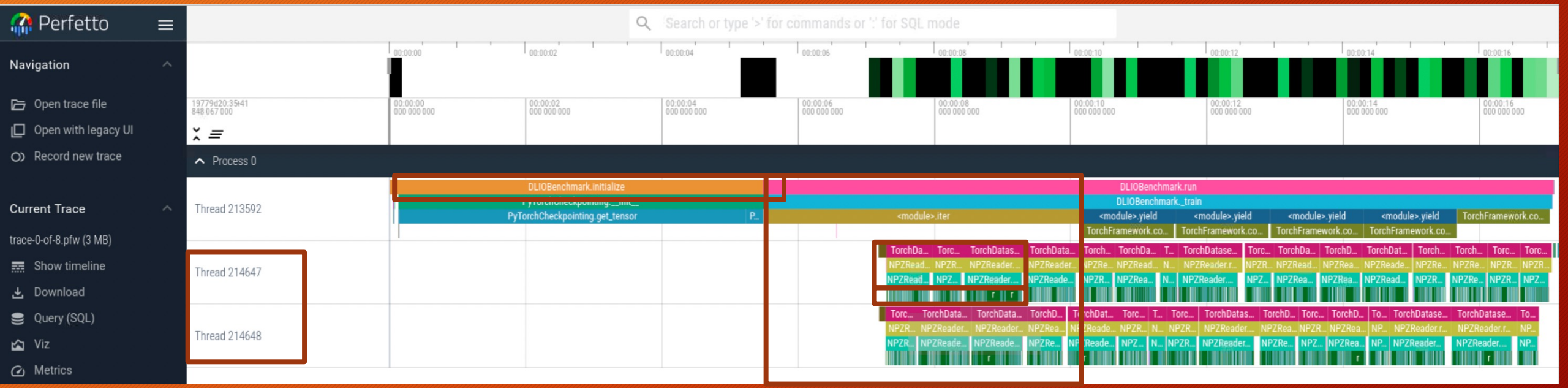
- Need for handling dynamic processes
- Multi-level profiling
  - Python
  - I/O Workers
  - C/C++ Code
  - System Calls
- Metadata Tagging



DLIO Profiler (High-Level Design)



Unet3D (Perfetto UI by Google)

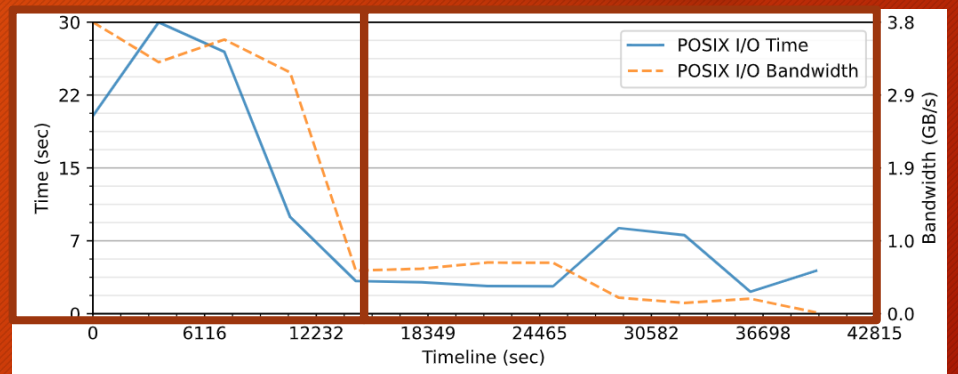


MuMMI (AI-driven MD workflow)

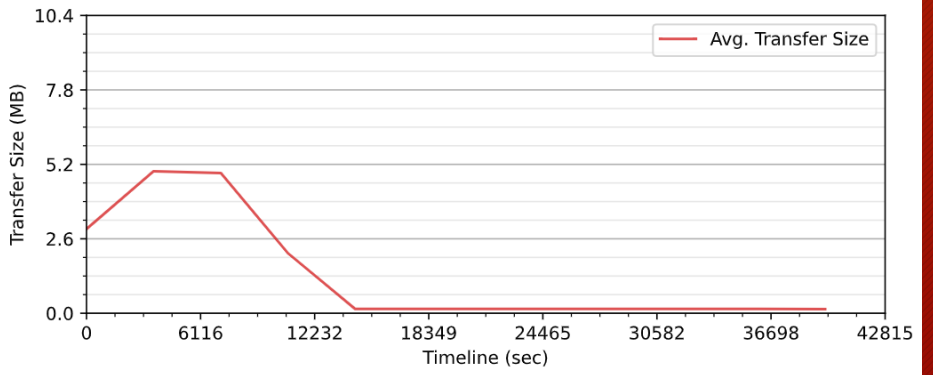

```
Scheduler Allocation Details
├── Nodes: 32
├── Processes: 22949
├── Thread allocations across nodes (includes dynamically created threads)
│   ├── Compute: 0
│   └── I/O: 2
└── Events Recorded: 4M
Description of Dataset Used
└── Files: 1276825
Behavior of Application
├── Split of Time in application
│   ├── Total Time: 42815.186 sec
│   └── Overall I/O: 372.411 sec
└── Metrics by function
```

Function	count	size					
		min	25	mean	median	75	max
__xstat64	3M	NA as these operations do not have bytes transfered					
open64	183K						
__fxstat64	134K						
lseek64	994K						
read	27K	NA	NA	11MB	NA	6KB	588MB
close	133K						
chdir	1K						
write	131K	1	2KB	148KB	2KB	9KB	3MB
mkdir	101K						
opendir	1K	NA as these operations do not have bytes transfered					
__lxstat64	613						
fcntl	613						
rmdir	613						

(c) High-level Summary.



(a) POSIX I/O call Timeline.



(b) Average Transfer Size Timeline.

DLIO Profiler Performance

	Score-P	Darshan	Recorder	DLIO Profiler (40 workers)
# Events Captured (out of 1.1M)	68,752	189	1,389	1.1M
Overhead	13%	23%	13%	7%
Load Time for 1M events	3.56 min	1.6 min	13 min	62 sec
Load Time for 10M events	34.6 min	17 min	2.2 hr	1.3 min
Load Time for 100M events (hr)	6.1 hr	3.3 hr	>12 hr	3.4 min
Trace size: 1M events	59 MB	9 MB	20 MB	8.4 MB
Trace size: 10M events	610 MB	76 MB	169 MB	56 MB
Trace size: 100M events	5 GB	638 MB	1.3 GB	556 MB

Conclusions

1. We demonstrated the need for a new benchmark and tool for AI workloads
2. Described the modular design of DLIO benchmark and the supported workloads.
3. Described the DLIO Profiler design along with analysis friendly trace format
4. Demonstrated some visualizations possible using the DLIO Profiler trace.

A list of all observations

Q&A

Benchmark

SCAN ME

Thank you

hariharandev1@llnl.gov

Profiler

SCAN ME

This material is based upon work supported by the National Science Foundation under Grant no. OCI-1835764 and CSR-1814872. Also, this work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under the DOE Early Career Research Program (LLNL-PRES-861305). Finally, this research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.