Oak Ridge National Laboratory

OLCF Training

AI on Frontier

Junqi Yin

Analytics and AI Methods at Scale

Aug 24, 2023

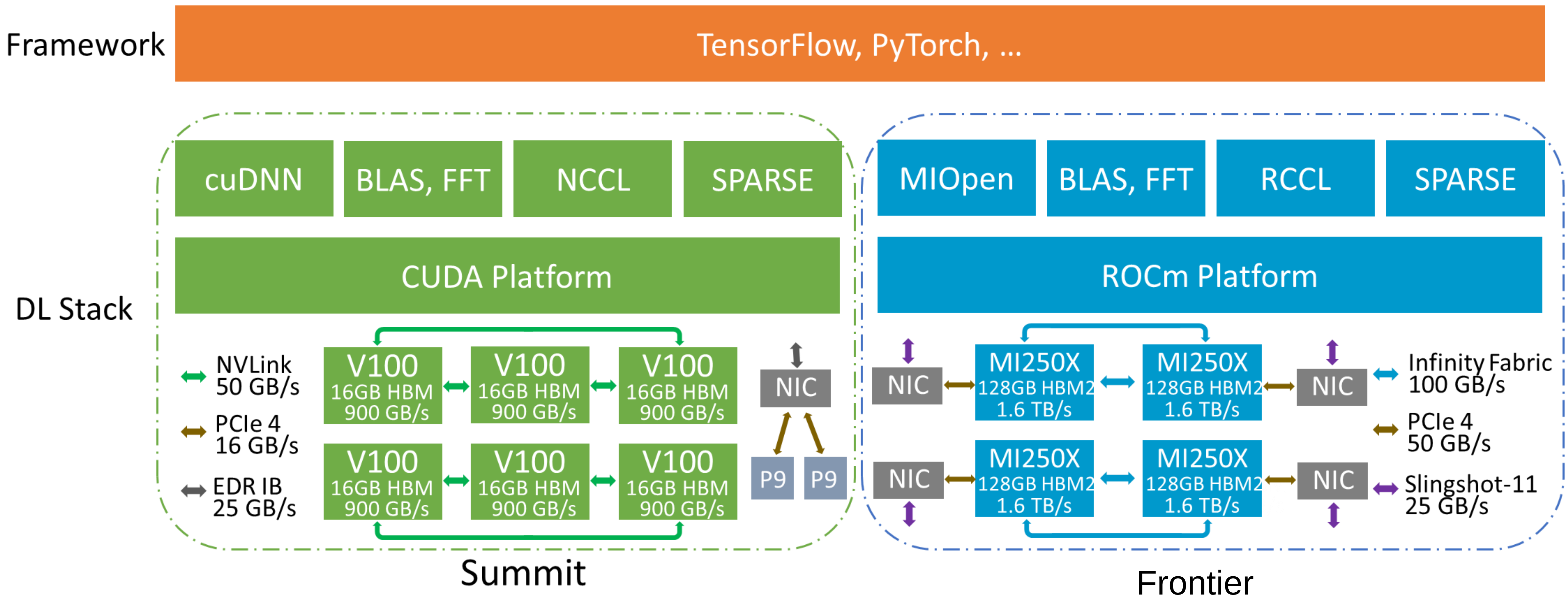
ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Outline

- Frontier DL Environment
- Distributed Training
- Examples
  - PyTorch
  - TensorFlow
- Training Large Language Models (LLMs)

Deep Learning Stacks



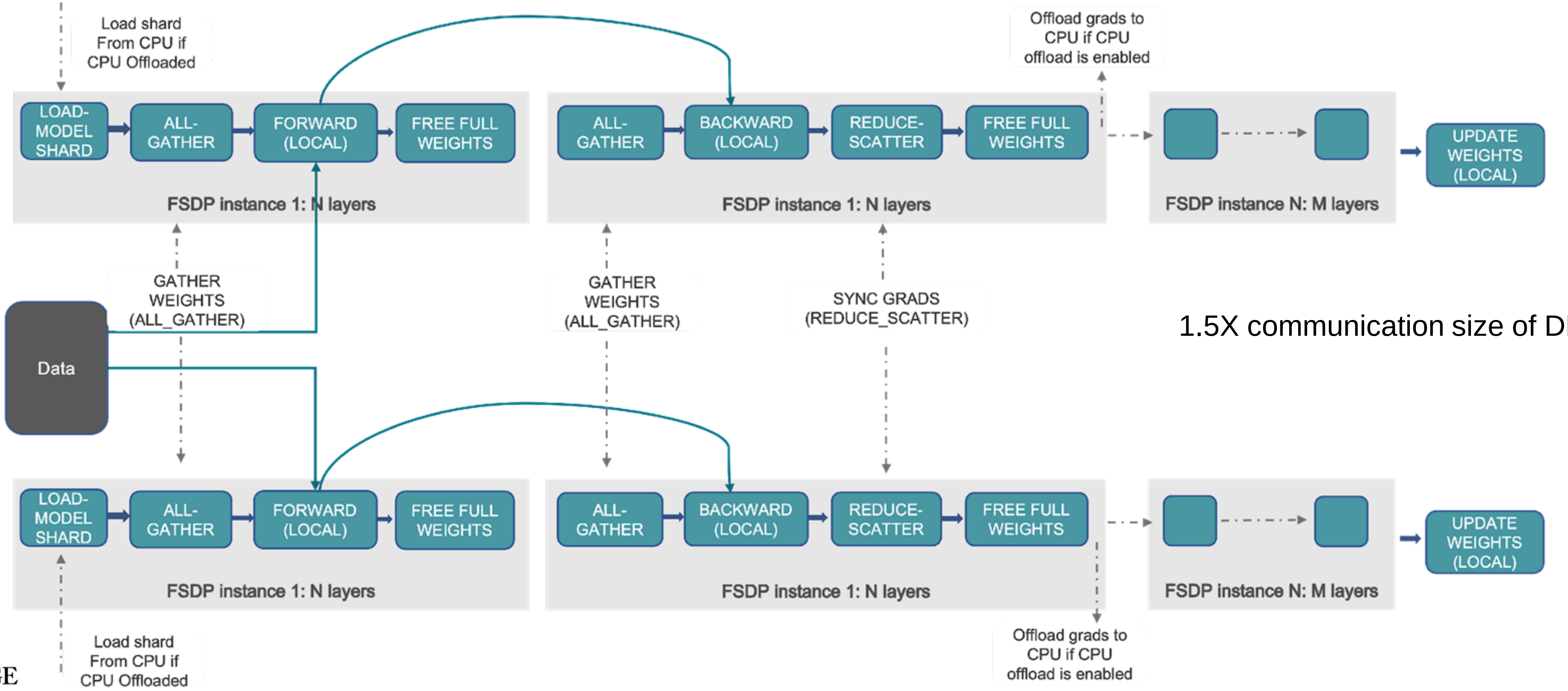
- Most DL codes work on Frontier as is

Distributed Training Strategies

- Data Parallel
  - PyTorch DDP, TensorFlow Multiworker Mirrorred strategy
  - Horovod
- Model Parallel
  - PyTorch FSDP
  - DeepSpeed-Megatron 3D parallelism

Distributed Training Strategies: FSDP

- FSDP – fully sharded data parallel, similar to ZeRO

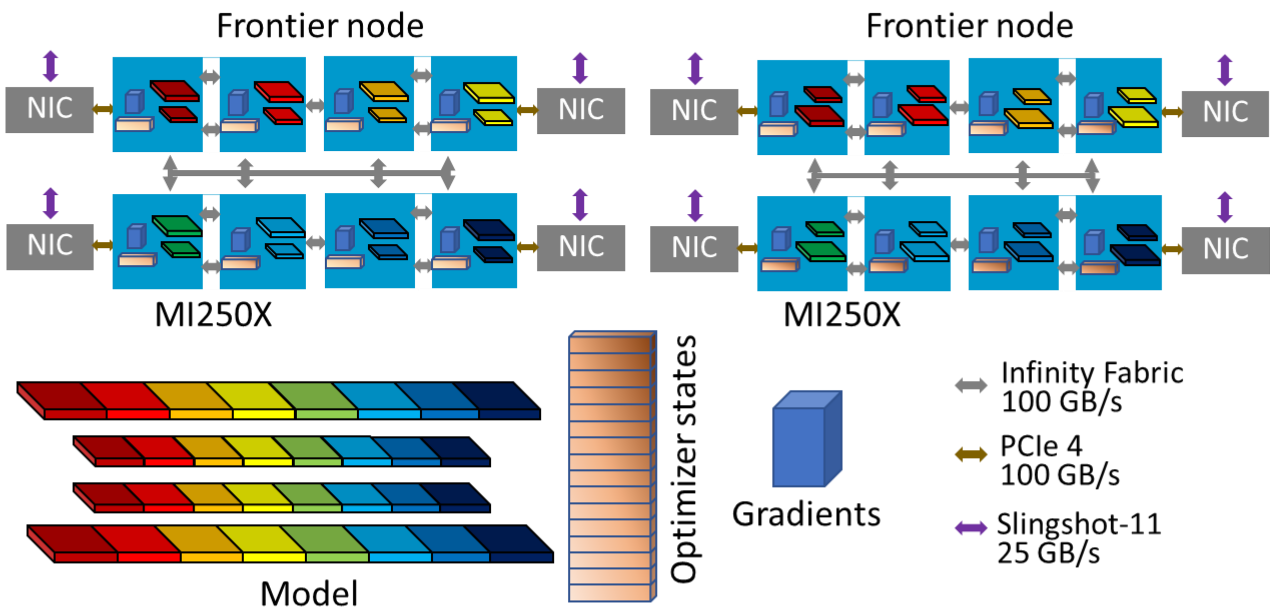


1.5X communication size of DDP

OAK RIDGE National Laboratory

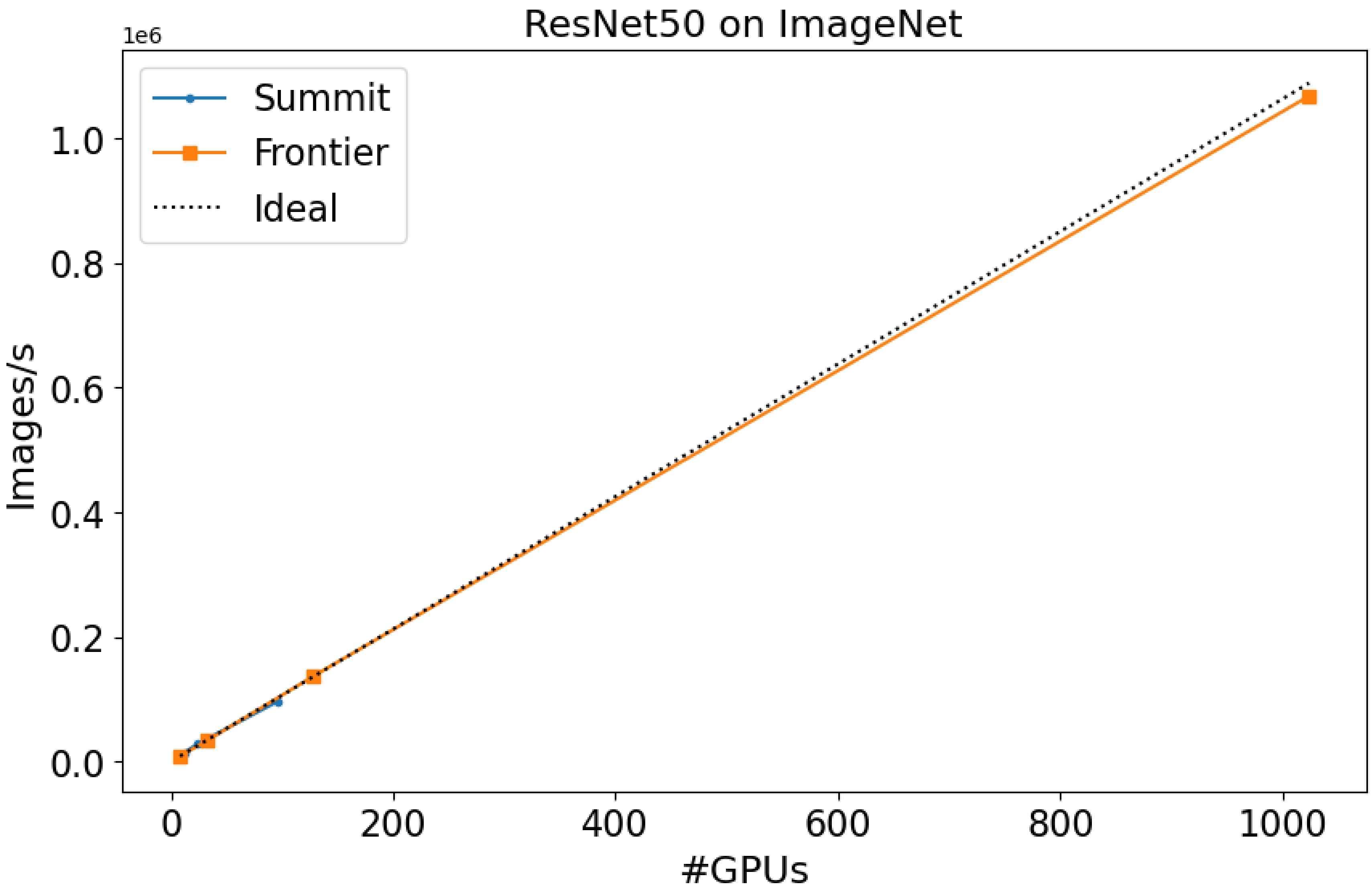
Distributed Training Strategies: DeepSpeed+Megatron

- DeepSpeed + Megatron
  - Sharded data parallel – ZeRO, shared optimizer states, gradients, and parameters
  - Pipeline parallel – layer placement + micro-batch (Gpipe)
  - Tensor parallel – sharded tensor



Performance baselines: data-parallel

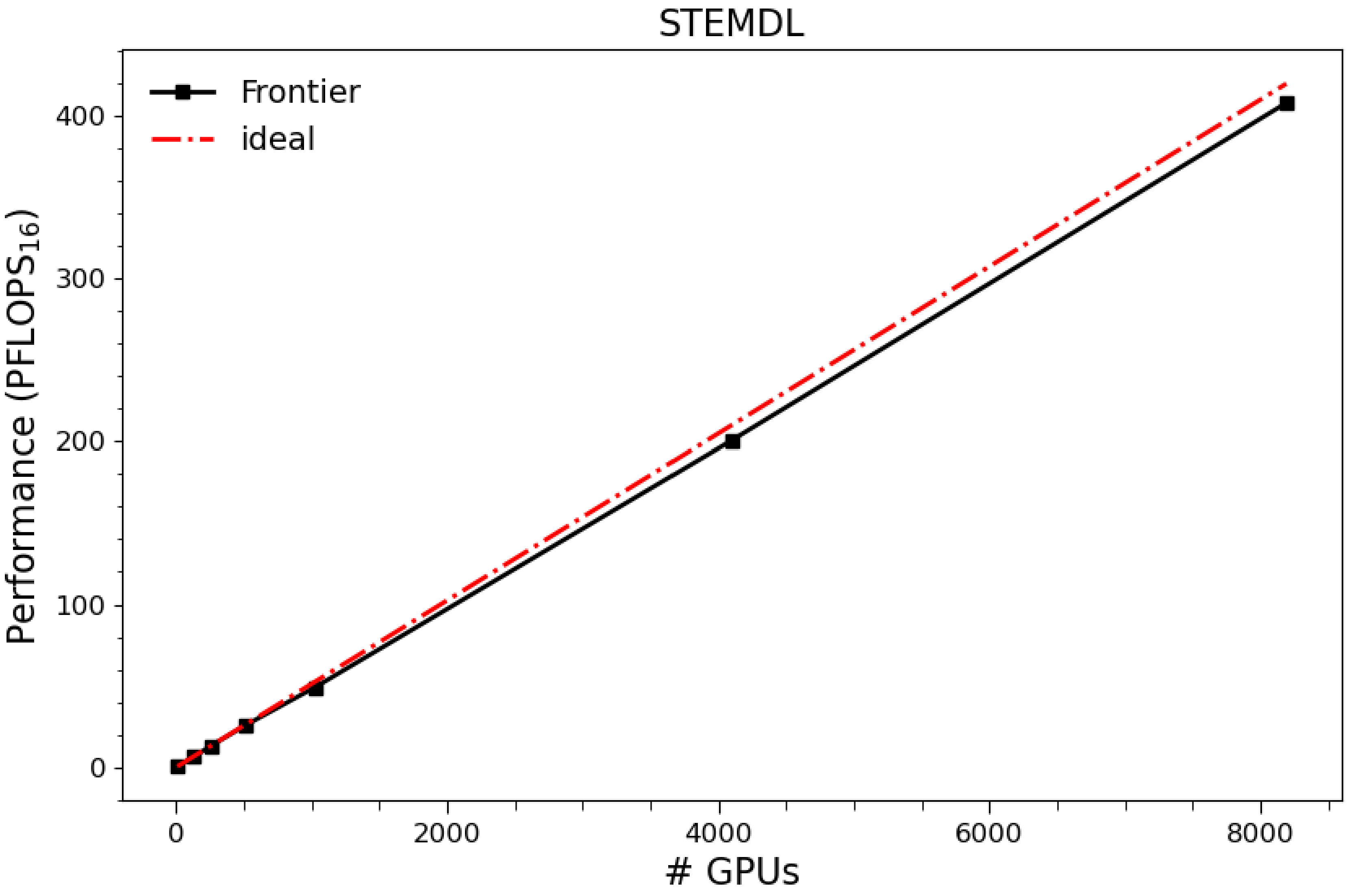
- ResNet50
  - Mixed
  - ~1.0x per GCD
  - 98% at 1024



Performance baselines: data-parallel

- STEMDL
  - Tiramisu network
  - 220M parameters
  - 97% at 8192 GPUs

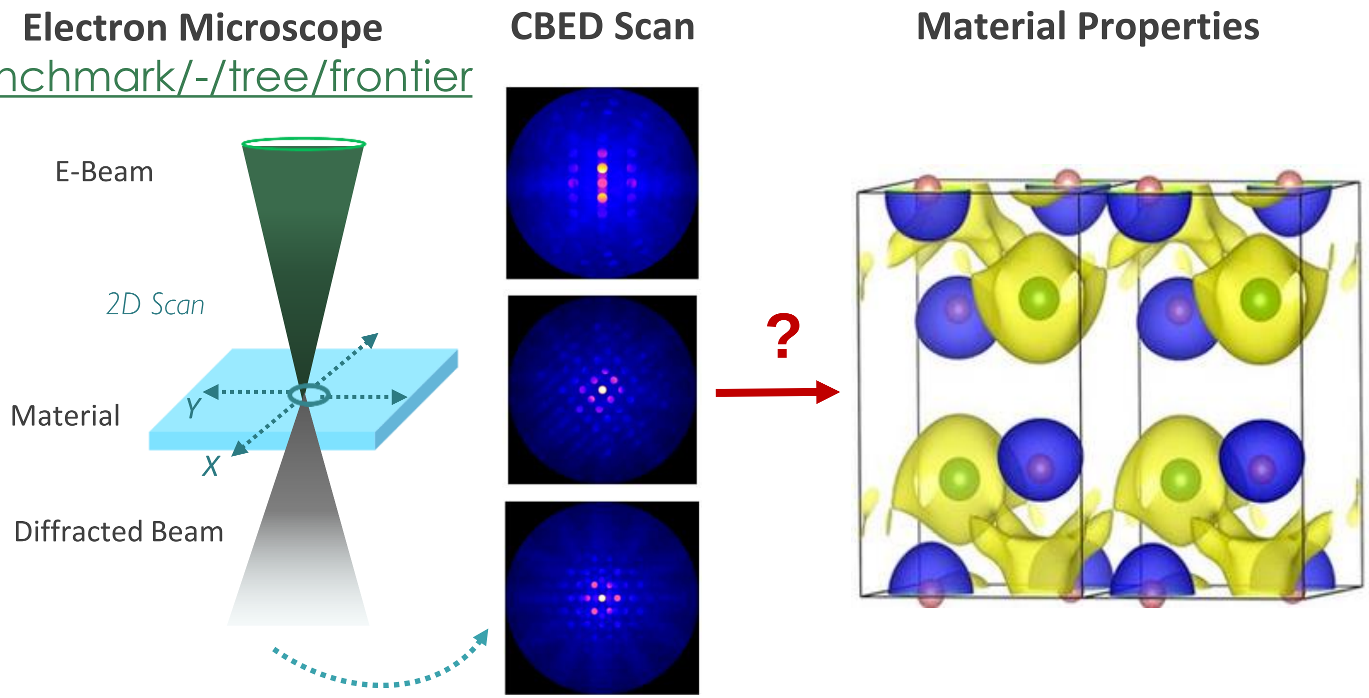
[Accelerating Collective Communication in Data Parallel Training across Deep Learning Frameworks. USENIX NSDI'22, 2022]



Oak Ridge National Laboratory

Distributed Training Examples: STEMDL Benchmark

- One of MLCommon Science benchmarks
- Code on Frontier
  - https://code.ornl.gov/jqyin/stemdl-benchmark/-/tree/frontier
  - Classification – PyTorch
  - Reconstruction – TensorFlow
  - Built/job scripts
  - Conda environment



Distributed Training Examples: STEMDL Benchmark

code.ornl.gov/jqyin/stemdl-benchmark/-/tree/frontier?ref_type=heads

Search GitLab /

Software Environment

A pre-built enviroment is provided at `/lustre/orion/world-shared/stf218/junqi/miniconda`.

```
module load rocm/5.1.0
BUILD=/lustre/orion/world-shared/stf218/jur
export PATH=${BUILD}/bin:$PATH
source ${BUILD}/etc/profile.d/conda.sh
```

To activate PyTorch env:

```
conda activate ${BUILD}/envs/pyt_env
```

To activate TensorFlow env:

```
conda activate ${BUILD}/envs/tf_env
```

To build from source, please follow the scripts in `utils`

Output

The clasificaiton task outputs the mlperf and tensorboard logs to `logs` folder (default)


```
#stemdl_classification.log
:::MLLOG {"namespace": "", "time_ms": 1692106306826, "event_type": "POINT_IN_TIME", "key": "eval_accuracy", "value": 7.01891407370
:::MLLOG {"namespace": "", "time_ms": 1692106306826, "event_type": "POINT_IN_TIME", "key": "f1_score", "value": 2.2504812106490135
:::MLLOG {"namespace": "", "time_ms": 1692106306826, "event_type": "INTERVAL_END", "key": "eval_stop", "value": null, "metadata":
:::MLLOG {"namespace": "", "time_ms": 1692106307223, "event_type": "INTERVAL_START", "key": "epoch_stop", "value": null, "metadata
```


The reconstruction task outputs the training preformance in FLOPS and loss values to `output_logs` folder (default)

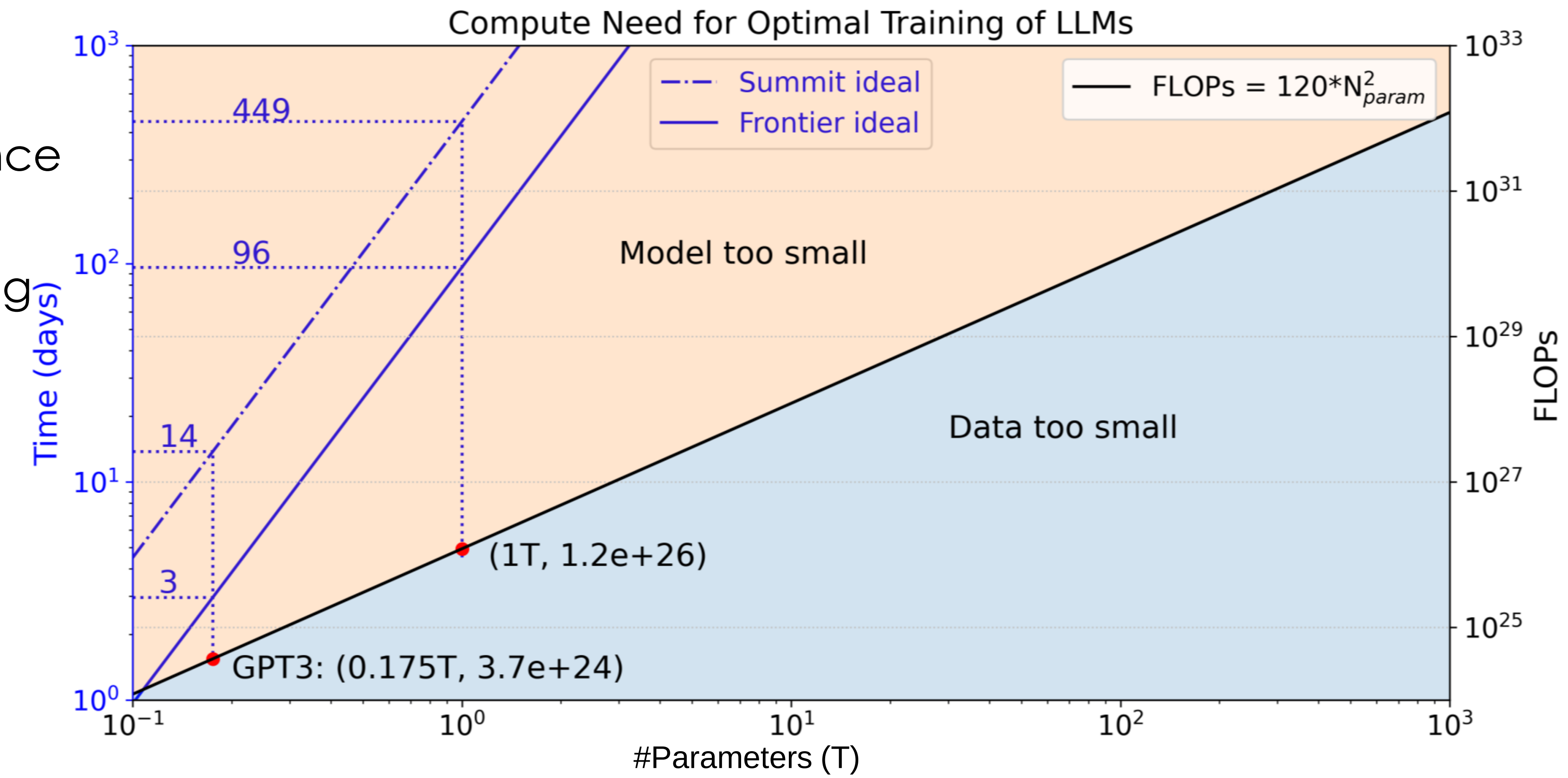
```
58.0, step= 10, epoch= 9.55e-03, loss= 0.52, lr= 1.00e-06, step_time= 15.93 sec, ranks= 8, examples/sec= 4.0, flops = 4.40e+13, ave
75.1, step= 20, epoch= 1.91e-02, loss= 0.46, lr= 1.00e-06, step_time= 1.71 sec, ranks= 8, examples/sec= 37.5, flops = 4.10e+14, aver
92.3, step= 30, epoch= 2.87e-02, loss= 0.53, lr= 1.00e-06, step_time= 1.71 sec, ranks= 8, examples/sec= 37.4, flops = 4.09e+14, aver
09.4, step= 40, epoch= 3.82e-02, loss= 0.52, lr= 1.00e-06, step_time= 1.71 sec, ranks= 8, examples/sec= 37.4, flops = 4.09e+14, aver
```

Data

Sample data (pre-processed) are located at `/lustre/orion/world-shared/stf218/junqi/stemdl-data`

Training LLMs on Frontier

- Ideal case
  - Peak performance
  - Linear scaling
  - Chinchilla scaling

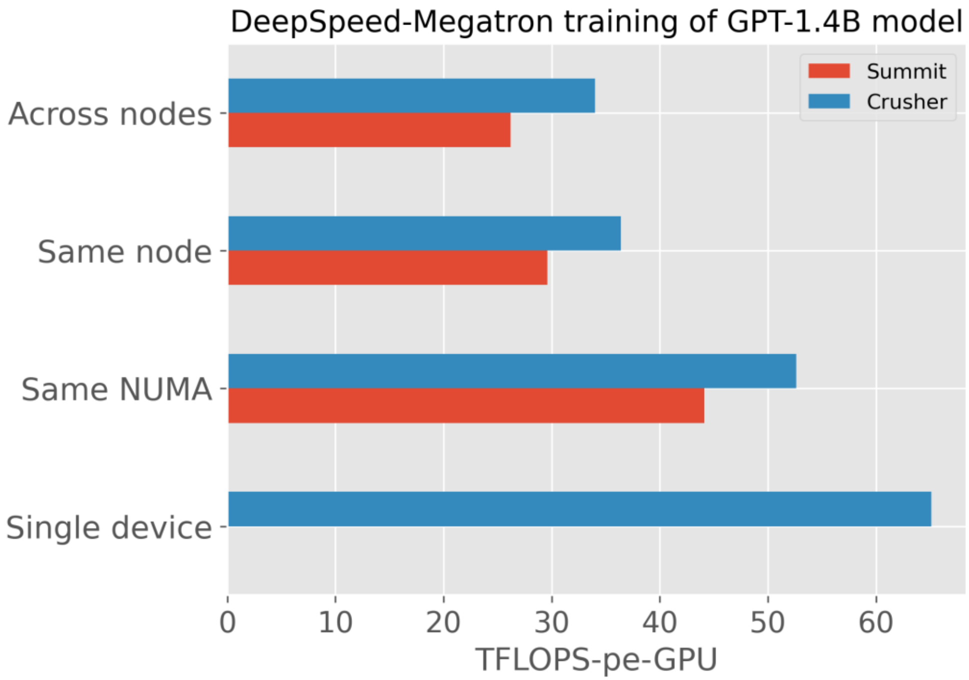


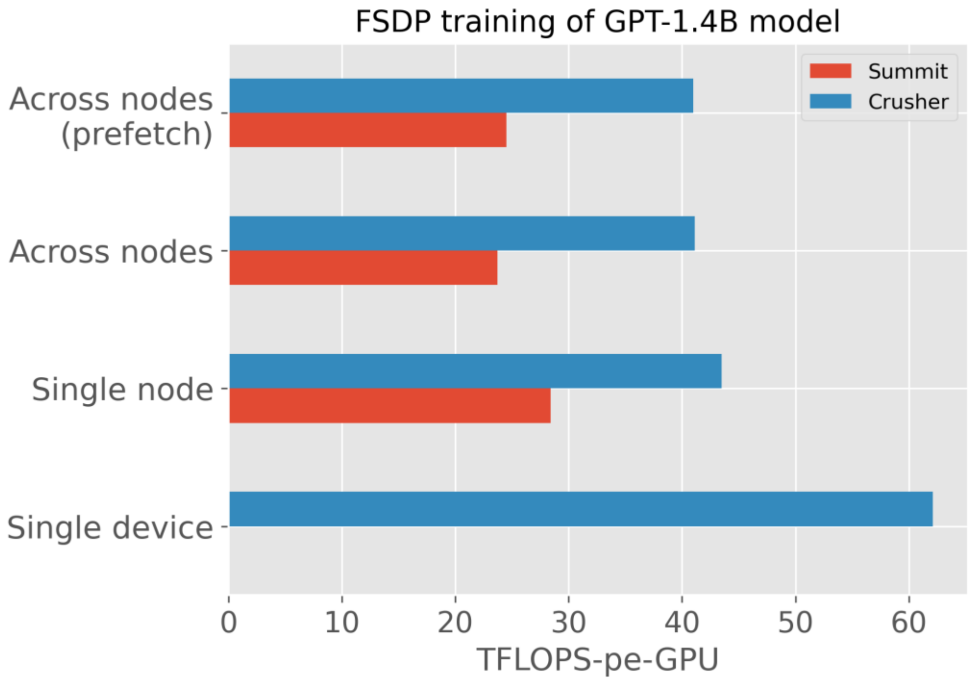
Evaluation of Pre-Training Large Language Models on Leadership Platforms, Journal of Supercomputing, 2023

OAK RIDGE National Laboratory

Training LLMs on Frontier

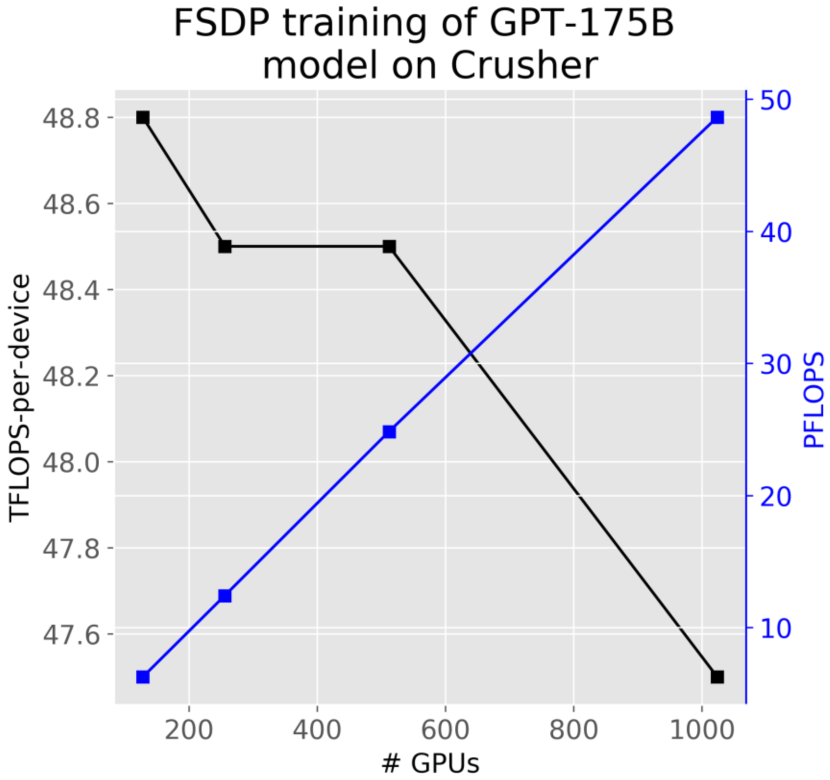
- Actual measurement
  - 2 frameworks: FSDP, DeepSpeed-Megatron
  - 3D parallelisms: DP, TP, PP

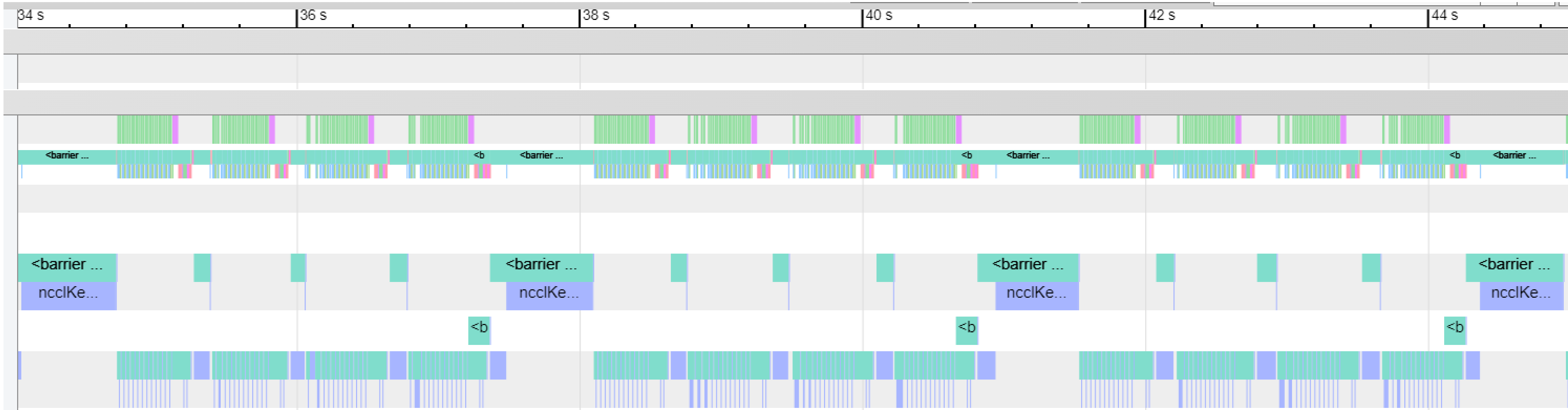




Training LLMs on Frontier

- Scaling
  - DP is preferable
  - Large batch
  - bf16 more stable





OAK RIDGE National Laboratory

Training LLMs on Frontier

- Computation and energy needed to pretrain LLMs

Model	Summit			Frontier		
	Time (node hours)	Energy (MWh)	TFLOPS/ Watt	Time (node hours)	Energy (MWh)	TFLOPS/ Watt
GPT-13B	23K	30	0.185	12K	24	0.239
GPT-175B	[0.6M, 12M]	[309, 6176]	0.165	[0.1M, 2.6M]	[217, 4338]	0.235
GPT-1T	[1B, 21B]	[3.7e5, 7.4e6]	0.01	[8M, 160M]	[4727, 94539]	0.343

[# tokens = # params, # tokens = 20*# params]

- Takeaway: pre-train O(13B) ~ O(DD), pre-train O(175B) ~ O(INCITE)

Evaluation of Pre-Training Large Language Models on Leadership Platforms, Journal of Supercomputing, 2023

Simulation-ML Integration

Method	Pro	Con
Framework C++ API (TensorFlow/PyTorch C++)	• Portable • Better latency • Easy to deploy	• Not flexible
Framework Server (TensorFlow Serving/TorchServe)	• Flexible • Better throughput • Options to deploy	• High maintenance
Third-party API (SmartRedis/RedisAI)	• Easy integration • More functionality • Model support	• Portability

[Strategies for Integrating Deep Learning Surrogate Models with HPC Simulation Applications, IPDPSW 2022]

Questions ?

yinj@ornl.gov