Analysis of Large-Scale Multi-Tenant GPU Clusters for DNN Training Workloads

Myeongjae Jeon, Shivaram Venkataraman, Amar Phanishayee,

Junjie Qian, Wencong Xiao, Fan Yang

ULSAN NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY

Microsoft

WISCONSIN UNIVERSITY OF WISCONSIN–MADISON

北京航空航天大學 BEIHANG UNIVERSITY

Deep Learning at a Large Enterprise

Speech, Image, Ads, NLP, Web Search …

DL training jobs require large GPU clusters

Cortana XBOX Office 365 Bing

Philly: *Cluster manager for DL workloads on large shared GPU clusters*

		Motivated by observations in Philly	
Recent Cluster Managers	**Optimus [EuroSys 18]**	**Gandiva [OSDI 18]**	**Tiresias [NSDI 19]**
Objective	Average JCT	Consolidation	Average JCT
Scheduler	SRTF	Time-sharing	Gittins Index

Microsoft Philly

Significant increase in scale during 2017

10.5 × in DL training jobs

5 × in GPU cluster size

PyTorch TensorFlow

Philly cluster manager

Resource scheduling (GPU, network)

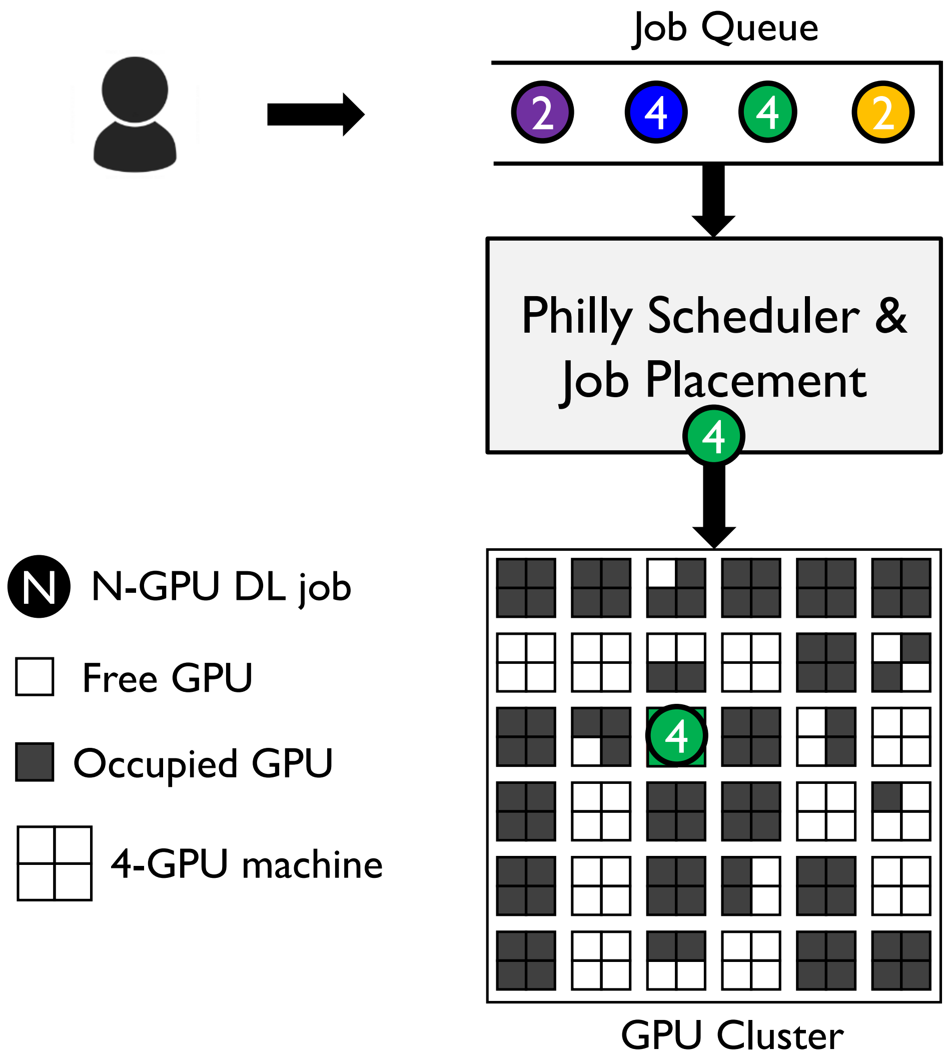
Storage for data & model ckpt

Failure handling

Multi-tenancy

….

Job Lifecycle in Philly



Contributions

1. First characterization study of large-scale GPU clusters for DNN training

2. Study cluster utilization and how effectively GPUs are used

3. Present lessons for better cluster manager designs

Contributions

1. First characterization study of large-scale GPU clusters for DNN training

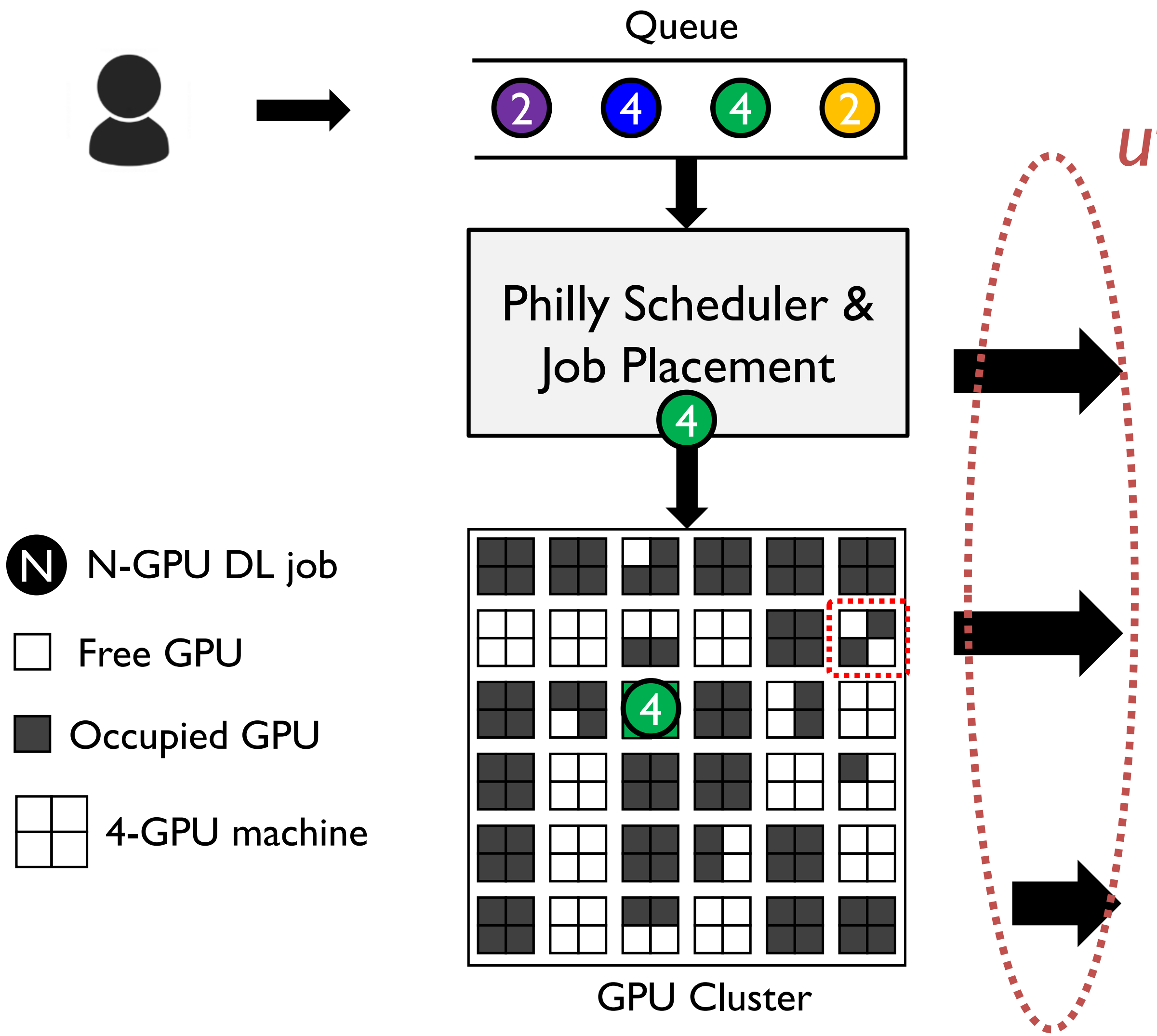
75-day period from Oct. 2017 to Dec. 2017

Total of 96,260 jobs across thousands of users

2. Study cluster utilization and how effectively GPUs are used

3. Present lessons for better cluster manager designs

Study Details



Track scheduling decision and utilization info during job lifecycle

Scheduler logs

– Job arrival, GPU alloc, finish status

HW perf counters

– GPU, CPU, memory utilization

AI engine logs

– stderr/stdout for executed jobs

Contributions

1. First characterization study of large-scale GPU clusters for DNN training

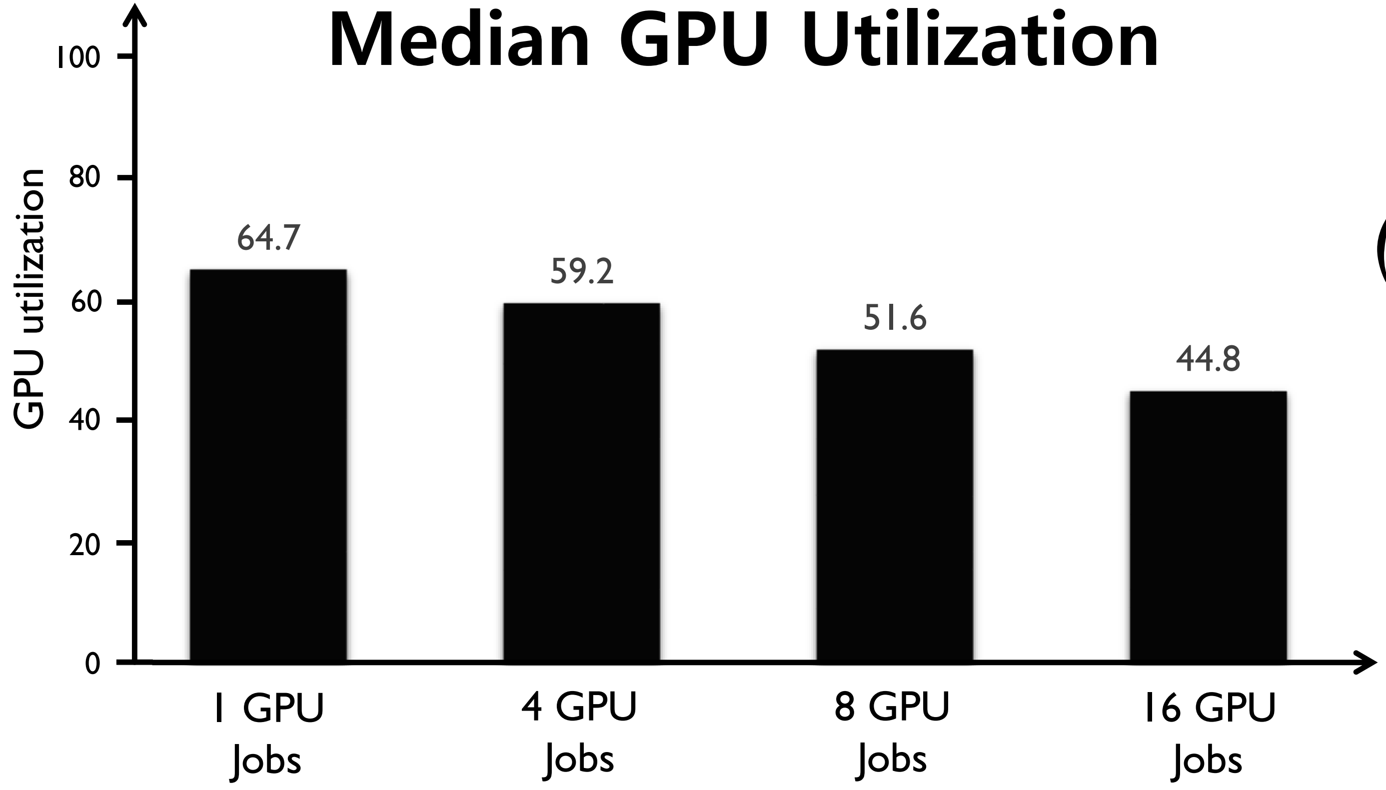
2. Study cluster utilization and how effectively GPUs are used

3. Present lessons for better cluster manager designs

Most GPUs in the cluster are allocated

How effectively are the GPUs utilized for DNN training?

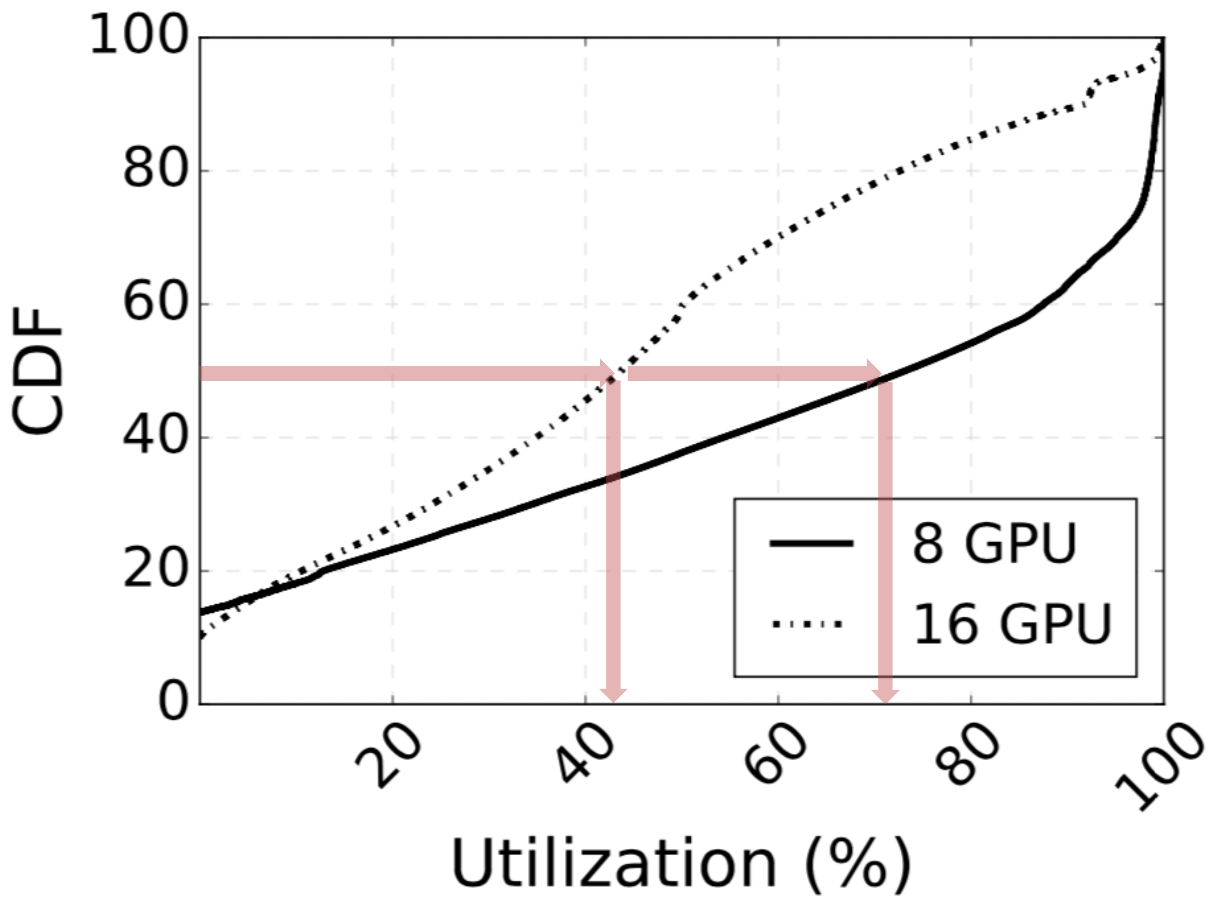
GPU Utilization for Job Sizes



GPU utilization is low!
(Lower in distributed training)

Two reasons:
- *Distribution across servers* ✔
- *Intra-server interference*

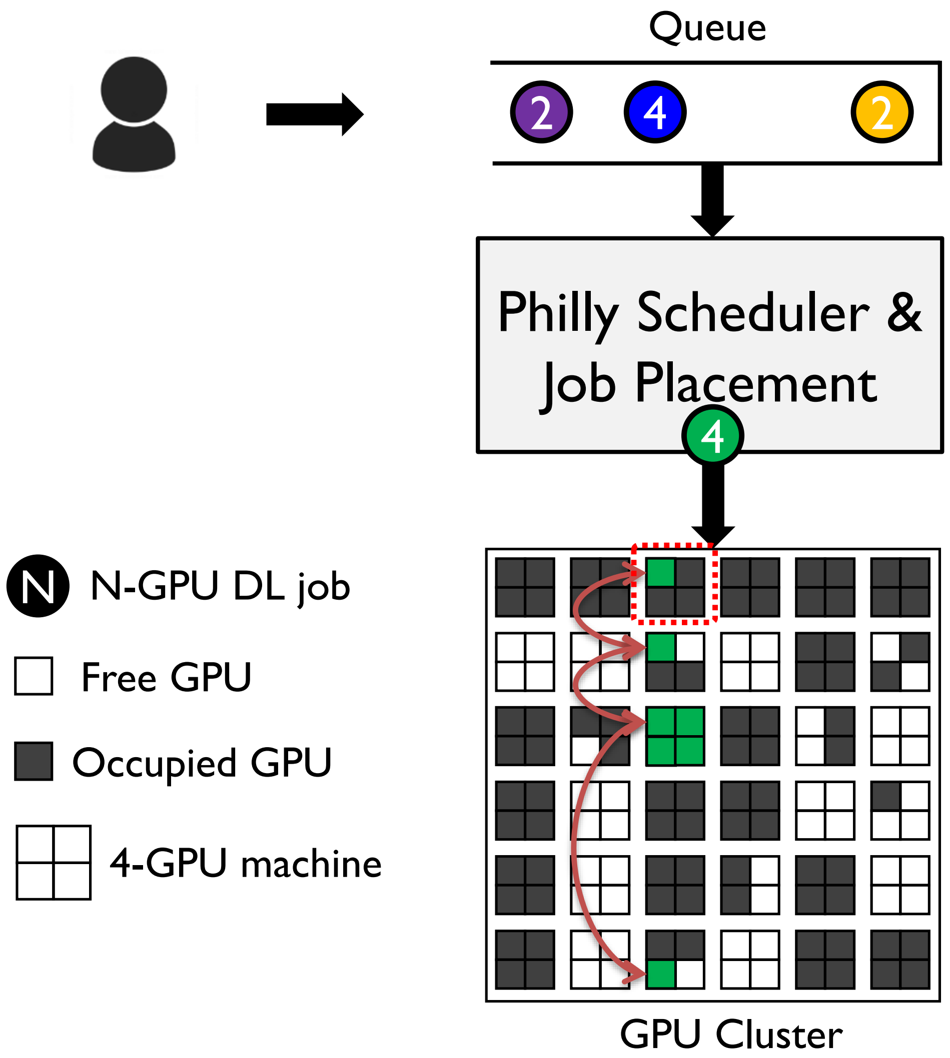
Effect of Distribution on Dedicate Servers



Dedicate servers →
No other jobs on this server

Distributed training itself causes utilization to go lower!

Scheduling Distributed Training



Relaxing locality constraints

High intra-server locality

– High communication efficiency

– Long queueing time

Low intra-server locality

– Low queueing time

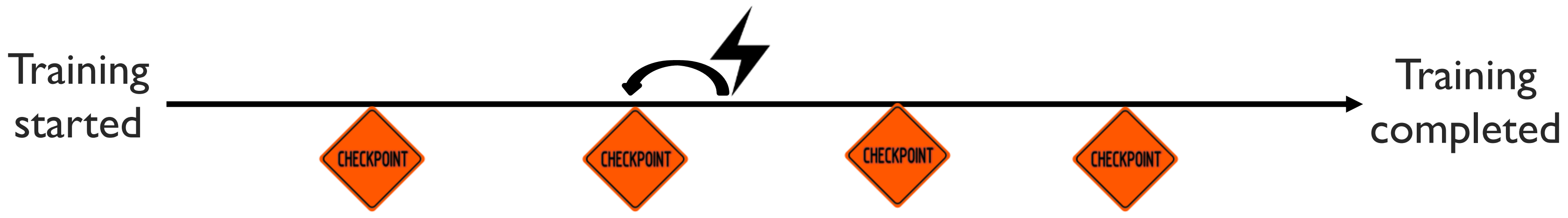
– Contention in the use of network

– Risk of intra-server interference (across jobs)

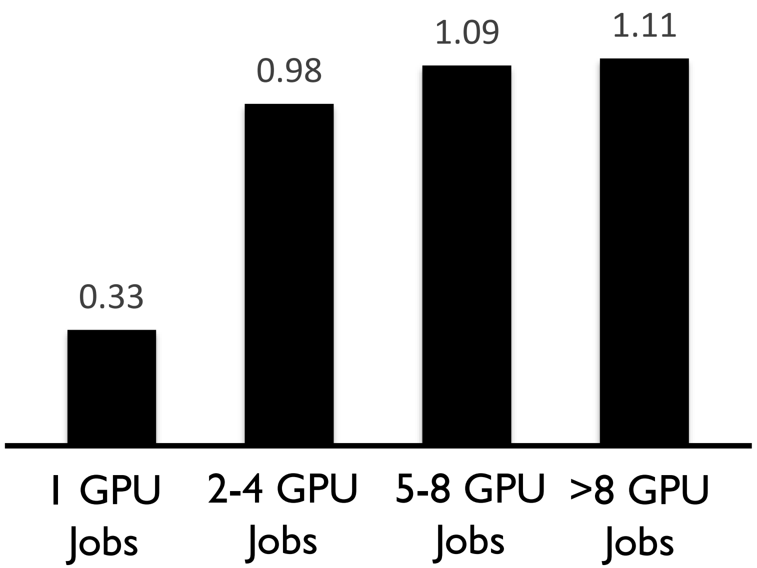
Failures occur during training

How do job failures affect cluster utilization?

Failures Can Reduce Cluster Utilization



A job is unsuccessful if it repeatedly fails (waste resources)



Average of one failure per distributed training job

Challenge: Failures across Stack

Infrastructure

Resource Scheduler

AI Engine

PyTorch

TensorFlow

mxnet

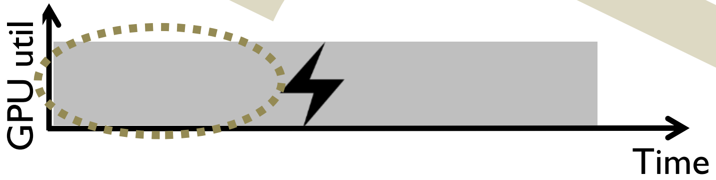
User Program

```
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1 = nn.Conv2d(1, 10, kernel_size=5)
        self.conv2 = nn.Conv2d(10, 20, kernel_size=5)
        self.conv2_drop = nn.Dropout2d()
        self.fc1 = nn.Linear(320, 50)
        self.fc2 = nn.Linear(50, 10)
```

Our study: classify into failure types and identify utilization impacts

Improve failure handling

Failure Classifier



stderr/stdout

Who - job & user ID

GPU hours - # of GPUs x Time to failure

Where - Infra? AI engine? User?

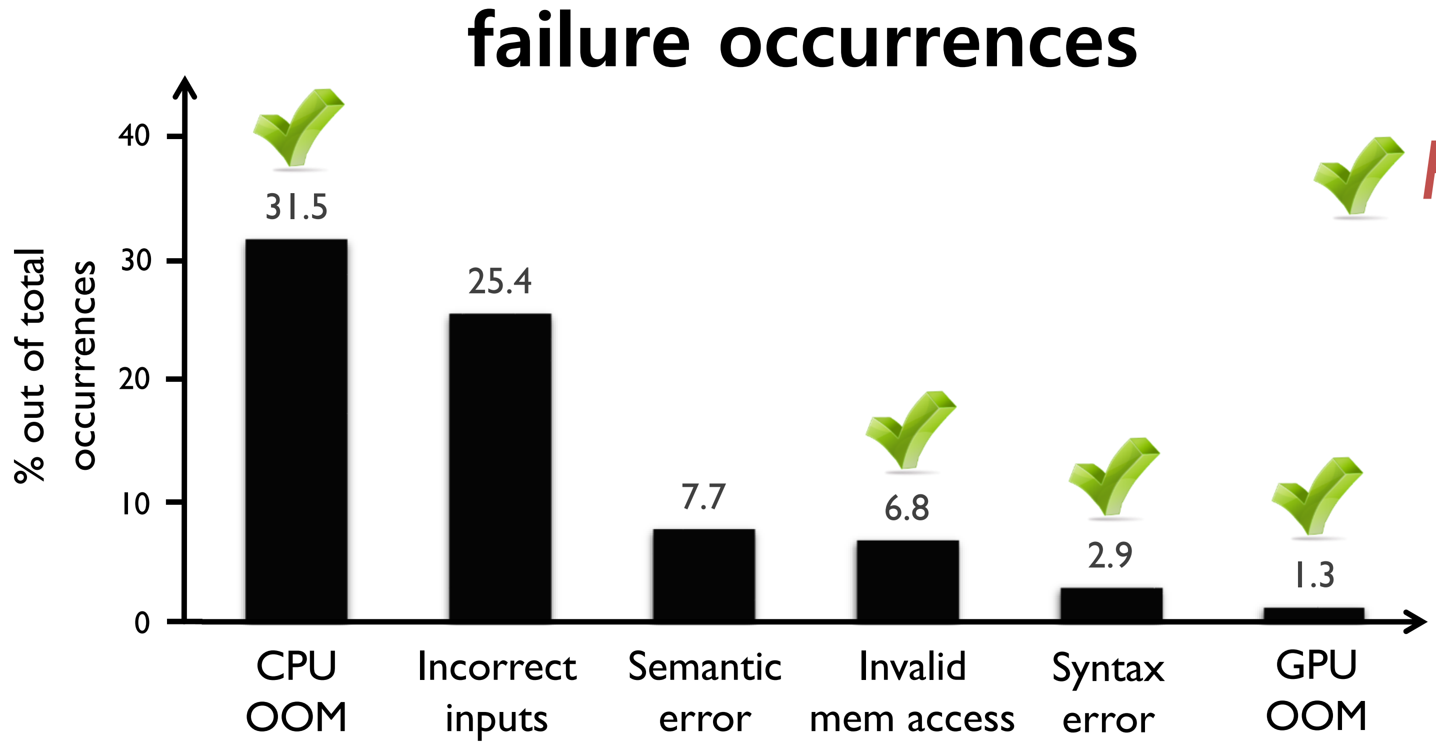
(signature, failure category)

>230 signatures

```
(["valueerror"] , "semantic_error"),
(["job preempted"] , "job_preempted"),
(["error reading from file"] , "incorrect_inputs"),
(["indexerror", "index out of range"] , "invalid_memory_access"),
(["cannot open file", "for writing"] , "model_checkpoint_error"),
(["lost communication with its daemon"] , "mpi_runtime_failure"),
....
```

Failures in High Frequency

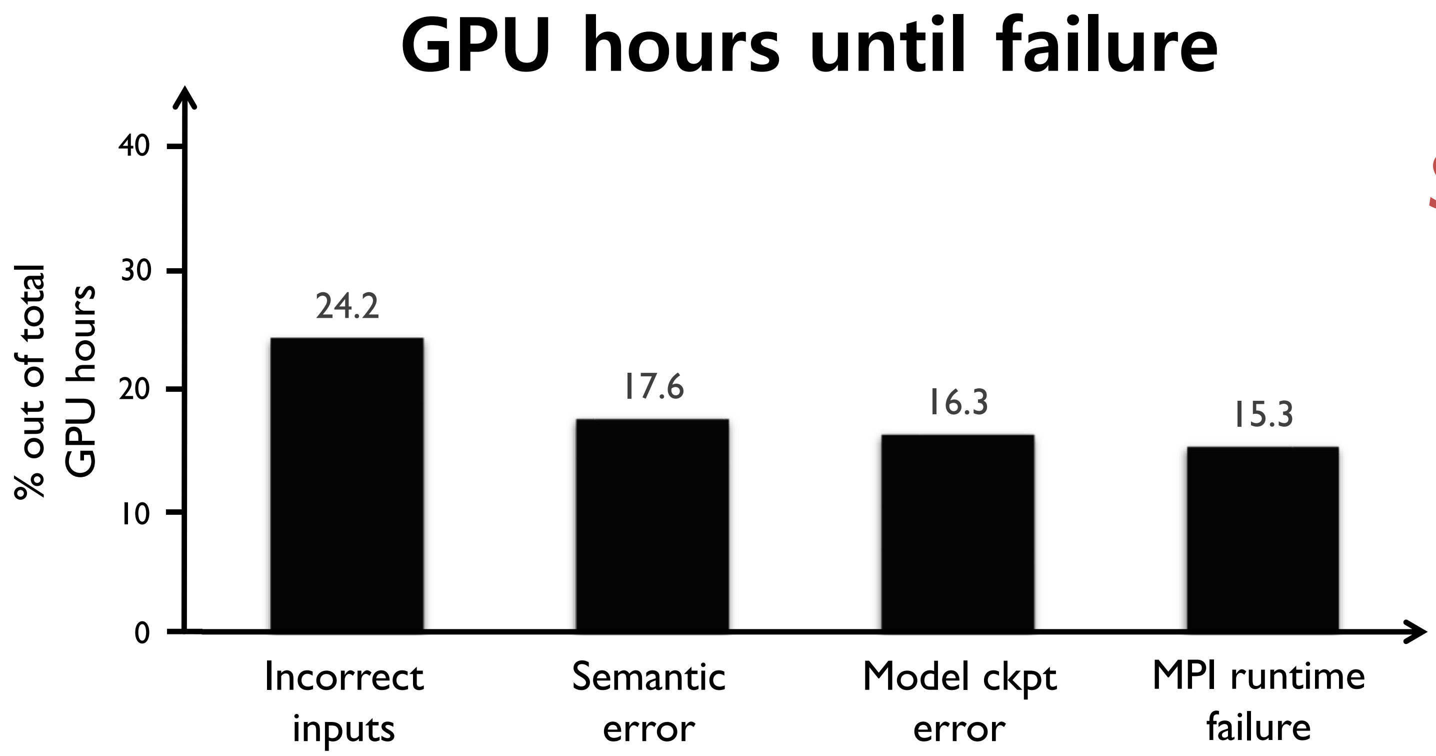
Reason: User errors in code or configuration



Repetitive and appearing early

Failures in High Resource Use

Reason: Infrastructure failures and semantics errors



Spread across many layers of system stack

Contributions

1. First characterization study of large-scale GPU clusters for DNN training

2. Study cluster utilization and how effectively GPUs are used

3. Present lessons for better cluster manager designs

Locality v.s. Waiting Time

Users prefer lower queuing delays

Initial delays can outweigh giving up locality for *long-running jobs*

		Queueing	*Run time*
example	**Low locality**	**(0 hour)**	**24 hours**
	High locality	**(1 hour)**	**16 hours**

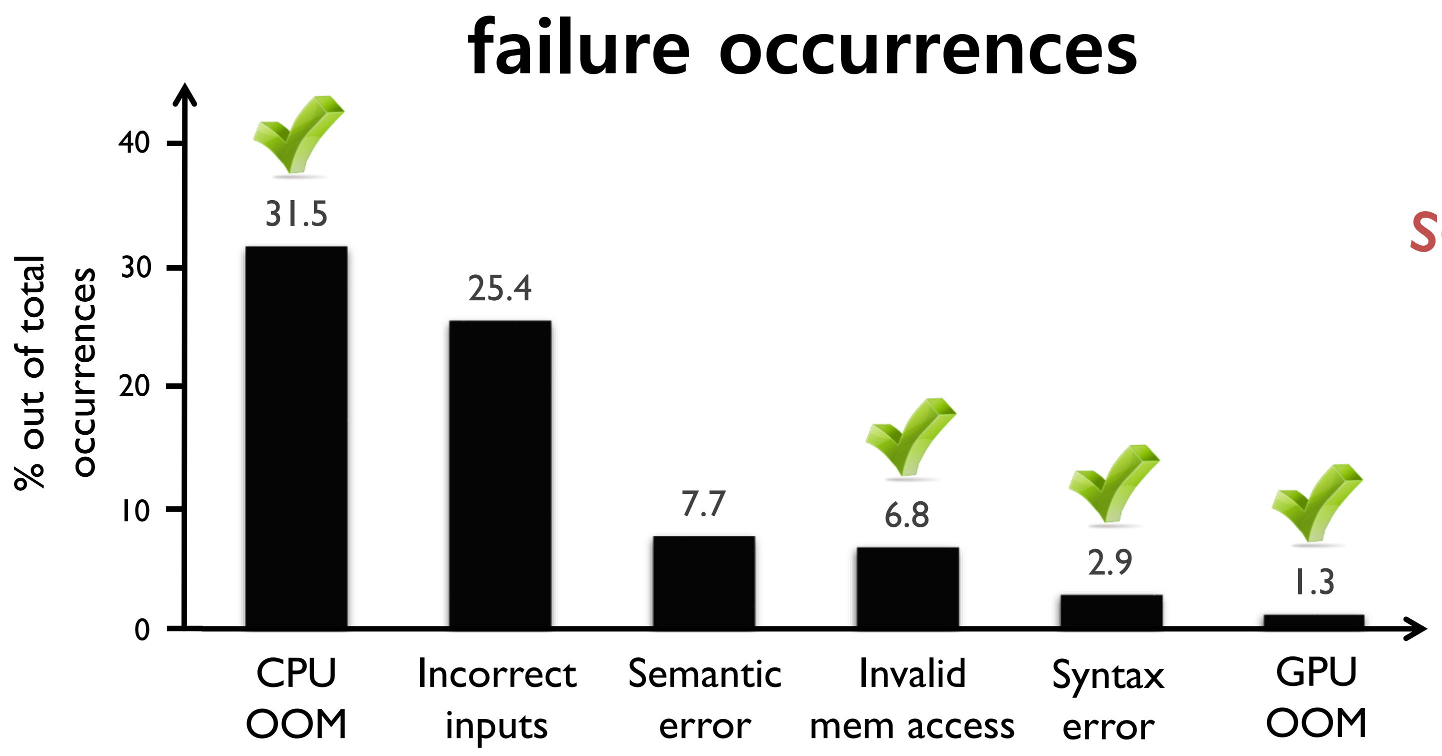
Scheduler needs to consider:

1) trade-off between queueing delay and locality-aware scheduling

2) incorporating job migration

Job Pre-Run before Scheduling

Reason: User errors in code or configuration



Simple validation before scheduling (e.g., pre-run) avoids a majority of these failures

More in the Paper

Job queueing

– Fair-share delay v.s. fragmentation delay

– Impact of out-of-order scheduling on job queueing

Job failures

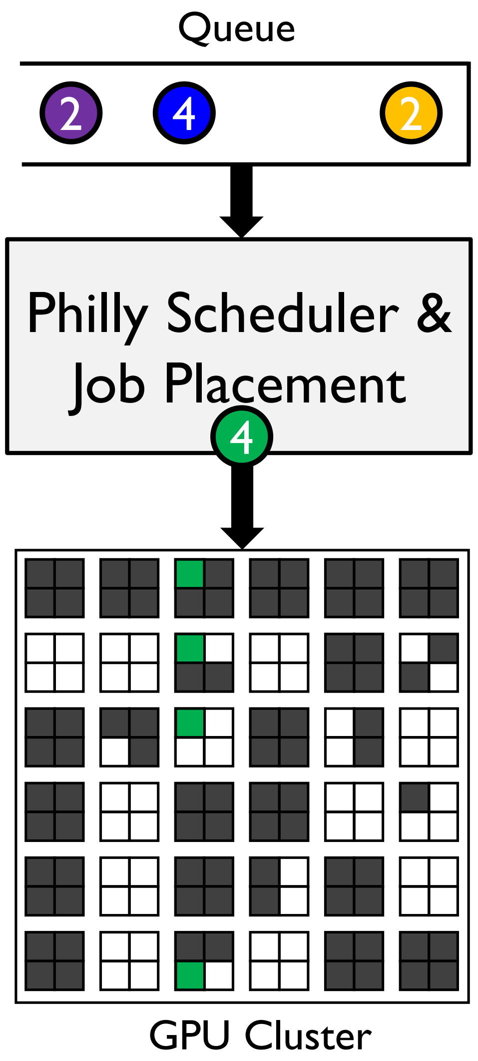
– Full classification of failures and detailed statistics

– How to mitigate failures by proactively analyzing failures at runtime

Effectiveness of the last epochs

– Opportunity to not perform the last bunch of epochs

Conclusion



1. **First characterization study of large-scale GPU clusters for DNN training**

2. **Inefficiencies come from multiple factors**

3. **Lessons on locality-awareness and failure handling**

Traces available! ☺

https://github.com/msr-fiddle/philly-traces