Tiresias

A GPU Cluster Manager for Distributed Deep Learning

Juncheng Gu, Mosharaf Chowdhury, Kang G. Shin,

Yibo Zhu, Myeongjae Jeon, Junjie Qian, Hongqiang (Harry) Liu, Chuanxiong Guo

UNIVERSITY OF MICHIGAN

Microsoft ByteDance UNIST ULSAN NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY Alibaba Group

GPU Cluster for Deep Learning Training

- Deep learning (DL) is popular
  - *10.5×* increase of DL training jobs in Microsoft
  - DL training jobs require GPU
    - Distributed deep learning (DDL) training with multiple GPUs

Google Lens

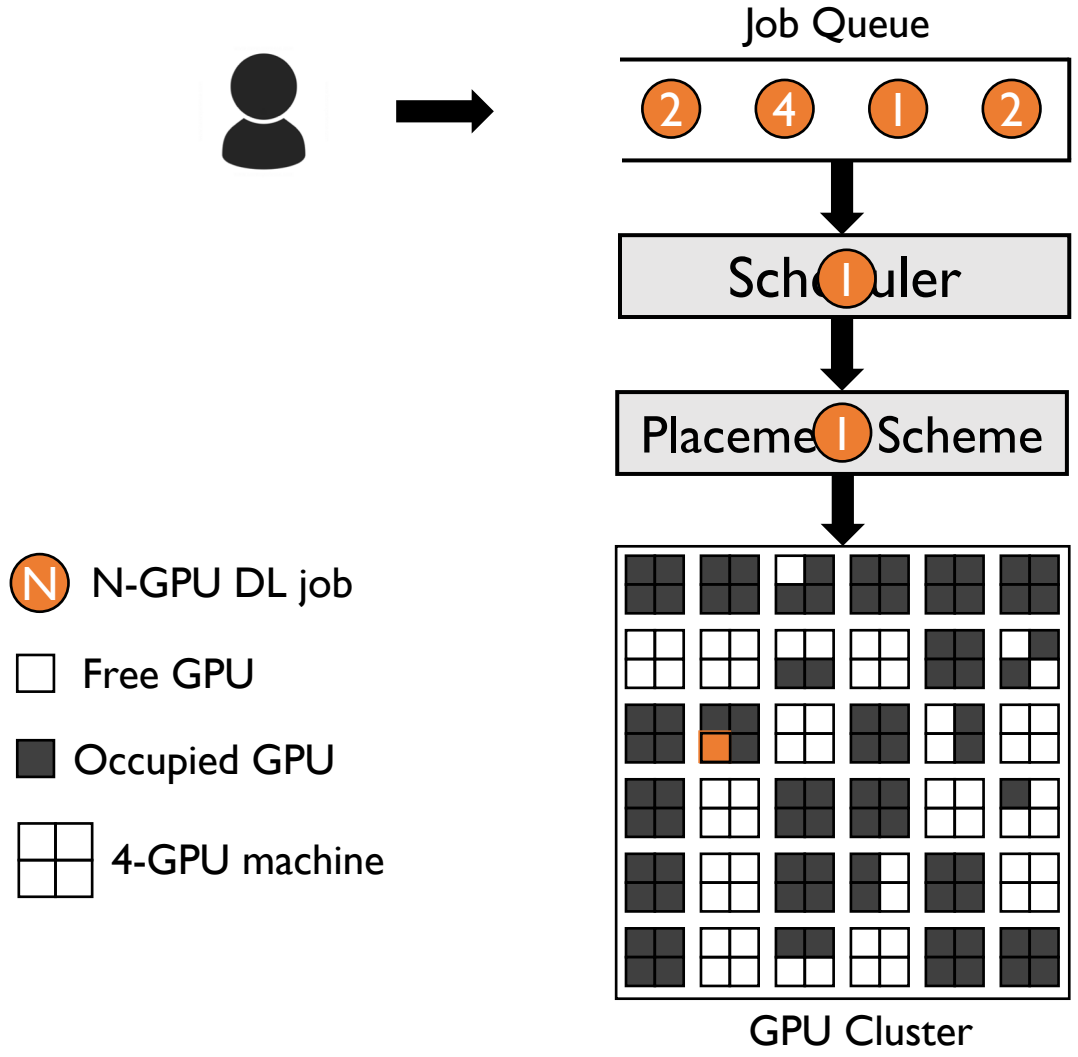
Siri

- GPU cluster for DL training
  - *5×* increase of GPU cluster scale in Microsoft [1]

How to efficiently manage a GPU cluster for DL training jobs?

[1]. Analysis of Large-Scale Multi-Tenant GPU Clusters for DNN Training Workloads. https://arxiv.org/abs/1901.05758

GPU Cluster Manager



Design Objectives

Minimize

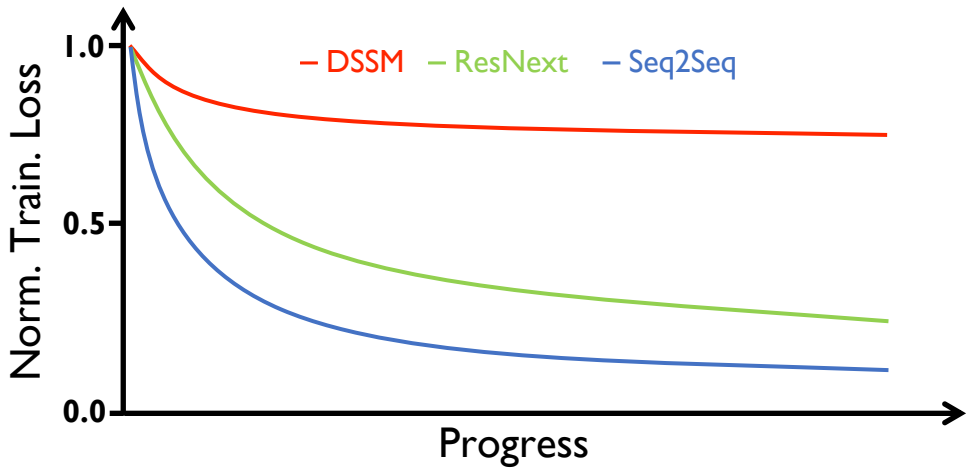
Cluster-Wide Average Job Completion Time (JCT)

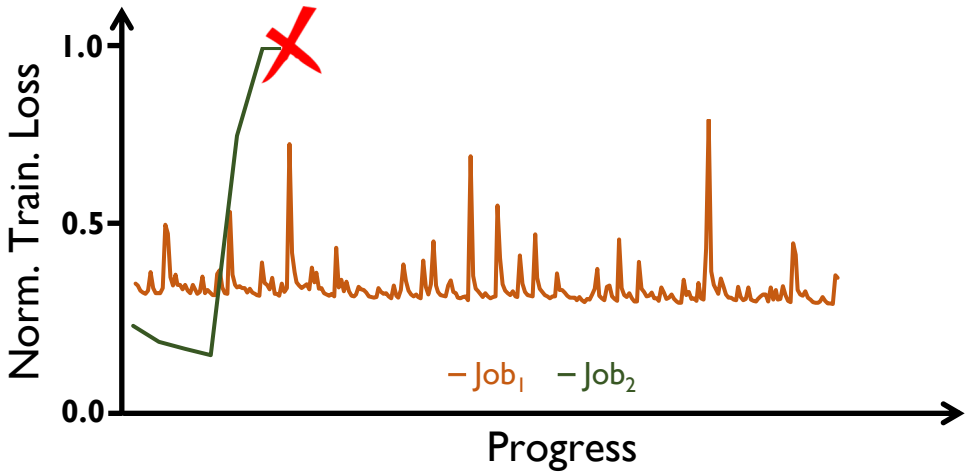
Achieve

High Resource (GPU) Utilization

Challenge I: Unpredictable Training Time

- Unknown execution time of DL training jobs
  - Job execution time is useful when minimizing JCT
- Predict job execution time
  - Use the smooth loss curve of DL training jobs (*Optimus* [1])

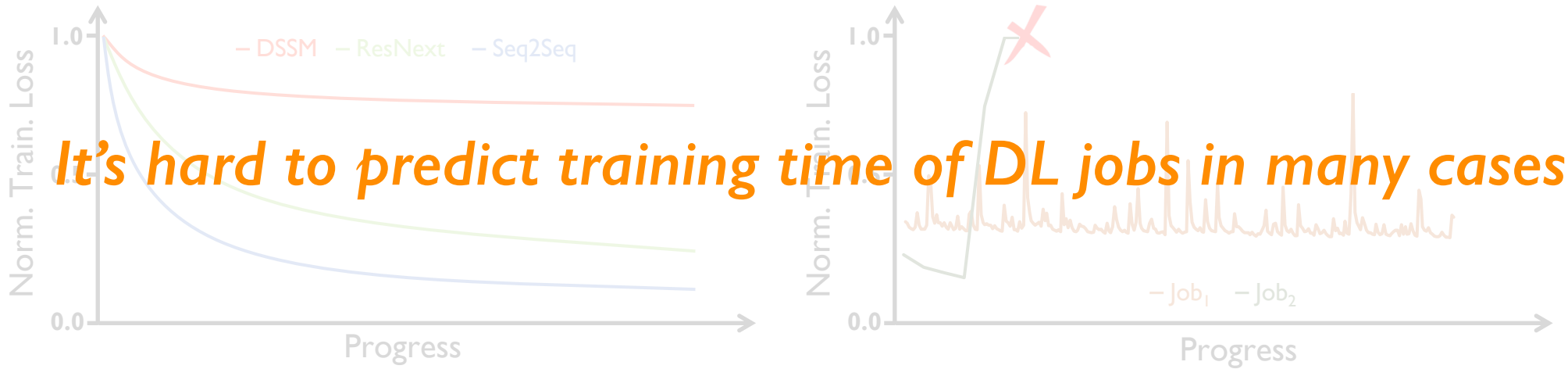




[1]. Optimus: An Efficient Dynamic Resource Scheduler for Deep Learning Clusters, EuroSys'18

Challenge I: Unpredictable Training Time

- Unknown execution time of DL training jobs
  - Job execution time is useful when minimizing JCT
- Predict job execution time
  - Use the smooth loss curve of DL training jobs (*Optimus* [1])

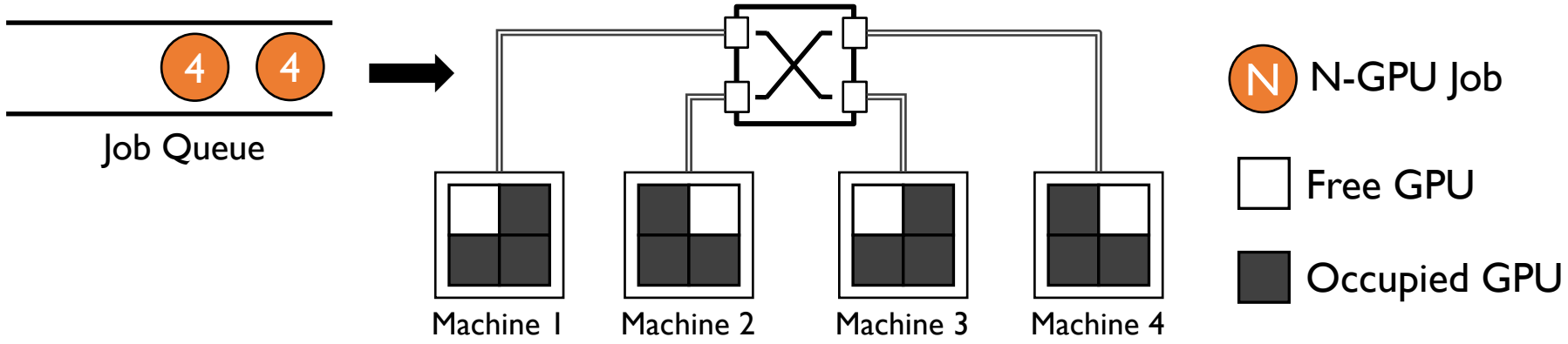


It's hard to predict training time of DL jobs in many cases

[1]. Optimus: An Efficient Dynamic Resource Scheduler for Deep Learning Clusters, EuroSys'18

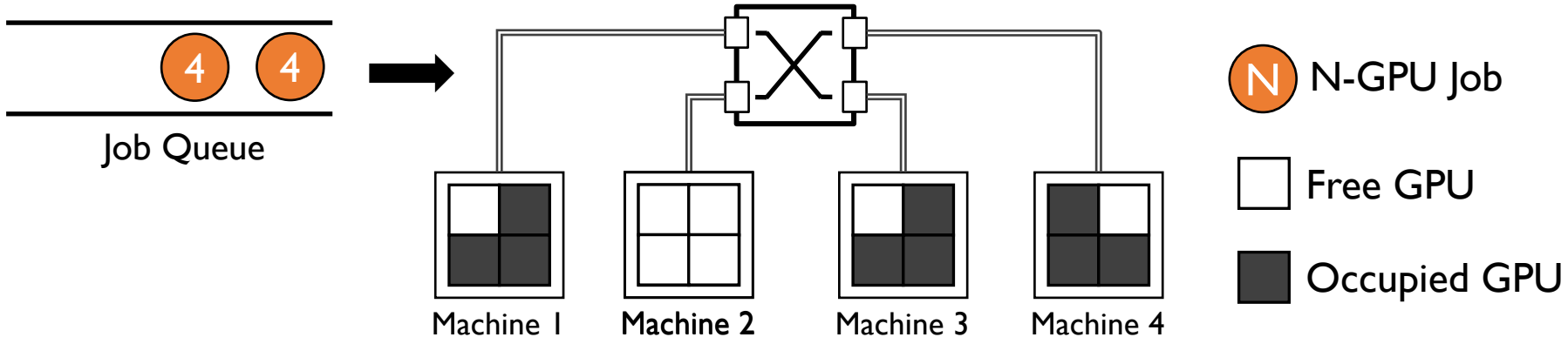
Challenge II: Over-Aggressive Job Consolidation

- Network overhead in DDL training



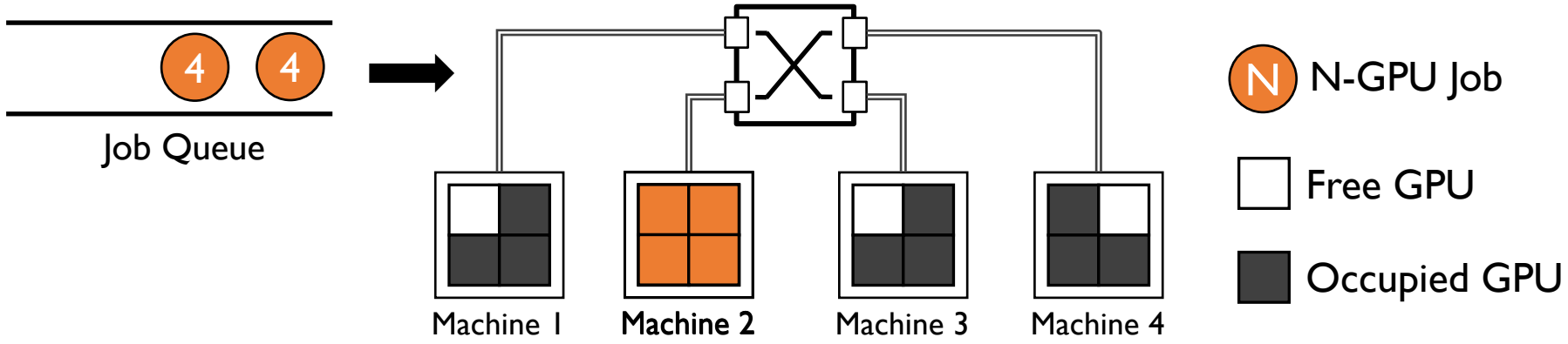
Challenge II: Over-Aggressive Job Consolidation

- Network overhead in DDL training
- *Consolidated placement* for good training performance



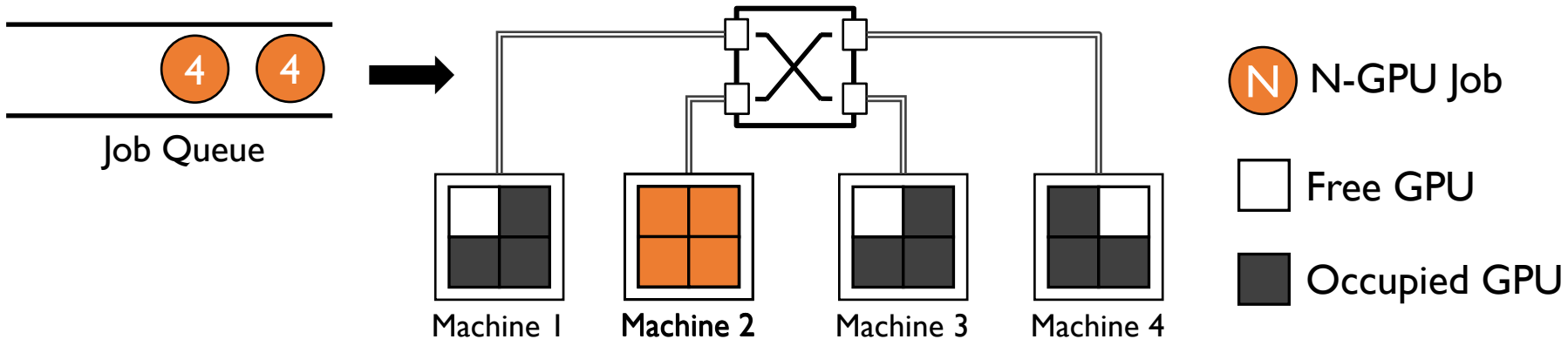
Challenge II: Over-Aggressive Job Consolidation

- Network overhead in DDL training
- *Consolidated placement* for good training performance



Challenge II: Over-Aggressive Job Consolidation

- Network overhead in DDL training
- *Consolidated placement* for good training performance
  - *Fragmented free GPUs in the cluster*
  - *Longer queuing delay*



Prior Solutions

	I. *Unpredictable Training Time* (*Scheduling*)	II. *Over-Aggressive Job Consolidation* (*Job Placement*)
Optimus[1]	None	None
YARN-CS	*FIFO*	None
Gandiva[2]	*Time-sharing*	*Trial-and-error*

[1]. Optimus: An Efficient Dynamic Resource Scheduler for Deep Learning Clusters, EuroSys'18
[2]. Gandiva: Introspective Cluster Scheduling for Deep Learning, OSDI'18

Tiresias

A GPU cluster manager for Distributed Deep Learning Without Complete Knowledge

1. Age-Based Scheduler

   Minimize JCT without complete knowledge of jobs

2. Model Profile-Based Placement

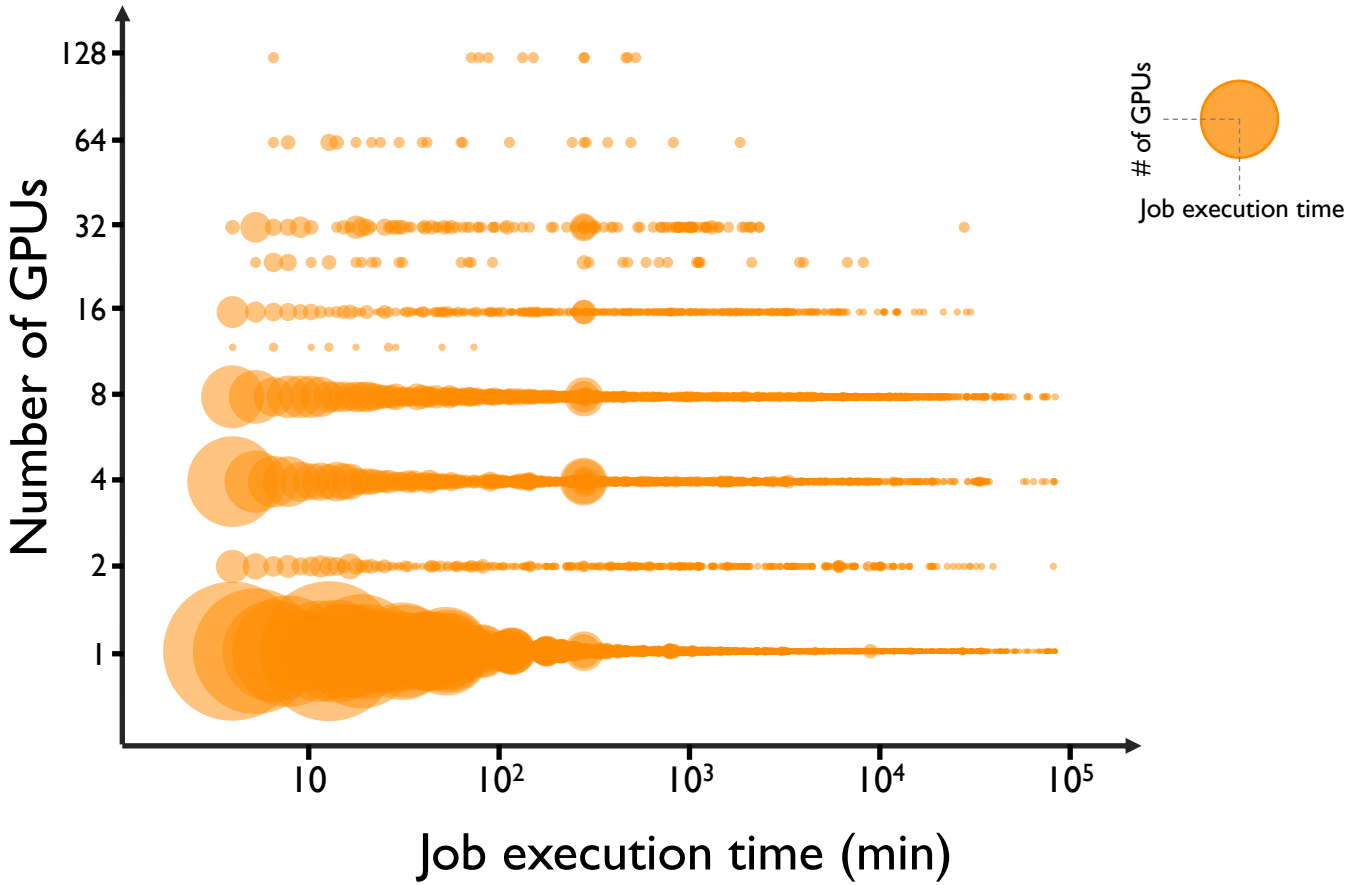
   Place jobs without additional information from users

Challenge 1

How To Schedule DL Training Jobs Without Complete Job Information?

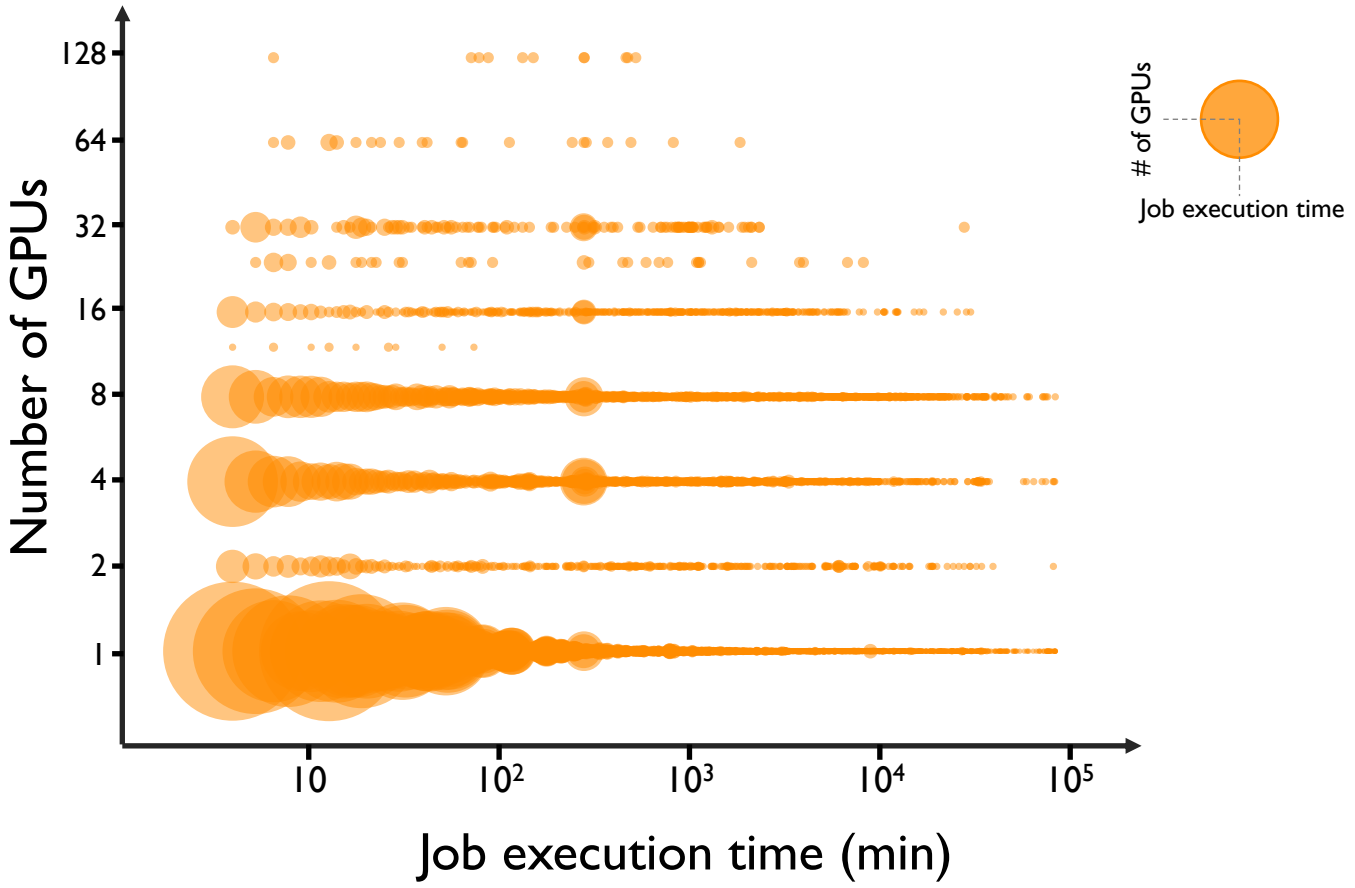
Characteristics of DL Training Jobs

- Variations in both temporal and spatial aspects



Characteristics of DL Training Jobs

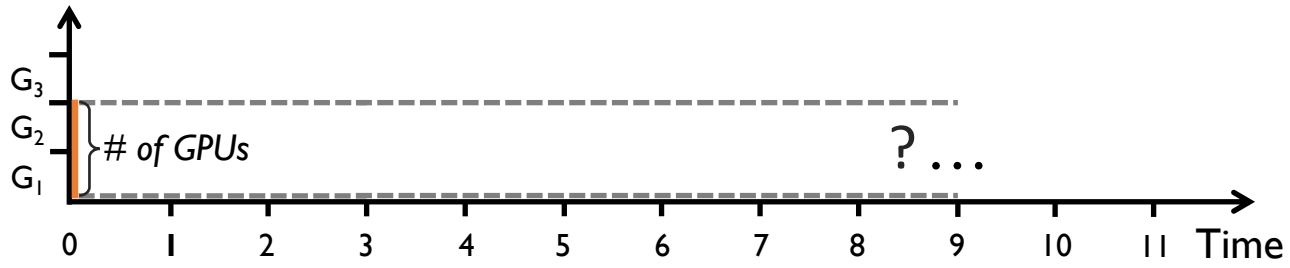
- Variations in both temporal and spatial aspects



Scheduler should consider both ***temporal and spatial*** *aspects of DL training jobs*

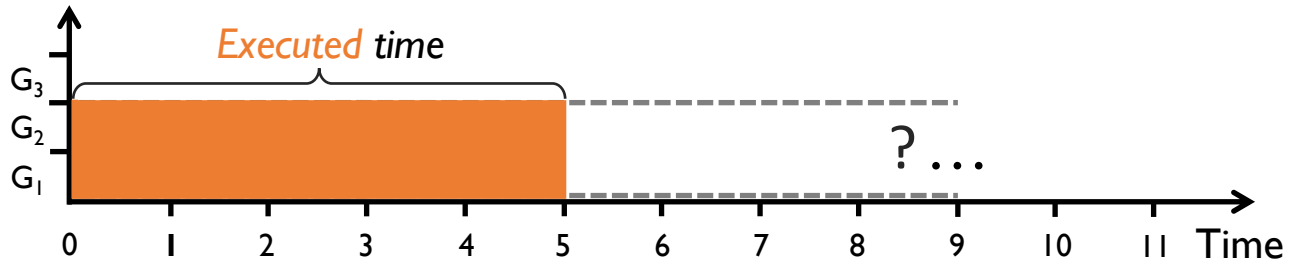
Available Job Information

1. Spatial: number of GPUs



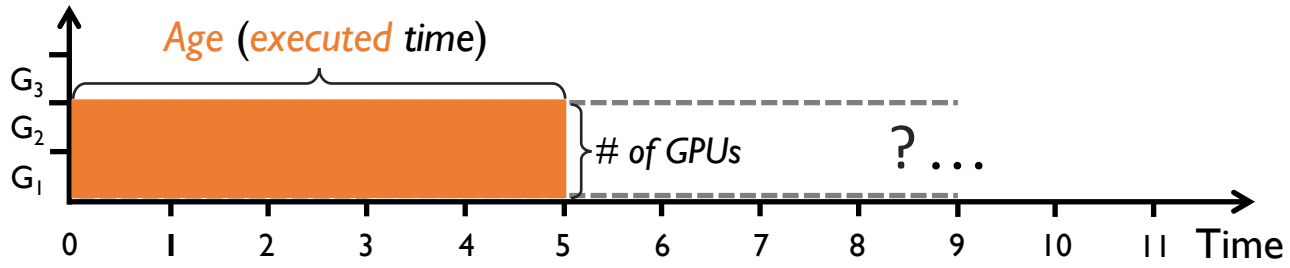
Available Job Information

1. Spatial: number of GPUs
2. Temporal: ***executed*** time



Age-Based Schedulers

- *Least-Attained Service*[1] (LAS)
  - Prioritize job that has the shortest executed time
- *Gittins Index policy*[2]
  - Need the ***distribution of job execution time***
  - Prioritize job that has the highest probability to complete in the near future



[1]. Feedback queueing models for time-shared systems. JACM, 1968
[2]. Multi-armed bandit allocation indices. Wiley, Chichester, 1989

Two-Dimensional Age-Based Scheduler (2DAS)

- Age calculated by two-dimensional attained service
  - i.e., a job's *total executed GPU time* (# of GPUs × executed time)
- No prior information
  - *2D-LAS*
- With partial information: distribution of job GPU time
  - *2D-Gittins Index*

2D-Gittins Index: Partial Information

- Higher ***probability to complete*** (***Gittins Index***), higher priority

	# of GPUs	Execution time
J_1	2	2
J_2	1	8
J_3	2	6

2D-Gittins Index: Partial Information

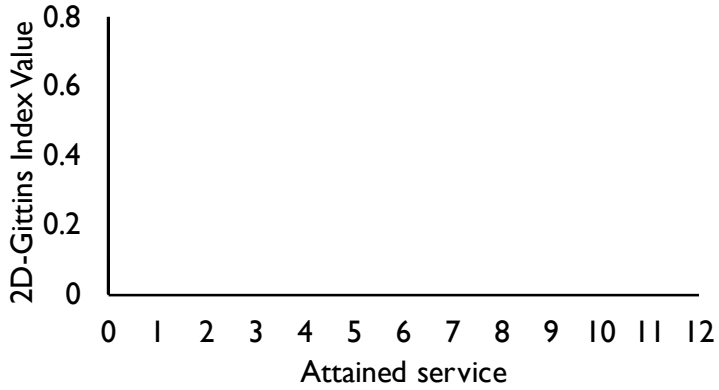
- Higher ***probability to complete*** (***Gittins Index***), higher priority

	# of GPUs	Distribution
J_1	2	2
J_2	1	(4, 8, 12)
J_3	2	6

2D-Gittins Index: Partial Information

- Higher ***probability to complete*** (***Gittins Index***), higher priority

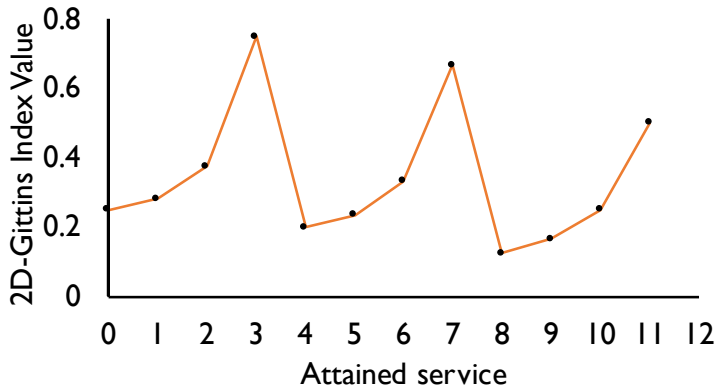
	# of GPUs	Distribution	Attained Service
J_1	2	2	0
J_2	1	(4, 8, 12)	0
J_3	2	6	0



2D-Gittins Index: Partial Information

- Higher ***probability to complete*** (***Gittins Index***), higher priority

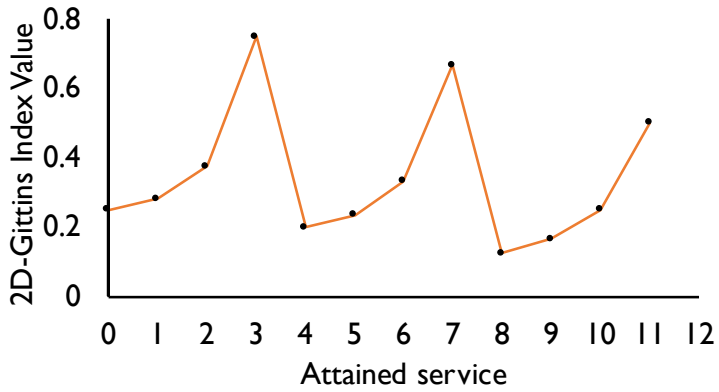
	# of GPUs	Distribution	Attained Service
J_1	2	2	0
J_2	1	(4, 8, 12)	0
J_3	2	6	0



2D-Gittins Index: Partial Information

- Higher ***probability to complete*** (***Gittins Index***), higher priority

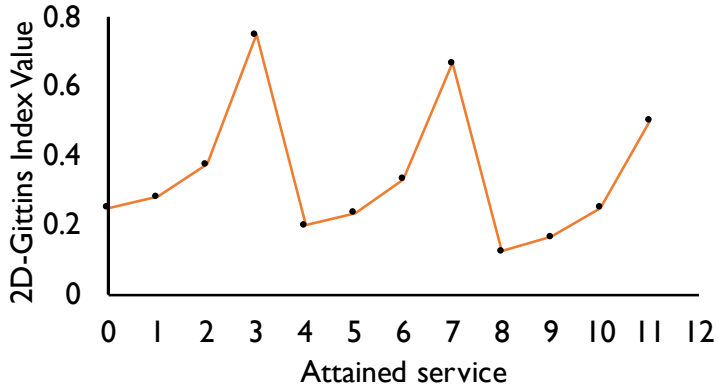
	# of GPUs	Distribution	Attained Service	Gittins Index
J_1	2	2	0	0.25
J_2	1	(4, 8, 12)	0	0.25
J_3	2	6	0	0.25

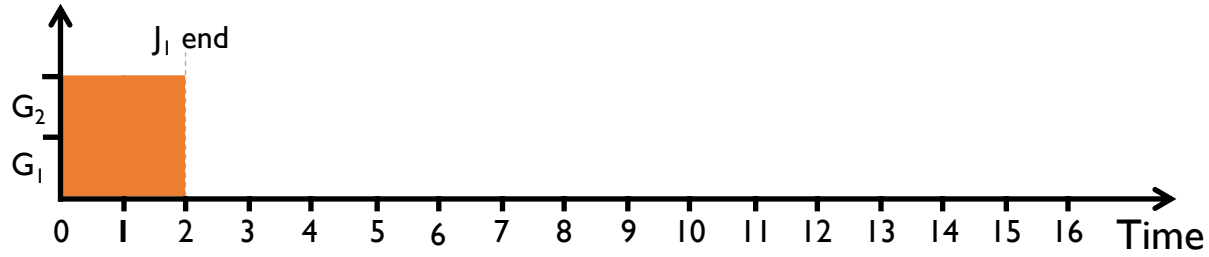


2D-Gittins Index: Partial Information

- Higher ***probability to complete*** (***Gittins Index***), higher priority

	# of GPUs	Distribution	Attained Service	Gittins Index
J_1	2	2	4	0.2
J_2	1	(4, 8, 12)	0	0.25
J_3	2	6	0	0.25

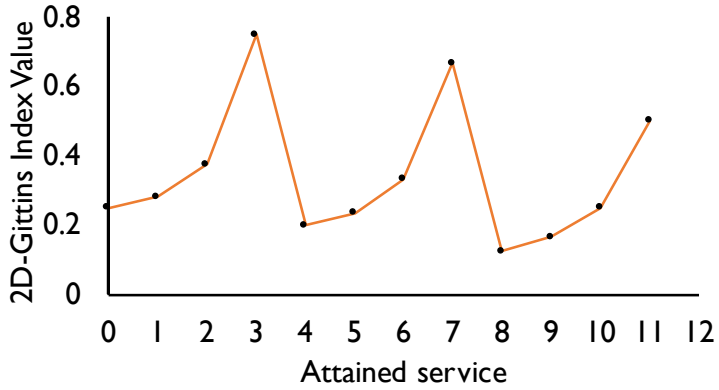


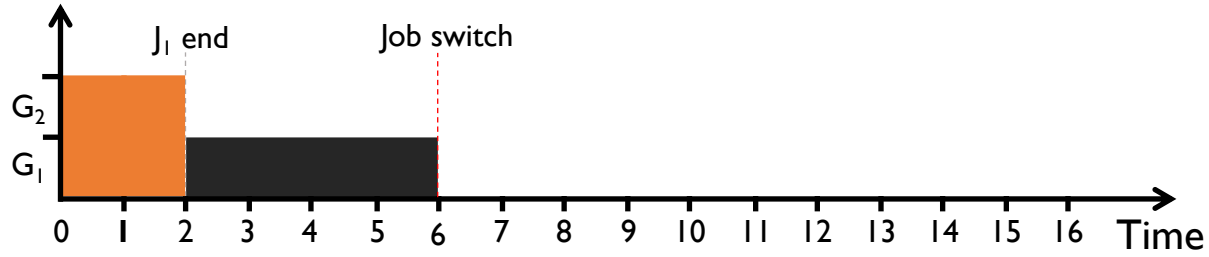


2D-Gittins Index: Partial Information

- Higher ***probability to complete*** (***Gittins Index***), higher priority

	# of GPUs	Distribution	Attained Service	Gittins Index
J_1	2	2	4	0.2
J_2	1	(4, 8, 12)	4	0.2
J_3	2	6	0	0.25

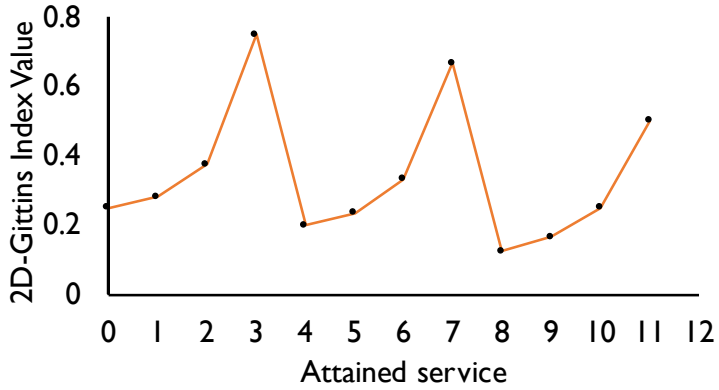


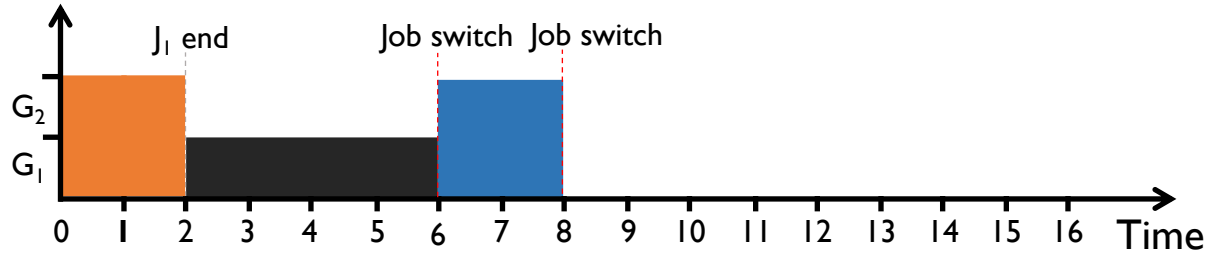


2D-Gittins Index: Partial Information

- Higher ***probability to complete*** (***Gittins Index***), higher priority

	# of GPUs	Distribution	Attained Service	Gittins Index
J_1	2	2	4	0.2
J_2	1	(4, 8, 12)	4	0.2
J_3	2	6	4	0.2

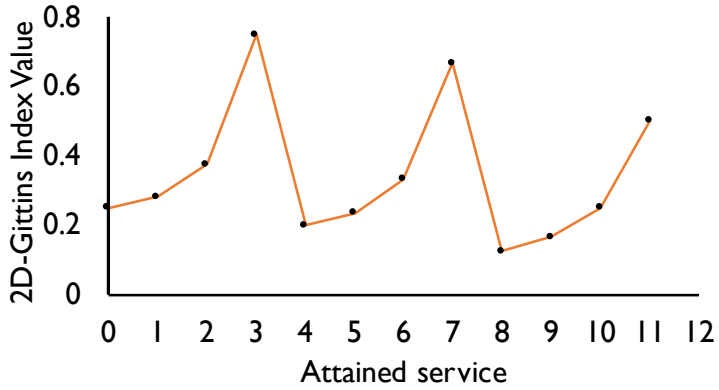


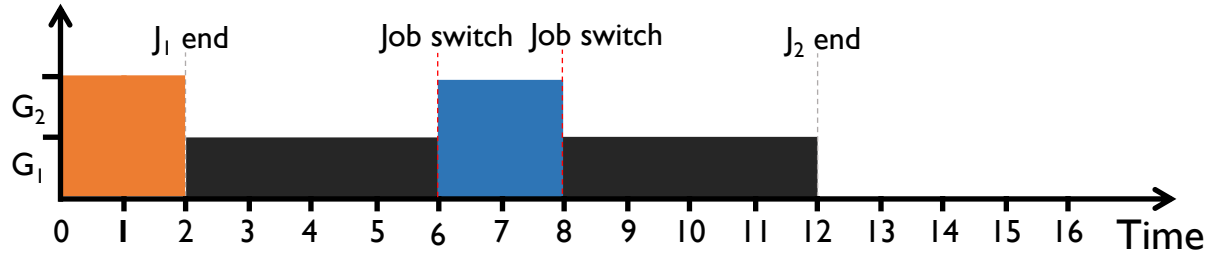


2D-Gittins Index: Partial Information

- Higher ***probability to complete*** (***Gittins Index***), higher priority

	# of GPUs	Distribution	Attained Service	Gittins Index
J_1	2	2	4	0.2
J_2	1	(4, 8, 12)	8	0.125
J_3	2	6	4	0.2

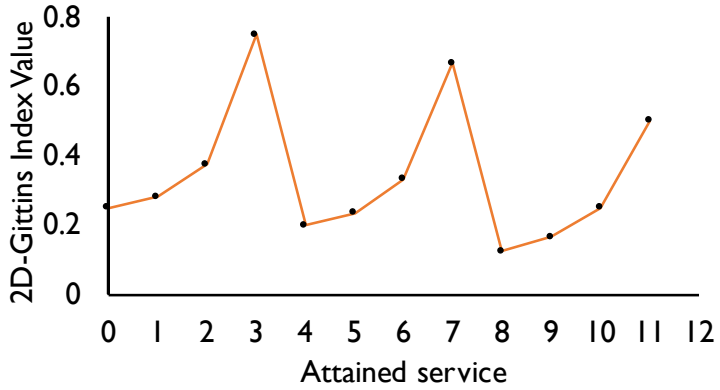


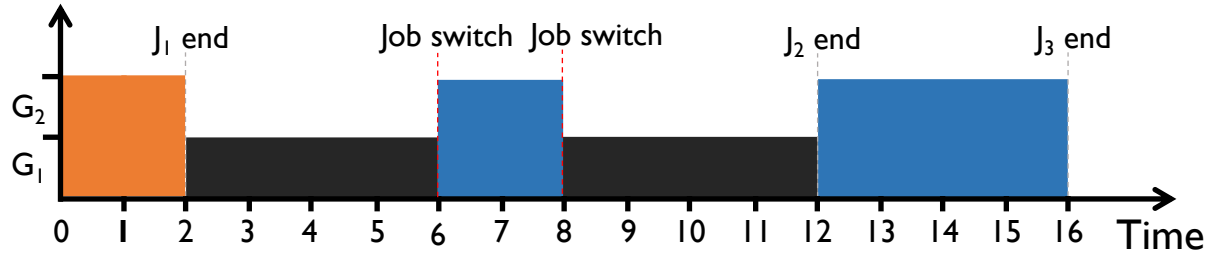


2D-Gittins Index: Partial Information

- Higher ***probability to complete*** (***Gittins Index***), higher priority

	# of GPUs	Distribution	Attained Service	Gittins Index
J_1	2		4	0.2
J_2	1	(4, 8, 12)	8	0.125
J_3	2		12	N/A



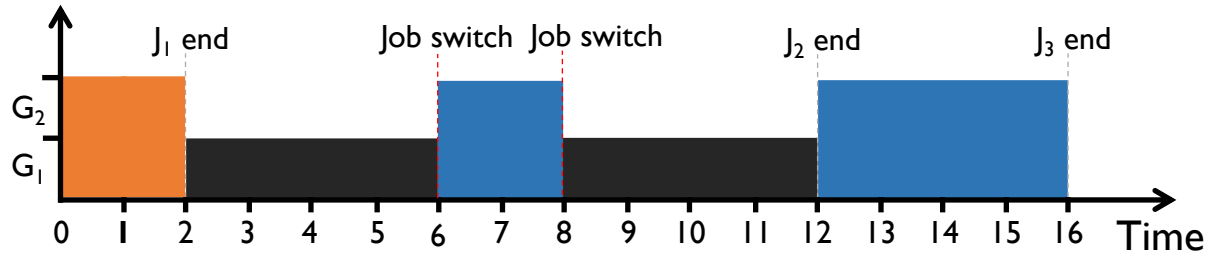


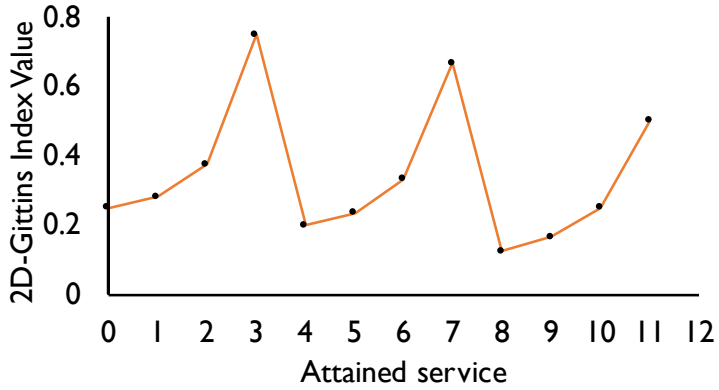
	Extra Information	Avg. JCT
2D-Gittins Index	GPU time distribution	10.0

2D-Gittins Index: Partial Information

- Higher ***probability to complete*** (***Gittins Index***), higher priority

	# of GPUs	Distribution	Attained Service	Gittins Index
J_1	2	(4, 8, 12)	4	0.2
J_2	1		8	0.125
J_3	2		12	N/A





	Extra Information	Avg. JCT
2D-Gittins Index	GPU time distribution	10.0
2D-LAS	None	11.7

Two-Dimensional Age-Based Scheduler (2DAS)

- Age calculated by two-dimensional attained service
  - i.e., a job's *total executed GPU time* (# of GPUs × executed time)
- No prior information
  - *2D-LAS*
- With partial information: distribution of job GPU time
  - *2D-Gittins Index*
- Fewer job switches
  - Priority discretization: *Discretized-2DAS*

Prior Solutions

	I. *Unpredictable Training Time* (*Scheduling*)	II. *Over-Aggressive Job Consolidation* (*Job Placement*)
Optimus[1]	None	None
YARN-CS	*FIFO*	None
Gandiva[2]	*Time-sharing*	*Trial-and-error*
Tiresias	*Discretized-2DAS*	

[1]. Optimus: An Efficient Dynamic Resource Scheduler for Deep Learning Clusters, EuroSys'18
[2]. Gandiva: Introspective Cluster Scheduling for Deep Learning, OSDI'18

Prior Solutions

	I. *Unpredictable Training Time* (*Scheduling*)		II. *Over-Aggressive Job Consolidation* (*Job Placement*)
Optimus[1]	None		None
YARN-CS	*FIFO*		None
Gandiva[2]	*Time-sharing*		*Trial-and-error*
Tiresias	*LAS*	*Gittins Index*	*?*

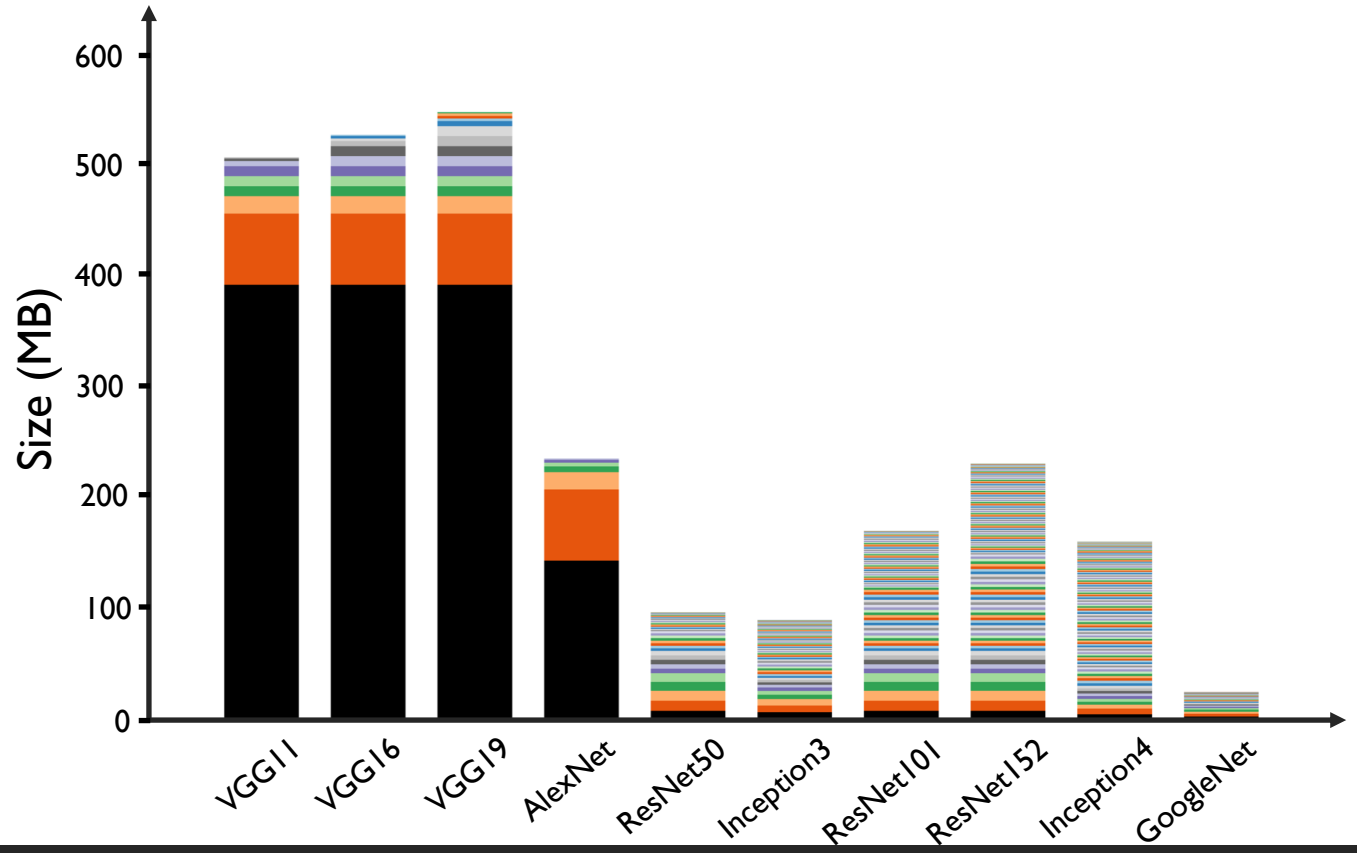
[1]. Optimus: An Efficient Dynamic Resource Scheduler for Deep Learning Clusters, EuroSys'18
[2]. Gandiva: Introspective Cluster Scheduling for Deep Learning, OSDI'18

Challenge II

How to Place DL Jobs Without Hurting Training Performance?

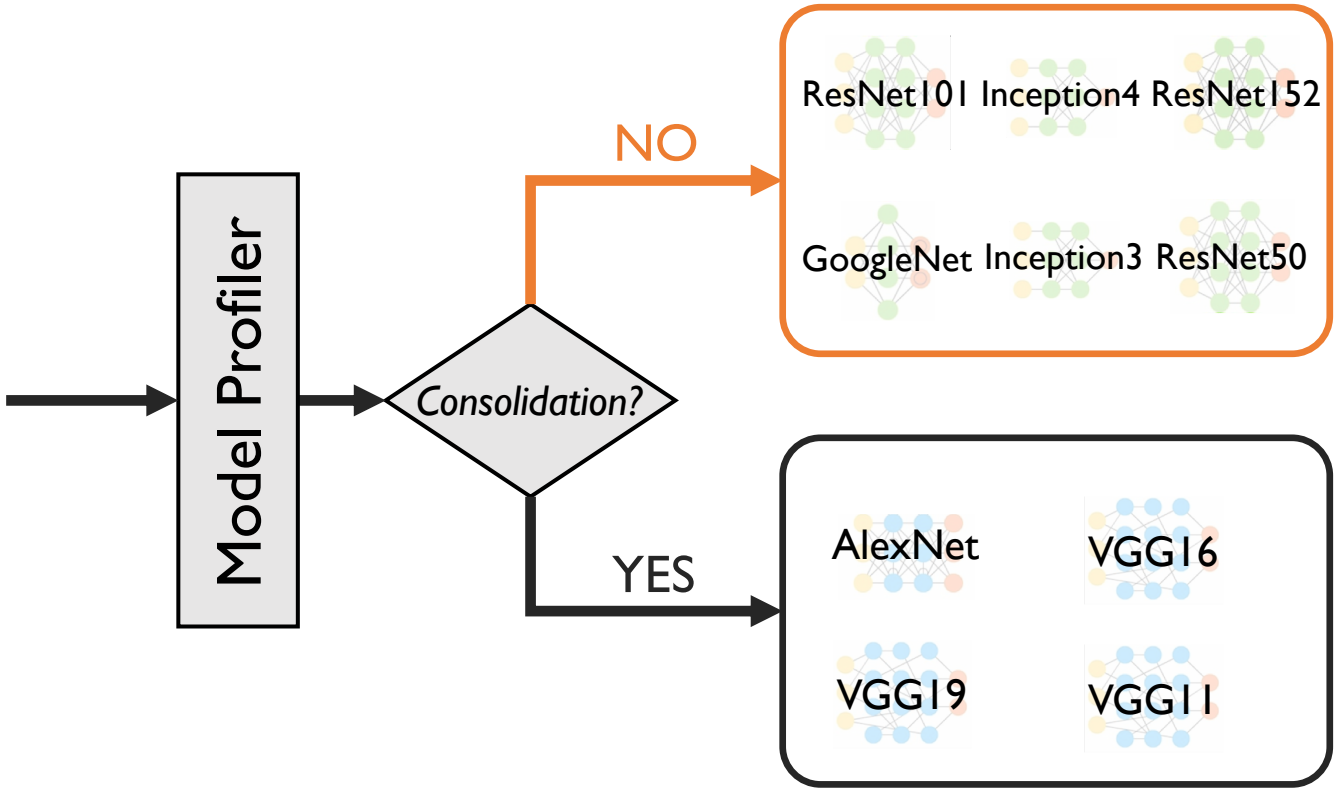
Characteristics of DL Models

- Tensor size in DL models
  - *Large tensors* cause network imbalance and contention



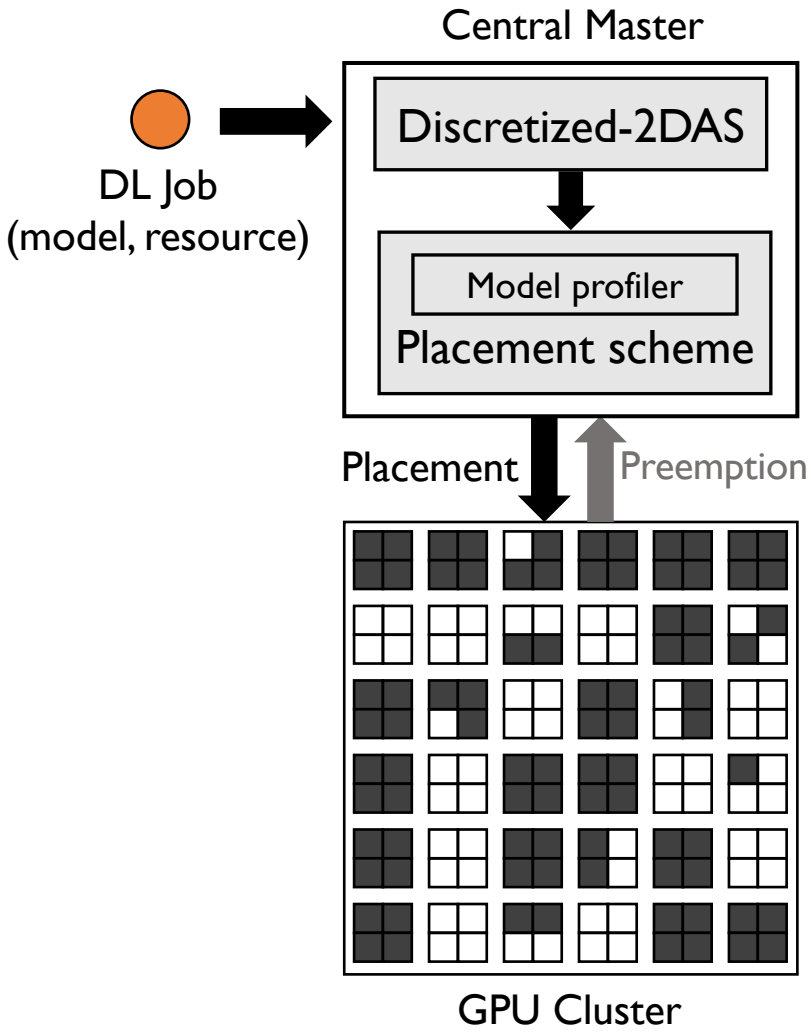
Consolidated placement is needed when the model is *highly skewed* in its tensor size

Model Profile-Based Placement



Tiresias

Central Master
Network-Level Model Profiler



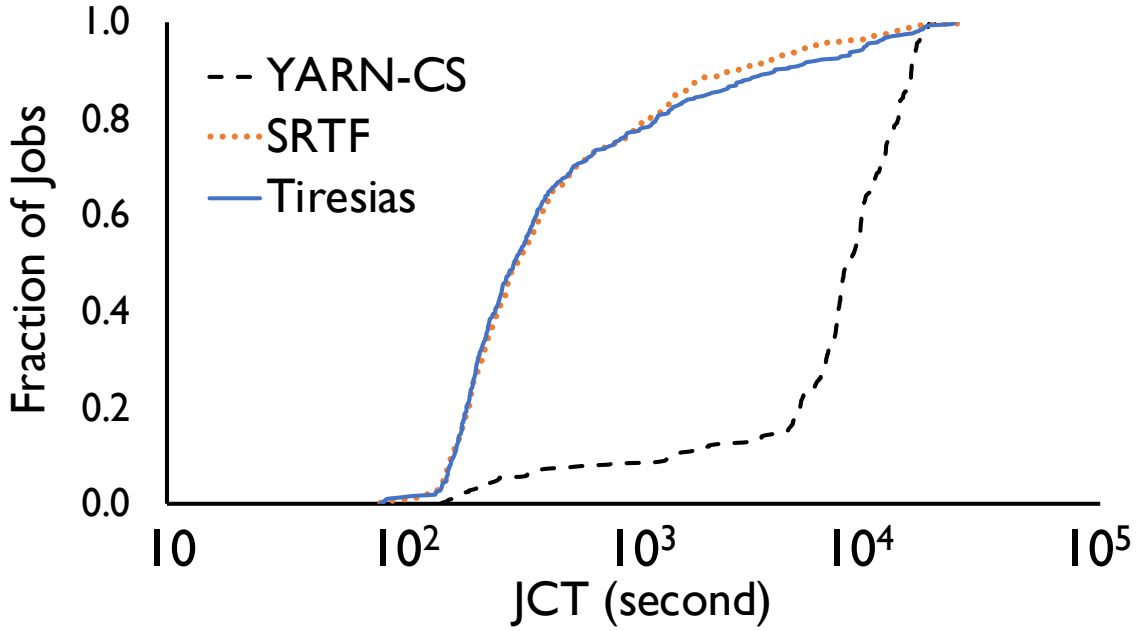
Evaluation

60-GPU Testbed Experiment

Large-scale & *Trace-driven* Simulation

JCT Improvements in Testbed Experiment

- Testbed – Michigan ConFlux cluster
  - 15 machines (4 GPUs each)
  - 100 Gbps RDMA network

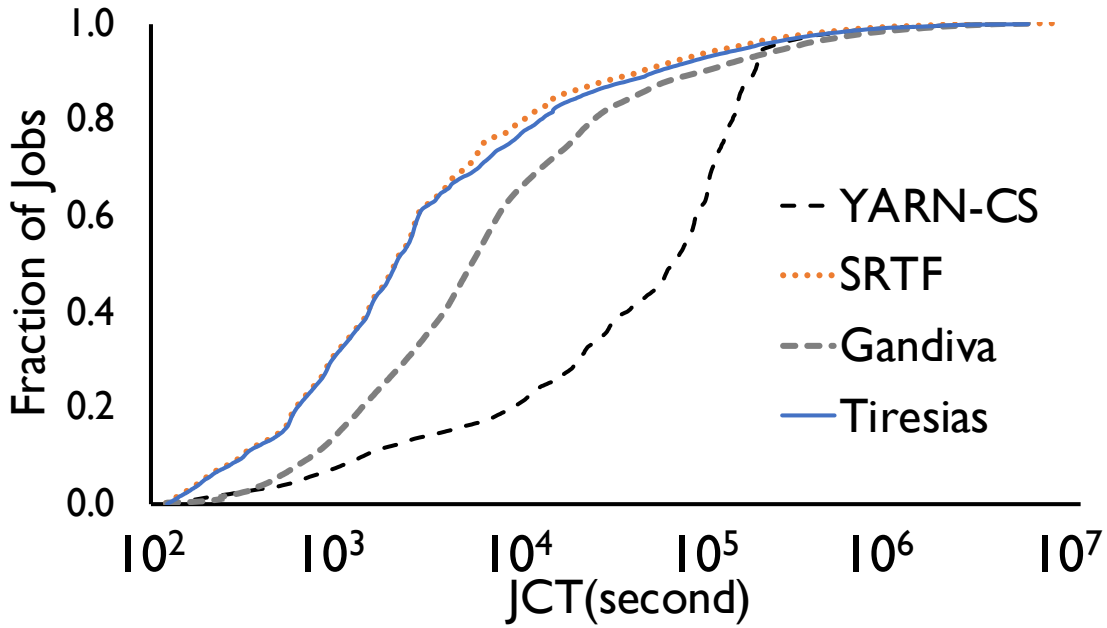


Avg. JCT improvement (w.r.t. YARN-CS): 5.5×

Comparable performance to SRTF

JCT Improvements in Trace-Driven Simulation

- Discrete-time simulator
  - 10-week job trace from Microsoft
  - 2,000-GPU cluster



Avg. JCT improvement (w.r.t. Gandiva): 2×

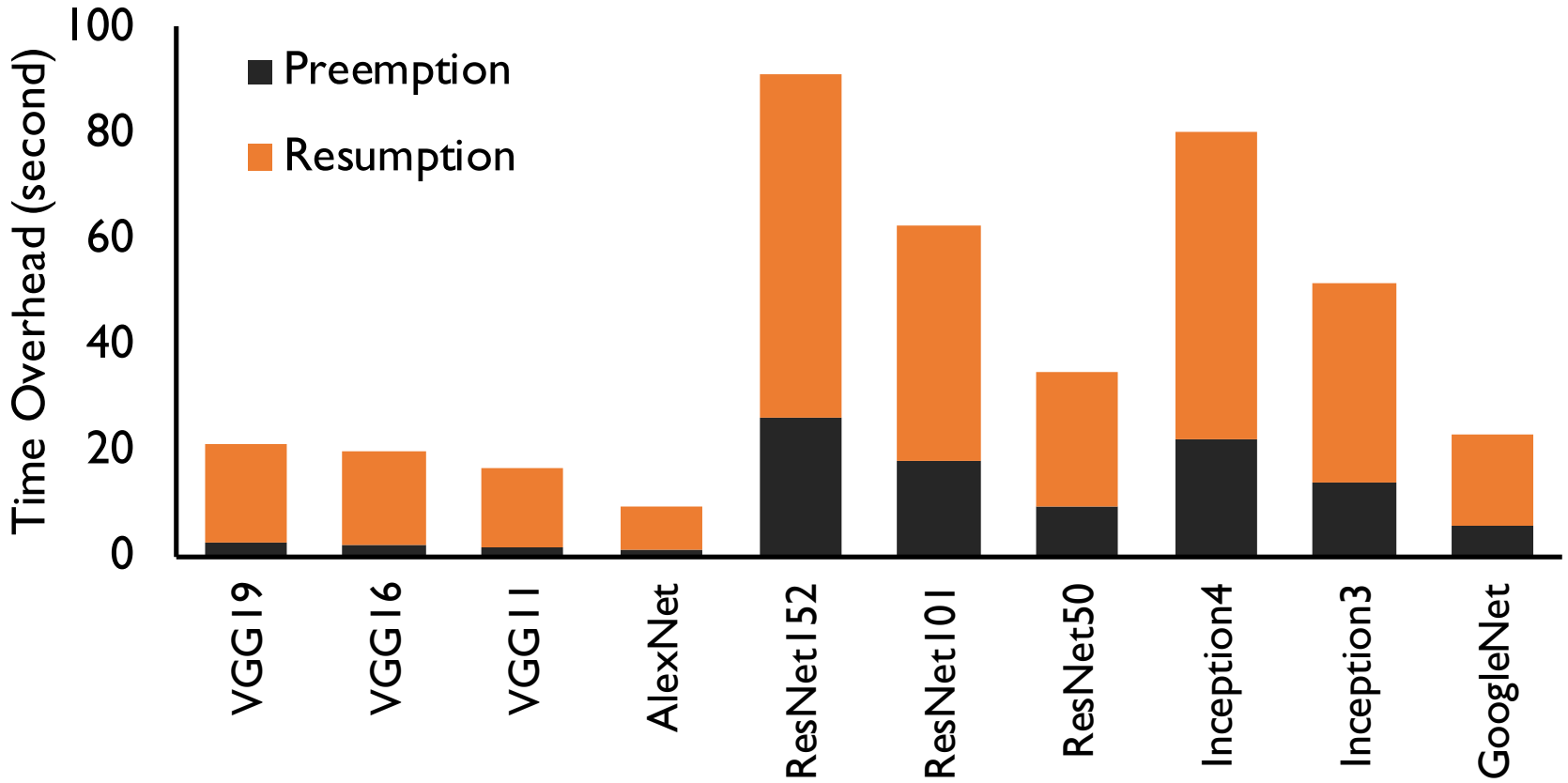
Tiresias

A GPU cluster manager for Distributed Deep Learning Without Complete Knowledge

- Optimize JCT with no or partial job information
- Relax placement constraint without hurting training performance
- Simple, practical, and with significant performance improvements

https://github.com/SymbioticLab/Tiresias

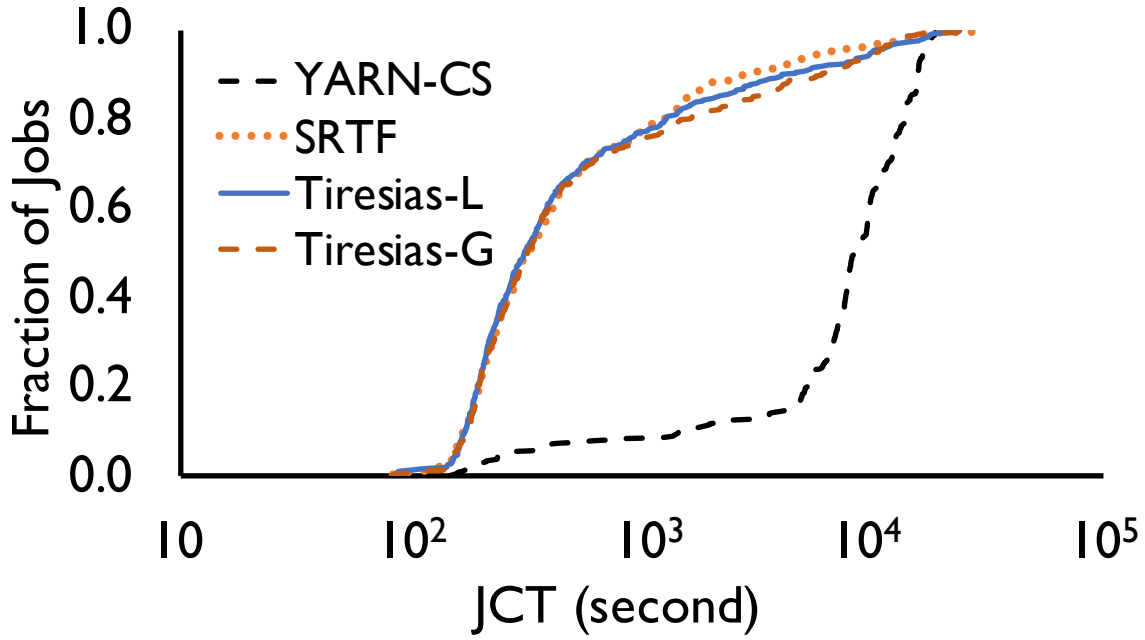
Time Overhead of Job Switch



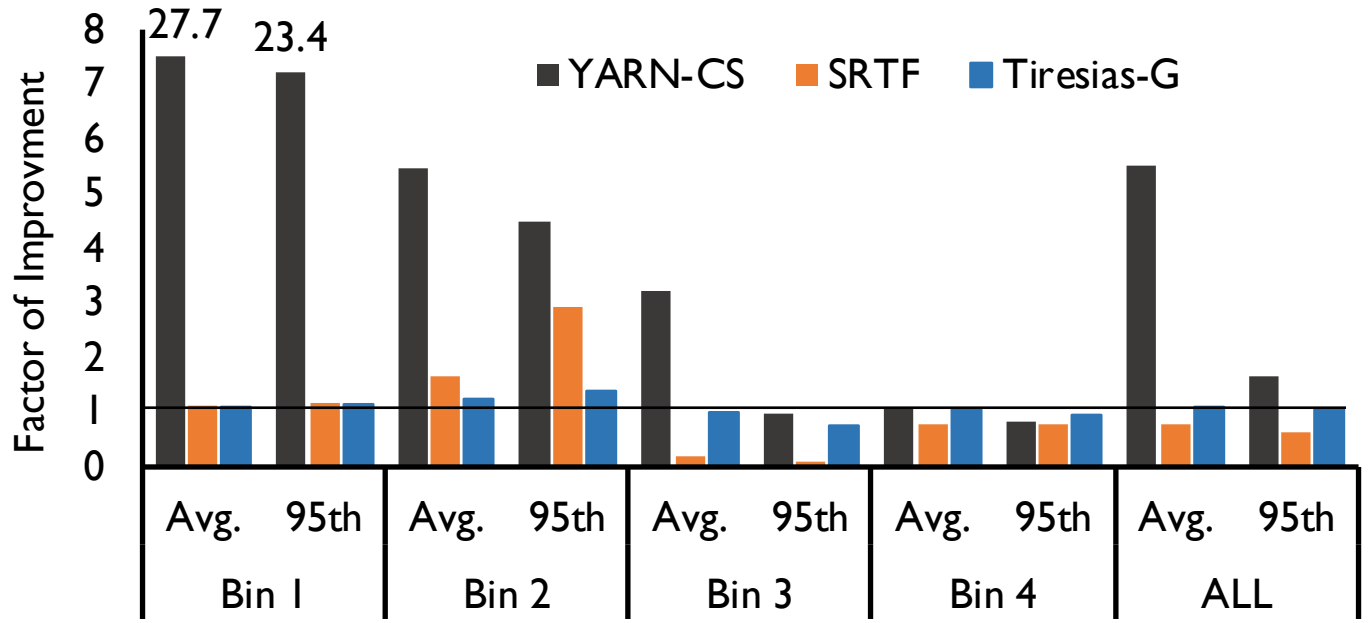
DL Models

Model	Total size (MB)	Largest tensor (MB)
VGG19	548	382
VGG16	527	392
VGG11	506	392
AlexNet	235	144
ResNet152	230	9
ResNet101	170	9
ResNet50	98	9
Inception4	163	6
Inception3	91	8
GoogleNet	27	4

JCT in Testbed Experiment



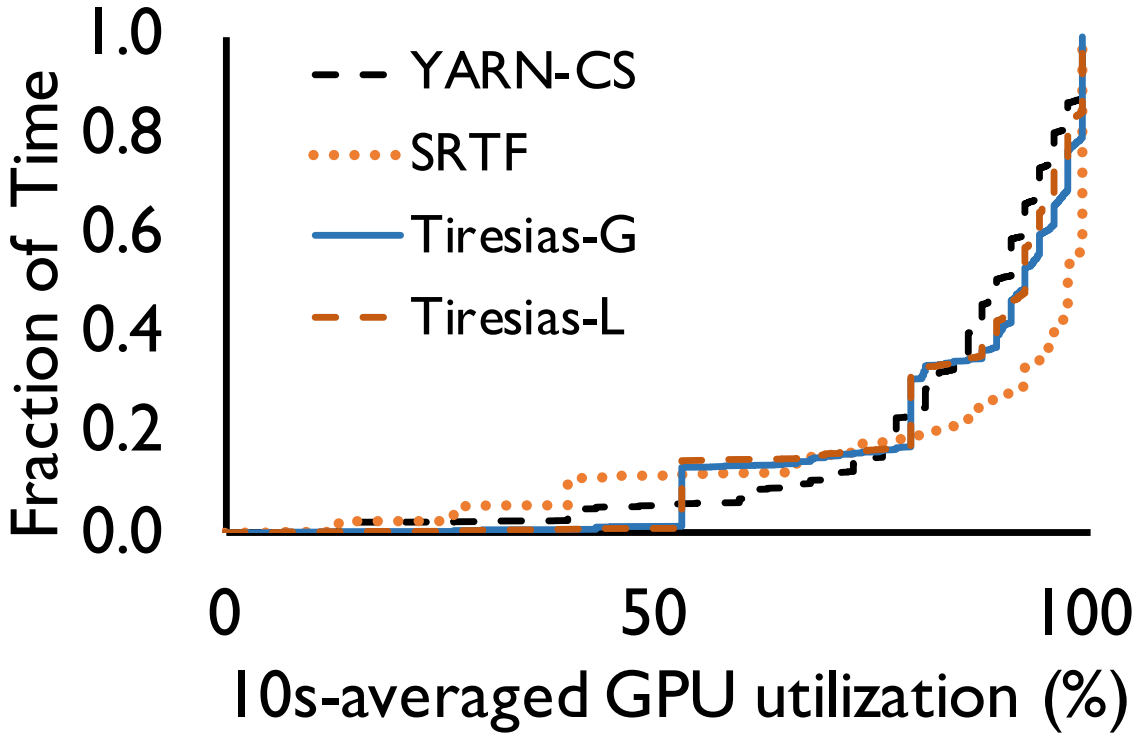
JCT Improvements in Testbed Experiment



Bins	1(Small-Short)	2(Small-Long)	3(Large-Short)	4(Large-Long)
% of Jobs	63.5%	12.5%	16.5%	7.5%

GPU Utilization in Testbed Experiment

- The makespan is improved by 1.21× (w.r.t. YARN-CS)



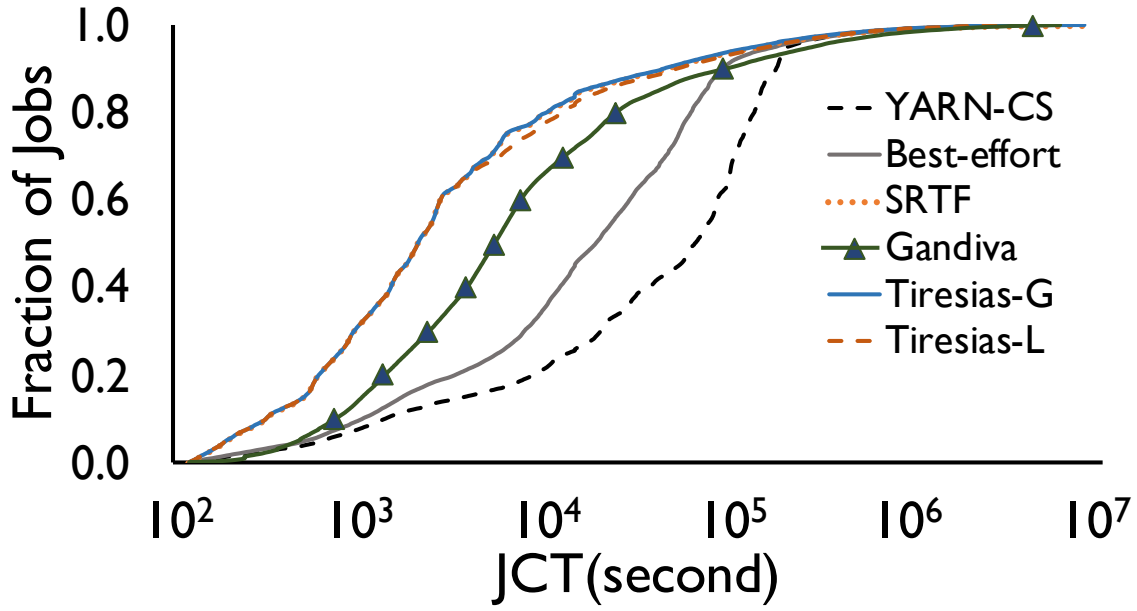
Queuing Delay in Testbed Experiment

	Average	Median	95th
YARN-CS	8146s	7464s	15327s
SRTF	593s	32s	3133s
Tiresias-G	1005s	39s	7933s
Tiresias-L	963s	13s	7755s

Training Performance in Testbed Experiment

- Training time when Tiresias-L running with and without placement

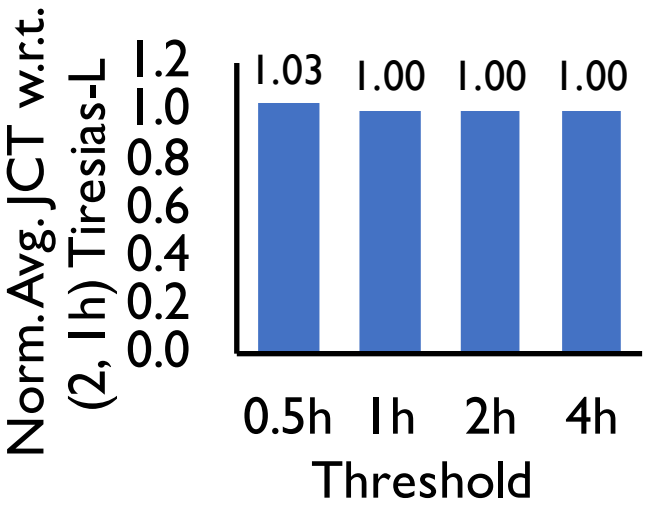
JCT in Trace-Driven Simulation

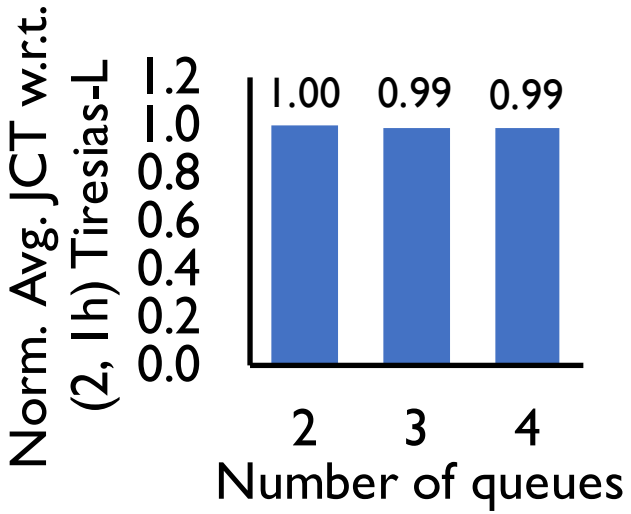


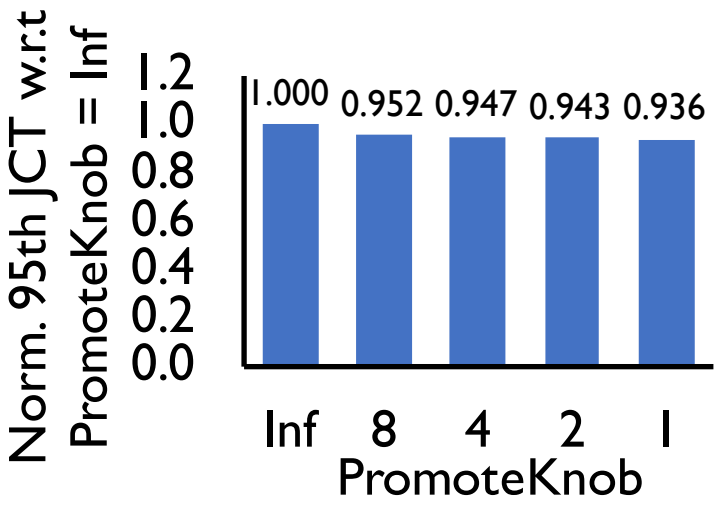
JCT Improvements in Trace-Driven Simulation

	Average	Median	95th
YARN-CS	2.41×	30.85×	1.25×
SRTF	1.00×	1.00×	0.84×
Gandiva	2.00×	2.59×	2.08×
Tiresias-G	0.97×	1.00×	0.85×

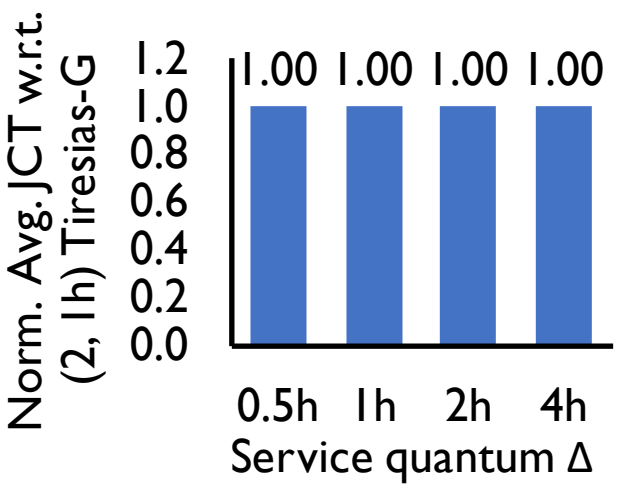
Sensitivity Analysis of 2D-LAS

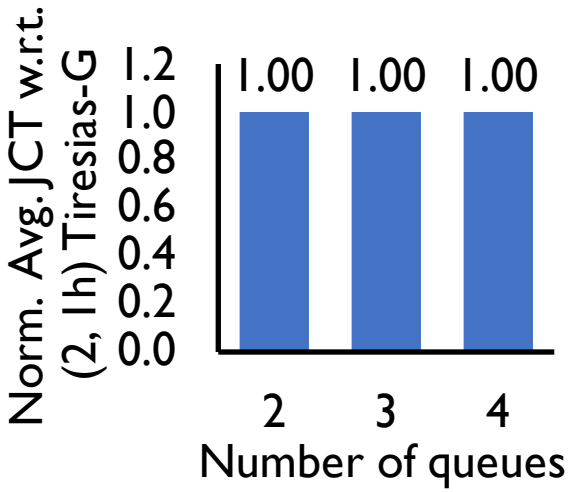


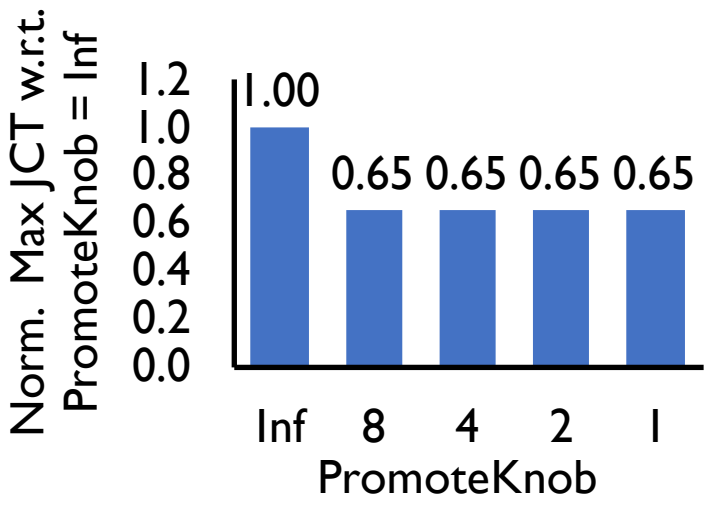




Sensitivity Analysis of 2D-Gittins Index







Gittins Index

$$\mathrm{GI_J} = \sup_{\Delta>0} \frac{\mathrm{P}\left(\mathrm{S} - \mathrm{a_J} \leq \Delta \,\middle|\, \mathrm{S} > \mathrm{a_J}\right)}{\mathrm{E}\left[\min\left\{\mathrm{S} - \mathrm{a_J}, \Delta\right\} \,\middle|\, \mathrm{S} > \mathrm{a_J}\right]}$$

- P is the probability that J can complete with in Δ
- E is the expected service (cost) of J to be complete with in Δ
- Δ is the next service quantum
- P and E are calculated from the distribution of job GPU time