Improving Shared GPU Clusters for DL Training Workloads

Myeongjae Jeon

UNIST CSE

Deep Learning at Enterprise

Deep learning (DL) is popular

- Speech, Image, Ads, NLP, Web Search …
- ***10.5×*** increase of DL training jobs in Microsoft

DL training jobs require GPU

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

Hi, I'm Cortana.

Office 365

XBOX

Bing ads

Bing

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

[1]. Analysis of Large-Scale Multi-Tenant GPU Clusters for DNN Training Workloads. USENIC ATC 2019

State-of-the-art DL Cluster Managers

	Gandiva [OSDI 2018]	Philly [ATC 2019]	Optimus [EuroSys 2018]	Tiresias [NSDI 2019]
Objective	Consolidation	Consolidation	Average JCT	Average JCT
Job Property	Any	Any	Converging	Any
Sched. Algorithm	Time-sharing	Locality-based	SRTF	Gittins Index
Input	N/A	Arrival time	Remaining time	Attained service

Most used Microsoft trace[1], will be open for public soon! ☺

[1]. Analysis of Large-Scale Multi-Tenant GPU Clusters for DNN Training Workloads. USENIC ATC 2019

Widespread Support by Open Source

Schedule GPUs

Kubernetes includes **experimental** support for managing AMD and NVIDIA GPUs spread across nodes. T

backwards incompatible iterations. The support for AMD GPUs was added in v1.9 via device plugin.

This page describes how users can consume GPUs across different Kubernetes versions and the current

First Class GPUs support in Apache Hadoop 3.1, YARN & HDP 3.0

by Wangda Tan & Vinod Kumar Vavilapalli

IF YOU'RE INTERESTED IN LEARNING MORE, GO TO OUR RECAP BLOG here!

THIS BLOG IS ALSO CO-AUTHORED BY ZIAN CHEN AND SUNIL GOVINDAN FROM HORTONWORKS.

INTRODUCTION – APACHE HADOOP 3.1, YARN, & HDP 3.0

Open Platform for AI (OpenPAI)

build passing coverage 60%

OpenPAI is an open source platform that provides complete AI model training and resource management capabilities, it is easy to extend and supports on-premise, cloud and hybrid environments in various scale.

Table of Contents

When to consider OpenPAI

1. When your organization needs to share powerful AI computing resources (GPU/FPGA farm, etc.) among teams.
2. When your organization needs to share and reuse common AI assets like Model, Data, Environment, etc.
3. When your organization needs an easy IT ops platform for AI.
4. When you want to run a complete training pipeline in one place.

Why choose OpenPAI

The platform incorporates the mature design that has a proven track record in Microsoft's large-scale production environment.

Today's Talk

Overall architecture of GPU cluster

Comm cost of distributed training and job placement

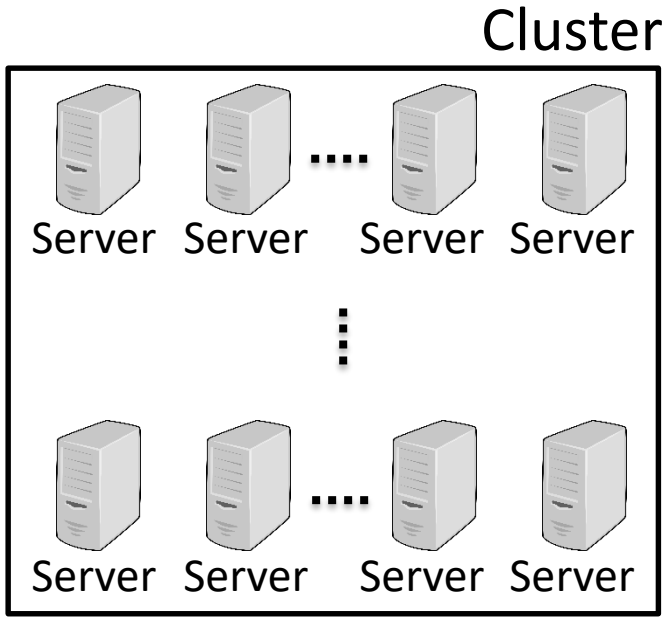
Strategy in Philly and Tiresias

Raising a few issues for the future

- Comm efficiency
- Failure handling
- More accessibility on HW

Shared GPU Cluster Architecture

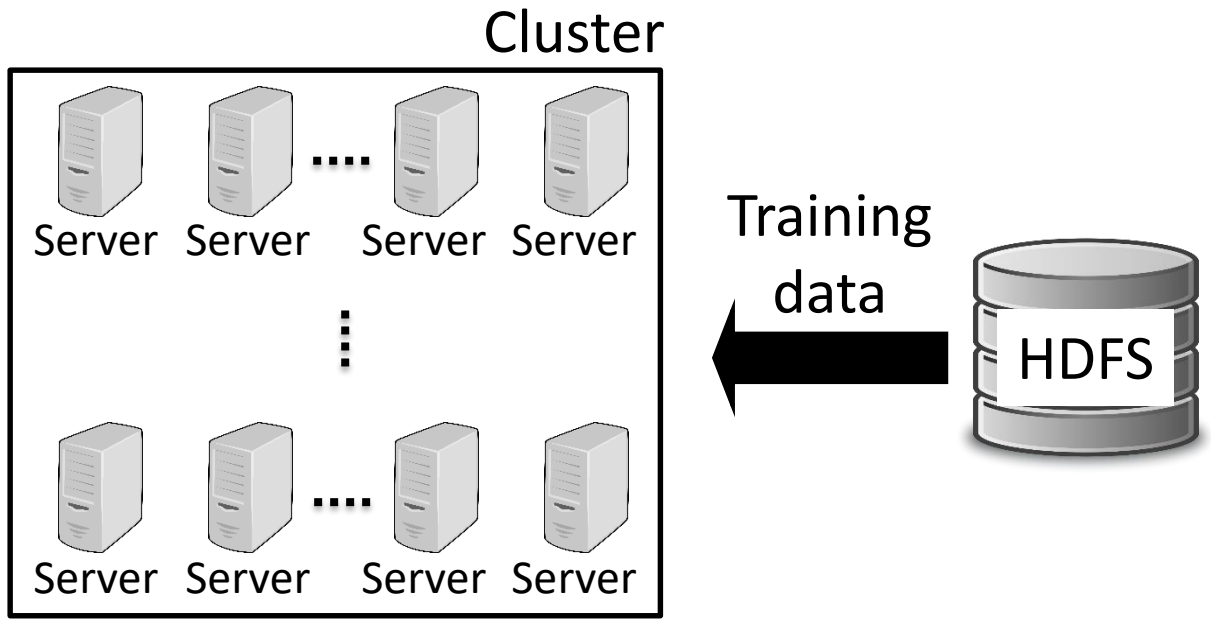
GPU cluster 100s of servers and thousands of GPUs



Shared GPU Cluster Architecture

GPU cluster 100s of servers and thousands of GPUs

HDFS Distributed "shared" storage for training data (and models)

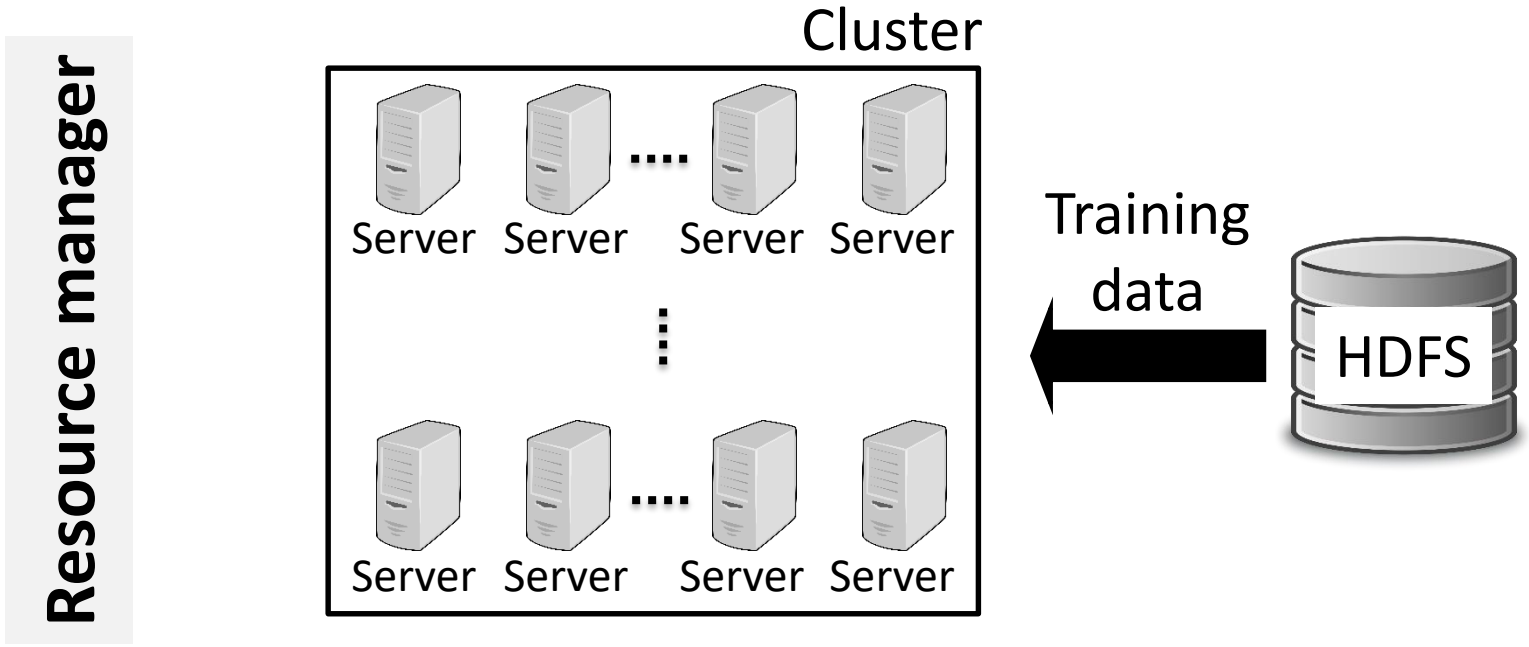


Shared GPU Cluster Architecture

GPU cluster 100s of servers and thousands of GPUs

HDFS Distributed "shared" storage for training data (and models)

RM Managing system resources for jobs submitted online

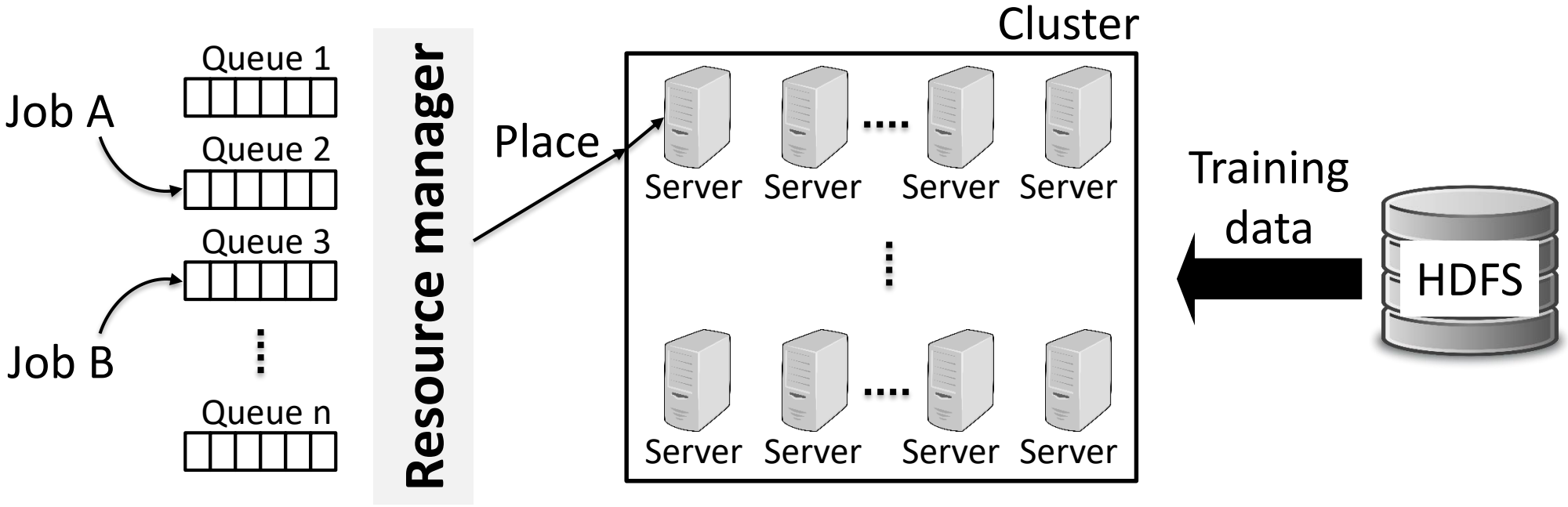


Shared GPU Cluster Architecture

Queues **Resource allocation (i.e., number of GPUs) for each group**

Managed by scheduler for fairness (e.g., Apache YARN's Fair Scheduler)

Allocate idle GPUs to a queue which has additional demand

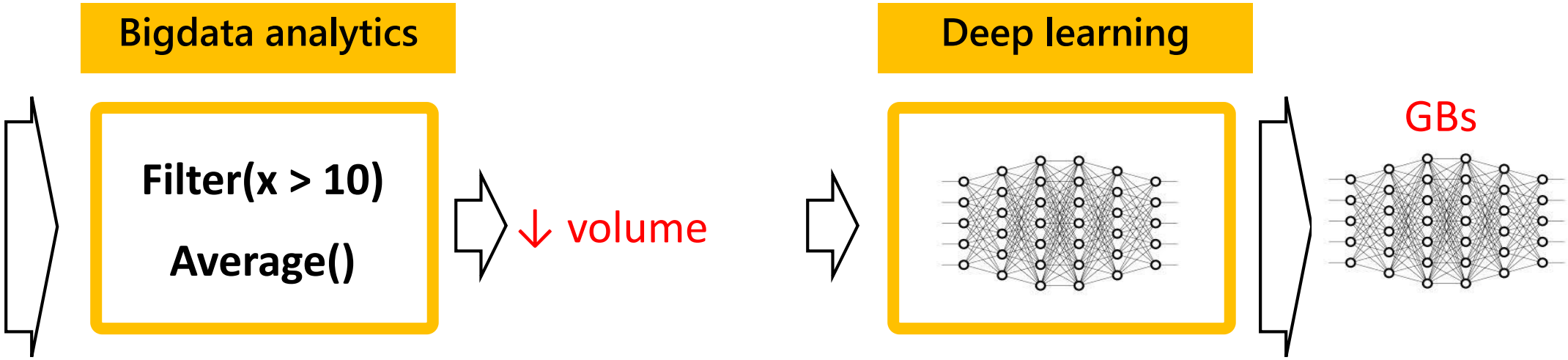


Comm Cost in Distributed Training

Data parallelism is most widely used in DL clusters

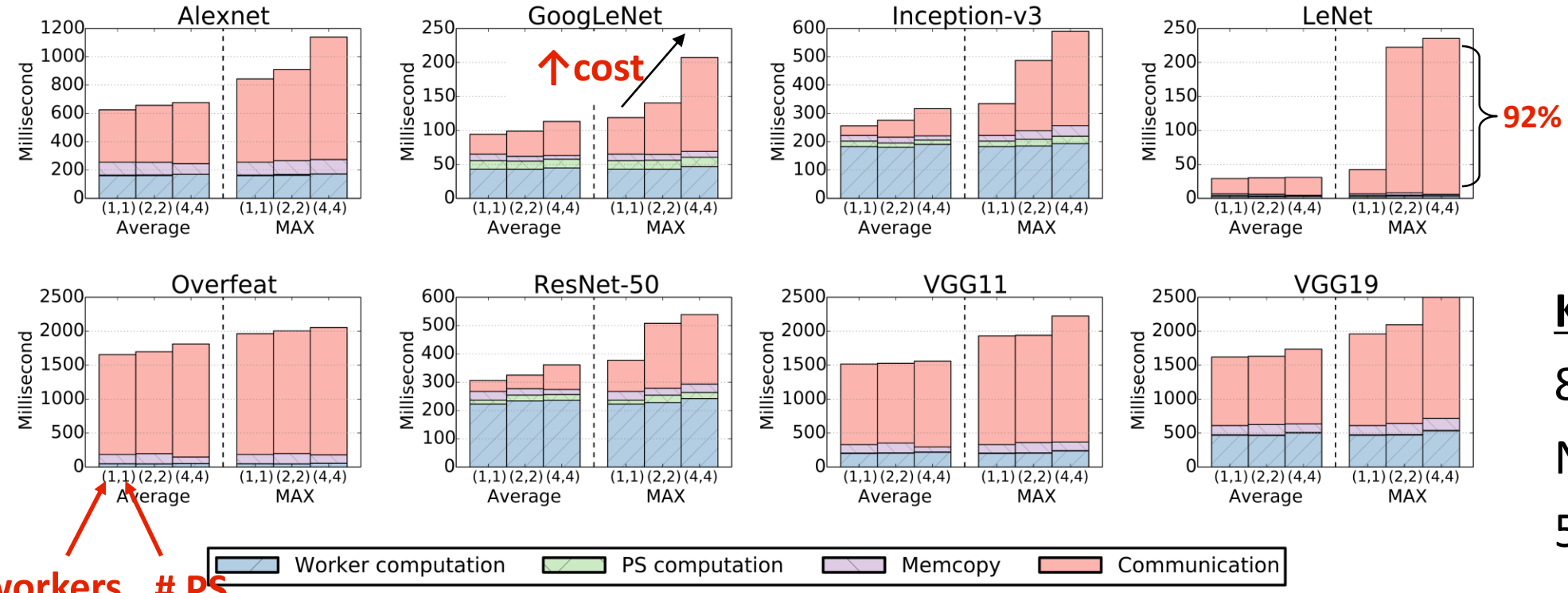
Periodic voluminous communication

– Workers running on multiple GPUs synchronize training progress



Network Cost in Data Parallel Training

Data parallelism is most widely used in DL clusters



Key HW conf

8 4-GPU servers

Nvidia Titan Xp

56 Gbps InfiniBand

Communication taking 58% on average!

Today's Talk

Overall architecture of GPU cluster

Comm cost of distributed training and job placement

Strategy in Philly and Tiresias

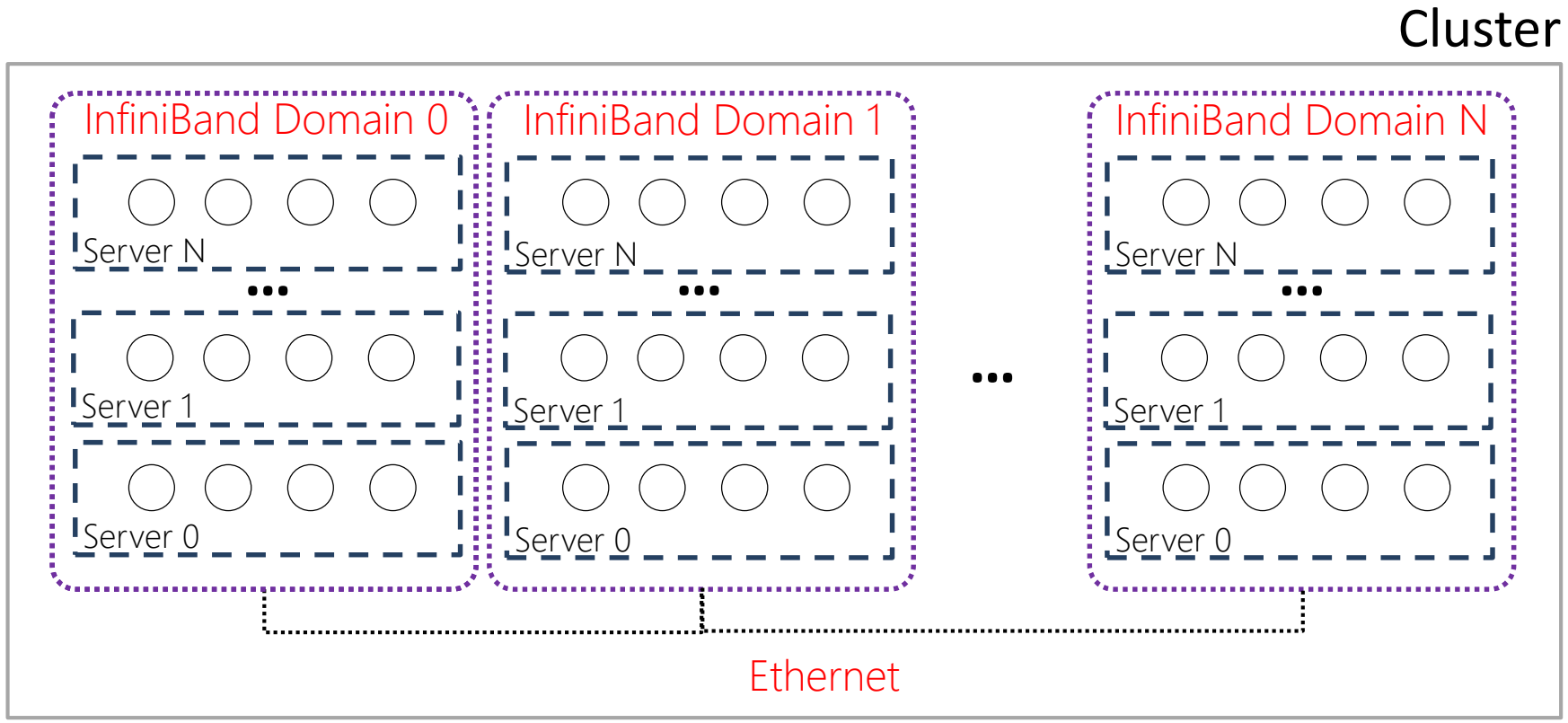
Raising a few issues for the future

- Comm efficiency
- Failure handling
- More accessibility

Deeper into Comm Heterogeneity

1. High-speed network (i.e., InfiniBand) is rack-localized

2. Intra-server GPU comm is only for 4 or 8 GPUs



Job Placement: High-level Objective

Job placement must be locality-aware!

High-speed network channel is rack-localized

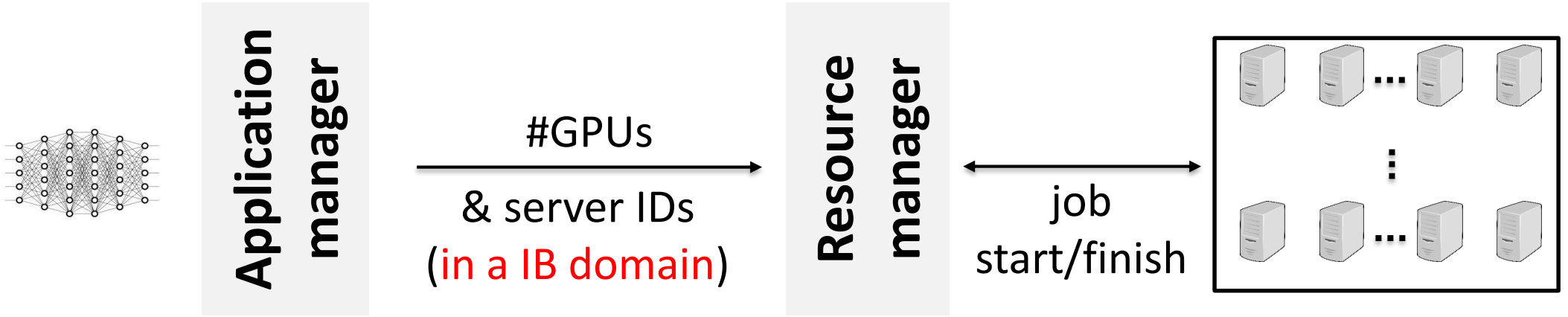
→ ***Pack a job's GPUs within a single InfiniBand domain***

Each server has 4 or 8 GPUs

→ ***Pack a job's GPUs onto the smallest number of servers possible***

Resource Negotiation

Each job has AM to negotiate resources from RM



CPU/memory assigned proportional to # of GPUs

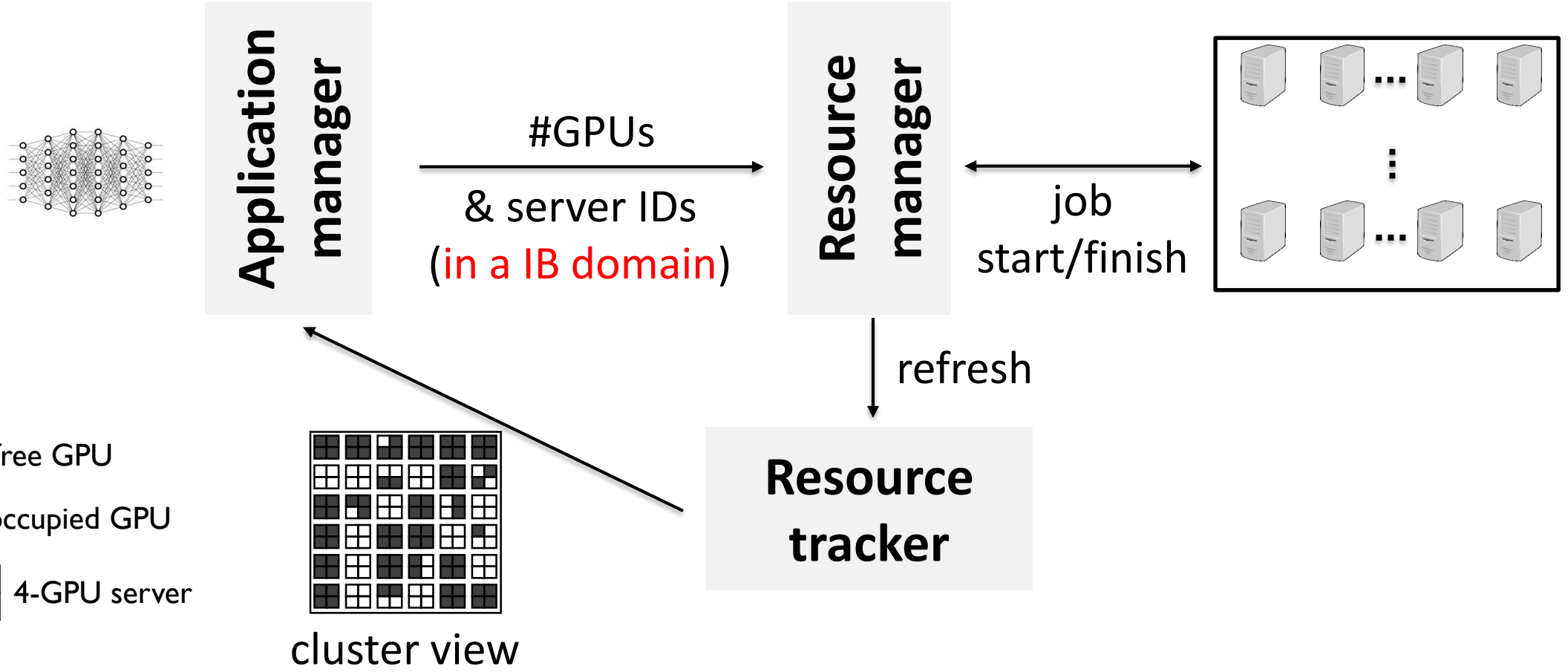
- Simplicity; Easier resource packing

Specific servers in a IBM domain selected

- Originally near-data processing in Bigdata; Now comm locality in deep learning

Decentralized Approach

Let each AM have the global view of the cluster



Scheduling Workflow for Distributed Training

Step 1: Make a request to RM

Calculate # of servers required

"Highest locality" at the beginning (i.e., using the fewest servers)

Pick a rack that has such servers most available

Pick a set of servers

Step 2: Not all GPUs ready until timeout?

Release any acquired GPUs and take a back-off

} **Avoid starvation**

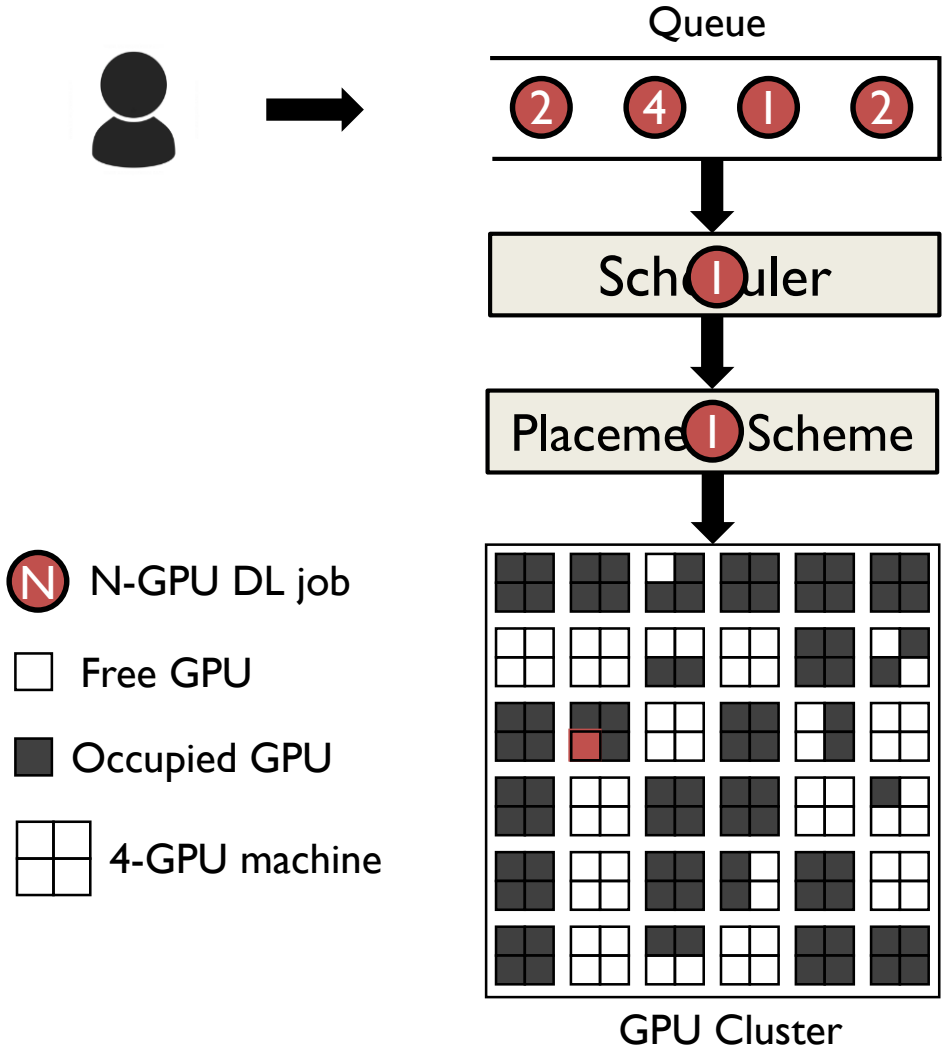
Step 3: Retry request

Increasingly "relax locality" constraints

Allowing more inter-server communication

} **Trade-off training efficiency for lower waiting**

Custer Manager in Tiresias



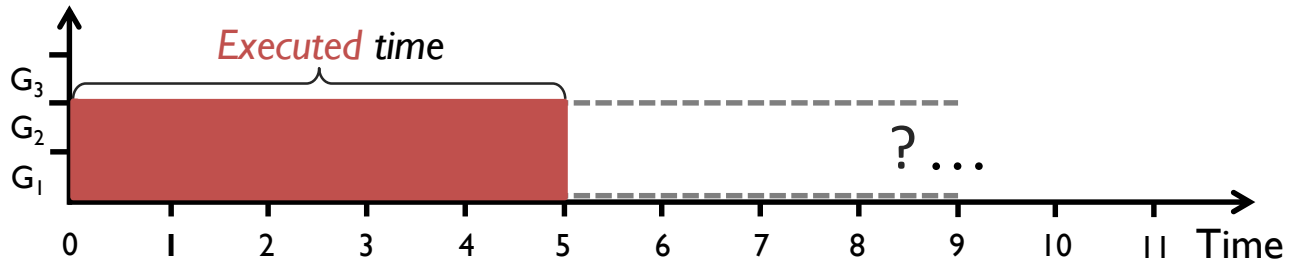
In Philly, placement requirements (i.e. locality) limit job scheduling

1. Age-based Scheduler — *Minimize average JCT*

2. Model Profile-based Placement — *Enable locality selectively*

Available Job Information

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



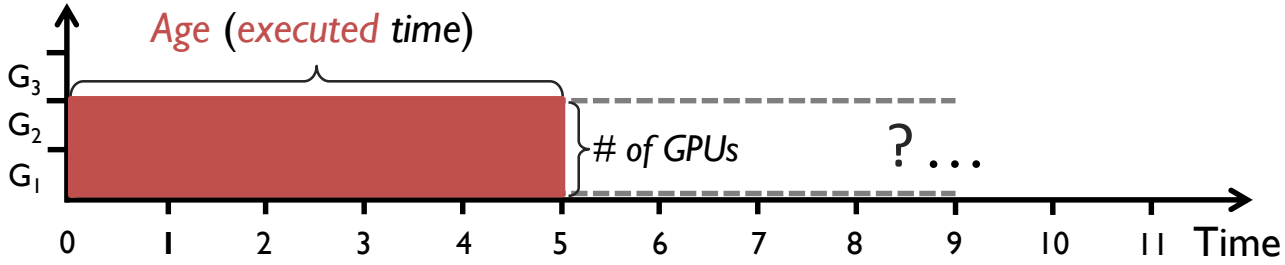
Age-based Scheduler

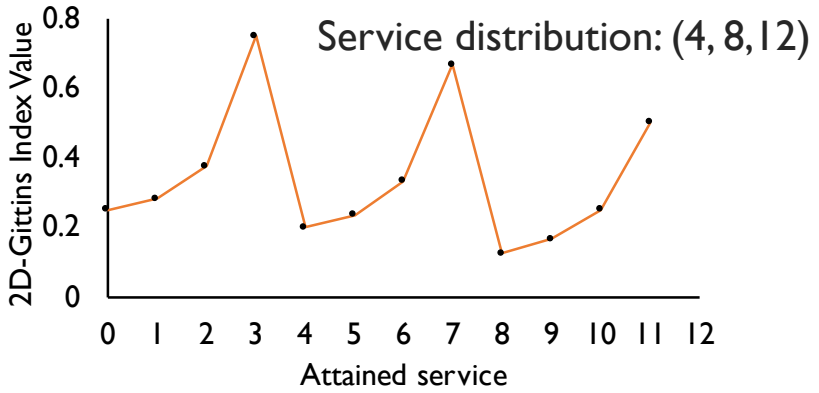
Gittins Index

- Need the *distribution of job execution time*
- **Probability** to complete in the near future based on current age

2D-Gittins Index policy

- Age calculated by attained service (# of GPUs × executed time)
- Prioritize a job that has the largest Gittins Index

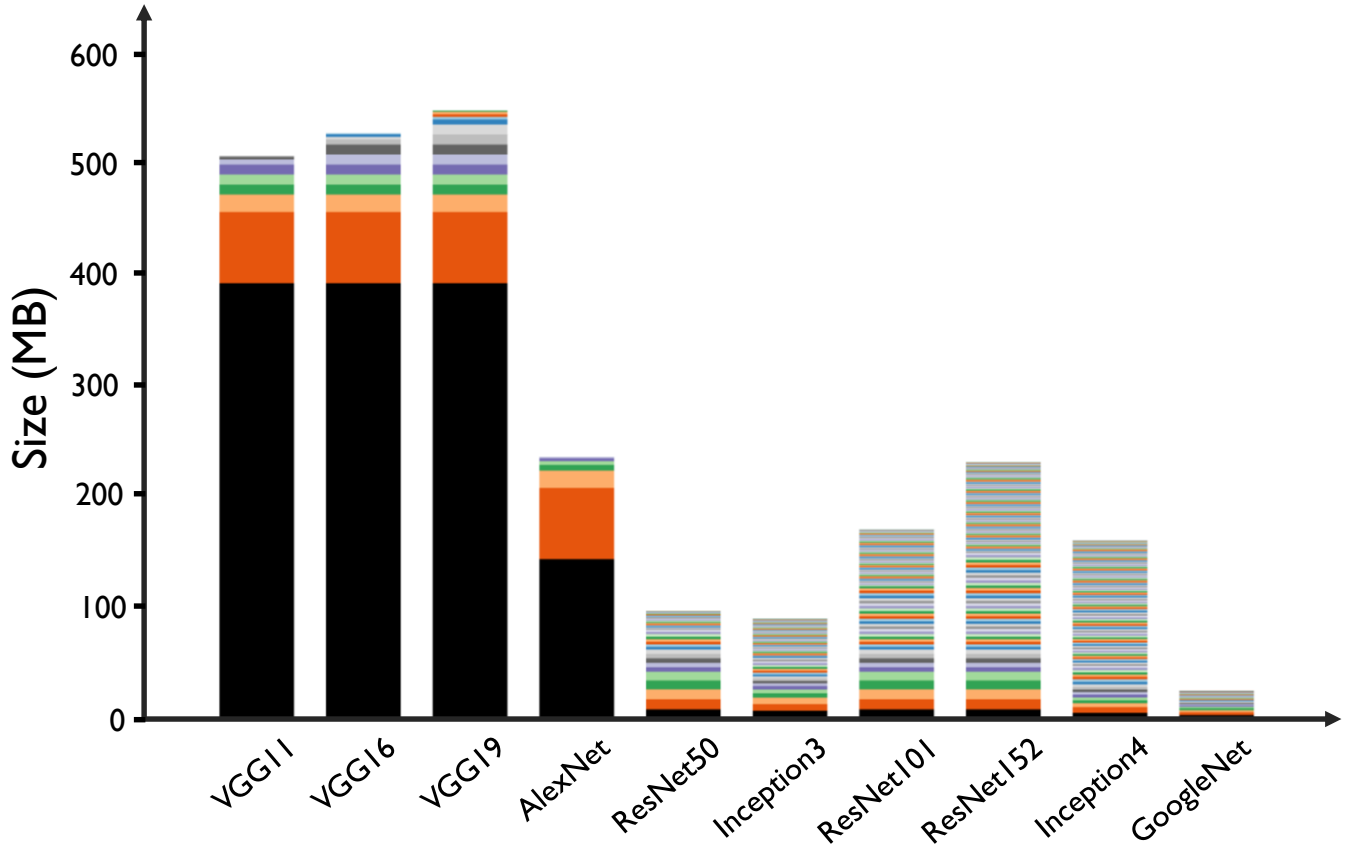




Model Profile-based Placement

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

Today's Talk

Overall architecture of GPU cluster

Comm cost of distributed training and job placement

Strategy in Philly and Tiresias

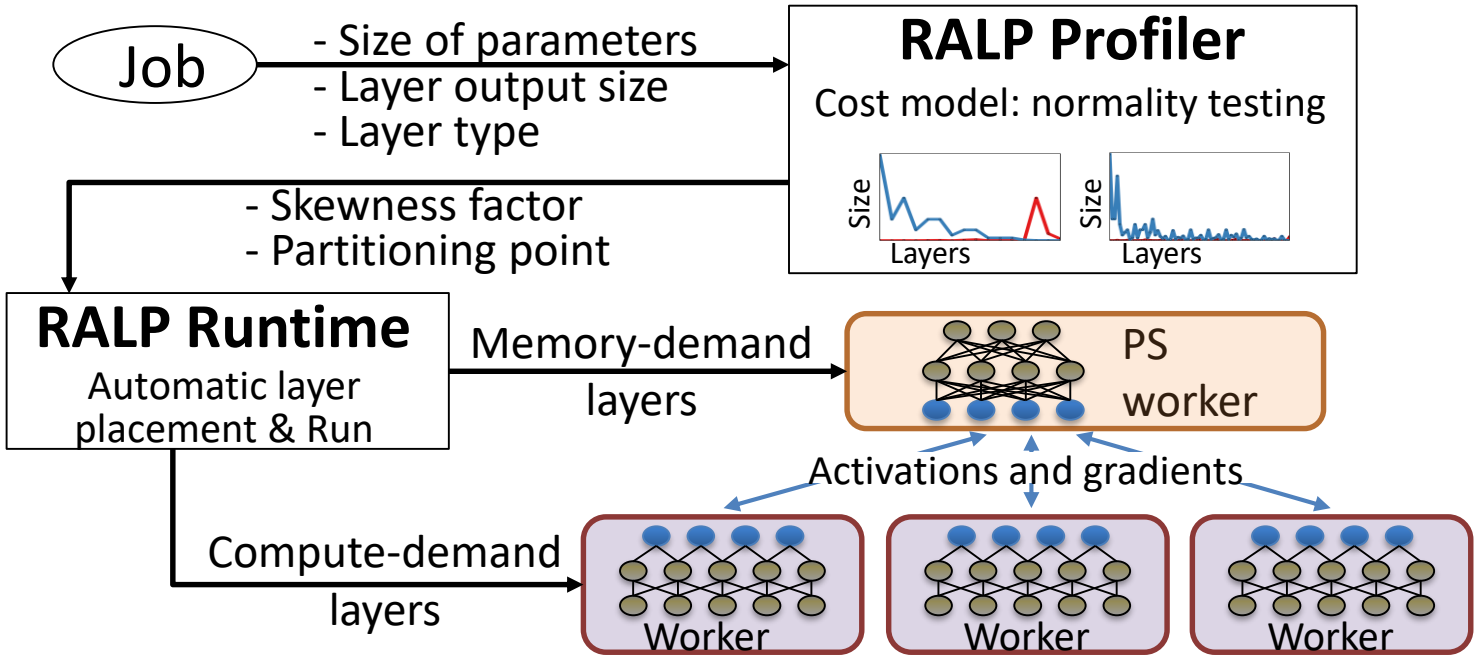
Raising a few issues for the future

- Comm efficiency
- Failure handling
- More accessibility

Mitigating Comm Cost

Optimize model training for low comm cost

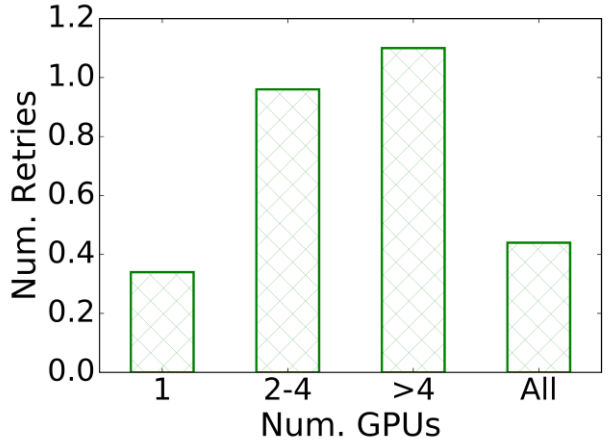
- Need other efficient types of parallelism
- Optimize various types of model (beyond CNNs)



Failure Handling by Cluster Manager

Job failures are frequent, and

more common for larger jobs



Per job User errors in code or configuration

→ *Need to pre-run the first iteration on a single GPU (cheap)*

Across jobs Input format errors or corrupted inputs

→ *Need blacklisting and stop retrying*

SW & Trace, then HW is Accessible?

Having open platforms are more than necessary!

1. Own training infrastructure setup
   - The number of GPUs in distributed training keeps increasing
     - 32 (2016) → 128 (2017)
   - 128 GPUs = $1M

2. Borrowing resources from cloud
   - 128 GPUs for 12 hours = $5K

Summary

Shared GPU cluster is coming popular for DL training

– Need to design cluster managers for diverse circumstance

Network cost during distributed training is detrimental

– Worse with increasing use of many GPUs

– Cluster managers can mitigate the cost

Many improvements are desired for better future

Thank You!

mjjeon@unist.ac.kr