Synergy: Looking Beyond GPUs for DNN Scheduling on Multi-Tenant Clusters

Jayashree Mohan*, Amar Phanishayee*, Janardhan Kulkarni*, Vijay Chidambaran^  

*Microsoft Research  ^UT Austin
Deep Learning at scale

- Large enterprises train DL jobs on large GPU clusters
  - Multi-tenant: Cluster shared between several users/product groups
  - Variety of training jobs – speech, image, NLP, etc.

- A cluster manager allocates resources and schedules training jobs

- Scheduling Policy: FIFO, SRTF, LAS, FTF etc
- Cluster metrics: Job completion time (JCT), fairness, makespan, etc
GPU-Proportional Allocation

- Job specifies only GPU demand
- Auxiliary resources (CPU, memory) are allocated proportional to the GPUs requested
- Uses GPU proportional allocation
Motivation

• DNNs exhibit *varying levels of sensitivity* to CPU, DRAM allocation
Motivation

• DNNs exhibit **varying levels of sensitivity** to CPU, DRAM allocation
Motivation

• DNNs exhibit varying levels of sensitivity to CPU, DRAM allocation

Exploit the difference in resource requirement across jobs to perform disproportionate resource allocation
Challenges

What is the ideal aux. resource requirement for each job?

Efficiently using resources in DNN scheduling

How to pack these jobs with malleable aux. resource demands on a cluster?

To address these challenges in a scheduling policy agnostic manner, we build Synergy
Synergy

• **Resource-sensitivity** aware scheduler for DNN training jobs

• Identifies each job’s best-case CPU and memory requirements using an **optimistic profiling** technique.

• Packs these jobs on to the available servers along multiple resource dimensions using a close-to-optimal **heuristic scheduling mechanism**

• Improves cluster objectives by upto **3.4x** when compared to traditional GPU-proportional scheduling mechanism.
Outline

- Motivation
- **Design**
  - Profiling
  - Placement
- Evaluation
Synergy : Design

- Round-based scheduling

**Placement Policy**

Finds best placement, based on resource constraints

**Scheduling Policy**

```
plan(i)
getPlan(i)
runPlan(i)
startNextRound()
```

**Profiler**

Identifies best resource allocation

- Incoming queue
- Wait queue
- Round $i$

0 1 2 3

1 2 3
Outline

• Motivation
• Design
  • **Profiling**
  • Placement
• Evaluation

Synergy
Profiling

- Runs offline
- Assume there is a dedicated server (identical to the ones in cluster) for profiling that measures the sensitivity of incoming job to CPU, memory and data locality

To fill each point, need to run the training job for a few iterations
Profiling

- Runs offline
- Assume there is a dedicated server (identical to the ones in cluster) for profiling that measures the sensitivity of incoming job to CPU, memory and data locality

- To fill each point, need to run the training job for a few iterations
  - Measure only the CPU sensitivity
  - Model memory sensitivity based on:
    - cache size
    - memory bandwidth
    - storage bandwidth

- Runs offline
- Assume there is a dedicated server (identical to the ones in cluster) for profiling that measures the sensitivity of incoming job to CPU, memory and data locality

- Use MinIO [VLDB’21]
- Predictable per-epoch I/O
Profiling

- **CPU, DRAM allocation**: To find ideal resource allocation, find the least (CPU+mem) that reaches max performance
Outline

• Motivation
• Design
  • Profiling
    • Placement
• Evaluation
Job Placement

• What is the best placement for a set of jobs in the given round?
• Multi-dimensional bin packing – NP hard
Job Placement

• What is the best placement for a set of jobs in the given round?
• Multi-dimensional bin packing – NP hard

Optimal allocation that provides an upper bound on the achievable cluster throughput

• Formulate our problem as a linear program (LP)
• 2 levels of LP:
  1. Idealized setting: All resources present in one (super) machine
     • Given profile matrix, find the allocation that maximizes overall throughput
  2. Construct a feasible allocation across available machines.

- Solving two LPs per scheduling round is a computationally expensive task.
- Final allocation matrix can have fractional GPU allocations
Job Placement

• What is the best placement for a set of jobs in the given round?
• Multi-dimensional bin packing – NP hard
Scheduling with multiple resource demands

Sorted queue of runnable jobs

<table>
<thead>
<tr>
<th>CPU</th>
<th>Mem</th>
<th>GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>250GB</td>
<td>2</td>
</tr>
</tbody>
</table>

Fits in cluster? (YES/NO)

Tune job to GPU-prop share

Schedule job
Scheduling with multiple resource demands

Sorted Queue of runnable jobs

Inc

Fits in cluster?

Tune job to GPU-prop share

Schedule job

YES

NO
Scheduling with multiple resource demands

Sorted Queue of runnable jobs

- Fits in cluster?
  - NO
    - Tune job to GPU-prop share
    - Find underutilized server
    - Get jobs to tune

Inc
Scheduling with multiple resource demands

Sorted Queue of runnable jobs

Inc

Fits in cluster?

NO

Tune job to GPU-prop share

Find underutilized server

Get jobs to tune

Switch to GPU proportional share

Top k jobs to switch to GPU proportional share so that (3) can fit in this server
Scheduling with multiple resource demands

Sorted Queue of runnable jobs

Fits in cluster?

- YES
- NO

Tune job to GPU-prop share
Find underutilized server
Get jobs to tune

Schedule job
Scheduling with multiple resource demands

Sorted Queue of runnable jobs

1. Tune job to GPU - prop share
2. Find underutilized server
3. Get jobs to tune

Continue till no more free GPUs (or jobs) in the cluster

Fits in cluster? (YES/NO)

Schedule job
Outline

- Motivation
- Design
  - Profiling
  - Placement
- Evaluation
Evaluation

• Experiments are performed on servers from an internal GPU cluster at Microsoft
  • Each server has eight V100 GPUs (32GiB), 24 CPU cores, 500 GB DRAM, and SSD storage

• Consider 10 different models (CNNs, RNNs, LSTMs) across different tasks

<table>
<thead>
<tr>
<th>Image</th>
<th>Language</th>
<th>Audio</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet, ResNet18, ShuffleNet, MobileNet, ResNet50</td>
<td>GNMT, LSTM, Transformer-XL</td>
<td>DeepSpeech, M5</td>
</tr>
</tbody>
</table>

Microbenchmarks | Trace-driven simulations | Physical cluster deployment
Evaluation Questions

• Can Synergy’s optimistic profiling replicate real trends in job throughput?

• Can Synergy’s scheduling mechanism improve overall cluster metrics like Makespan and average JCT?

• How does Synergy’s heuristic tuning mechanism compare to optimal solution?

• How does Synergy react to varying workload compositions?

• How well does Synergy’s scheduling mechanism scale to larger clusters?

• How does Synergy compare to multi-resource big data scheduling policies like DRF?
Evaluation Questions

• Can Synergy’s optimistic profiling replicate real trends in job throughput?
• Can Synergy’s scheduling mechanism improve overall cluster metrics like Makespan and average JCT?
• How does Synergy’s heuristic tuning mechanism compare to optimal solution?
• How does Synergy react to varying workload compositions?
• How well does Synergy’s scheduling mechanism scale to larger clusters?
• How does Synergy compare to multi-resource big data scheduling policies like DRF?
1. Optimistic Profiling

- Profiles CPU and memory demand for ResNet18

Optimistic profiling is able to replicate the CPU and memory demand curve within 3% of empirical results.
2. Synergy improves cluster objectives

Physical cluster of 32 GPUs across 4 machines
## 2. Synergy improves cluster objectives

<table>
<thead>
<tr>
<th>Policy</th>
<th>Metric</th>
<th>Mechanism</th>
<th>Deploy (hrs)</th>
<th>Simulation (hrs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIFO</td>
<td>Makespan</td>
<td>Proportional</td>
<td>16</td>
<td>15.67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tuned</td>
<td>11.6</td>
<td>11.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Opt</td>
<td>-</td>
<td>11.01</td>
</tr>
</tbody>
</table>

Physical cluster of 32 GPUs across 4 machines

Synergy-Tune improves makespan by 1.4x
2. Synergy improves cluster objectives

<table>
<thead>
<tr>
<th>Policy</th>
<th>Metric</th>
<th>Mechanism</th>
<th>Deploy (hrs)</th>
<th>Simulation (hrs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIFO</td>
<td>Makespan</td>
<td>Proportional</td>
<td>16</td>
<td>15.67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tune</td>
<td>11.6</td>
<td>11.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Opt</td>
<td>-</td>
<td>11.01</td>
</tr>
<tr>
<td>SRTF</td>
<td>Avg JCT</td>
<td>Proportional</td>
<td>4.81</td>
<td>4.52</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tune</td>
<td>3.21</td>
<td>3.19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Opt</td>
<td>-</td>
<td>3.06</td>
</tr>
</tbody>
</table>

Synergy-Tune improves makespan by 1.4x, average JCT by 1.5x
2. Synergy improves cluster objectives

<table>
<thead>
<tr>
<th>Policy</th>
<th>Metric</th>
<th>Mechanism</th>
<th>Deploy (hrs)</th>
<th>Simulation (hrs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIFO</td>
<td>Makespan</td>
<td>Proportional</td>
<td>16</td>
<td>15.67</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tune</td>
<td>11.6</td>
<td>11.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Opt</td>
<td>-</td>
<td>11.01</td>
</tr>
<tr>
<td>SRTF</td>
<td>Avg JCT</td>
<td>Proportional</td>
<td>4.81</td>
<td>4.52</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tune</td>
<td>3.21</td>
<td>3.19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Opt</td>
<td>-</td>
<td>3.06</td>
</tr>
<tr>
<td></td>
<td>99th p JCT</td>
<td>Proportional</td>
<td>17.32</td>
<td>16.85</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tune</td>
<td>8.29</td>
<td>8.54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Opt</td>
<td>-</td>
<td>8.21</td>
</tr>
</tbody>
</table>

Synergy-Tune improves makespan by 1.4x, average JCT by 1.5x and 99th percentile JCT by 2x
3. Synergy enables the cluster to support higher load

Synergy supports higher load by efficiently utilizing resources to finish jobs faster.
Conclusion

- Resource-sensitivity aware scheduler for DNN training
- Exploits heterogeneity in auxiliary resource requirement to perform workload-aware allocation
- Improves cluster-wide performance

https://github.com/msr-fiddle/synergy

Thanks!
Questions?

Contact : jamohan@microsoft.com