Hello

CS 744: GAVEL

Shivaram Venkataraman

Fall 2022

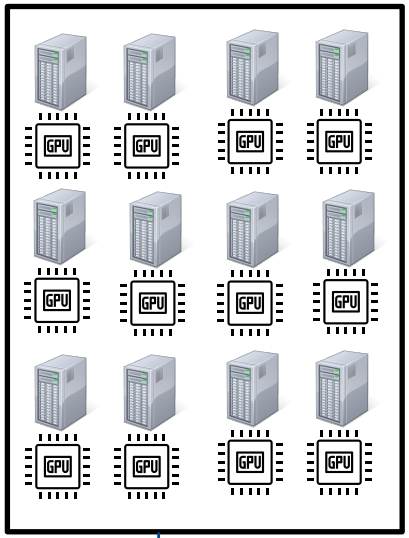
ADMINISTRIVIA

- Course project assignments
  - Emails will go out end of this week (Oct 14th)
  - Introductions due Oct 25th → Piazza

- Midterm Exam
  - In class on Oct 27th
  - Includes everything from beginning to the end of ML (including Nexus)

- Sample exams from prev. years

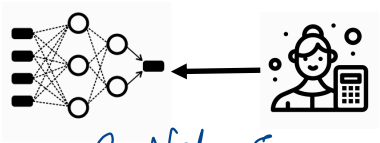
MACHINE LEARNING: TRAINING



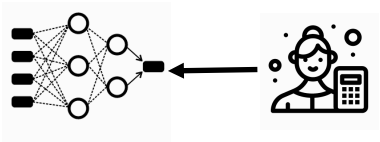
Shared cluster

APIs for programmers

Techniques to distribute / parallelize

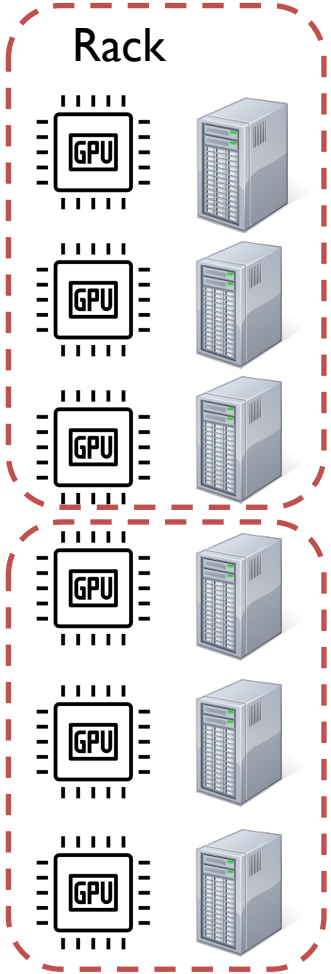


ResNet-50



BERT

WORKLOAD CHARACTERISTICS



→ many hours or even days to train one model

Long running tasks

→ All workers need to be active at the same time

Gang scheduling

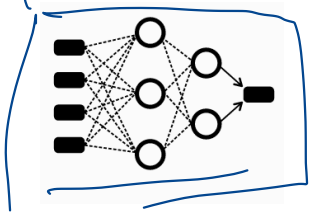
Heterogeneity?

↳ Hardware

↳ Placement or Sharing of Accelerators

DL SCHEDULER INTERFACE

Code or Container



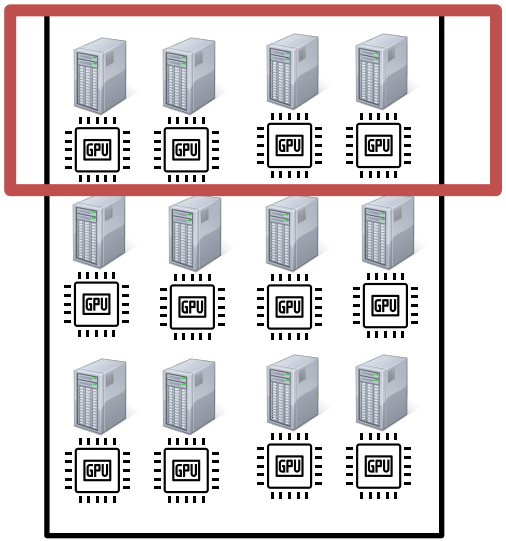
Run job Resnet18
With BatchSize = 64
on Num GPUs = 4

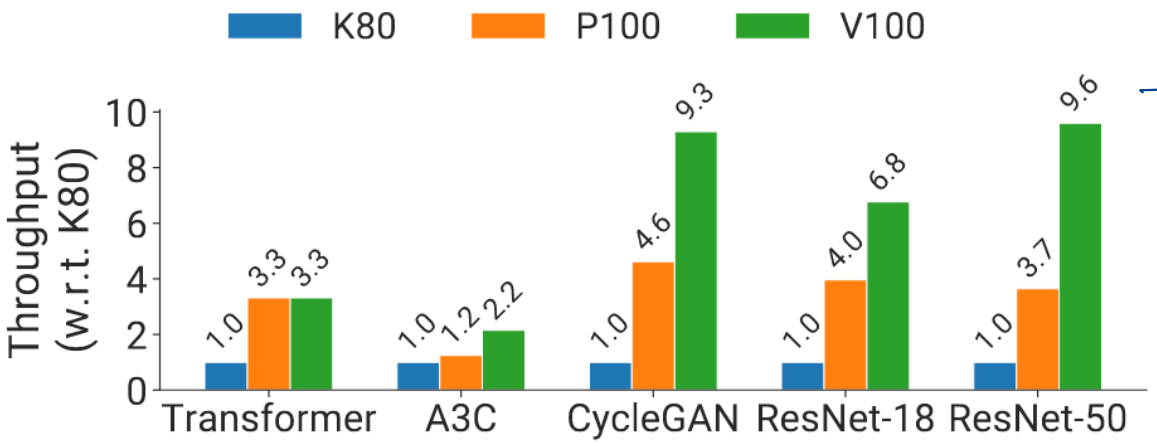
Centralized

Scheduler

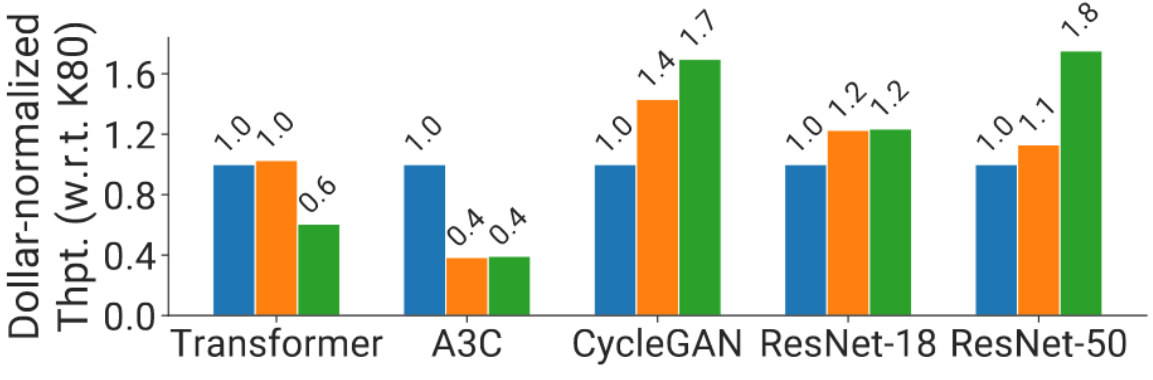
Goals:
- Maximize throughput
- Fairness
- Minimize JCT
- …

Job 1 allocated





(a) Throughput.



(b) Dollar-normalized.

→ Diff models get diff speedups across K80, P100, V100

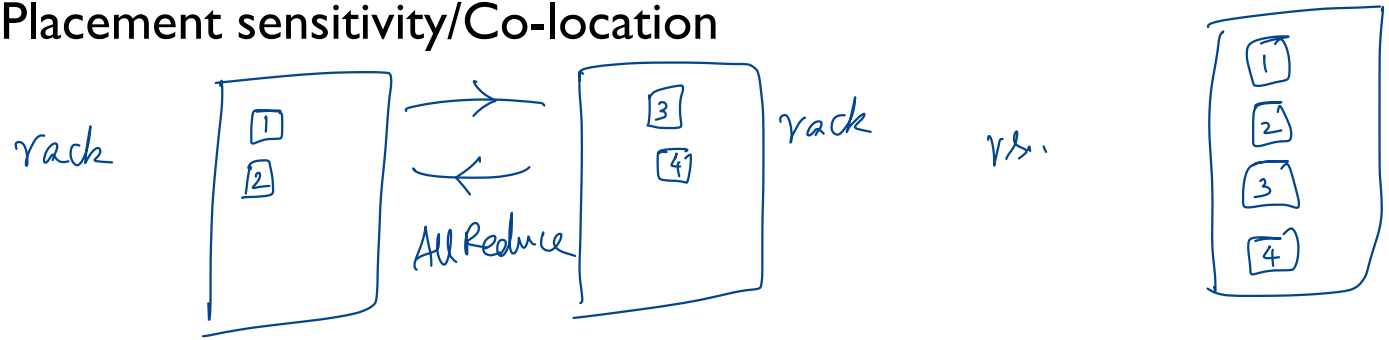
MOTIVATION: HETEROGENEITY

→ Diff accelerators are cost-effective A3C vs. Resnet-50

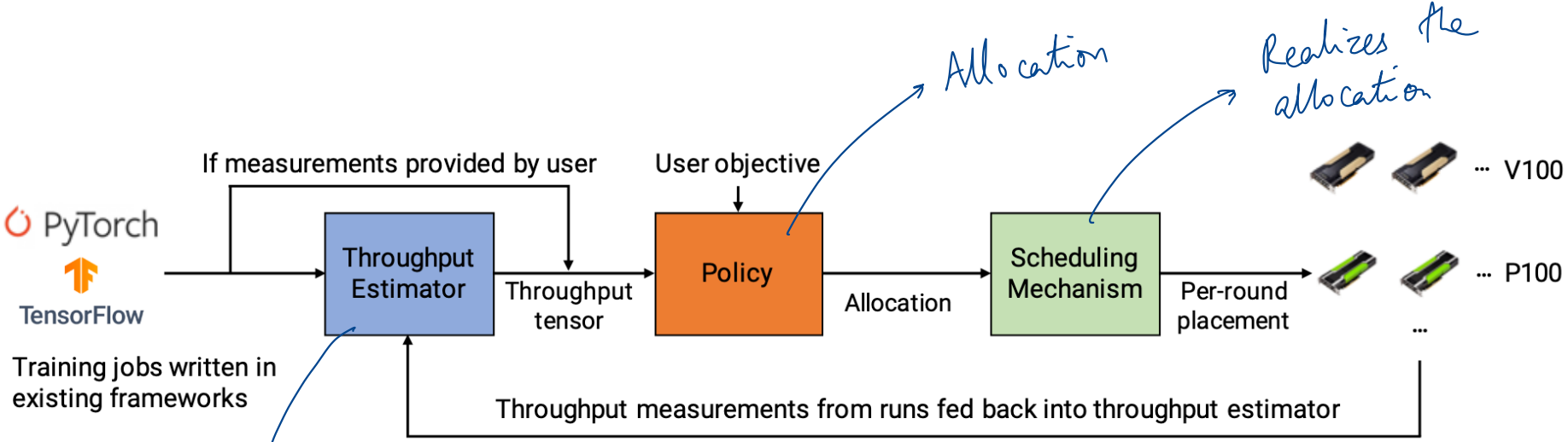
ADDITIONAL GOALS

- Support a wide range of objectives
  - Minimize makespan → Time to finish last job in trace
  - Average JCT → "utilization"
  - Fairness (Sharing incentive) → "responsiveness"
  - … → GPUs or accelerators

- Placement sensitivity/Co-location



GAVEL: SYSTEM DESIGN



SCHEDULING POLICY: OPTIMIZATION PROBLEM

$$\text{Maximize}_X \sum_{m \in \text{jobs}} \text{throughput}(m, X)$$

$$\text{throughput}(m, X) = \sum_{j \in \text{accelerator types}} T_{mj} \cdot X_{mj}$$

$$0 \leq X_{mj} \leq 1 \qquad \forall (m, j) \quad (1)$$

$$\sum_j X_{mj} \leq 1 \qquad \forall m \quad (2)$$

$$\sum_m X_{mj} \cdot \text{scale\_factor}_m \leq \text{num\_workers}_j \qquad \forall j \quad (3)$$

$$X^{\text{example}} = \begin{array}{c} \begin{array}{ccc} V100 & P100 & K80 \end{array} \\ \begin{pmatrix} 0.6 & 0.4 & 0.0 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.0 & 0.8 \end{pmatrix} \end{array} \begin{array}{l} \text{job 0} \\ \text{job 1} \\ \text{job 2} \end{array}$$

Least Attained Service (LAS)

POLICY: MAX-MIN FAIRNESS

Classic: Weighted max-min fairness based on accelerator hours consumed

$X_0 = 0.33$
$X_1 = 0.33$
$X_2 = 0.33$

$$\text{Maximize}_X \min_m \frac{1}{w_m} X_m$$

w_m → weight for a particular job

Gavel: Use weighted normalized effective throughputs

$$\text{Maximize}_X \min_m \frac{1}{w_m} \frac{\text{throughput}(m,X)}{\text{throughput}(m,X_m^{\text{equal}})}$$

X_m^{equal} → 1/n share

$$\text{throughput}(m,X) = \sum_{j \in \text{accelerator types}} T_{mj} \cdot X_{mj}$$

For a given job, its tput is weighted sum of allocations

EXAMPLE

Throughput matrix

estimator

$$T = \begin{array}{c} \begin{matrix} V100 & K80 \end{matrix} \\ \begin{pmatrix} 40.0 & 10.0 \\ 12.0 & 4.0 \\ 100.0 & 50.0 \end{pmatrix} \end{array} \begin{matrix} \text{job 0} \\ \text{job 1} \\ \text{job 2} \end{matrix}$$

$$X^{\text{hom.}} = \begin{array}{c} \begin{matrix} V100 & \end{matrix} \\ \begin{bmatrix} 0.33 & 0.33 \\ 0.33 & 0.33 \\ 0.33 & 0.33 \end{bmatrix} \end{array} \begin{matrix} J0 \\ J1 \\ J2 \end{matrix}$$

1.0

Eff TPUT =

J0 = 0.33 × 40 + 0.33 × 10 = 16.5

J1 = 5.28

J2 = 49.5

$$X^{\text{het.}} = \begin{array}{c} \begin{matrix} V100 & K80 \end{matrix} \\ \begin{pmatrix} 0.45 & 0.0 \\ 0.45 & 0.09 \\ 0.09 & 0.91 \end{pmatrix} \end{array} \begin{matrix} \text{job 0} \\ \text{job 1} \\ \text{job 2} \end{matrix}$$

Eff Tput

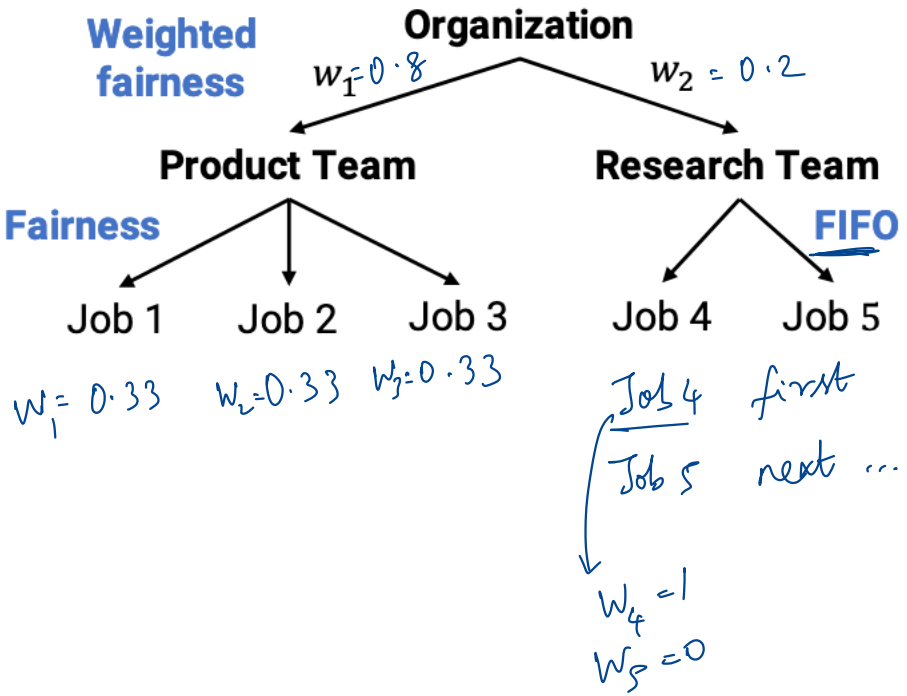
J0 = 0.45 × 40 + 0 × 10

= 18

J1 = 5.76

J2 = 54.5

HIERARCHICAL POLICIES



Share physical cluster among sub-organizations
Different policies at levels of hierarchy

Solve an LP problem across the organization
Weights constrained by policy within entity
(e.g., w4 = 1 and w5 = 0)

Use water-filling to remove bottlenecked jobs

MECHANISM: ROUND-BASED SCHEDULING

Schedule in "rounds" – every round is ~6 mins

In every round:

- Consider a list of schedulable jobs and X^{opt} (from policy)
- Decide which jobs are chosen to run in this round
  - Track time spent by job m on accelerator type j
- Give high priority to jobs which are farthest from X^{opt}

Greedy policy that converges across rounds

→ more jobs than available machines in cluster

→ realise X^{opt} over time because not feasible to realize at one go!

MECHANISM: PRIORITIES

$$X^{\text{example}} = \begin{array}{c} \begin{matrix} V100 & P100 & K80 \end{matrix} \\ \begin{pmatrix} 0.6 & 0.4 & 0.0 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.0 & 0.8 \end{pmatrix} \end{array} \begin{matrix} \text{job 0} \\ \text{job 1} \\ \text{job 2} \end{matrix}$$

→ Allocation we want to attain.

V100 | P100 | K80

$$\begin{pmatrix} 3 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 4 \end{pmatrix} \begin{matrix} \text{job 0} \\ \text{job 1} \\ \text{job 2} \end{matrix}$$

rounds_received_n

V100 | P100 | K80

$$\begin{pmatrix} 0.2 & 0.4 & 0 \\ 0.2 & 0.2 & \infty \\ \infty & 0 & 0.2 \end{pmatrix} \begin{matrix} \text{job 0} \\ \text{job 1} \\ \text{job 2} \end{matrix}$$

priorities_n

round_recv / priority

0.6 / 3 = 0.2

priority that shows which resource type is most important

V100 | P100 | K80

$$\begin{pmatrix} 3 & 2 & 0 \\ 1 & 3 & 1 \\ 1 & 0 & 4 \end{pmatrix} \begin{matrix} \text{job 0} \\ \text{job 1} \\ \text{job 2} \end{matrix}$$

$\text{rounds\_received}_{n+1}$

Jobs placed on resources where they have high priority (marked in **red**)

SUMMARY

DL training workloads properties

Clusters with mix of accelerators

Gavel: Framework to capture many scheduling goals

Mechanism based on round-based assignments

DISCUSSION

https://forms.gle/Y5J5NrD4ZwoKGjw76

What are some similarities or differences between Mesos/DRF and DL schedulers like Gavel?

Similarities ?

- Sharing incentive
  Pareto efficiency
-

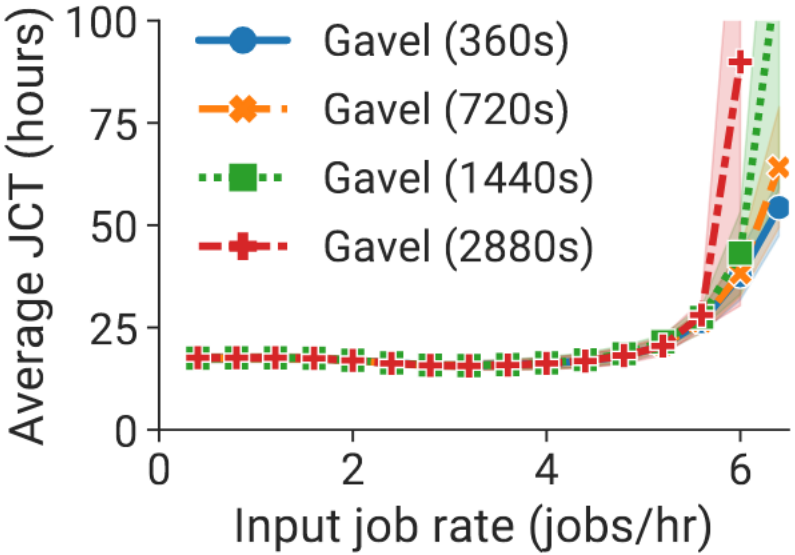
Differences ?

- Dominant resource ⇒ fair sharing
  Tput of jobs ↗ fair sharing
  ↘ JCT,
- Mesos master just makes offers (ignores heterogeneity)

- Allocate until task pre-emption. Mesos Round based scheduling (Gavel)
- Centralized vs. Decentralized

When input job rates are small

- Avg. JCT constant
- higher job rate exponentially goes up



NEXT STEPS

Next Class: Nexus

Course Project Introductions!

Midterm after that!