CS 744: GAVEL

Shivaram Venkataraman

Spring 2024

ADMINISTRIVIA

- Course project assignments
  - Emails will go out end of this week (March 1)
  - Introductions due March 8th

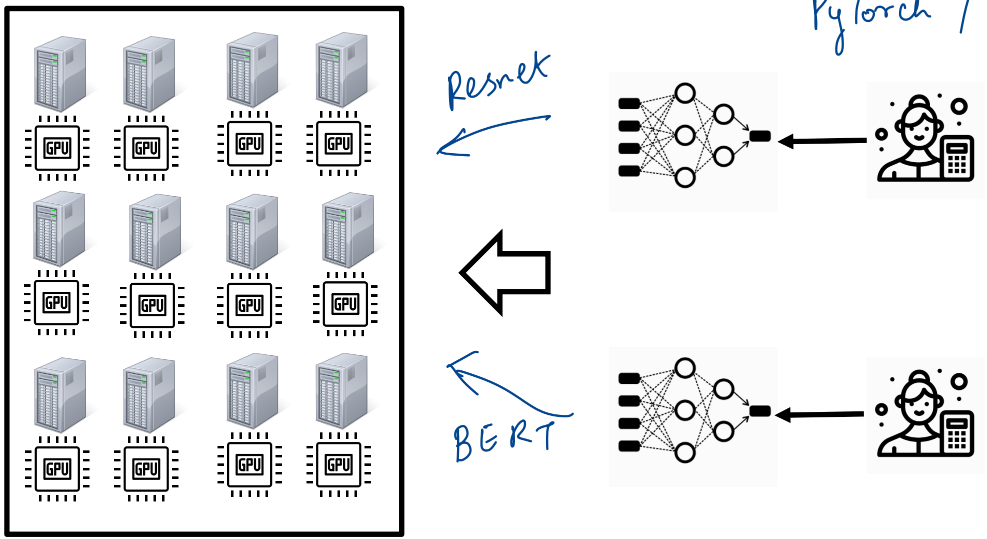
- Midterm Exam
  - In class on March 14th
  - Includes everything from beginning to the end of scheduling (including INFaaS)

DC as a computer

MACHINE LEARNING: TRAINING

Microsoft
University

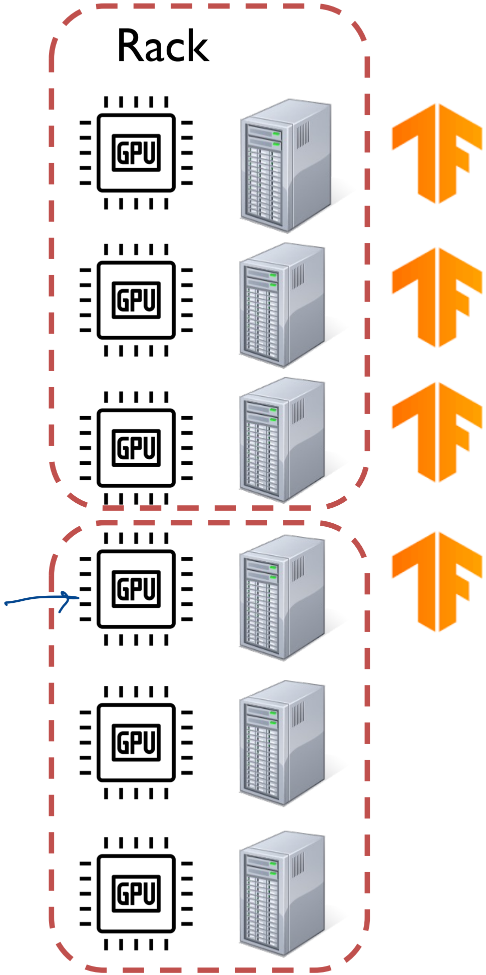
PyTorch / Pipedream

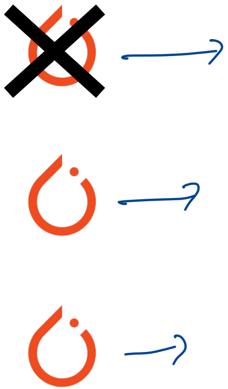


↳ GPU devices
accelerators

Setup is
more focused
on a specific
workload

WORKLOAD CHARACTERISTICS





Long running tasks ← hours / days

Gang scheduling → run together at the same time

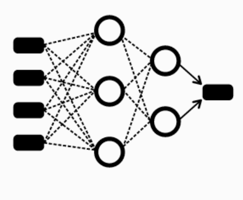
Heterogeneity? → Hardware generations K80, V100, A100

Task runs until training completes

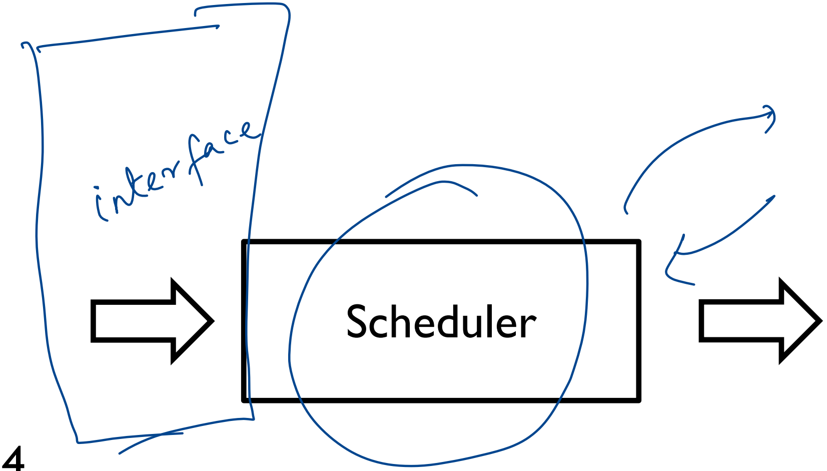
Created by Phonlaphat Thongsriphong from Noun Project

DL SCHEDULER INTERFACE

Job



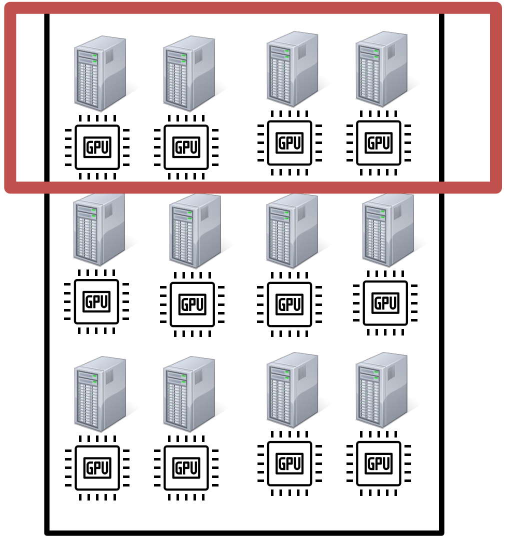
Run job Resnet18
With BatchSize = 64
on Num GPUs = 4



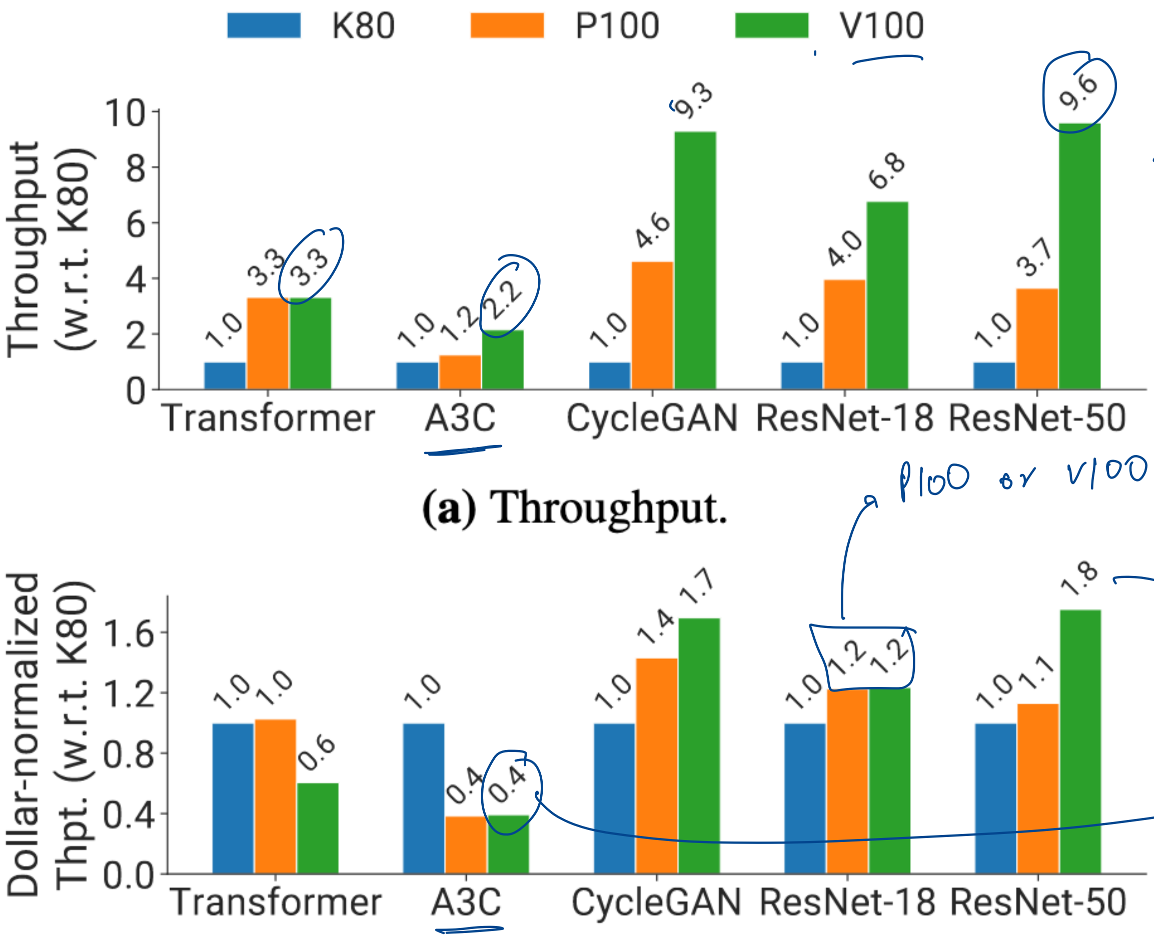
Goals:

- Maximize throughput
- Fairness
- Minimize JCT
- …

allocate these 4 GPUs to job 0



MOTIVATION: HETEROGENEITY



(a) Throughput.

(b) Dollar-normalized.

speed up with a newer GPU is diff for diff models

P100 or V100

lower than 9.6

worse to use V100 compared to K80

ADDITIONAL GOALS

list of jobs $J_1 \dots J_n$

utilization

- Support a wide range of objectives

  Minimize makespan → last job finishes

  responsiveness ← Average JCT

  Fairness (Sharing incentive) → DRF

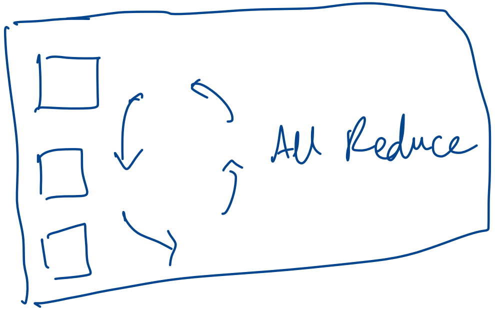
  …

General framework

→ Pluggable Policy

→

- Placement sensitivity/Co-location → locality MR task close to input



close to each other → same mc

same rack

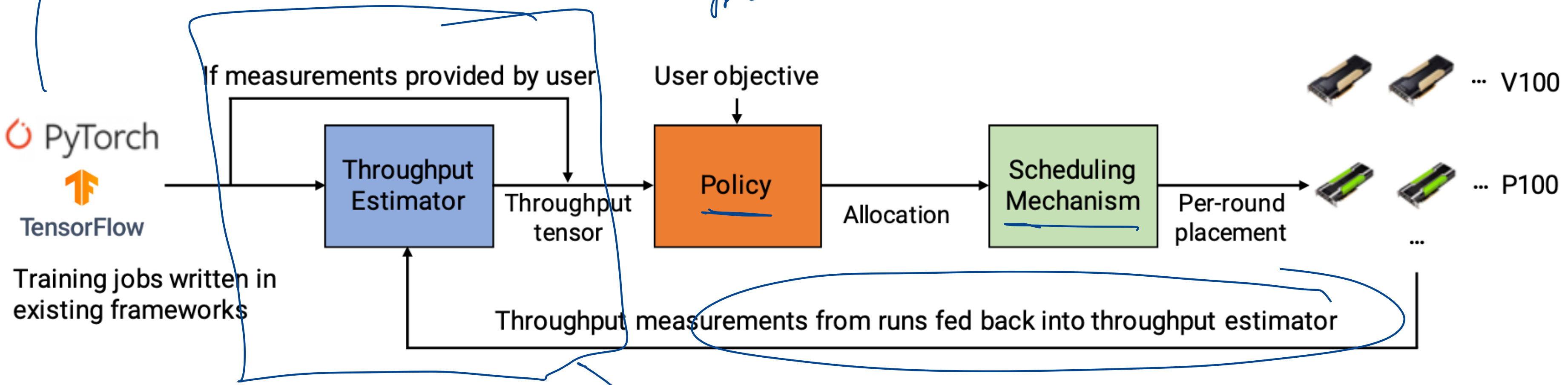
⋮

GAVEL: SYSTEM DESIGN

Simplicity
→ submits a job

Accuracy of profile affect the scheduler?

How fast does each model run on each hardware?



PyTorch

TensorFlow

Training jobs written in existing frameworks

If measurements provided by user

Throughput Estimator

Throughput tensor

User objective

Policy

Allocation

Scheduling Mechanism

Per-round placement

... V100

... P100

...

Throughput measurements from runs fed back into throughput estimator

profiling step
input to the
scheduler

What is
the overhead?

Throughput profiles
augmented using
feedback loop

SCHEDULING POLICY: OPTIMIZATION PROBLEM

obj: maximize total tput

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

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

weighted by acc. types in cluster

$$\begin{array}{lll} 0 \leq X_{mj} \leq 1 & \forall (m, j) & (1) \\ \sum_j X_{mj} \leq 1 & \forall m & (2) \\ \sum_m X_{mj} \cdot \text{scale\_factor}_m \leq \text{num\_workers}_j & \forall j & (3) \end{array}$$

total is <= 1

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

solve this opt problem

get back X → allocation that achieves this objective

POLICY: MAX-MIN FAIRNESS

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

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

$x_0 = 0.33$

$x_1 = 0.33$ → equal share if all GPUs are same

$x_2 = 0.33$

Gavel: Use weighted normalized effective throughputs

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

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

weighted sum per acc. type

EXAMPLE

Profiler or tput estimator

$$T = \begin{array}{cc} V100 & K80 & \\ \left(\begin{array}{cc} 40.0 & 10.0 \\ 12.0 & 4.0 \\ 100.0 & 50.0 \end{array}\right) & \begin{array}{l} \text{job } 0 \\ \text{job } 1 \\ \text{job } 2 \end{array} \end{array}$$

homogeneous

$$\underline{X}^{\text{hom.}} = \begin{array}{cc} \begin{bmatrix} 0.33 & 0.33 \\ 0.33 & 0.33 \\ 0.33 & 0.33 \end{bmatrix} & \begin{array}{l} J_0 \\ J_1 \\ J_2 \end{array} \end{array}$$

$$\underline{X}^{\text{het.}} = \begin{array}{cc} V100 & K80 & \\ \left(\begin{array}{cc} 0.45 & 0.0 \\ 0.45 & 0.09 \\ 0.09 & 0.91 \end{array}\right) & \begin{array}{l} \text{job } 0 \\ \text{job } 1 \\ \text{job } 2 \end{array} \end{array}$$

higher tput

Eff tput

$J_0 = 40 \times 0.33 + 10 \times 0.33 = 16.5$

$J_1 = 12 \times 0.33 + 4 \times 0.33 = 5.28$

$J_2 = 49.5$

$J_0 = 0.45 \times 40 = \boxed{18}$

$J_1 = 0.45 \times 12 + 0.09 \times 4 = \boxed{5.76}$

$J_2 = 0.09 \times 100 + 0.91 \times 50 = \boxed{54.5}$

HIERARCHICAL POLICIES

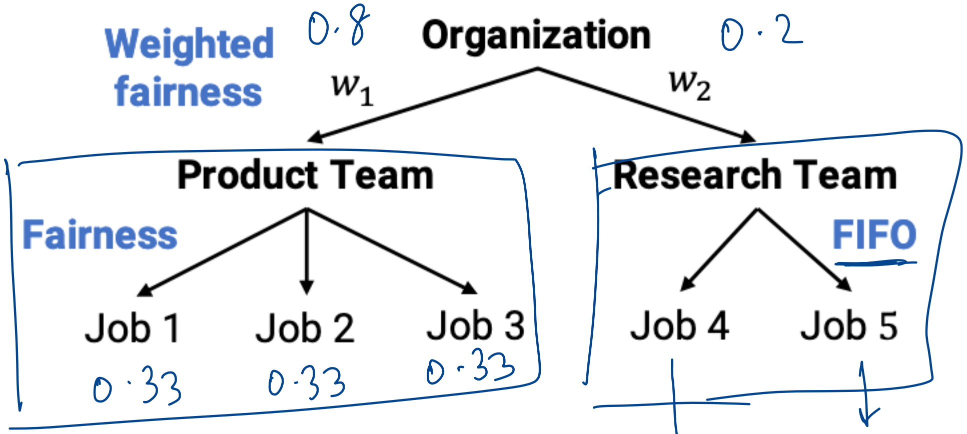
Single resource

GPUs

"Synergy"

Share physical cluster among sub-organizations

Different policies at levels of hierarchy



$w_4 = 1.0$ needs to run first

$w_5 = 0$

↳ only start after job 4 finishes

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

↳ Prior work in max min fairness

MECHANISM: ROUND-BASED SCHEDULING

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

↳ very long running tasks

↳ end of a round, pause stop tasks

solution opt problem

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 → Compute next round

Greedy policy that converges across rounds → Start jobs for next round

MECHANISM: PRIORITIES

recompute X_{opt} when jobs arrive/leave

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

→ X^{OPT} target allocation

$$
\begin{array}{ccc|c}
V100 & P100 & K80 & \\
3 & 1 & 0 & \text{job } 0 \\
1 & 3 & 0 & \text{job } 1 \\
0 & 0 & 4 & \text{job } 2
\end{array}
\quad \Rightarrow \quad
\begin{array}{ccc|c}
V100 & P100 & K80 & \\
0.2 & 0.4 & 0 & \text{job } 0 \\
0.2 & 0.2 & \infty & \text{job } 1 \\
\infty & 0 & 0.2 & \text{job } 2
\end{array}
$$

rounds_received$_n$ priorities$_n$

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

$$
\begin{array}{ccc|c}
V100 & P100 & K80 & \\
3 & 2 & 0 & \text{job } 0 \\
1 & 3 & 1 & \text{job } 1 \\
1 & 0 & 4 & \text{job } 2
\end{array}
$$

rounds_received$_{n+1}$

$$\text{Priority} = \frac{x_{jm}}{rr_{jm}}$$ element wise division

Job 1	K80
Job 2	V100
Job 0	P100

- Switch frequently
- Doesnt work jobs shorter than 1 round

SUMMARY

DL training workloads properties

Clusters with mix of accelerators

Gavel: Framework to capture many scheduling goals

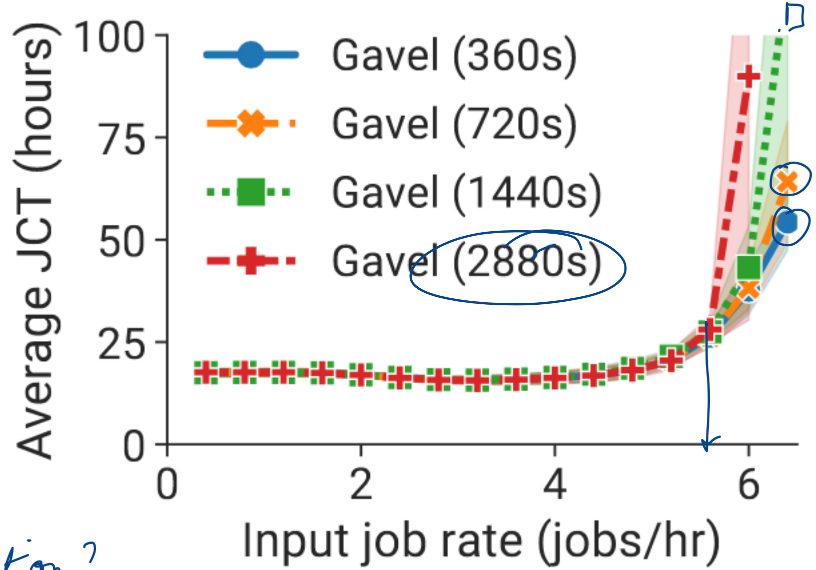
Mechanism based on round-based assignments

DISCUSSION

https://forms.gle/pYnFErGi54HEHcuj7

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

round length



round length is high JCT is high for > 5.5 jobs/hr

small round length

→ overhead of computing allocation?

→ Pre-emption overhead (< 10%?)

→ Increase seems to be linear!

→ round length is large

→ fragmentation jobs finish early in the round

NEXT STEPS

Next Class: INFaaS

Course Project Introductions!