Good morning!

CS 744: INFAAS

Shivaram Venkataraman

Spring 2024

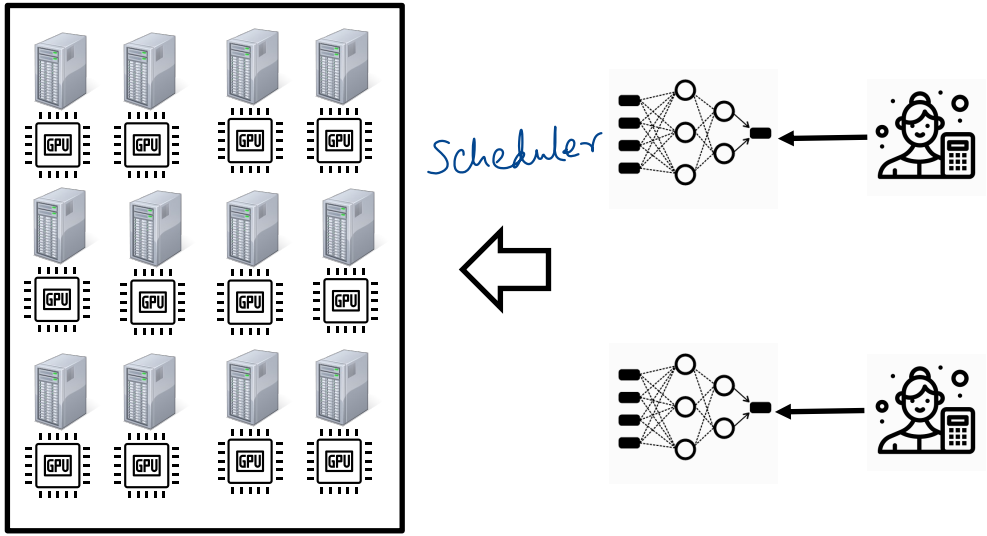
ADMINISTRIVIA

- Assignment 1 → office hours

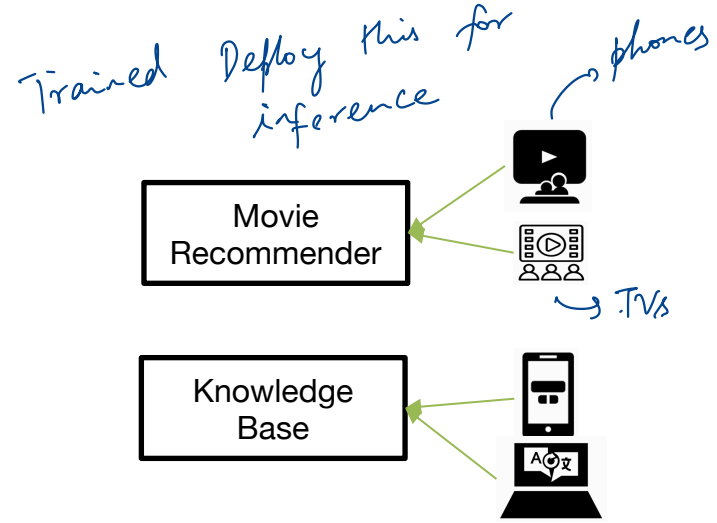
- Course project
  - Introductions due March 8th → template ~ 2 page

- Midterm Exam
  - In class on March 14th → Next Thursday
  - Includes everything from beginning to the end of scheduling (including INFaaS)
  - Sample papers on Piazza

MACHINE LEARNING: TRAINING, INFERENCE



TRAINING



INFERENCE

ML INFERENCE: MODEL-FUL?

Tensorflow Serving

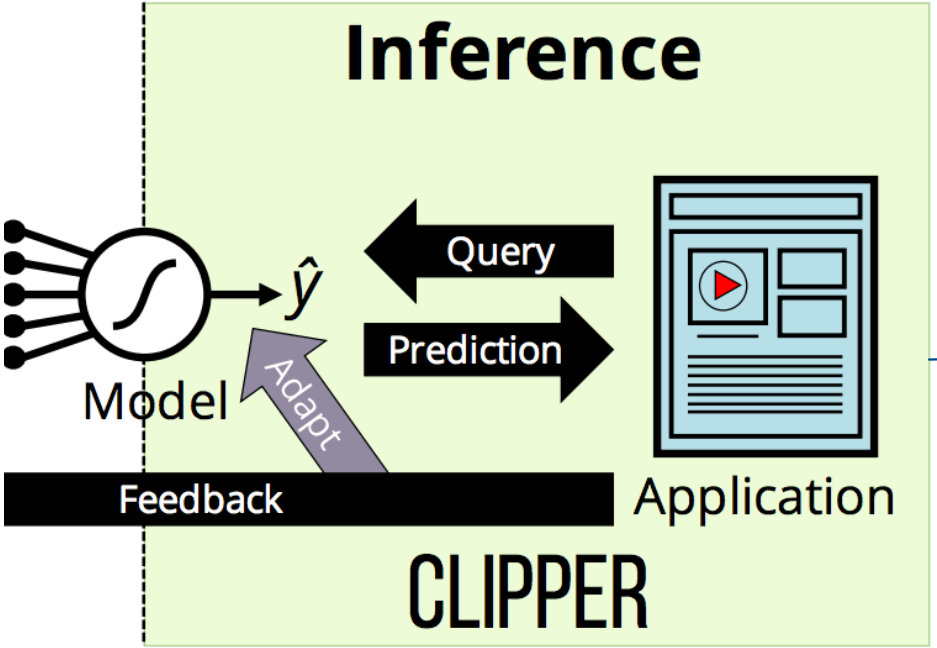
Clipper

Interface

inference (model name, input data)

model-ful?

↳ result of inference



inference ↳ slow? → latency SLO.

Developer needs to make informed choice

MOTIVATION: MODEL VARIANTS!

Scale

Image classification

Application	Accuracy	Latency	Cost
Social Media	High	Medium	Low
Visual Guidance	High	Low	High
Intruder Detection	Low	Low	Low

Diverse Applications

→ App requirements are diverse

Heterogeneous environments

Diverse model variants

Resnet-18, Resnet-50, EfficientNet

↓

batch size

CPU / GPU

etc.

Model Variants

Variant (hardware, framework)	Lat. (ms)	Req/s	Cost ($/s)
A (4 CPUs, TensorFlow)	200	5	1
→ B (1 Inferentia core, Neuron)	20	100	3
C (1 V100 GPU, TensorRT)	15	800	16

1 instance of B. Cost $3.
1 instance of C. Cost $16

2 instances of A. Cost $2/s

QPS	SLO (ms)	#Var. A	#Var. B	#Var. C	Cost ($/s)
10	300	2	0	0	$2
10	50	0	1	0	
1000	300	0	2	1	

↳ requirements → Assignment ↳ What we want to automate

INFAAS: MODEL-LESS SERVING

Insight: Opportunity to pick right model for a query

User specifies performance, cost, or accuracy requirements

System manages model variants at runtime

→ change the API being used so that system picks model variant!

INFAAS: OUTLINE

Model Variants

API Design & Architecture

Selecting, Scaling Model variants

MODEL VARIANT

Architecture → Resnet-18, VGG-19

Programming Framework → PyTorch, Tensorflow

Graph Optimizer → XLA, TensorRT → DAG of operators are in a model

Batch size → small batch size → low latency, not high utilization → low tput

vs

large

Hardware

↳ CPU, GPUs, new accelerators

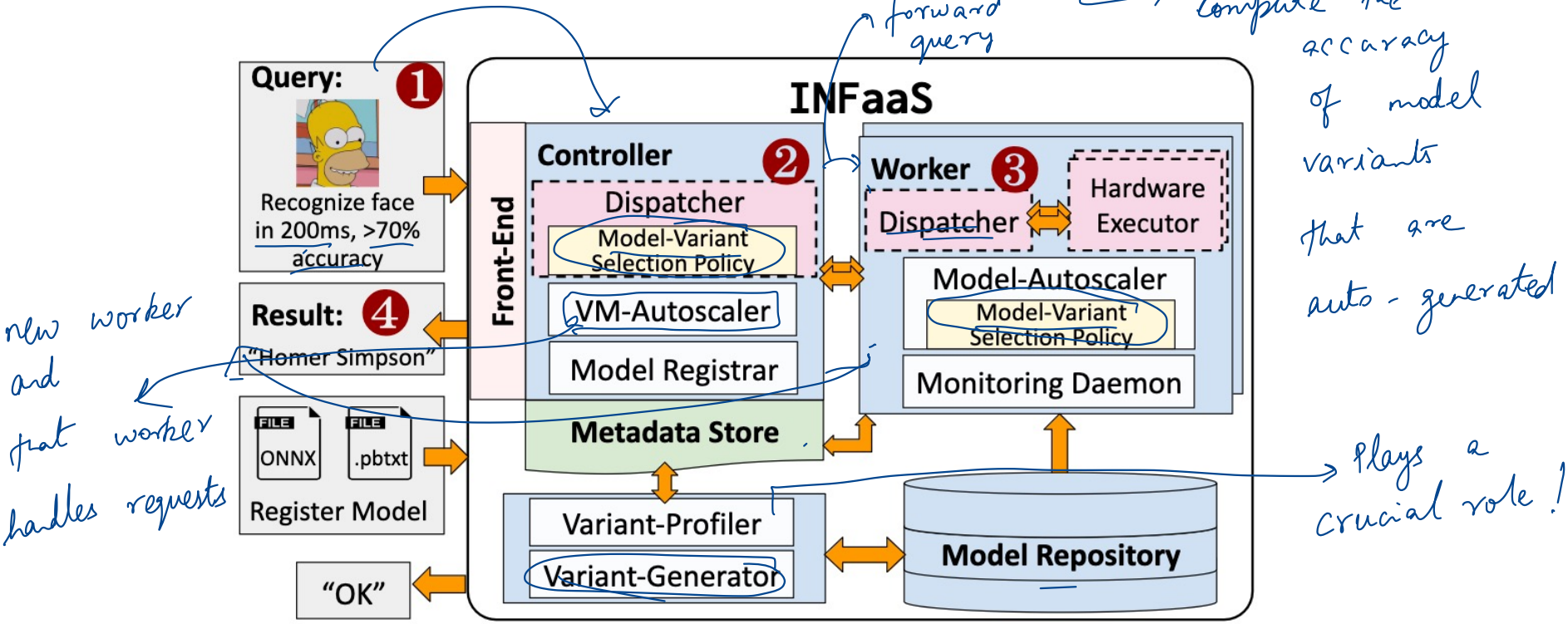
```
1 register_model("ResNet50",ResNet50.pt,valSet,detectFaceApp)
2 register_model("MobileNet",MobileNet.pt,valSet,detectFaceApp)
```

model weights

appID

compute the accuracy of model variants that are auto-generated

forward query



new worker and that worker handles requests

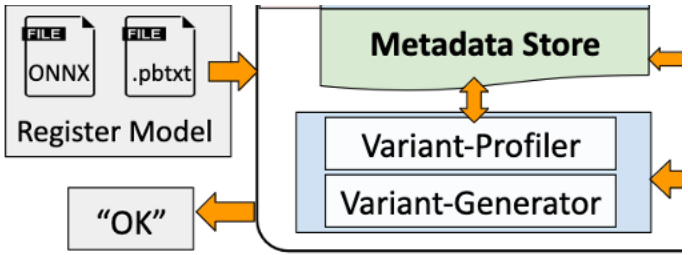
plays a crucial role!

VARIANT GENERATOR

Generates model variants

Accuracy using val set

Profiler used to measure latency, memory use, loading time etc.



MODEL VARIANT SELECTION POLICY

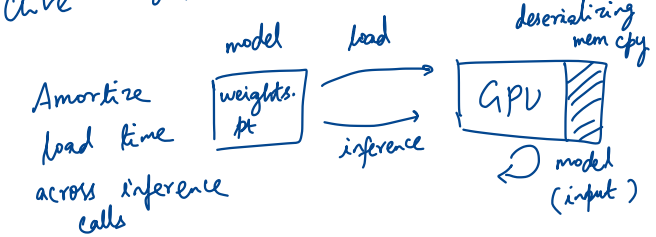
Case 1: Query arrival → Given an input query, forward this to appropriate worker

Goal: minimize time to select model variant → Critical path. Counts towards SLO

Approach

- Select least loaded active variant (no loading time)
- If not, select a variant with low loading+inference time and worker with low utilization → space to load model

→ Which model variants are "active" on which worker



MODEL VARIANT SELECTION POLICY

each → Worker

Case 2: Query load change

Options

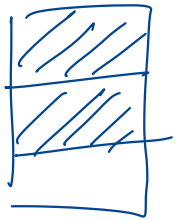
- Horizontal scaling, replication
- "Vertical" scaling!

how many QPS → Worker

1 variant Resnet-50

2 variant Resnet-50

GPU memory



2x QPS

1 variant R50 ↓ 2x R18 or 1 R50 or 2 R50 or 1 R18

↳ diff variant that can help meet requirements

classic load balancing FE

$$\text{Cost}(\delta_{ij}) = C_{ij}(\delta_{ij} + \lambda T_{ij}^{\text{load}} \max(\delta_{ij}, 0))$$

inference cost

loading cost

Optimization problem

GREEDY HEURISTIC

Prune the search space for variant selection!

- Calculate headroom available on each worker
- How to scale?
  - Estimate cost of horizontal, vertical scaling
  - Compute cost for both proposals
  - Check if it fits in the worker

req/s can this worker handle

trigger before worker gets overloaded

small number of options

memory budget

OPT problem

~ 30s or 1 min

want it to be ms instead!

SUMMARY

DL inference workloads properties

Model-variants search space large

INFaaS: Model-less serving

Scale model variants horizontally and vertically

DISCUSSION

https://forms.gle/gkNTqdrSpLVpa9rg8

List one similarities and one difference between training schedulers like Gavel and inference scheduler like INFaaS

Similarity

→ Both handle hardware heterogeneity

Focus on Cost in inference settings

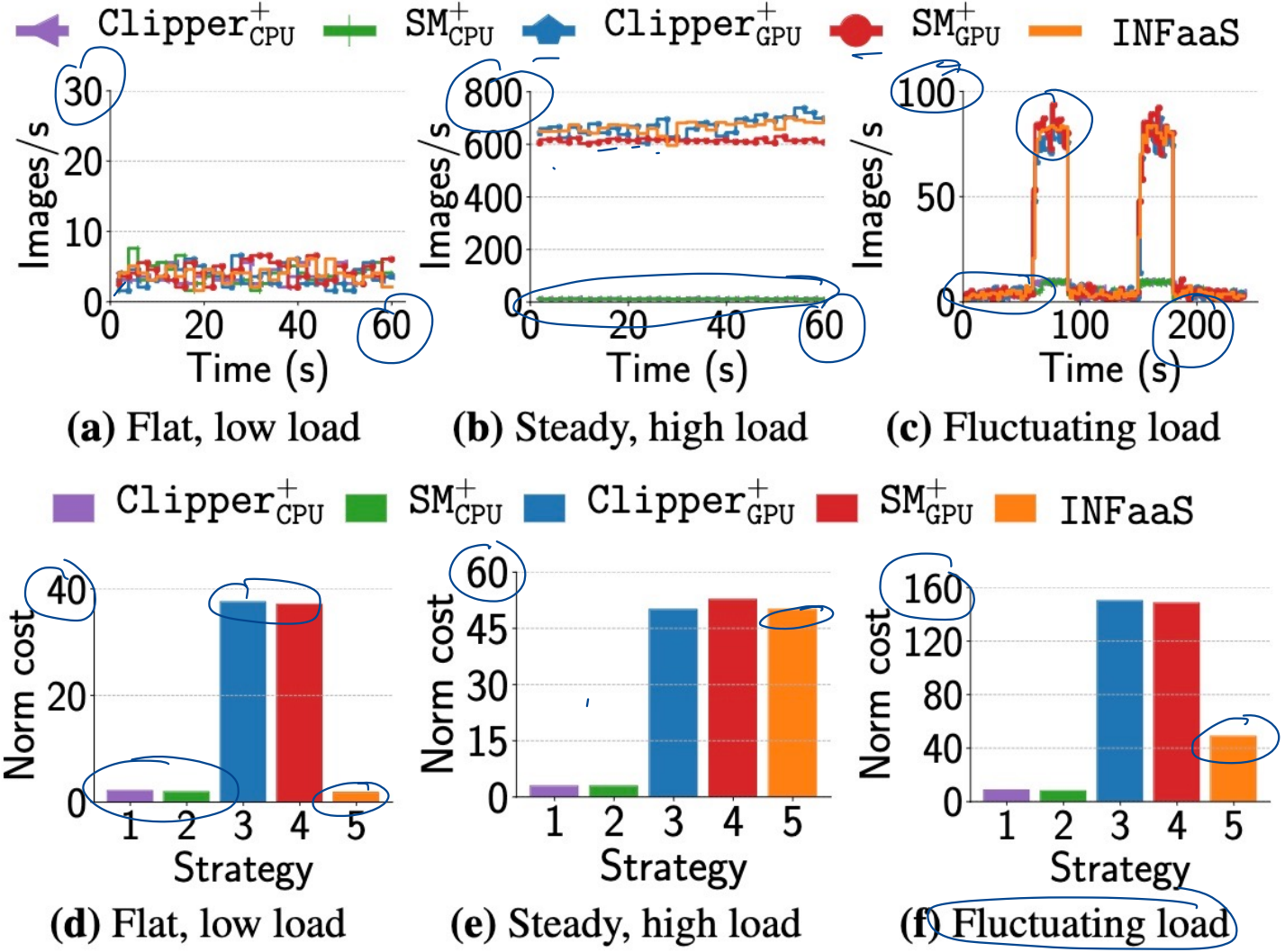
Difference

→ Gavel → fractions for each job limits of cluster

INFAAS → launch VMs / variant selection to meet tput

→ Infaas → fast decisions

Gavel slower decisions



GPUs can keep up with high load

meet tput requests

matches baselines for flat loads

reduce cost

CPU based good for low load

NEXT STEPS

Next Class: SQL

Course Project Introductions! → March 8th