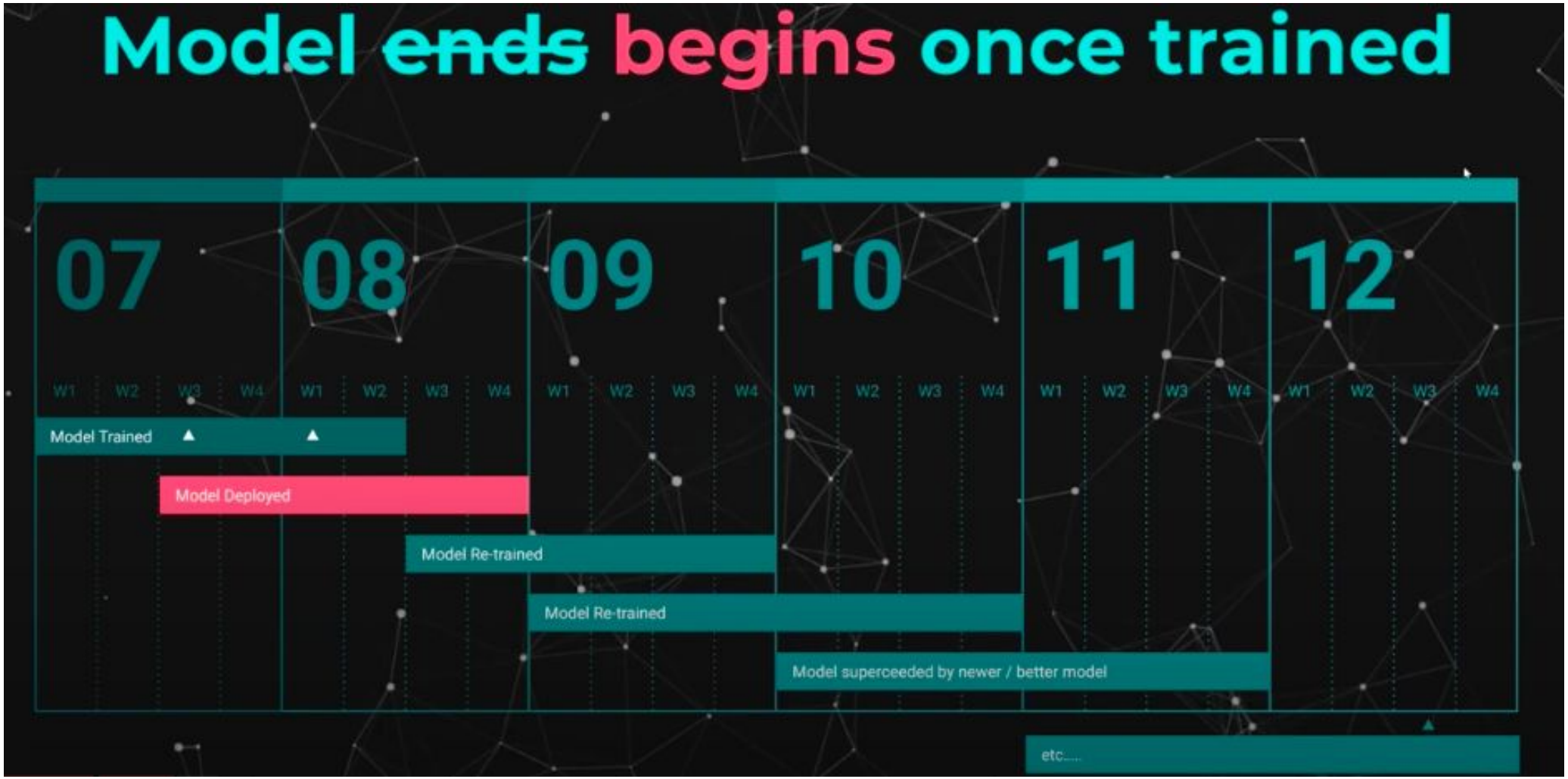
Machine Learning in Production and **AISys** Research

Saeid Ghafouri

ML Systems Course

Sep 14th 2023

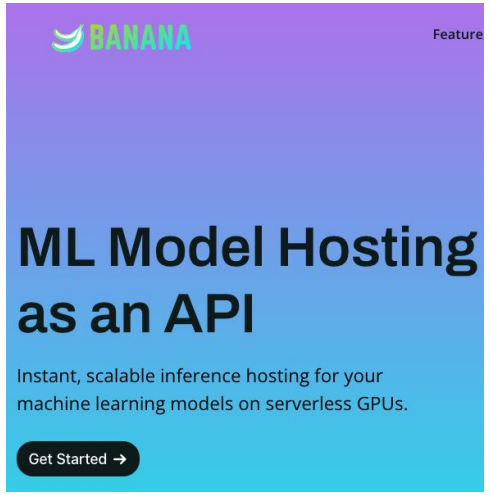
Production Machine learning life-cycle



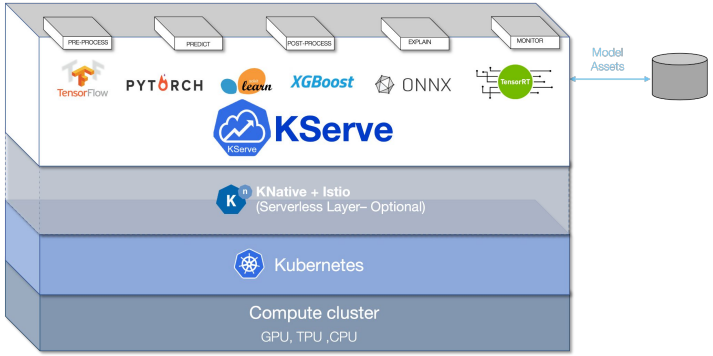
Source: https://youtu.be/QcevzK9ZuDg

ML Serving In production

- Optimizing the resources
  - Subject to the variable workload
  - Hardware
  - Budget
- Model drift detection
- Serving Infrastructure
- Model server
- Streaming Data
- Observibility of the model
- Automatic retraining
- Explainability
- ...



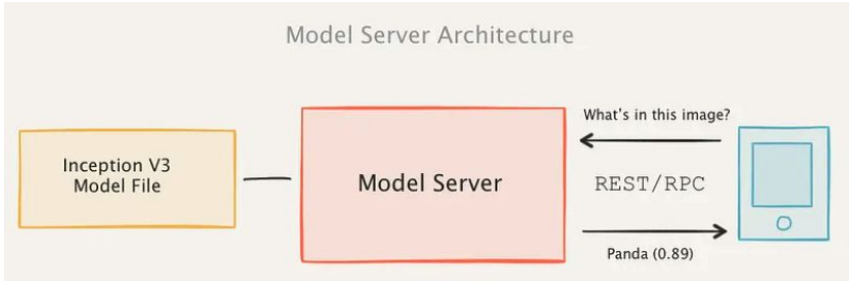
CORE

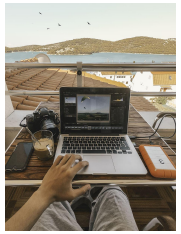


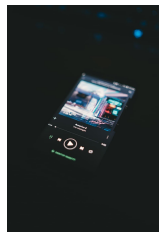
The AI community building the future.

Build, train and deploy state of the art models powered by the reference open source in machine learning.

ML Serving In production

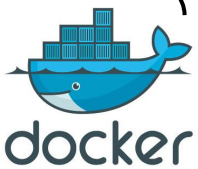






End User

web service



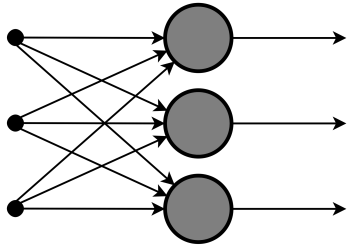
packaging

Datacenter

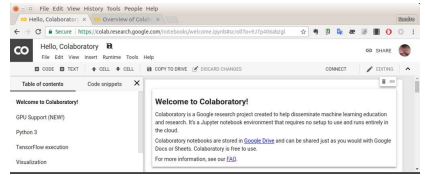
MLOps Engineer

Machine learning in Production

Exports the model in .pk .pt etc



ML Model



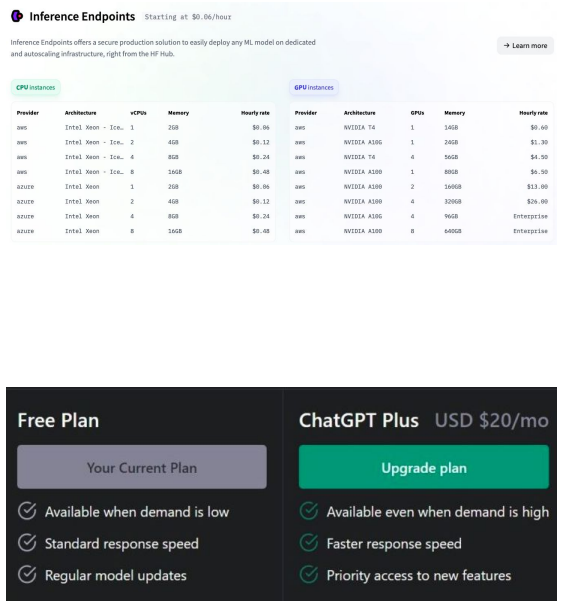
Developed in isolated environment

Data Scientist

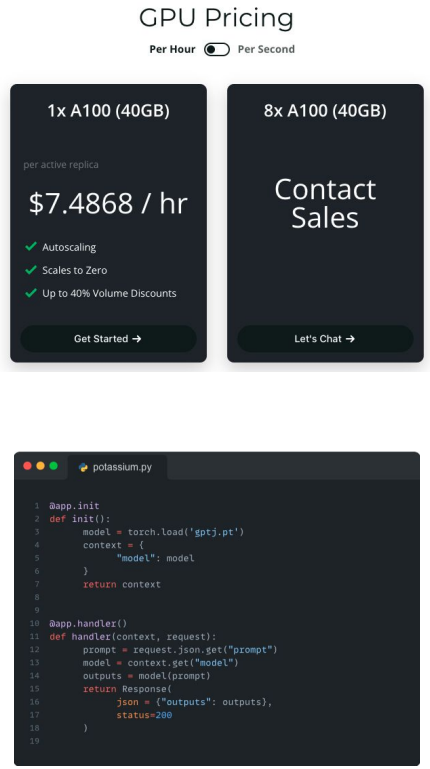
Theoretical Machine Learning

Different format of ML Deployment in production

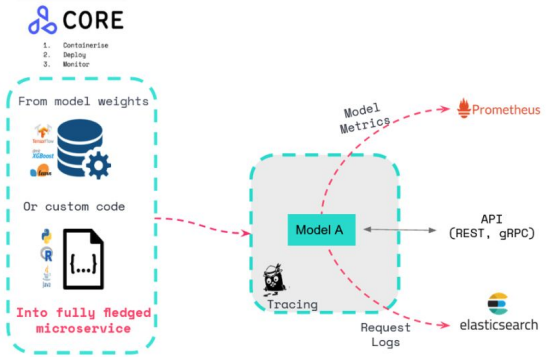
- Full API



- Serverless



- Self managed infrastructure



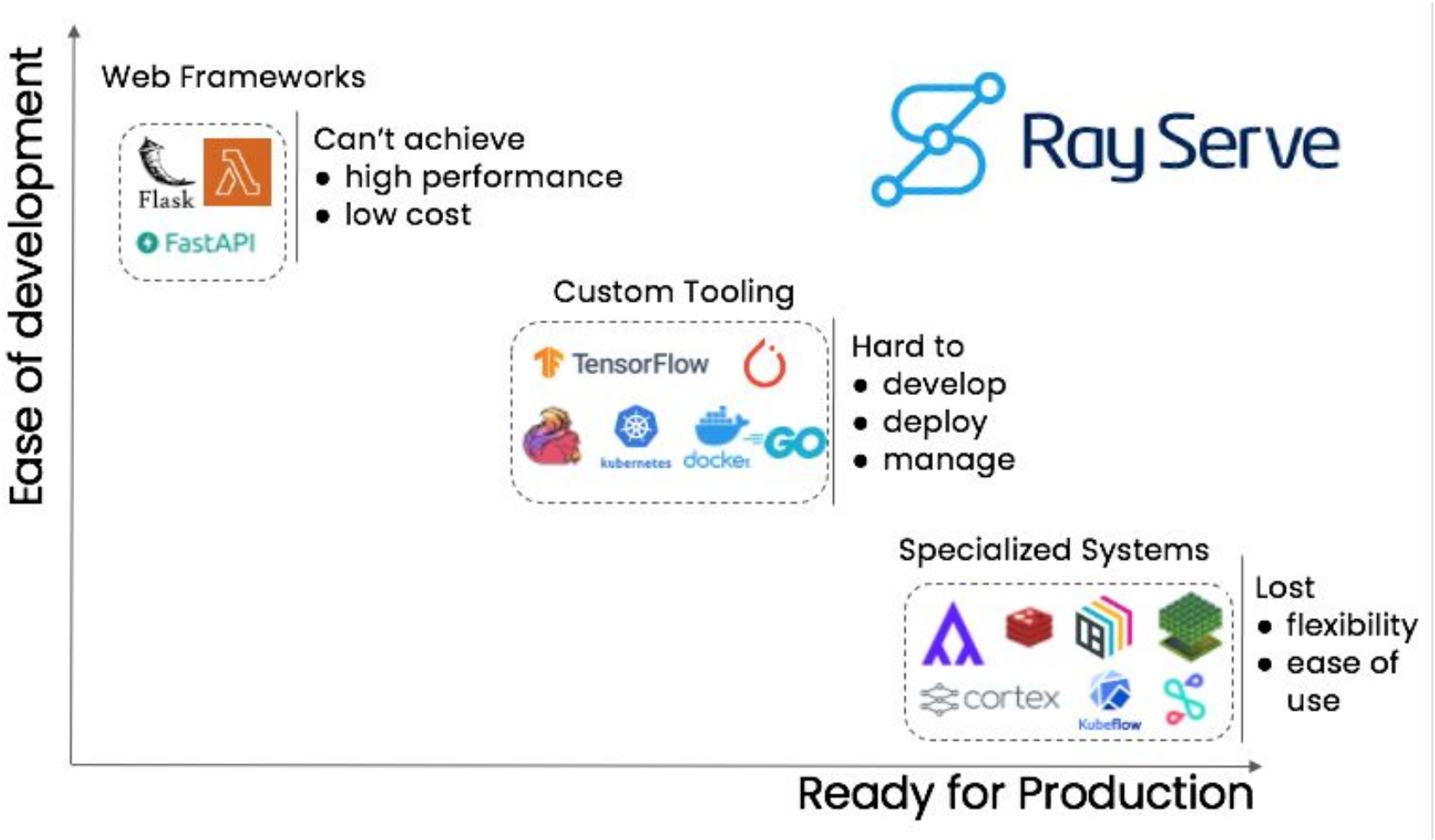
Main players

- Full API

- Serverless

- Self managed Infrastructure

Spectrum of different ways of deployment



https://www.anyscale.com/blog/serving-ml-models-in-production-common-patterns

Cloud Orchestration

Borg

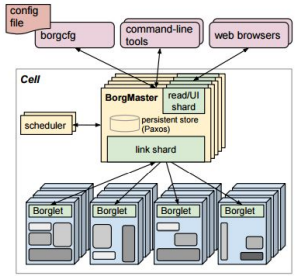


Figure 1: The high-level architecture of Borg. *Only a tiny fraction of the thousands of worker nodes are shown.*

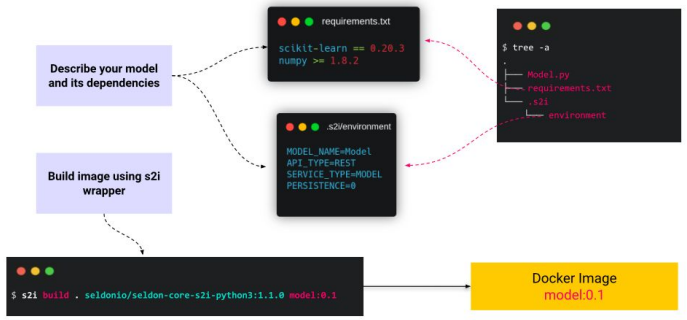
docker

Swarm

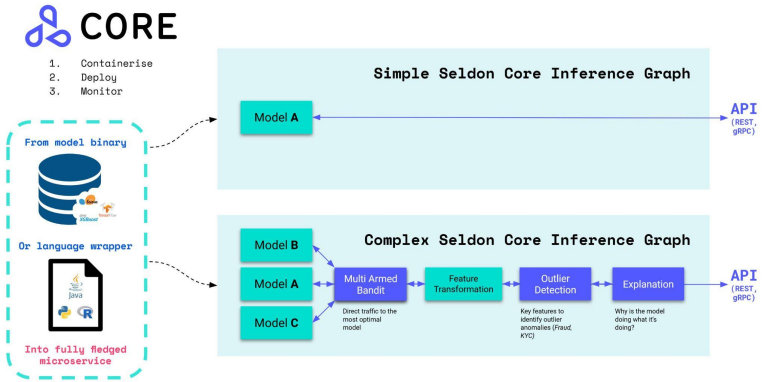
kubernetes

MESOS

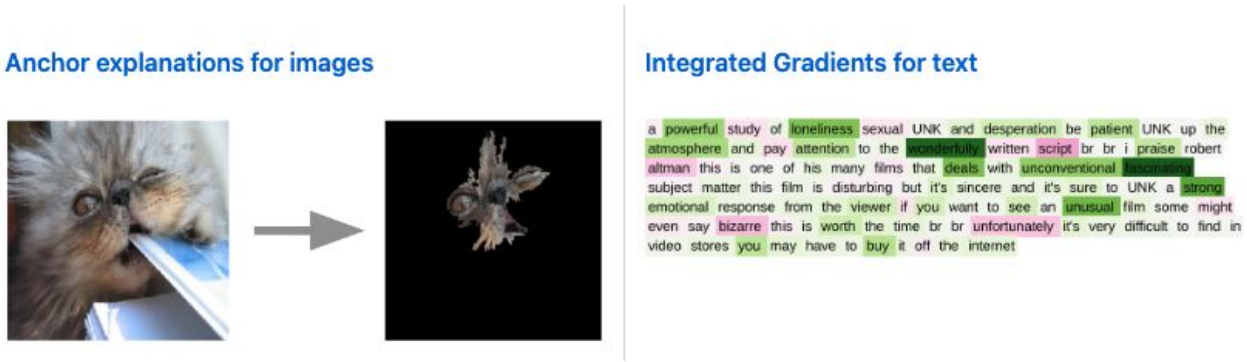
Seldon Core and ML Deployment on K8S



Kubernetes Deployment



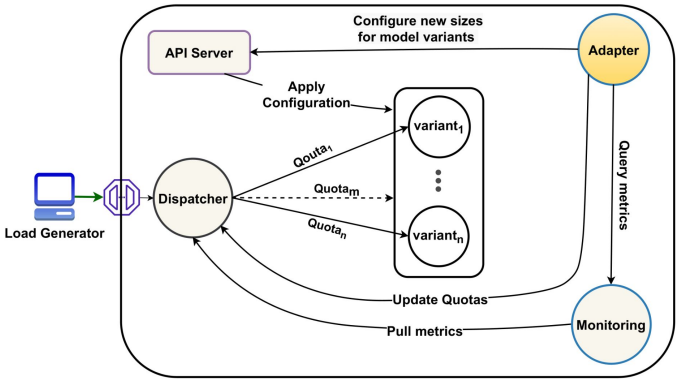
Inference pipelines



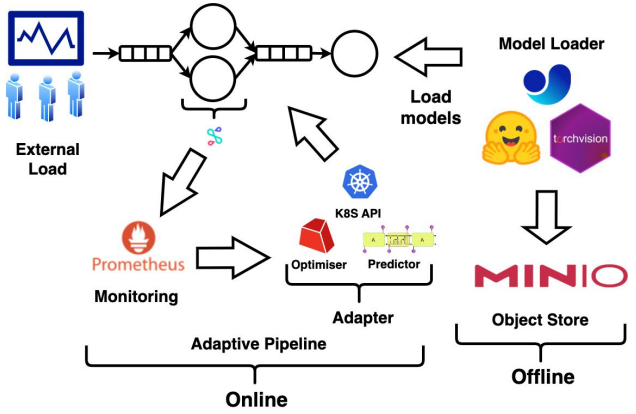
Model Explainability

Recent Projects of **AISys**

InfAdapter (EuroMLSys 2023)



IPA (under review)



Inference_x (In Progress)

InfAdapter: Reconciling High Accuracy, Cost-Efficiency, and Low Latency of Inference Serving Systems

Mehran Salmani, Saeid Ghafouri, Alireza Sanaee, Kamran Razavi

Joseph Doyle, Max Mühlhäuser, Pooyan Jamshidi, Mohsen Sharifi

“More than 90% of data center compute for ML workload, is used by inference services”

ML inference services have strict requirements

Highly Responsive!

ML inference services have strict requirements

Highly Responsive!

Cost-Efficient!

ML inference services have strict requirements

Highly Responsive!

Cost-Efficient!

Highly Accurate!

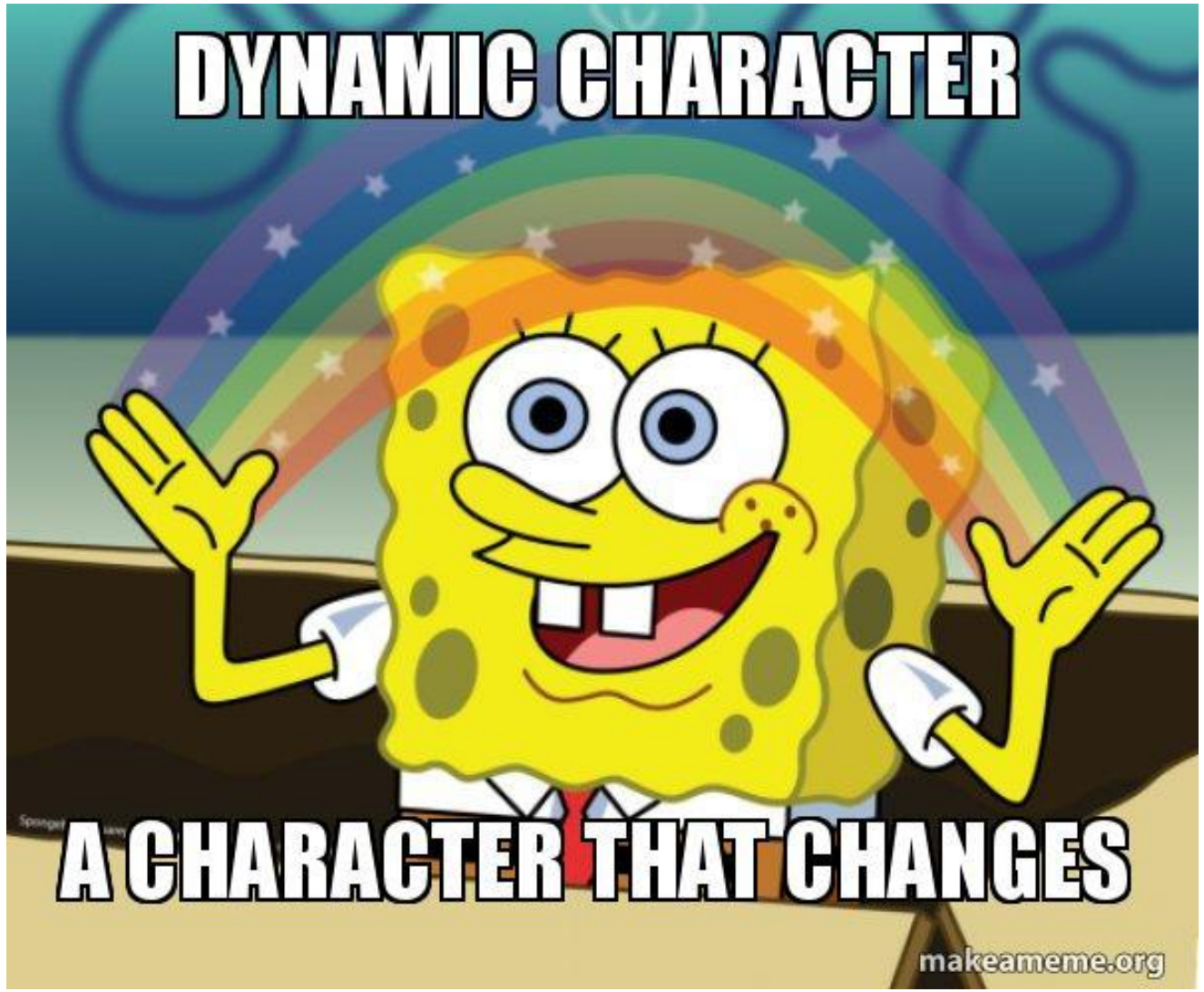
ML inference services have strict & **conflicting** requirements

Highly Responsive!

Cost-Efficient!

Highly Accurate!

More challenge: Dynamic workload

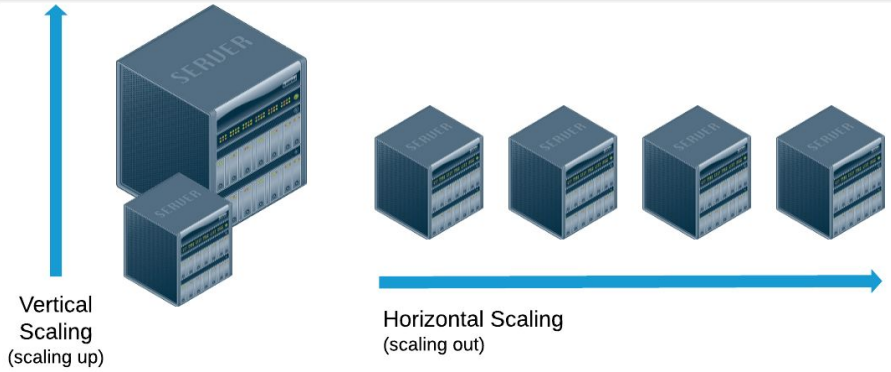


Existing adaptation mechanisms

Resource Scaling

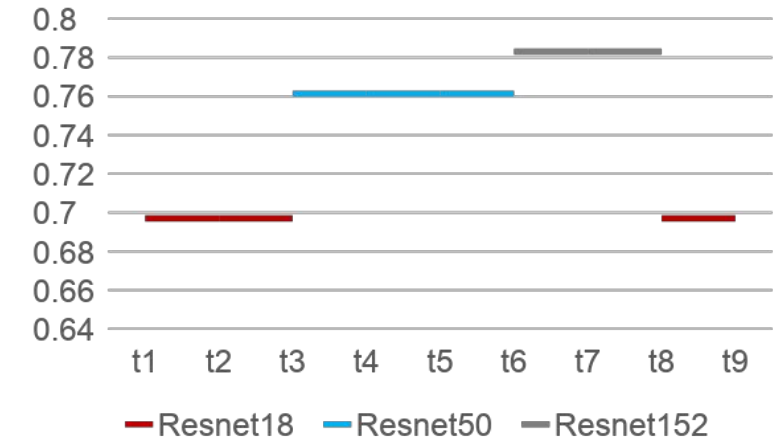
Vertical Scaling (AutoPilot EuroSys'20)

Horizontal Scaling (MArk ATC'19)



Quality Adaptation

Multi Variants (Model-Switching Hotcloud'20)

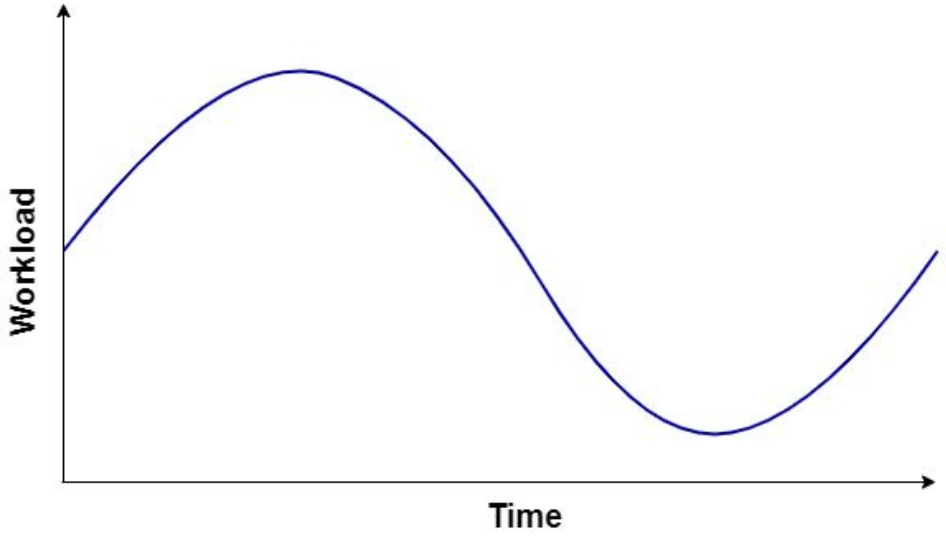


Resource allocation

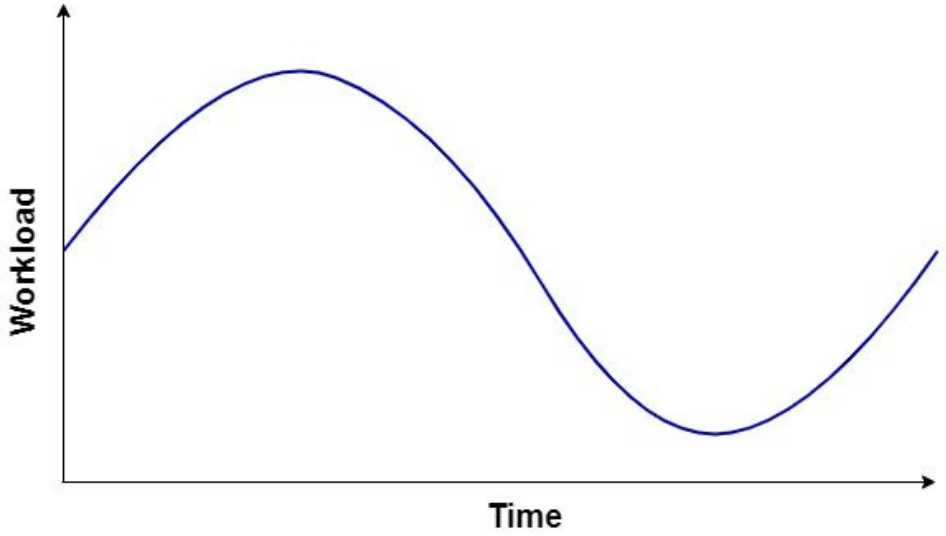
Over Provisioning

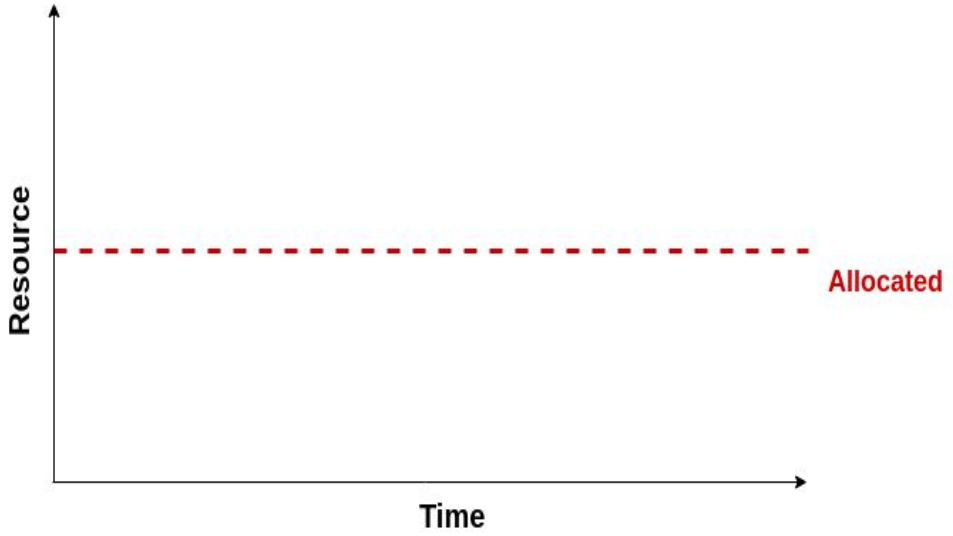
Under Provisioning

Resource allocation

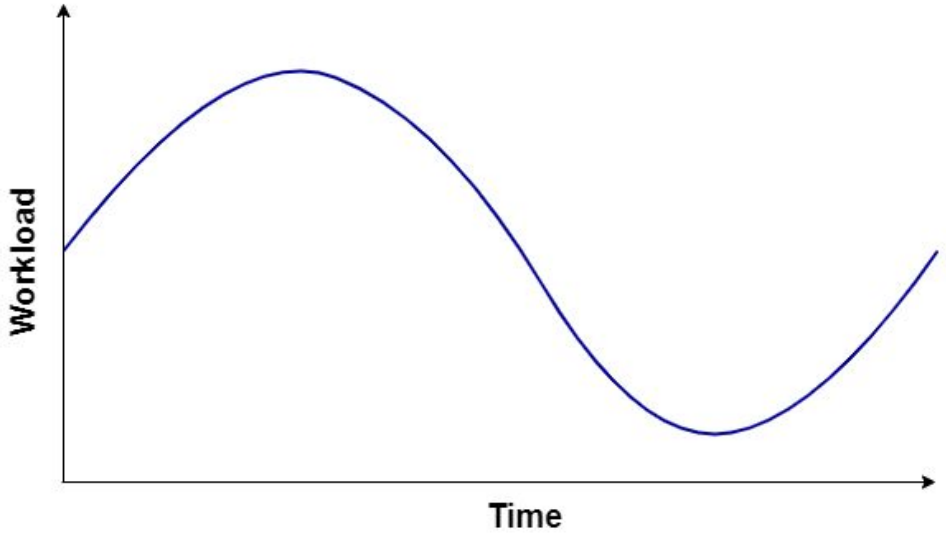


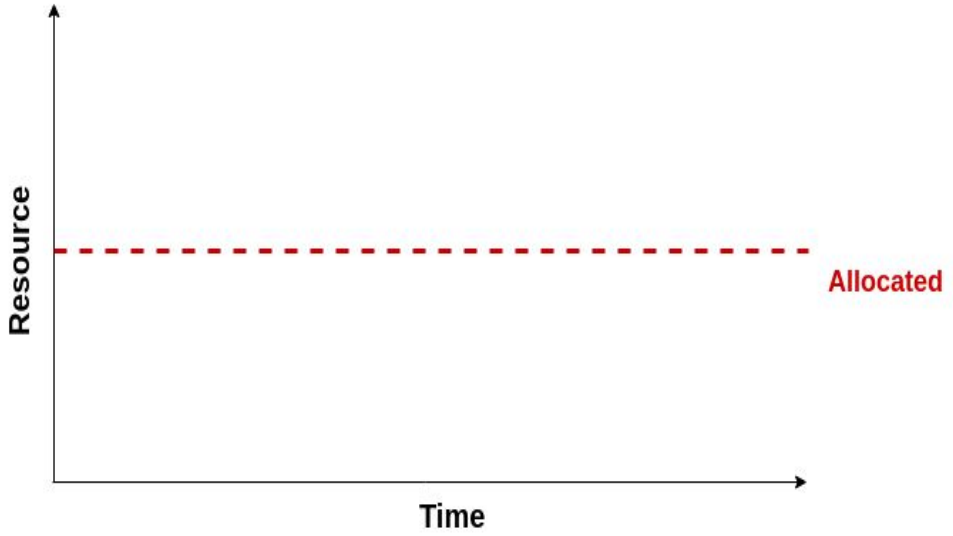
Resource allocation

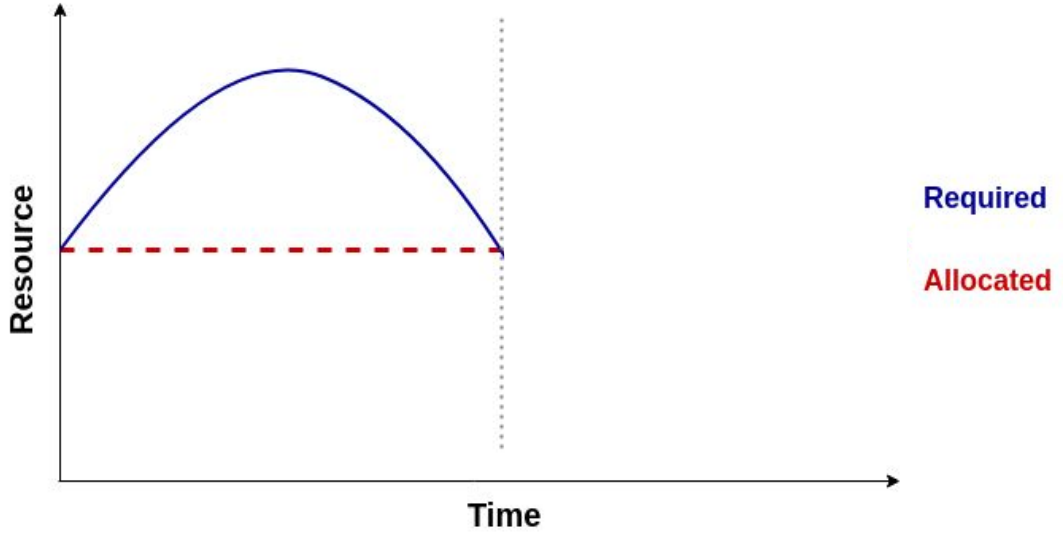




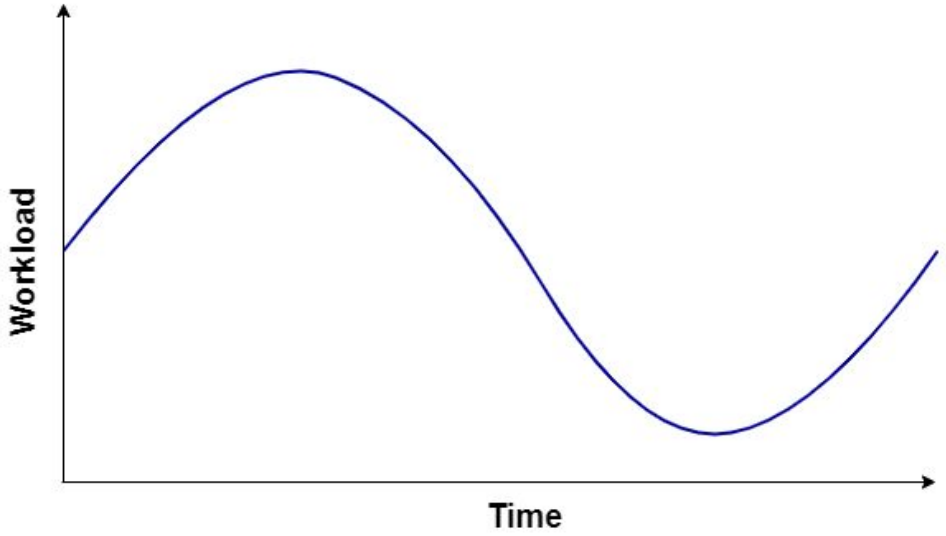
Resource allocation

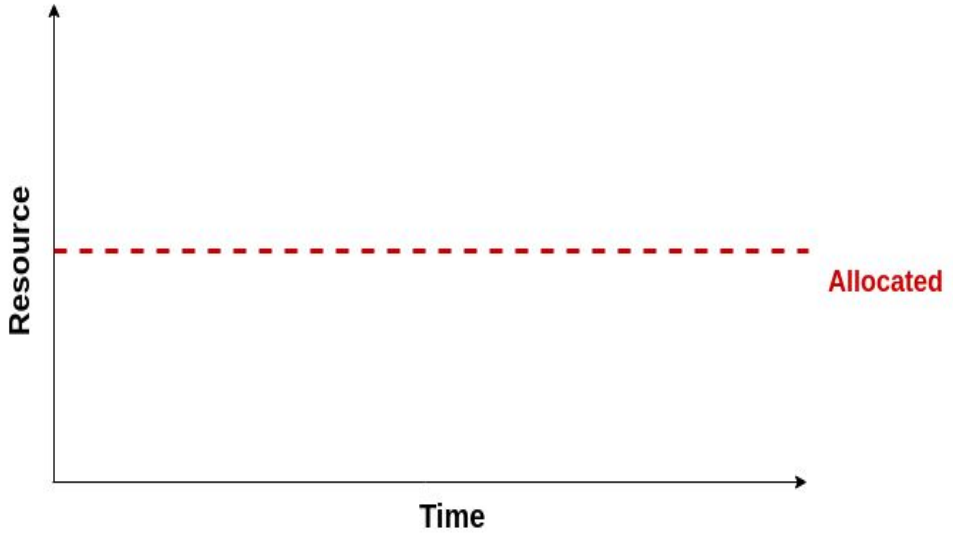


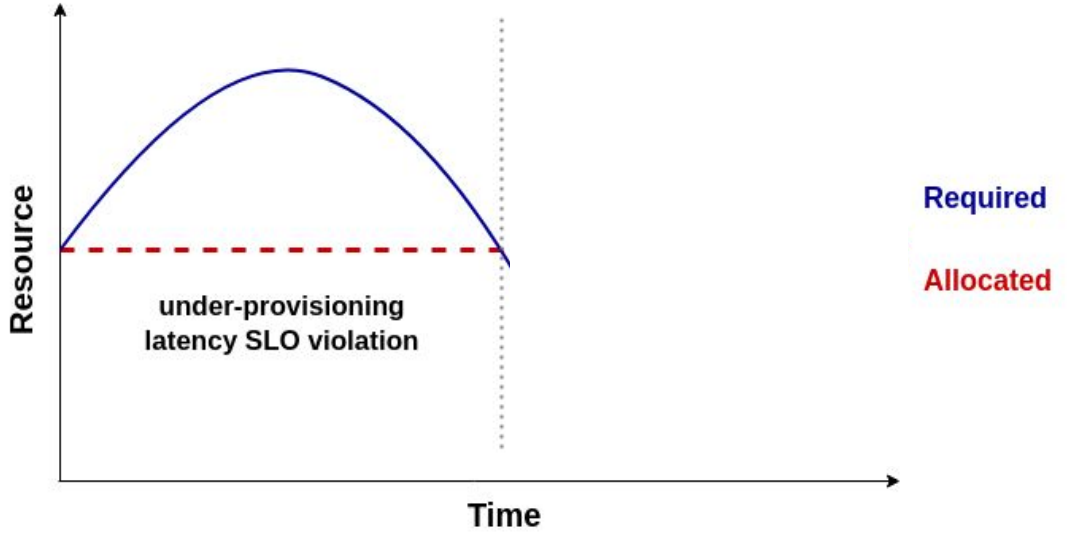




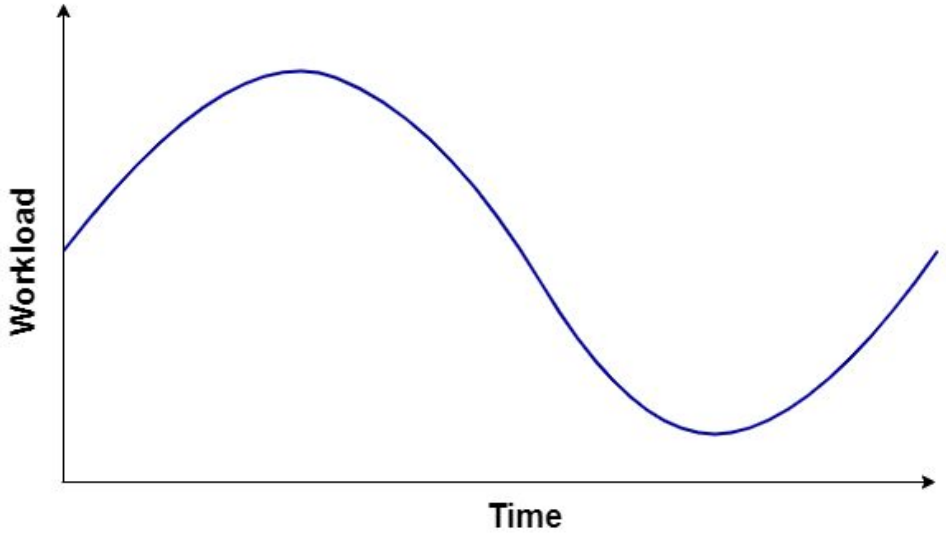
Resource allocation

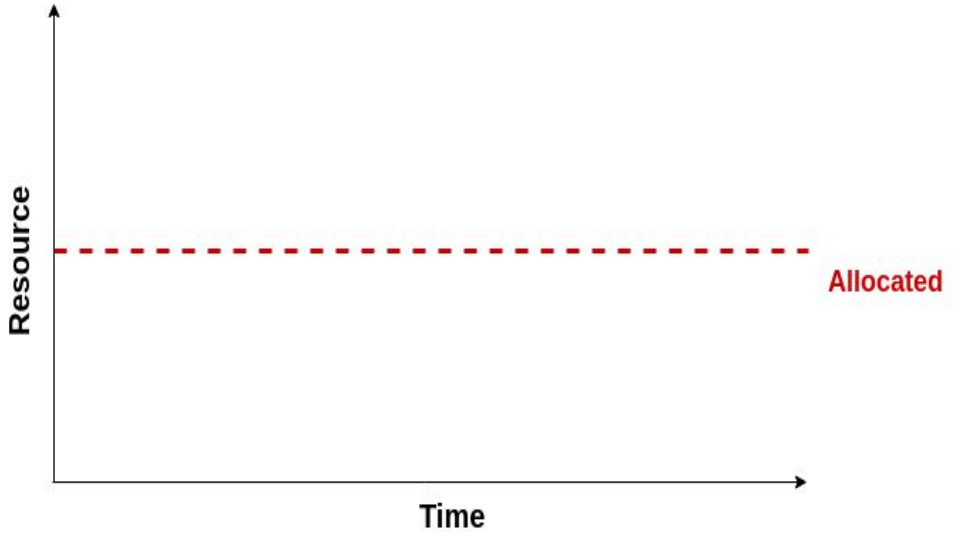


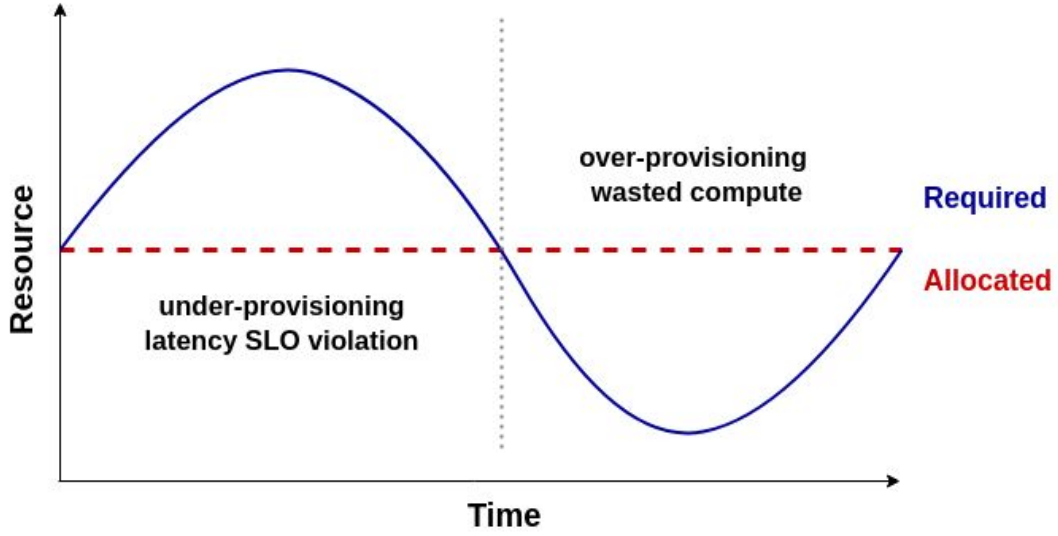




Resource allocation





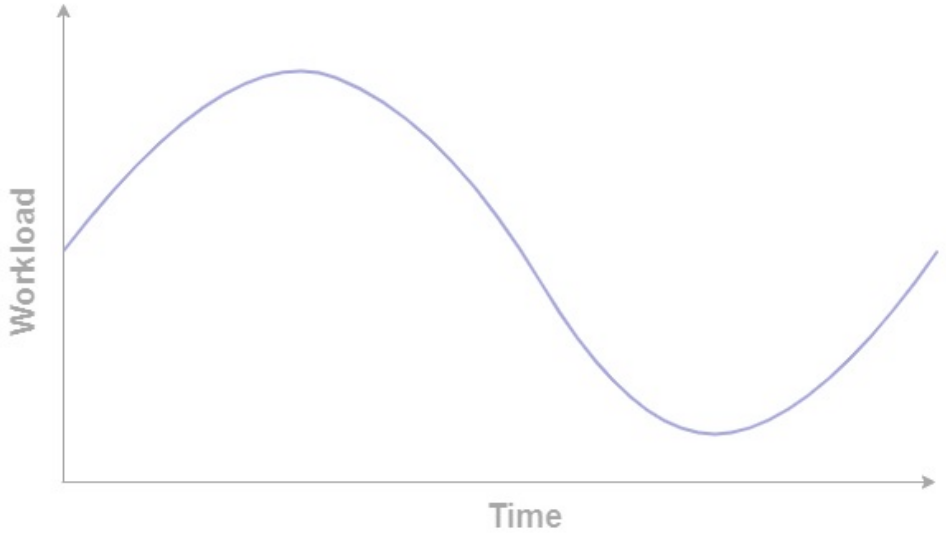


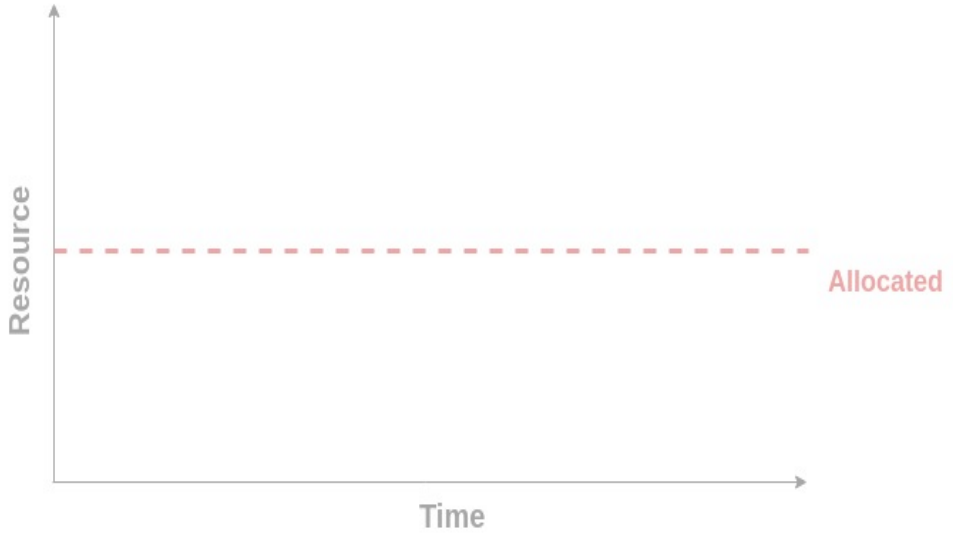
Quality adaptation

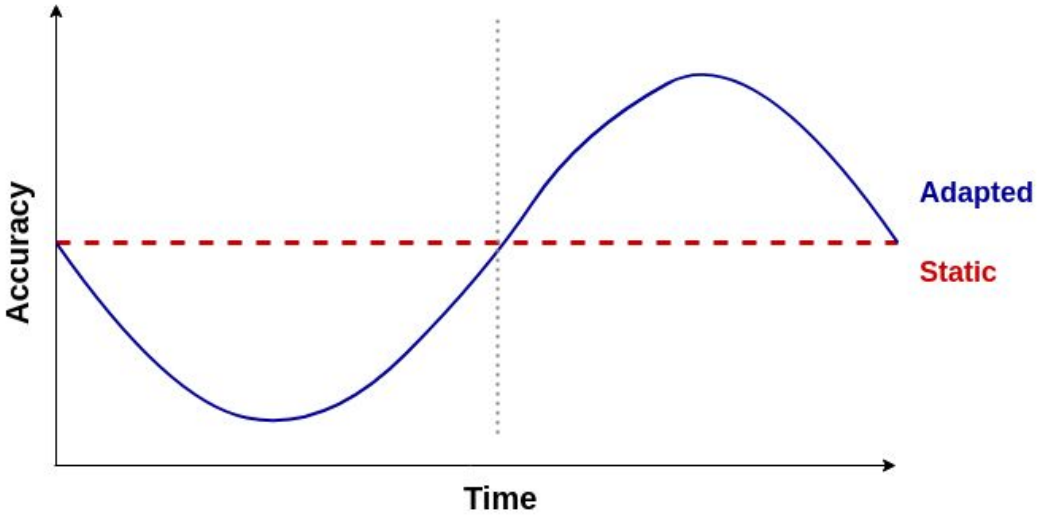
ResNet18: Tiger

ResNet152: Dog

Quality adaptation





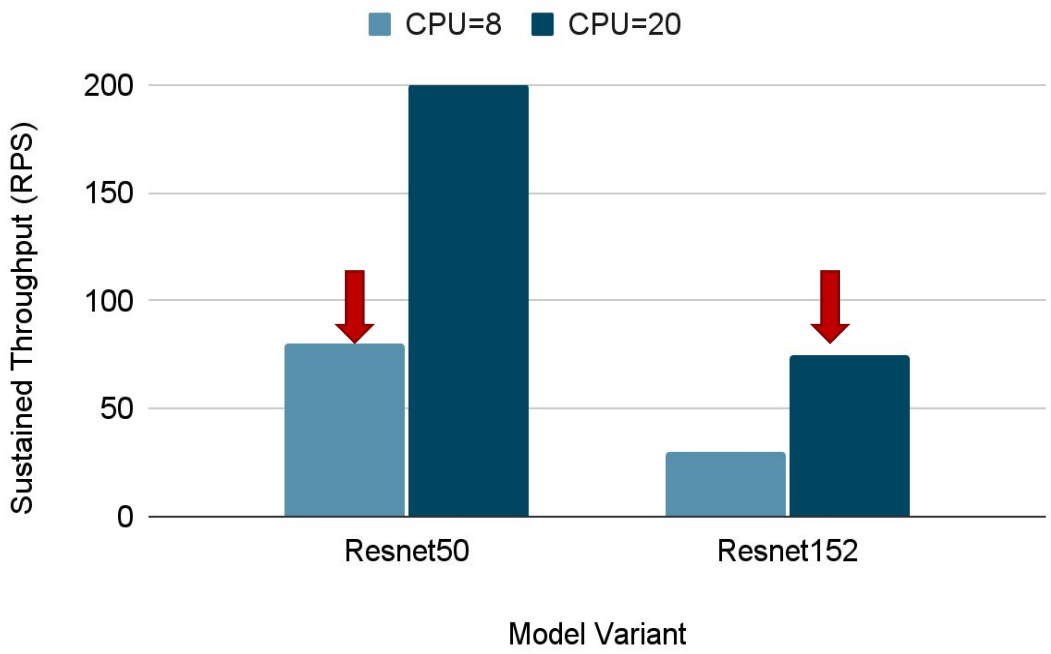


Solution: InfAdapter

InfAdapter is a latency SLO-aware, highly accurate, and cost-efficient inference serving system.

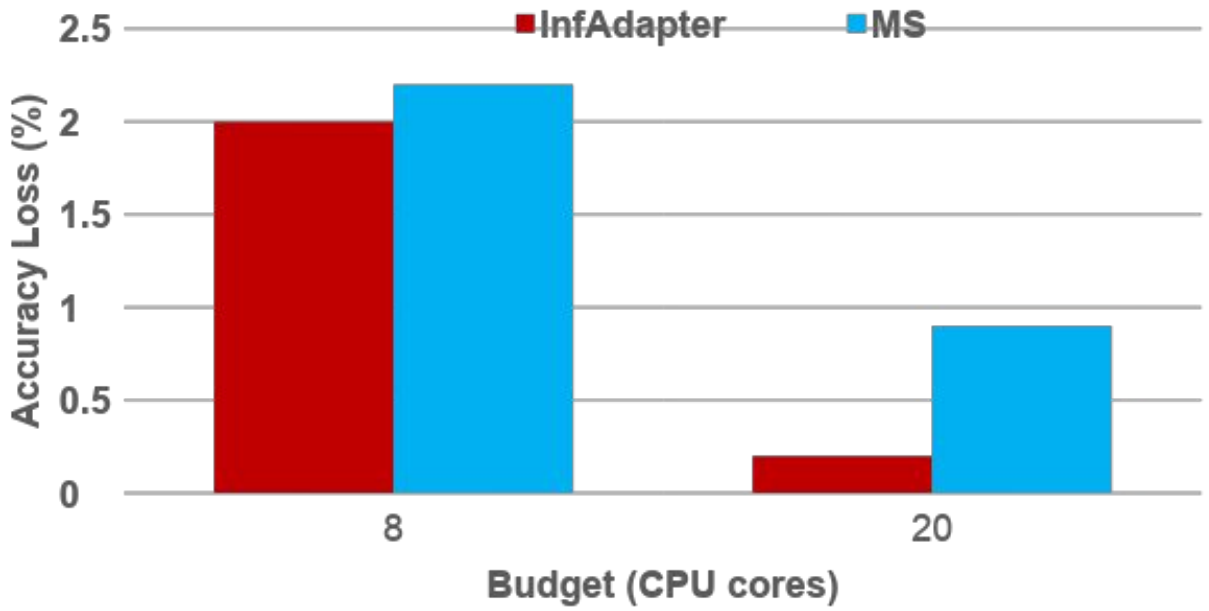
InfAdapter: Why?

Different throughputs with different model variants

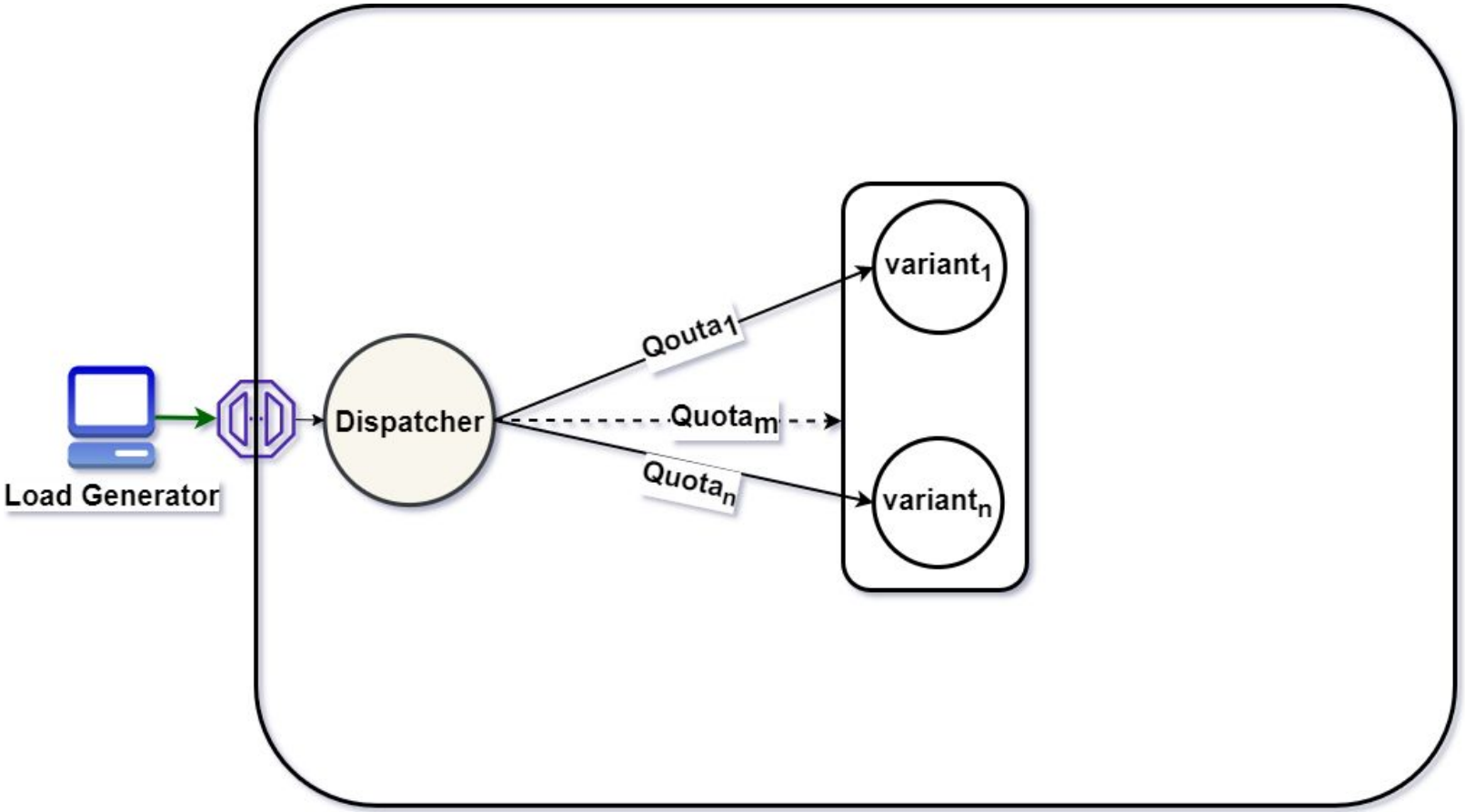


InfAdapter: Why?

Higher average accuracy by using multiple model variants



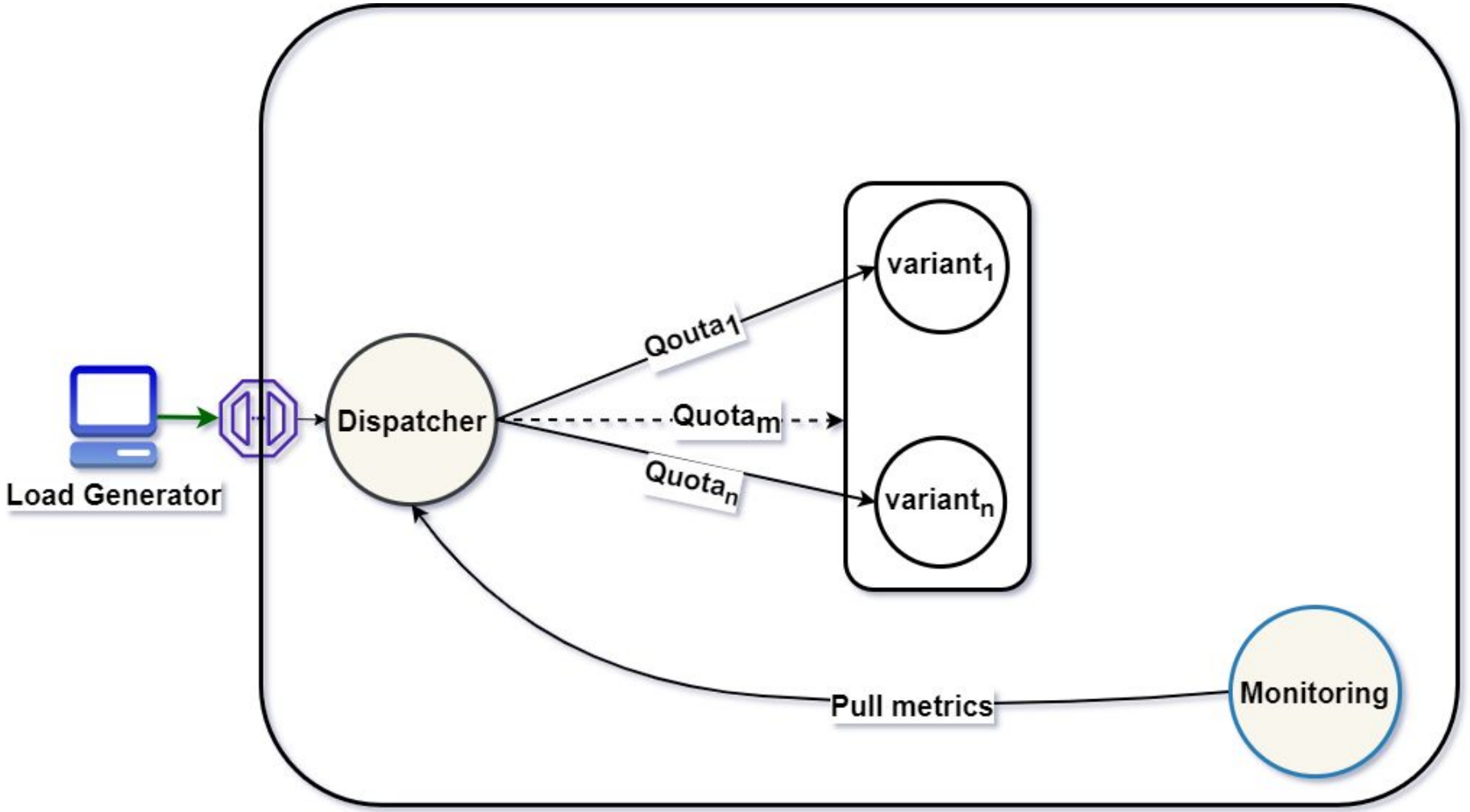
InfAdapter: How?



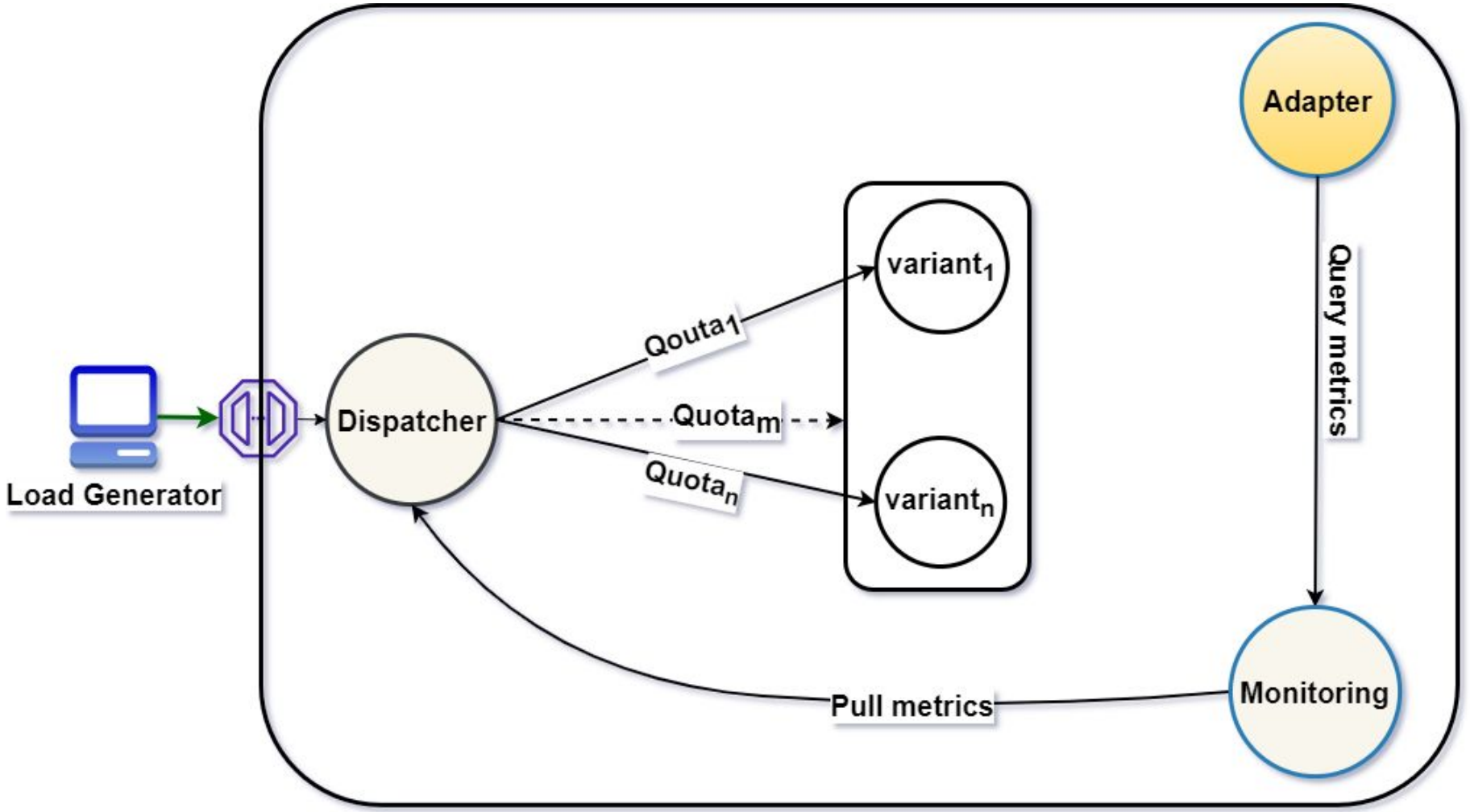
Selecting a subset of model variants, each having its own size

Meeting latency requirement for the predicted workload while maximizing accuracy and minimizing cost

InfAdapter: Design



InfAdapter: Design



InfAdapter: Formulation

$$\max \quad \alpha \cdot AA - (\beta \cdot RC + \gamma \cdot LC)$$

InfAdapter: Formulation

$$\max \quad \alpha \cdot AA - (\beta \cdot RC + \gamma \cdot LC)$$

Maximizing Average Accuracy

InfAdapter: Formulation

$$\max \quad \alpha \cdot AA - (\beta \cdot RC + \gamma \cdot LC)$$

Maximizing Average Accuracy

Minimizing Resource and Loading Costs

InfAdapter: Formulation

$$
\begin{aligned}
\max \quad & \alpha \cdot AA - (\beta \cdot RC + \gamma \cdot LC) \\
\text{subject to} \quad & \lambda \leq \sum_{m \in M} th_m(n_m), \\
& \lambda_m \leq th_m(n_m) \\
& p_m(n_m) \leq L, \forall m \in M, \\
& RC \leq B, \\
& n_m \in \mathbb{W}, \forall m \in M.
\end{aligned}
$$

InfAdapter: Formulation

$$\max \quad \alpha \cdot AA - (\beta \cdot RC + \gamma \cdot LC)$$

$$\text{subject to} \quad \lambda \leq \sum_{m \in M} th_m(n_m),$$

Supporting incoming workload

$$\lambda_m \leq th_m(n_m)$$

$$p_m(n_m) \leq L, \forall m \in M,$$

$$RC \leq B,$$

$$n_m \in \mathbb{W}, \forall m \in M.$$

InfAdapter: Formulation

$$\max \quad \alpha \cdot AA - (\beta \cdot RC + \gamma \cdot LC)$$

$$\text{subject to} \quad \lambda \leq \sum_{m \in M} th_m(n_m),$$

Supporting incoming workload

$$\lambda_m \leq th_m(n_m)$$

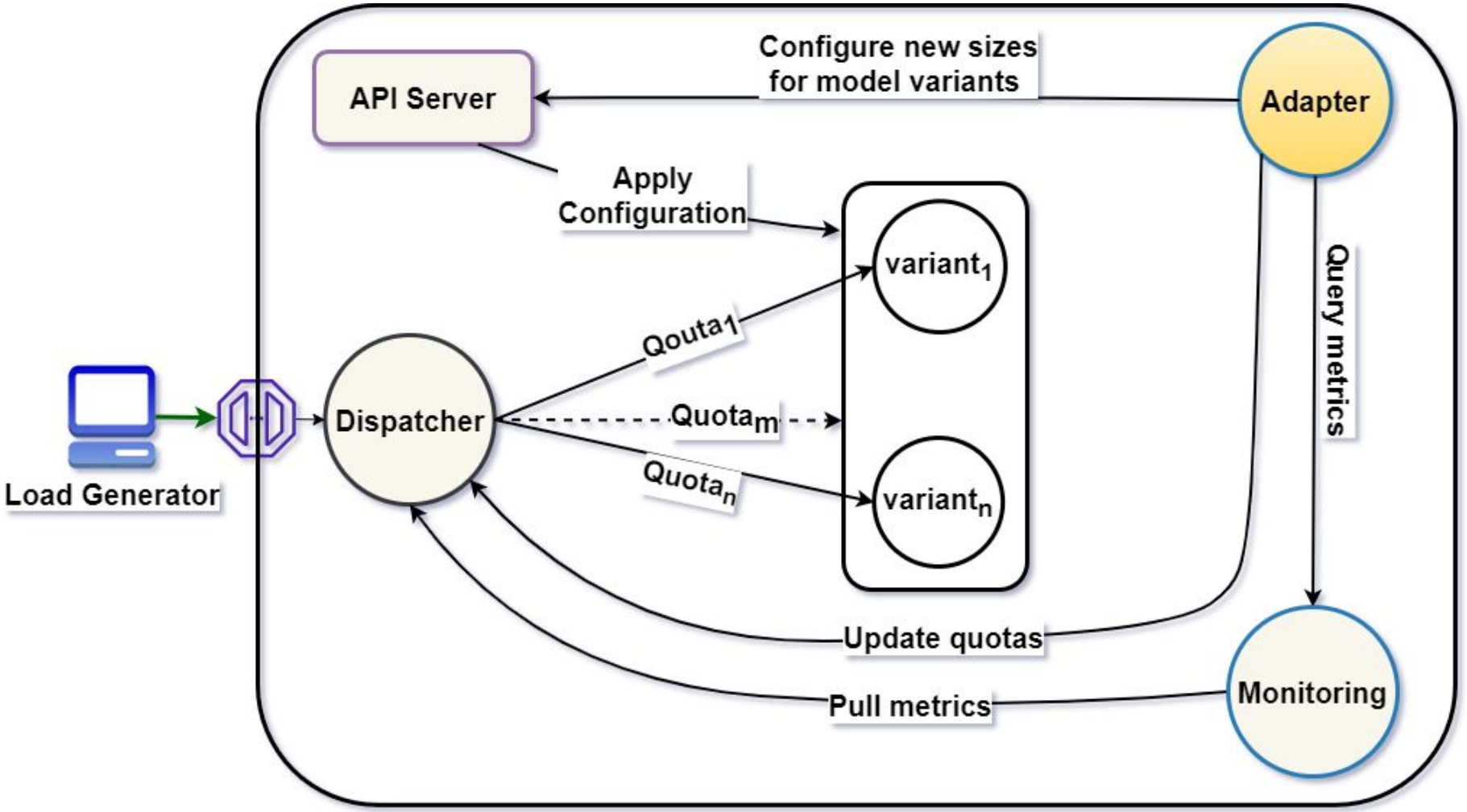
Guaranteeing end-to-end latency

$$p_m(n_m) \leq L, \forall m \in M,$$

$$RC \leq B,$$

$$n_m \in \mathbb{W}, \forall m \in M.$$

InfAdapter: Design



InfAdapter: Experimental evaluation setup

Twitter-trace sample (2022-08)

Baselines

- Kubernetes VPA and adapted Model-Switching

Used models

- Resnet18, Resnet34, Resnet50, Resnet101, Resnet152

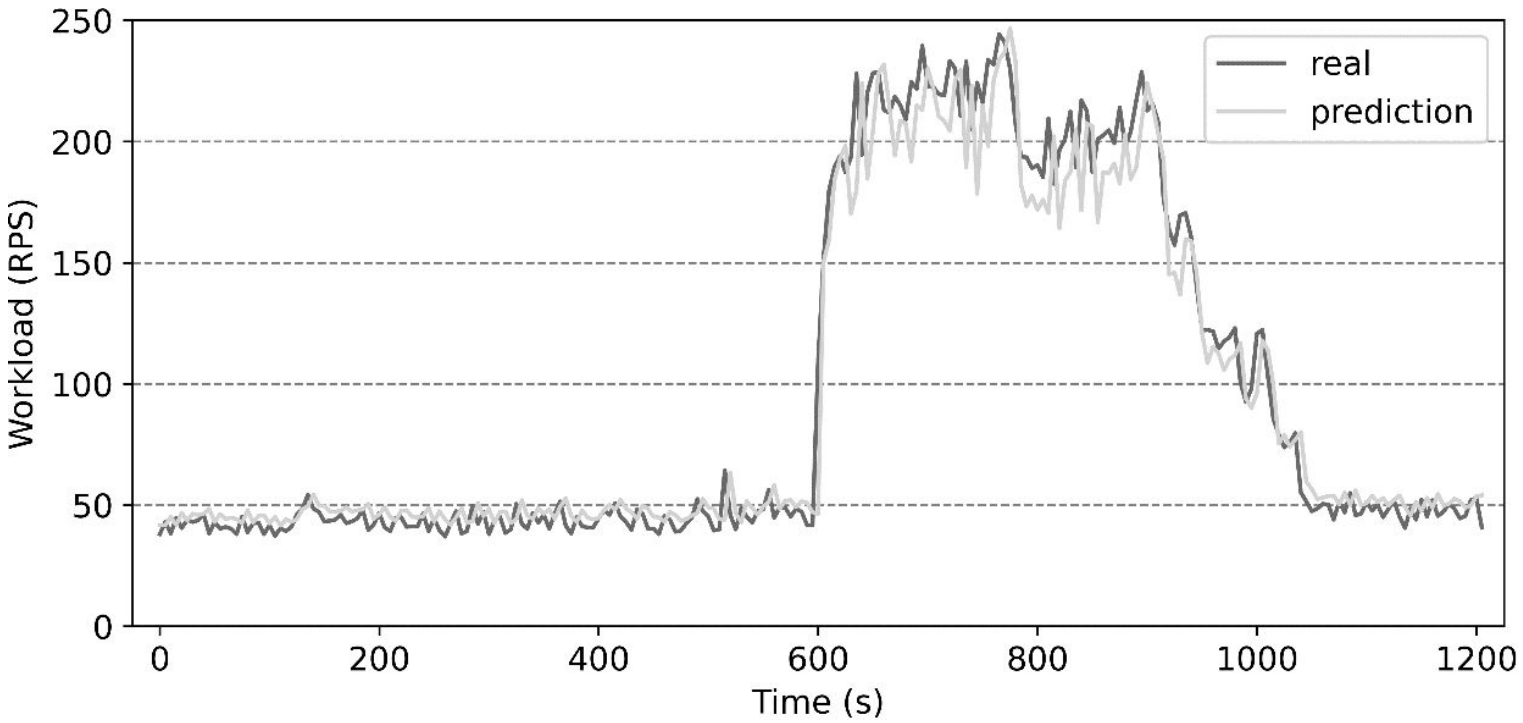
Interval adaptation

- 30 seconds

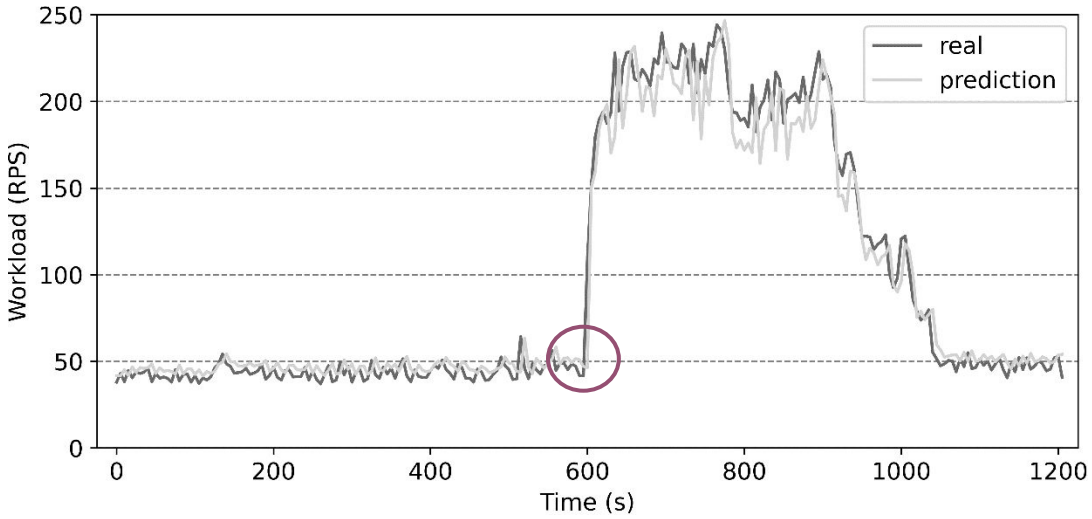
A Kubernetes cluster of 2 computing nodes

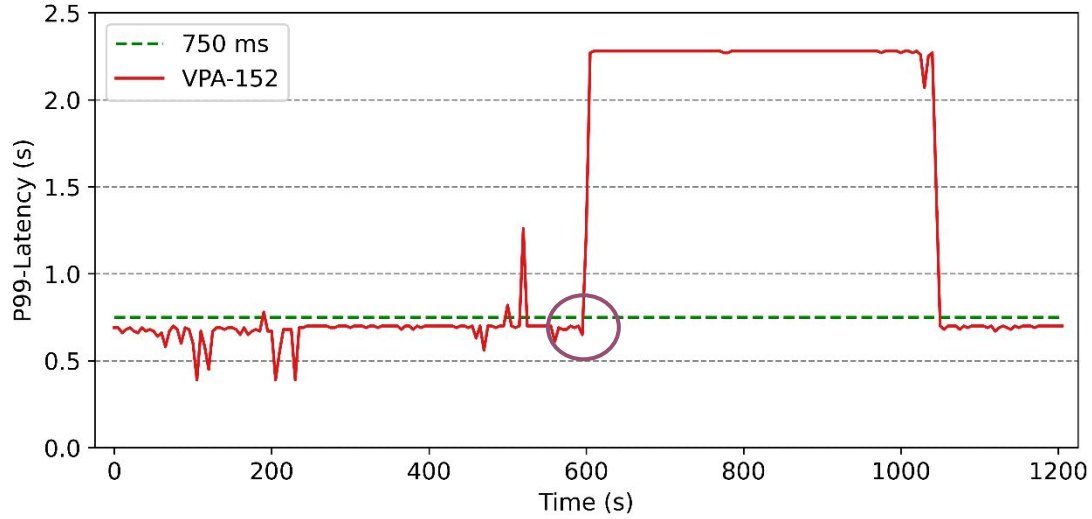
- 48 Cores, 192 GiB RAM

Workload Pattern

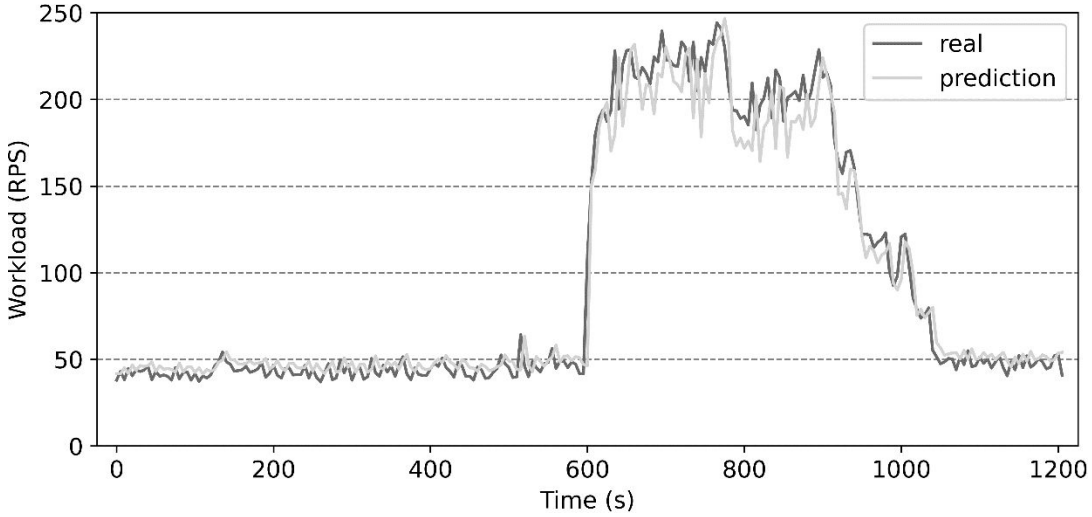


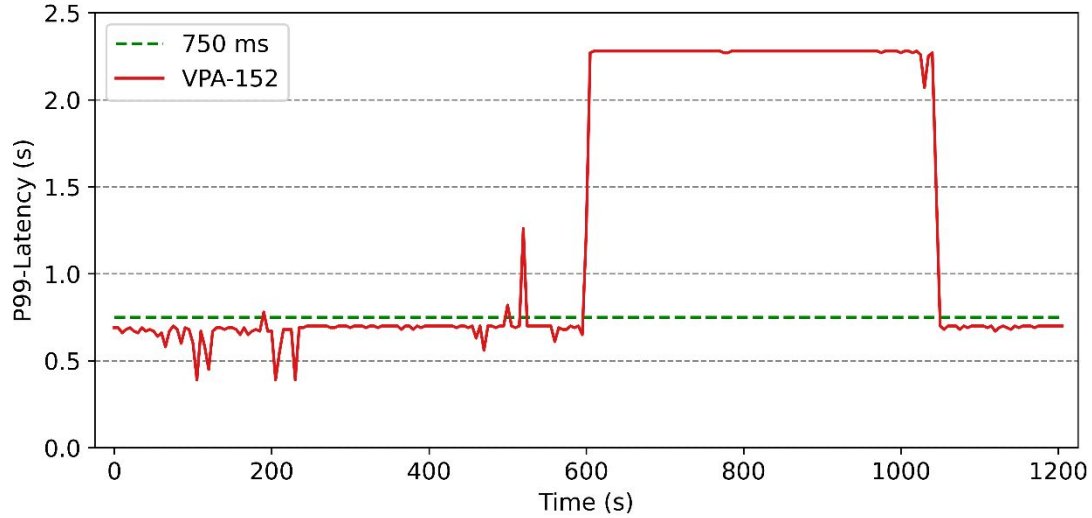
InfAdapter: P99-Latency evaluation



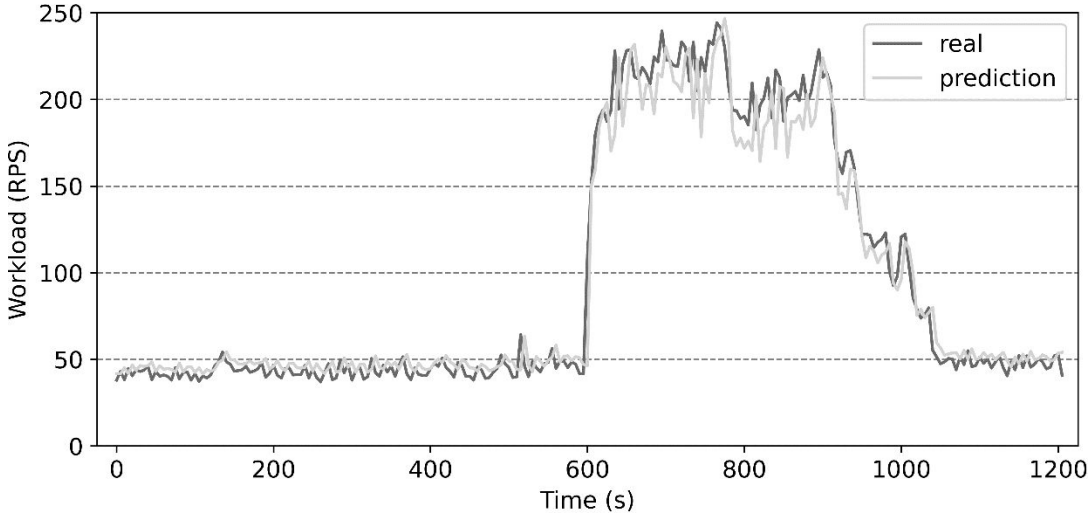


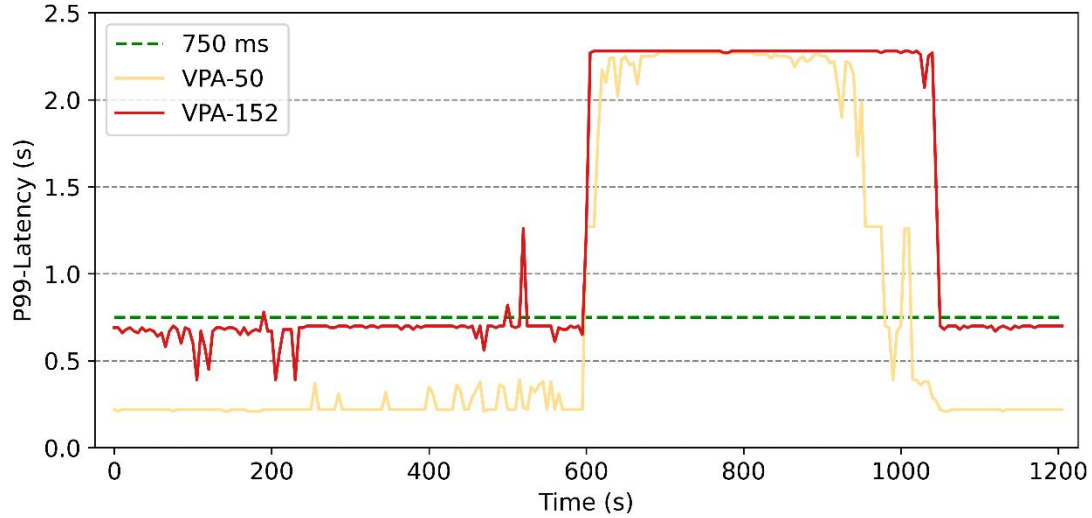
InfAdapter: P99-Latency evaluation



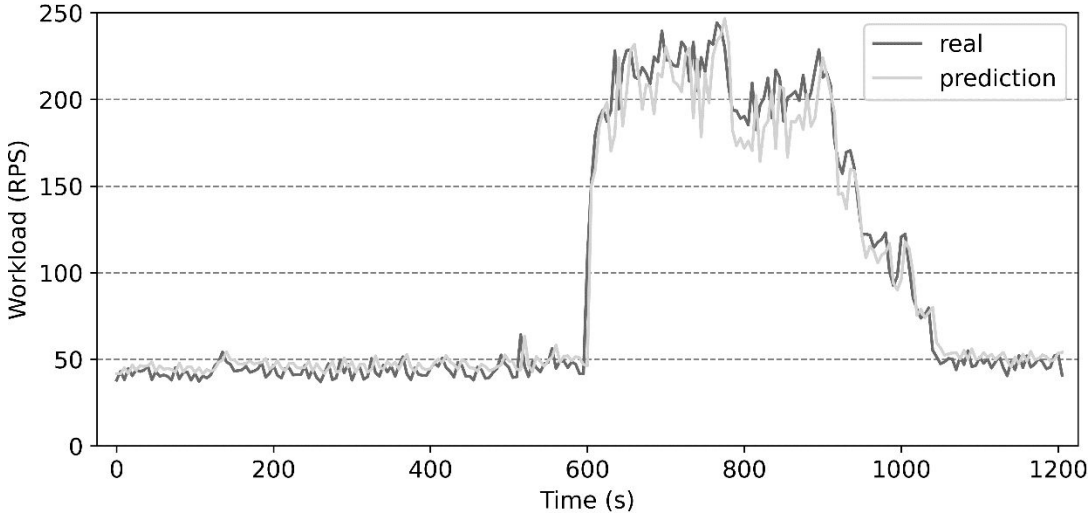


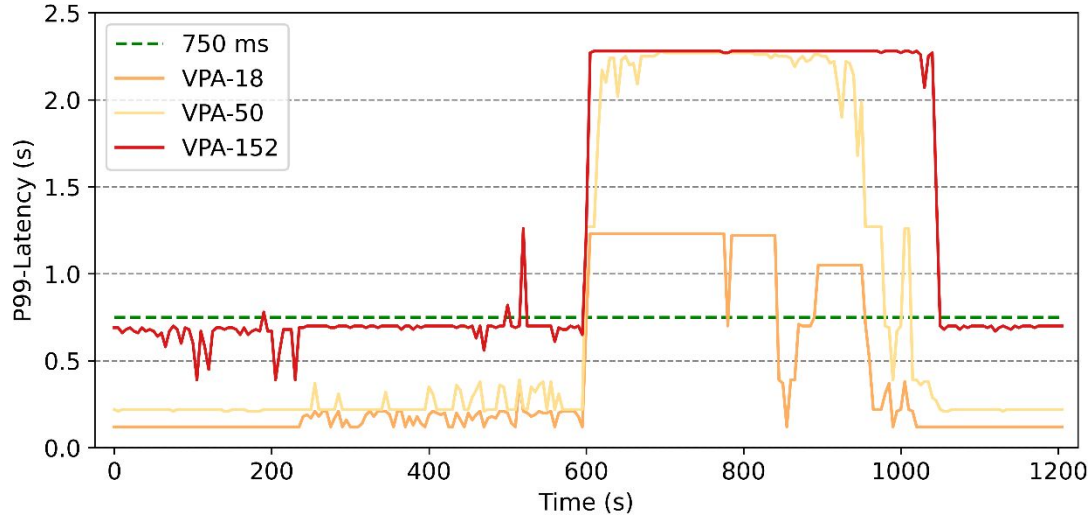
InfAdapter: P99-Latency evaluation



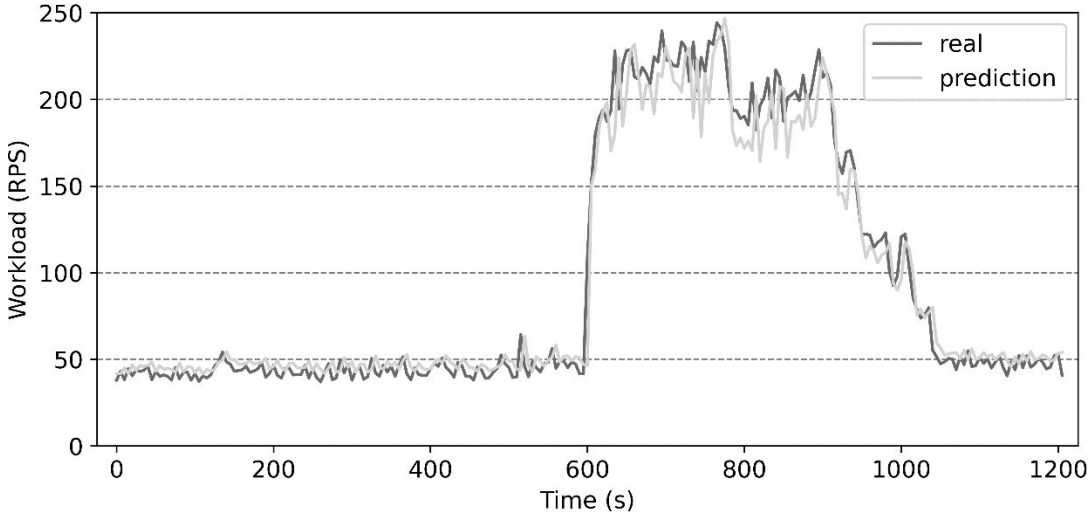


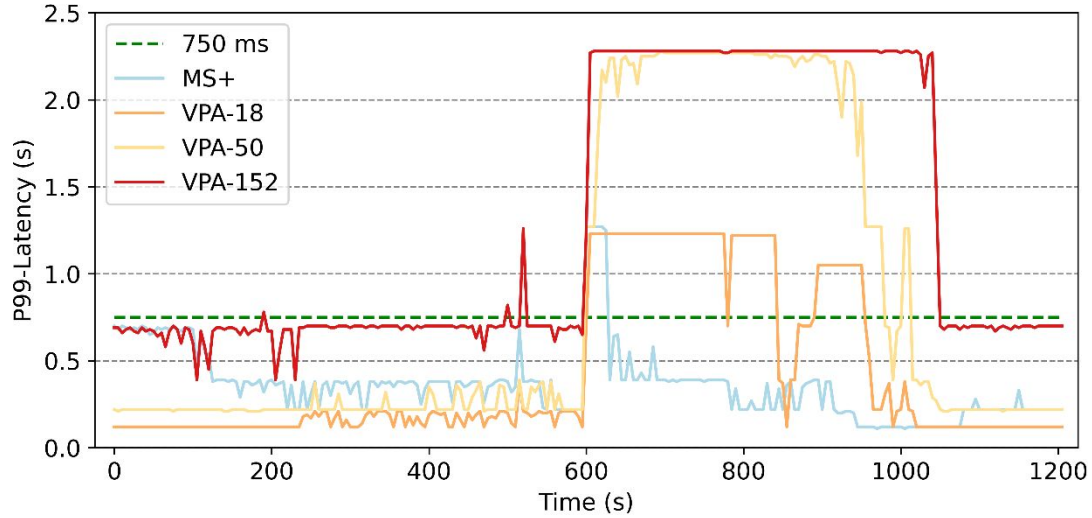
InfAdapter: P99-Latency evaluation



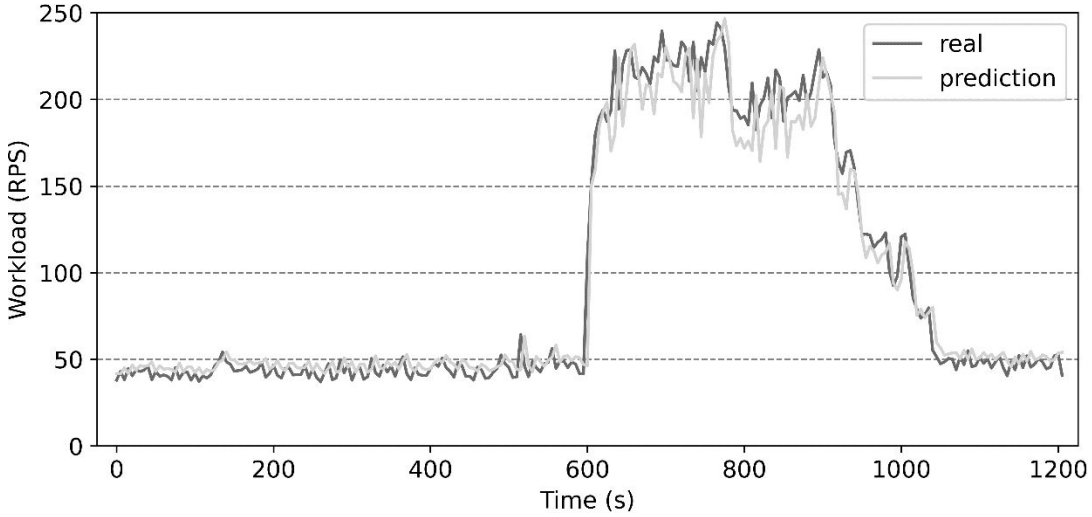


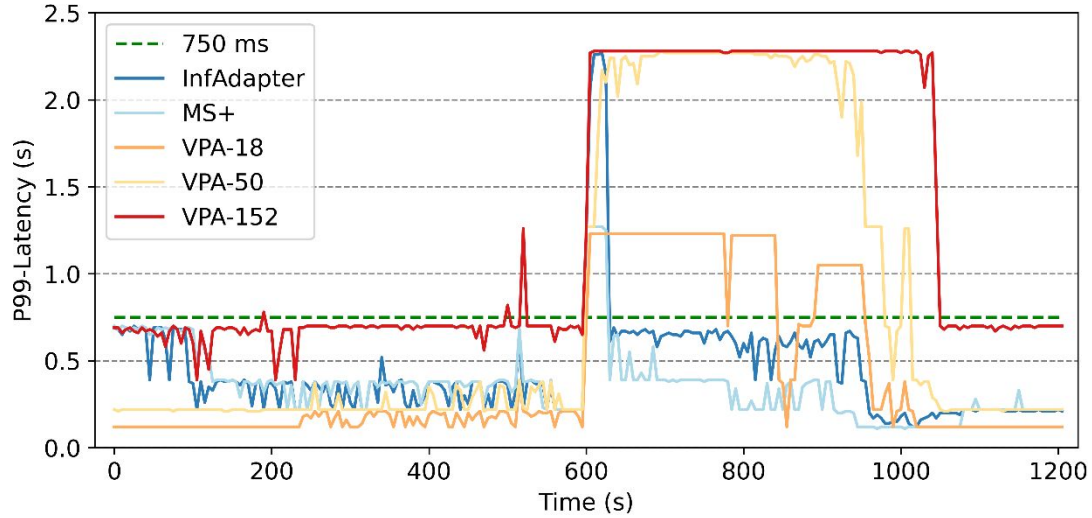
InfAdapter: P99-Latency evaluation



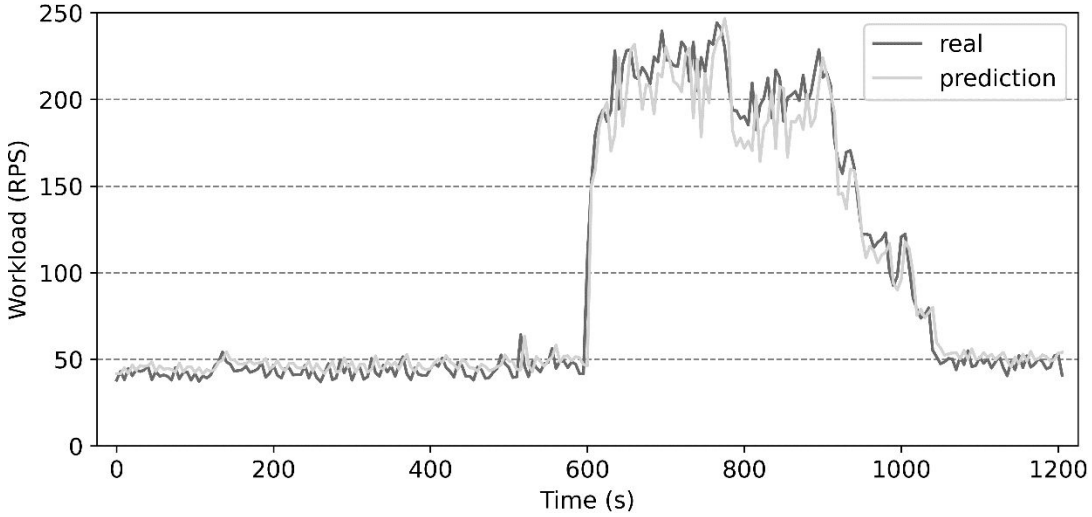


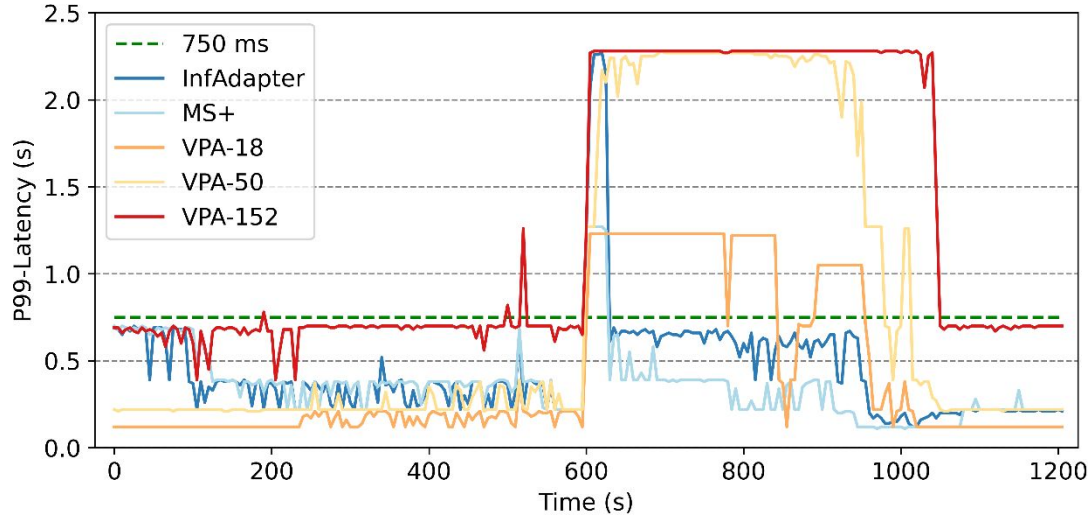
InfAdapter: P99-Latency evaluation



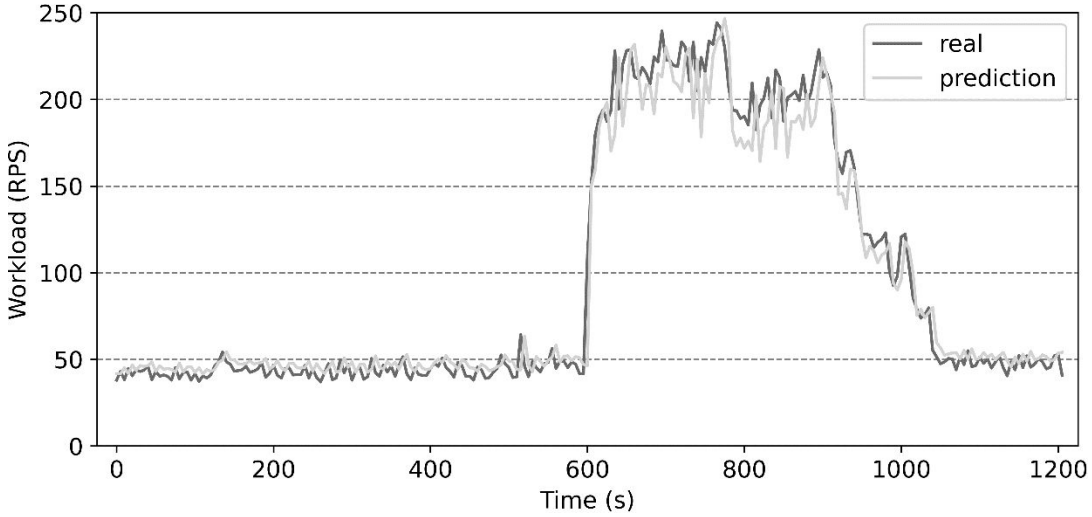


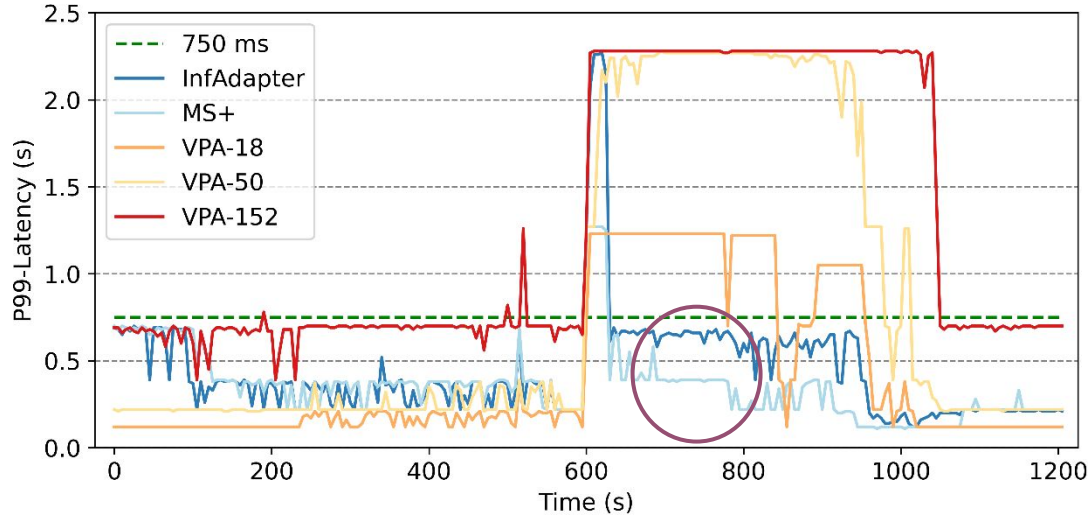
InfAdapter: P99-Latency evaluation



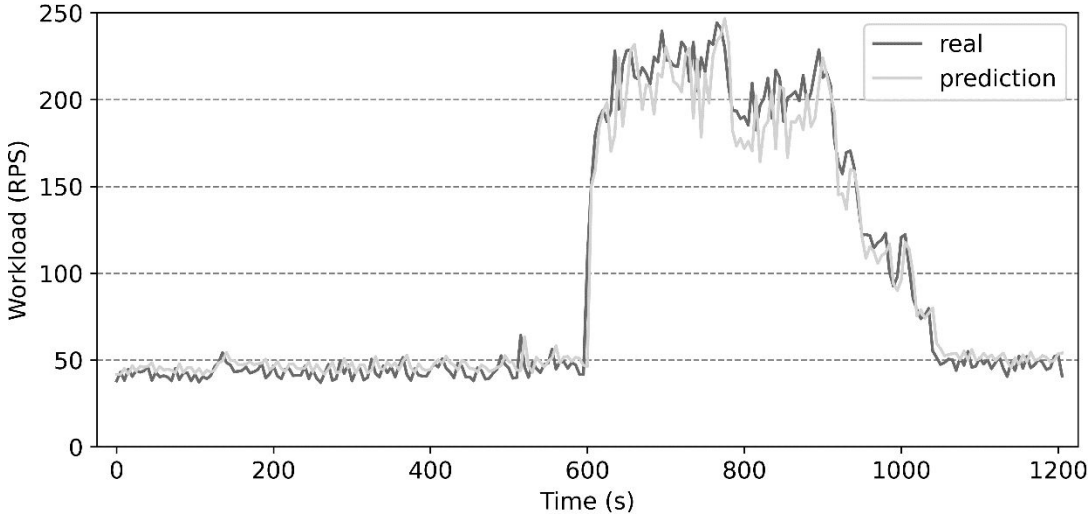


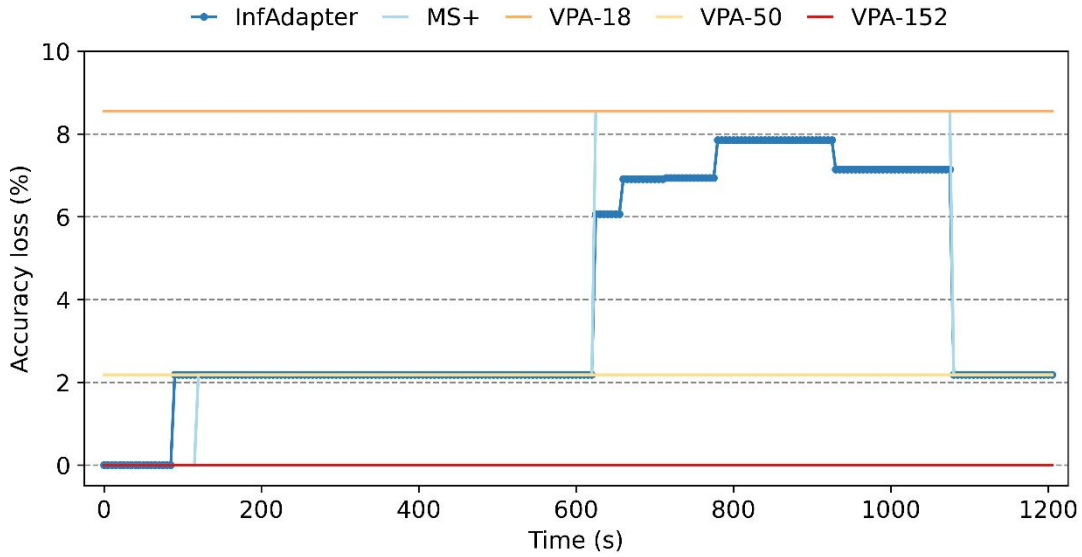
InfAdapter: P99-Latency evaluation



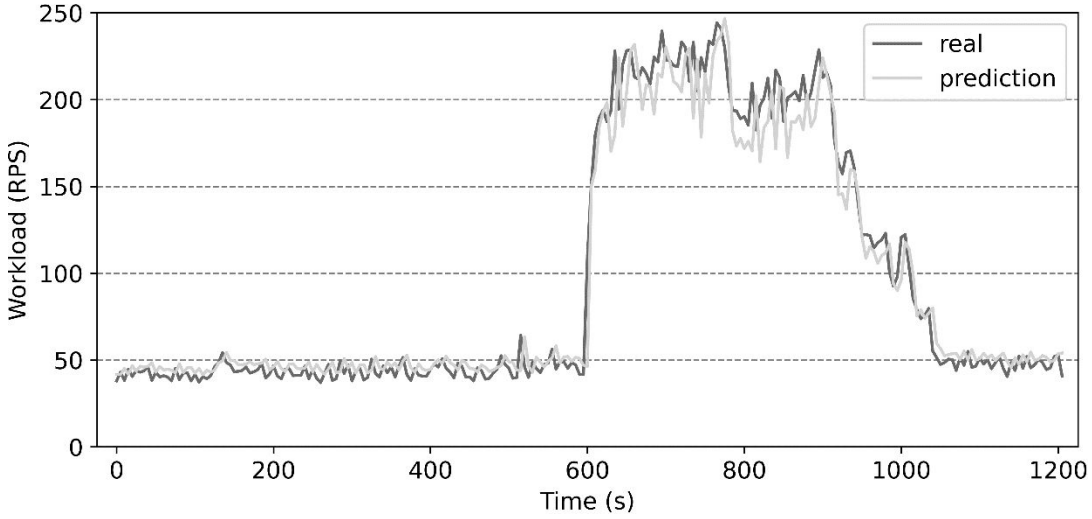


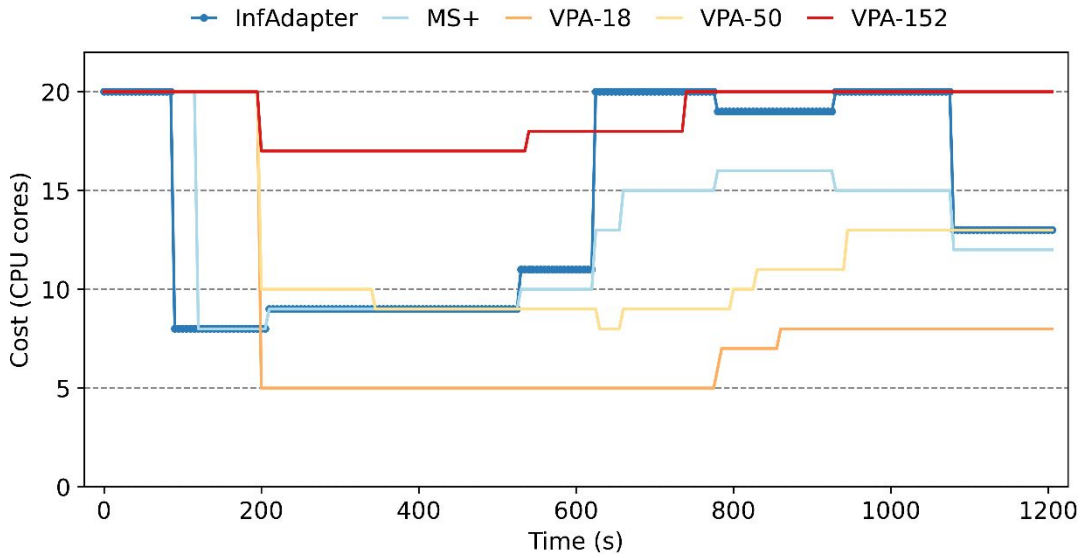
InfAdapter: Accuracy evaluation





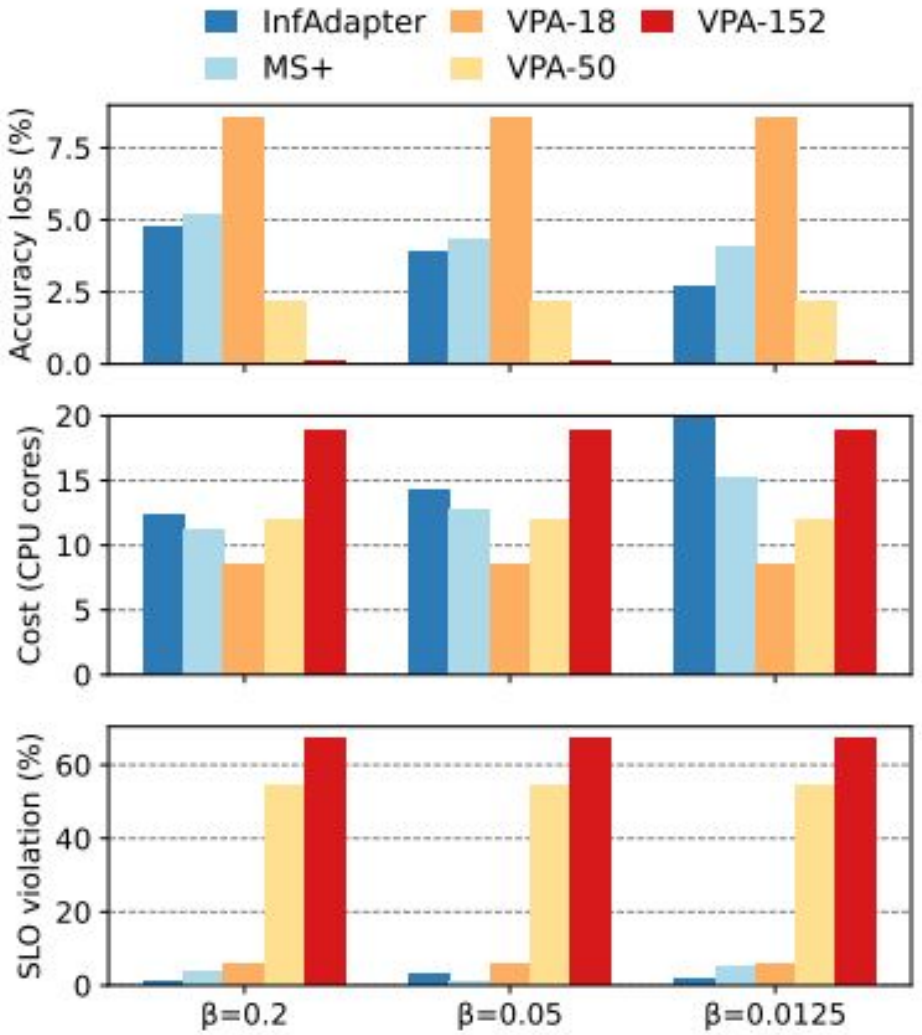
InfAdapter: Cost evaluation





InfAdapter: Experimental evaluation

Compare aggregated metrics of latency SLO violation, accuracy and cost with other works on different β values to see how they perform on different accuracy-cost trade-off



Takeaway

Inference Serving Systems should consider accuracy, latency, and cost at the same time.

Takeaway

Inference Serving Systems should consider accuracy, latency, and cost at the same time.

Model variants provide the opportunity to reduce resource costs while adapting to the dynamic workload.

Using a set of model variants simultaneously provides higher average accuracy compared to having one variant.

Takeaway

Inference Serving Systems should consider accuracy, latency, and cost at the same time.

Model variants provide the opportunity to reduce resource costs while adapting to the dynamic workload.

Using a set of model variants simultaneously provides higher average accuracy compared to having one variant.

InfAdapter!

https://github.com/reconfigurable-ml-pipeline/InfAdapter

ML inference services have strict & **conflicting** requirements

Highly Responsive! Cost-Efficient! Highly Accurate!

6

Takeaway

Inference Serving Systems should consider accuracy, latency, and cost at the same time.

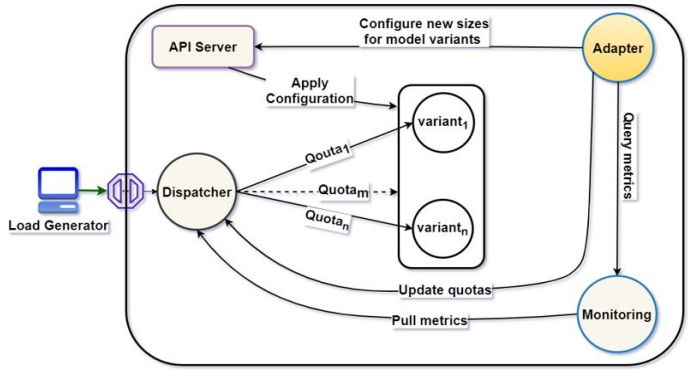
Model variants provide the opportunity to reduce resource costs while adapting to the dynamic workload.

Using a set of model variants simultaneously provides higher average accuracy compared to having one variant.

InfAdapter!

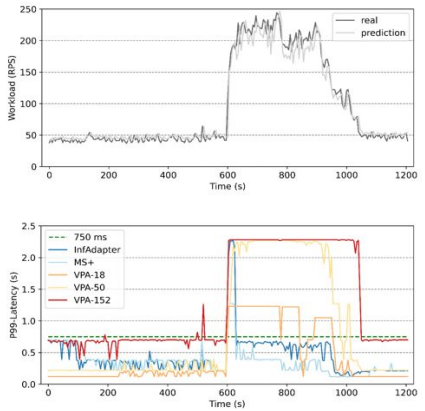
41

InfAdapter: Design



29

InfAdapter: P99-Latency evaluation



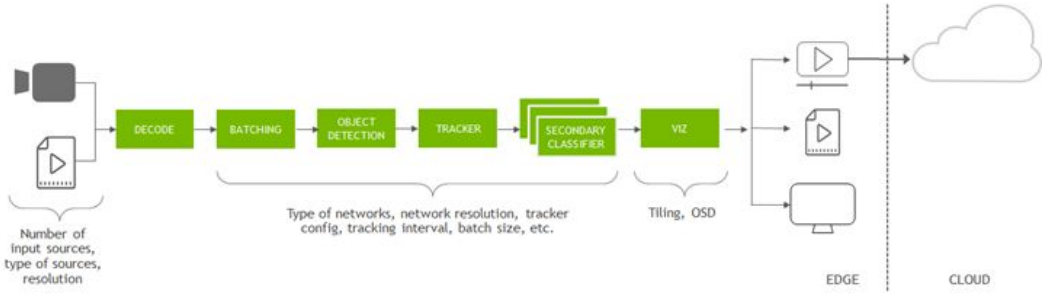
36

IPA: Inference Pipeline Adaptation to Achieve High Accuracy and Cost-Efficiency

Saeid Ghafouri, Kamran Razavi, Mehran Salmani, Alireza Sanaee

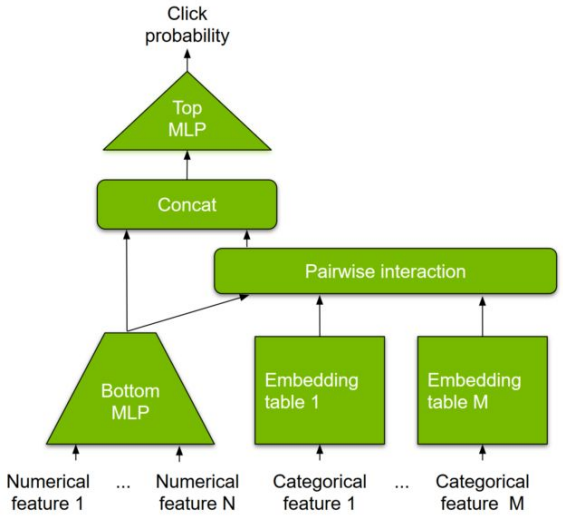
Tania Lorida-Botran, Lin Wang, Joseph Doyle, Pooyan Jamshidi

Inference Pipeline



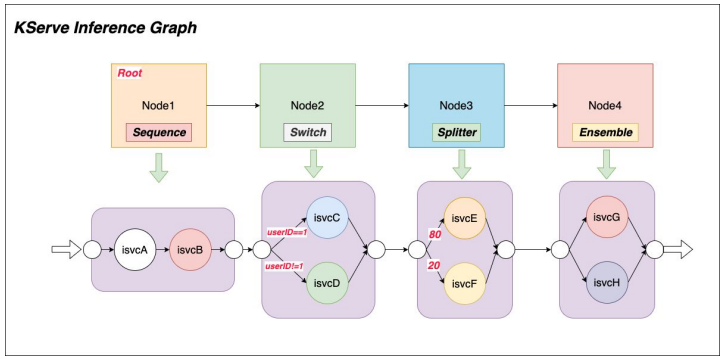
Video Pipelines

Source:
https://docs.nvidia.com/metropolis/deepstream/5.0/dev-guide/index.html#page/DeepStream_Development_Guide/deepstream_overview.html

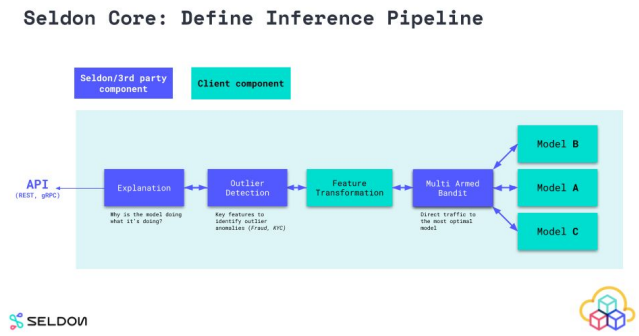


Recommender Systems

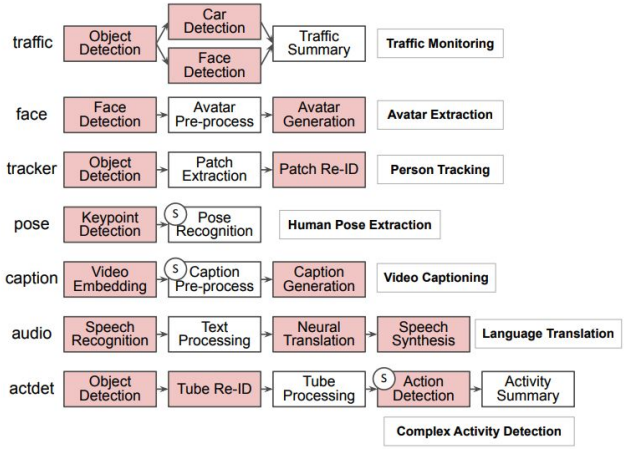
Source:
https://developer.nvidia.com/blog/optimizing-dlrm-on-nvidia-gpus/



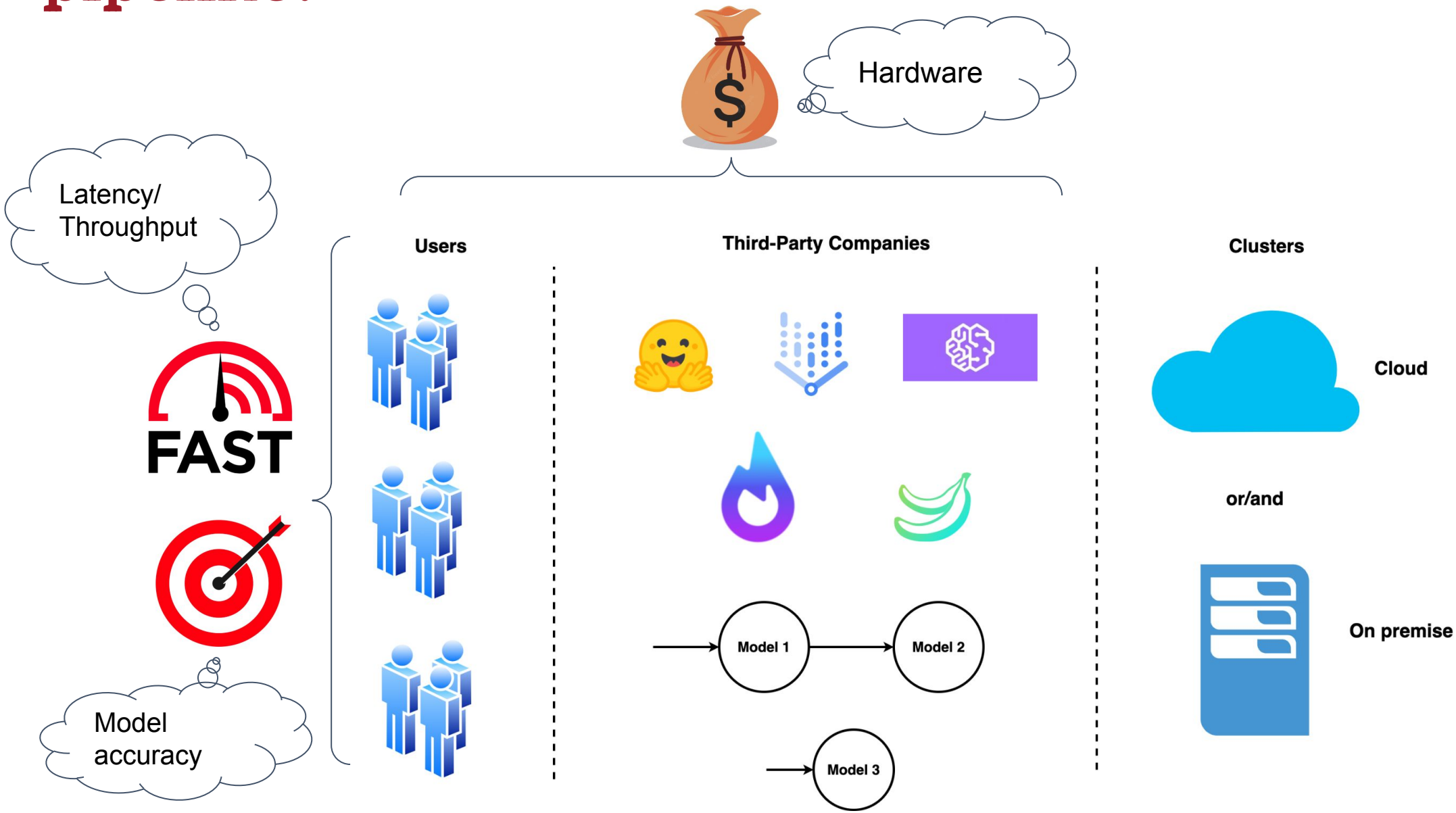
KServe



SELDON



What should be characteristic of an inference pipeline?



Autoscaling

Previous works have used auto scaling for cost optimization of inference pipeline

InferLine: Latency-Aware Provisioning and Scaling for Prediction Serving Pipelines

Daniel Crankshaw
Microsoft Research
dacranks@microsoft.com

Gur-Eyal Sela
UC Berkeley
ges@berkeley.edu

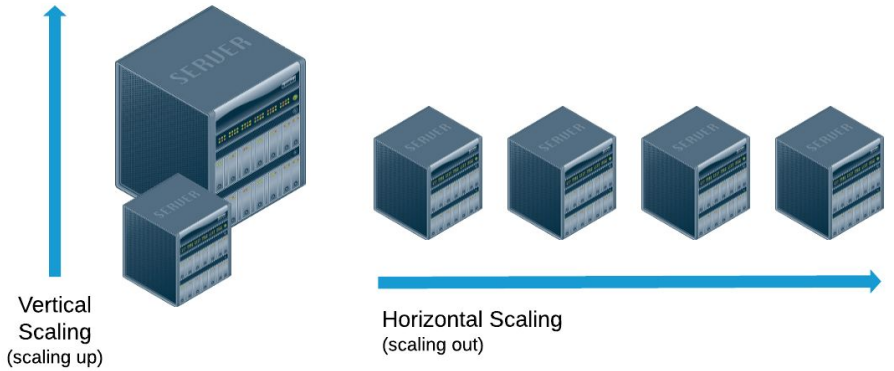
Xiangxi Mo
UC Berkeley, Anyscale
xmo@berkeley.edu

Corey Zumar
Databricks
czumar@berkeley.edu

Ion Stoica
UC Berkeley, Anyscale
istoica@berkeley.edu

Joseph Gonzalez
UC Berkeley
jegonzal@berkeley.edu

Alexey Tumanov
Georgia Tech
atumanov@gatech.edu

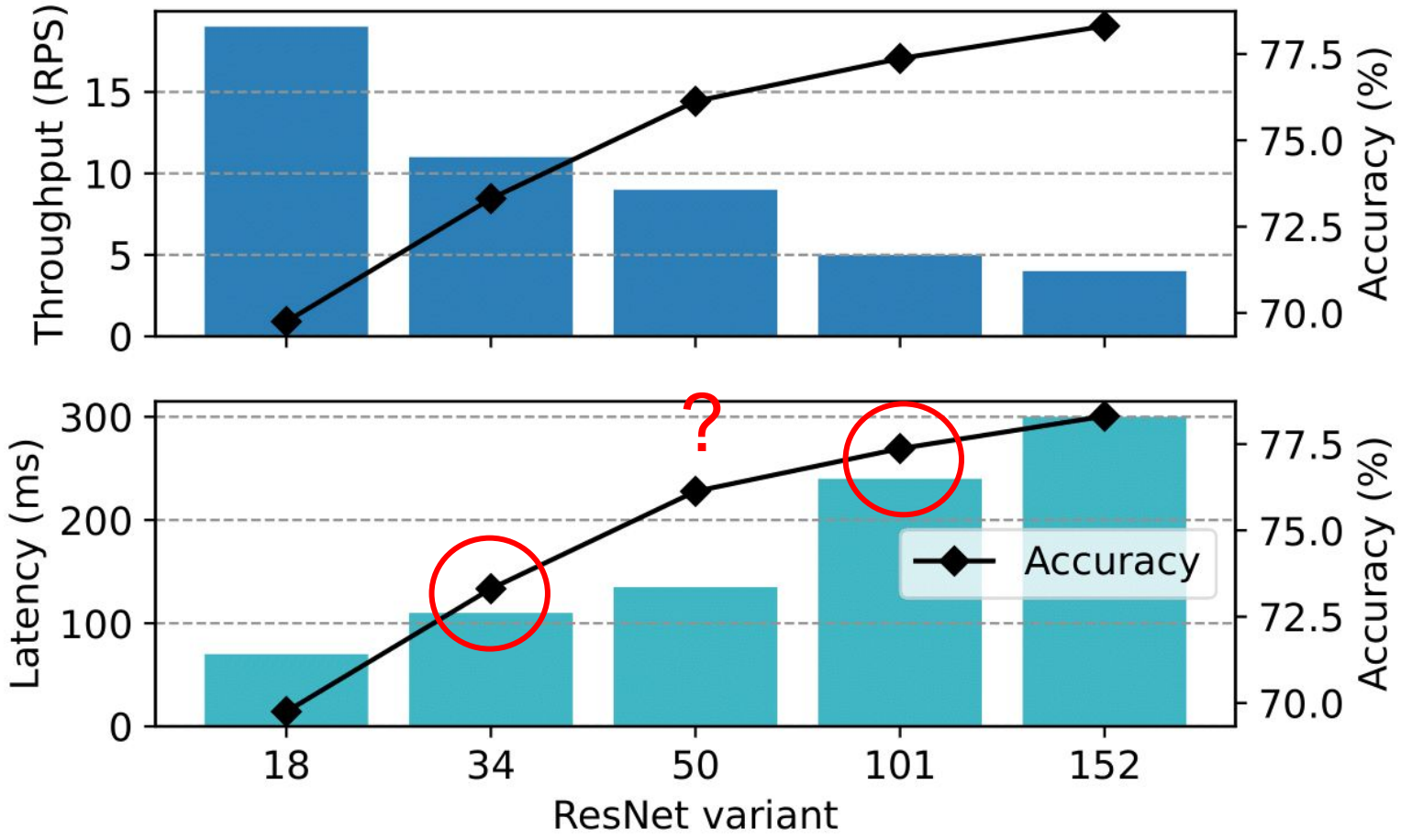


FA2: Fast, Accurate Autoscaling for Serving Deep Learning Inference with SLA Guarantees

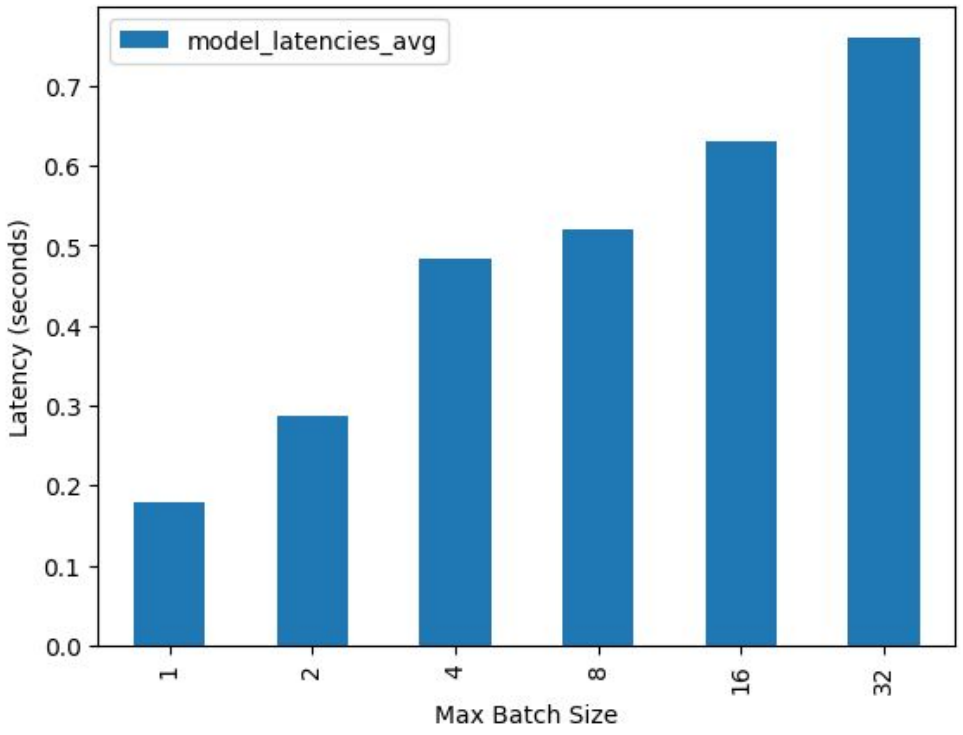
Kamran Razavi[†], Manisha Luthra[†], Boris Koldehofe[†,‡], Max Mühlhäuser[†], Lin Wang[†,§]

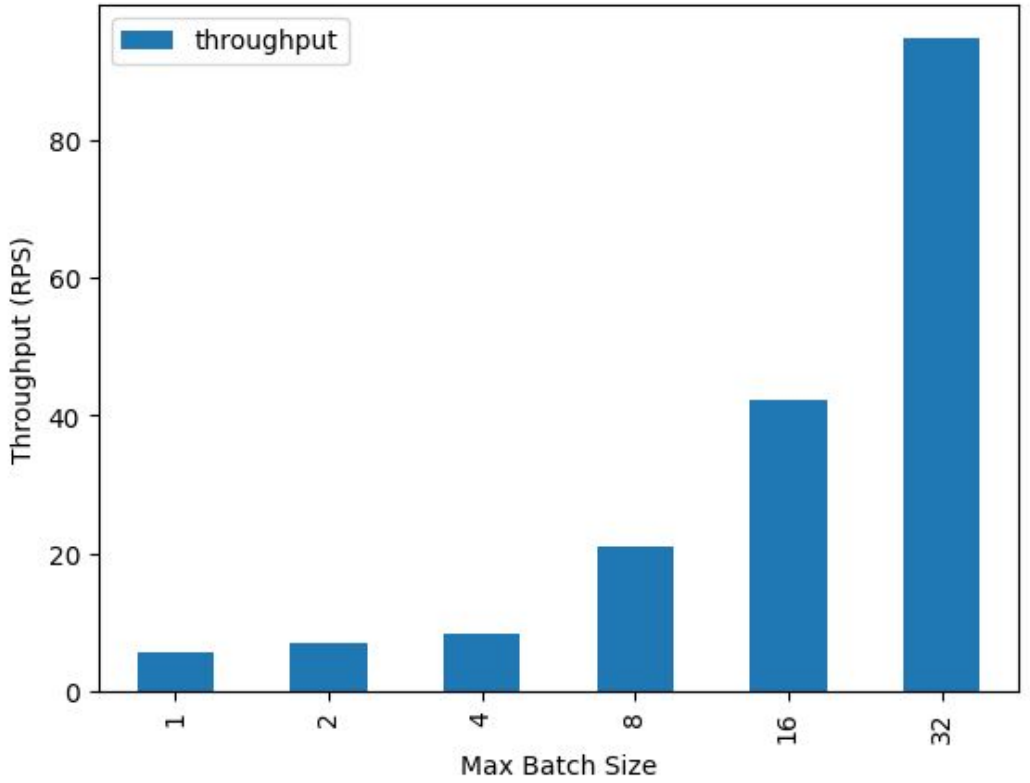
[†]Technische Universität Darmstadt
[‡]University of Groningen
[§]Vrije Universiteit Amsterdam

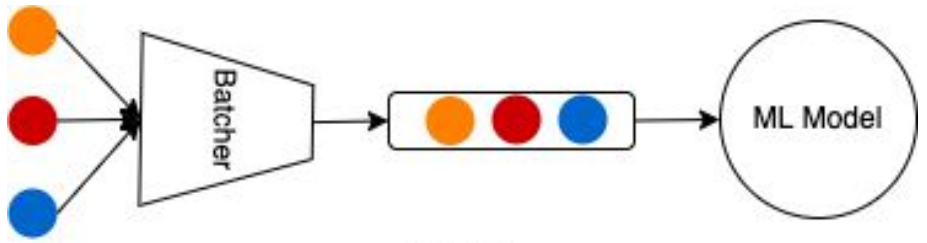
Is only scaling enough?



Effect of Batching

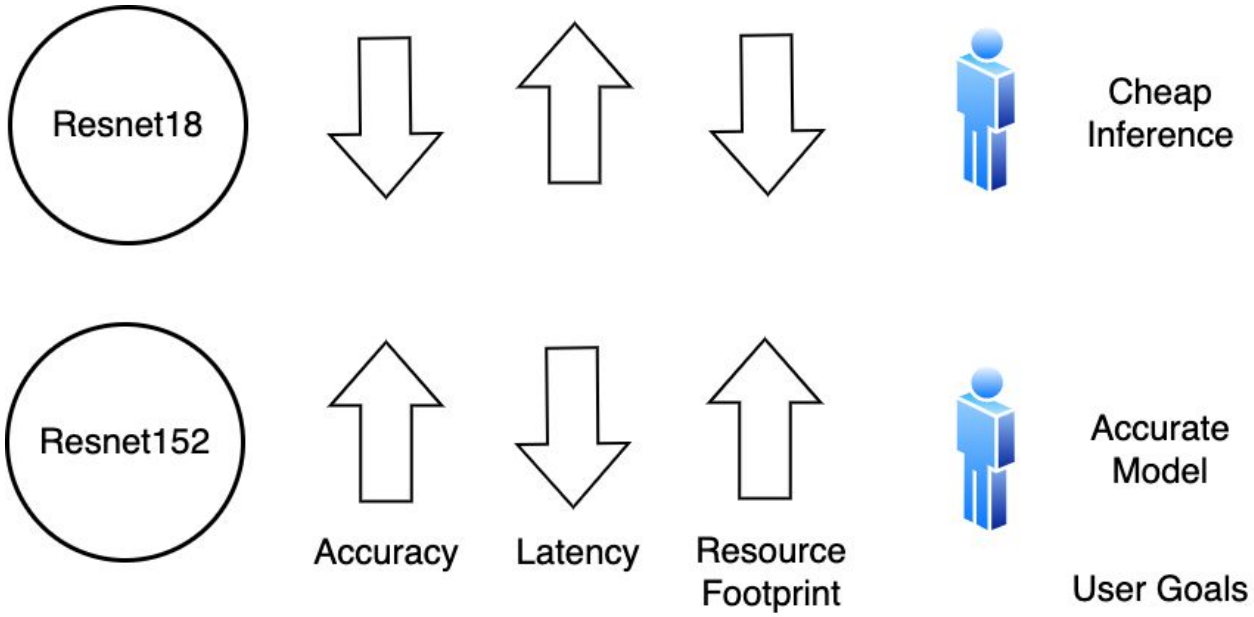




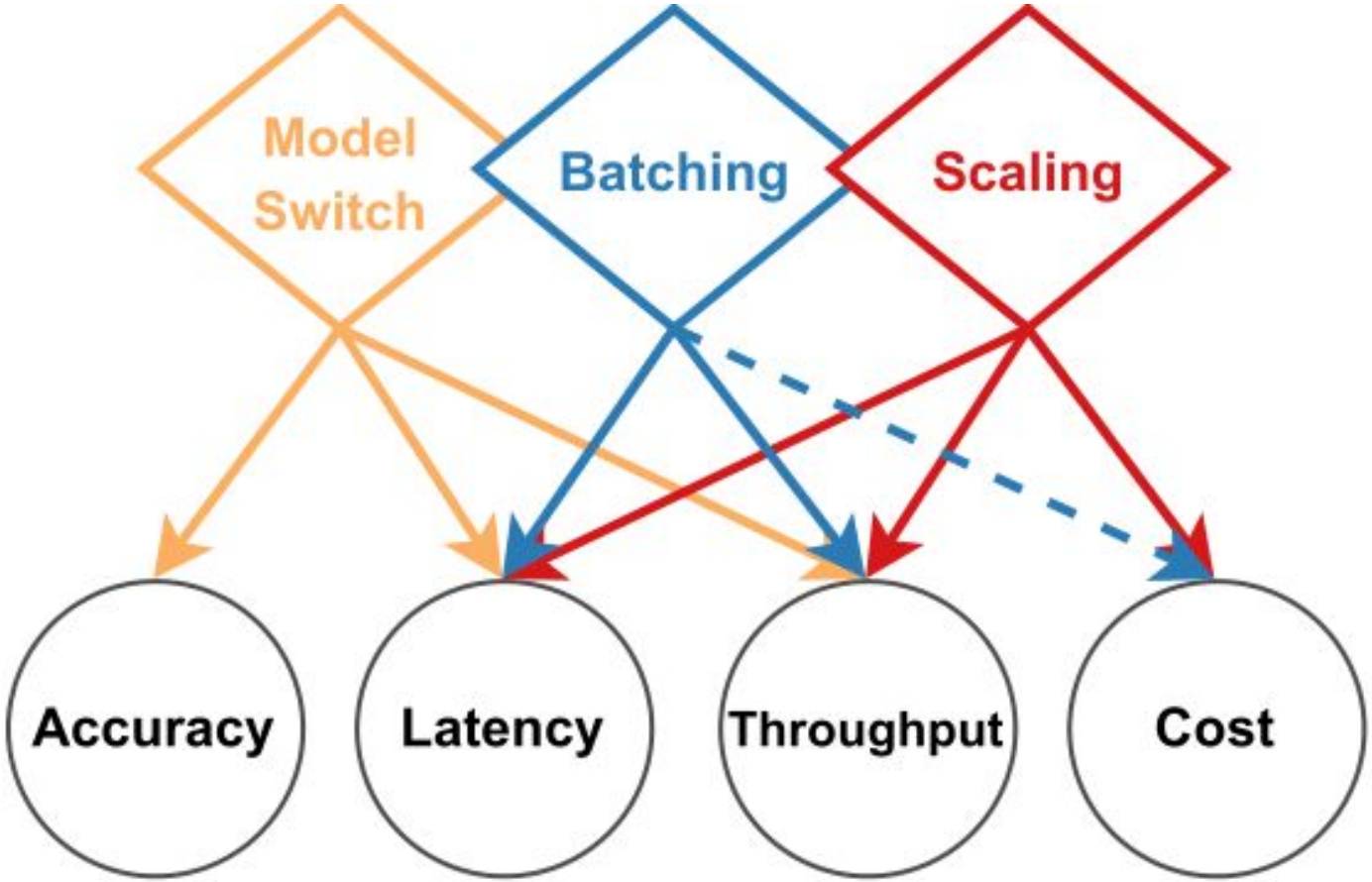


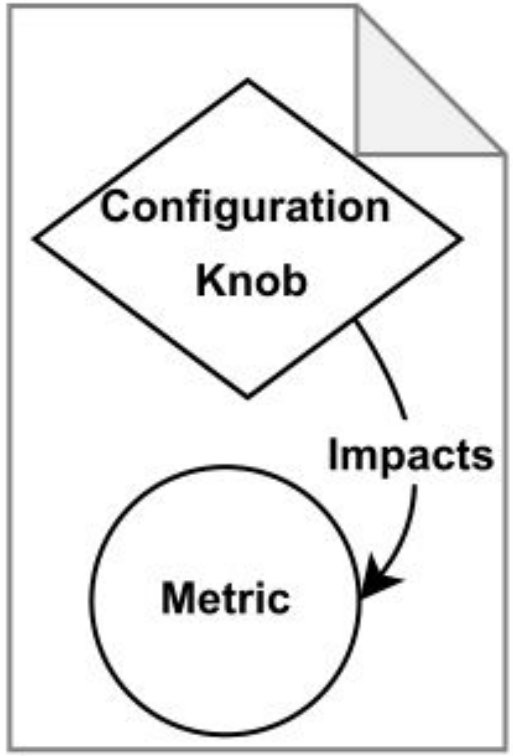
How to navigate Accuracy/latency trade off? Model Variants and Model Switching!

Previous works INFaaS and Model-Switch have proven that there is a big a latency-accuracy-resource footprint tradeoffs of models trained for the same task

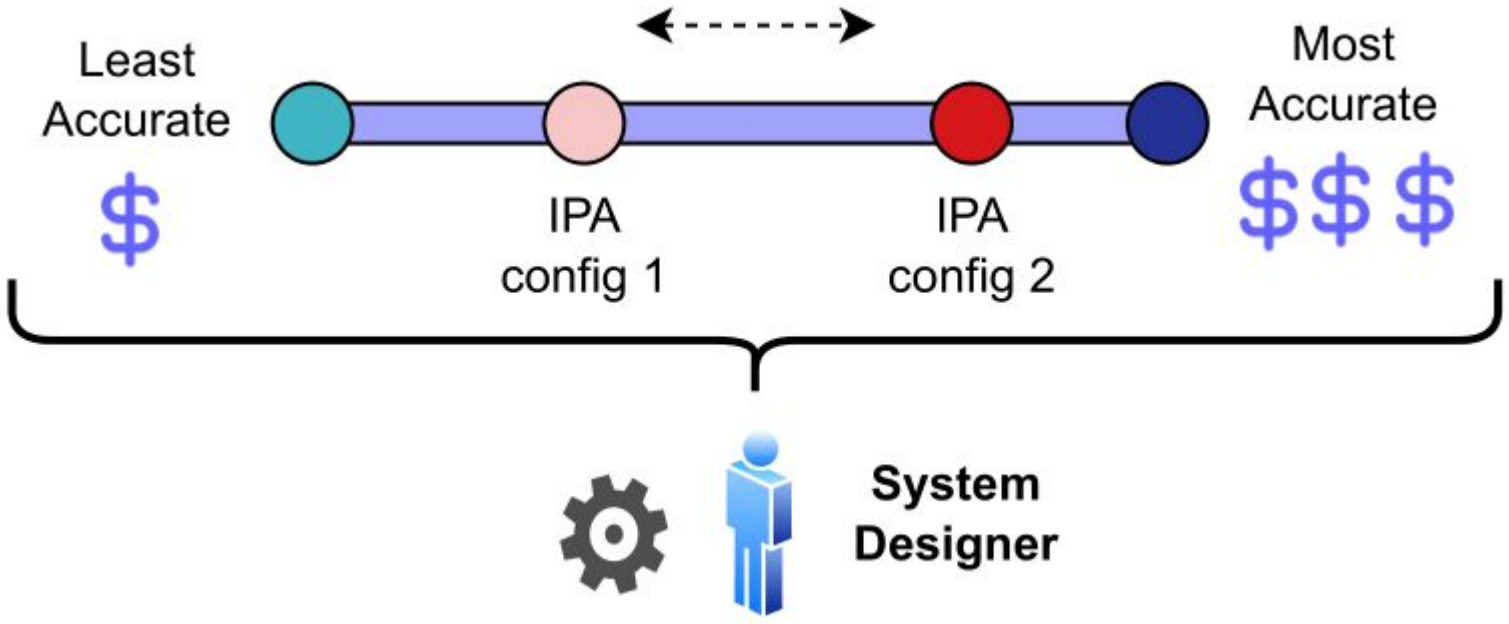


Search Space

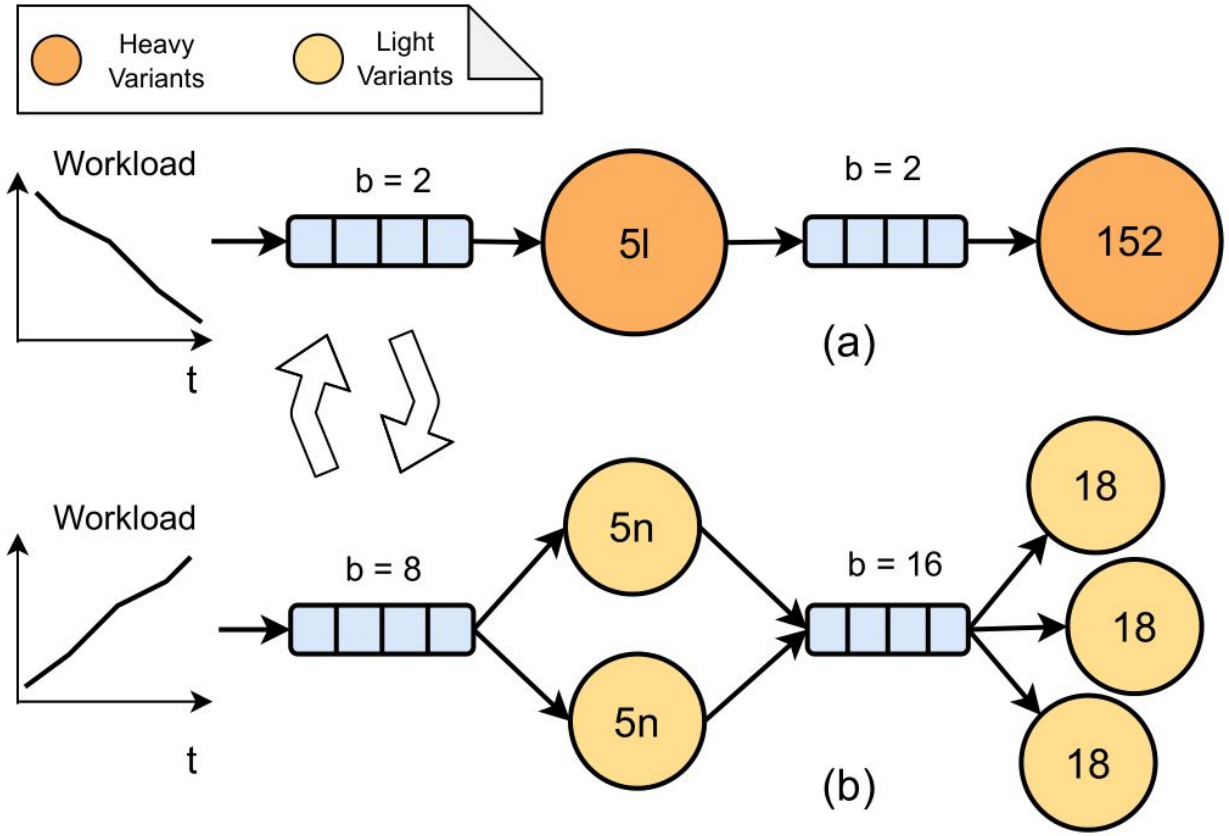




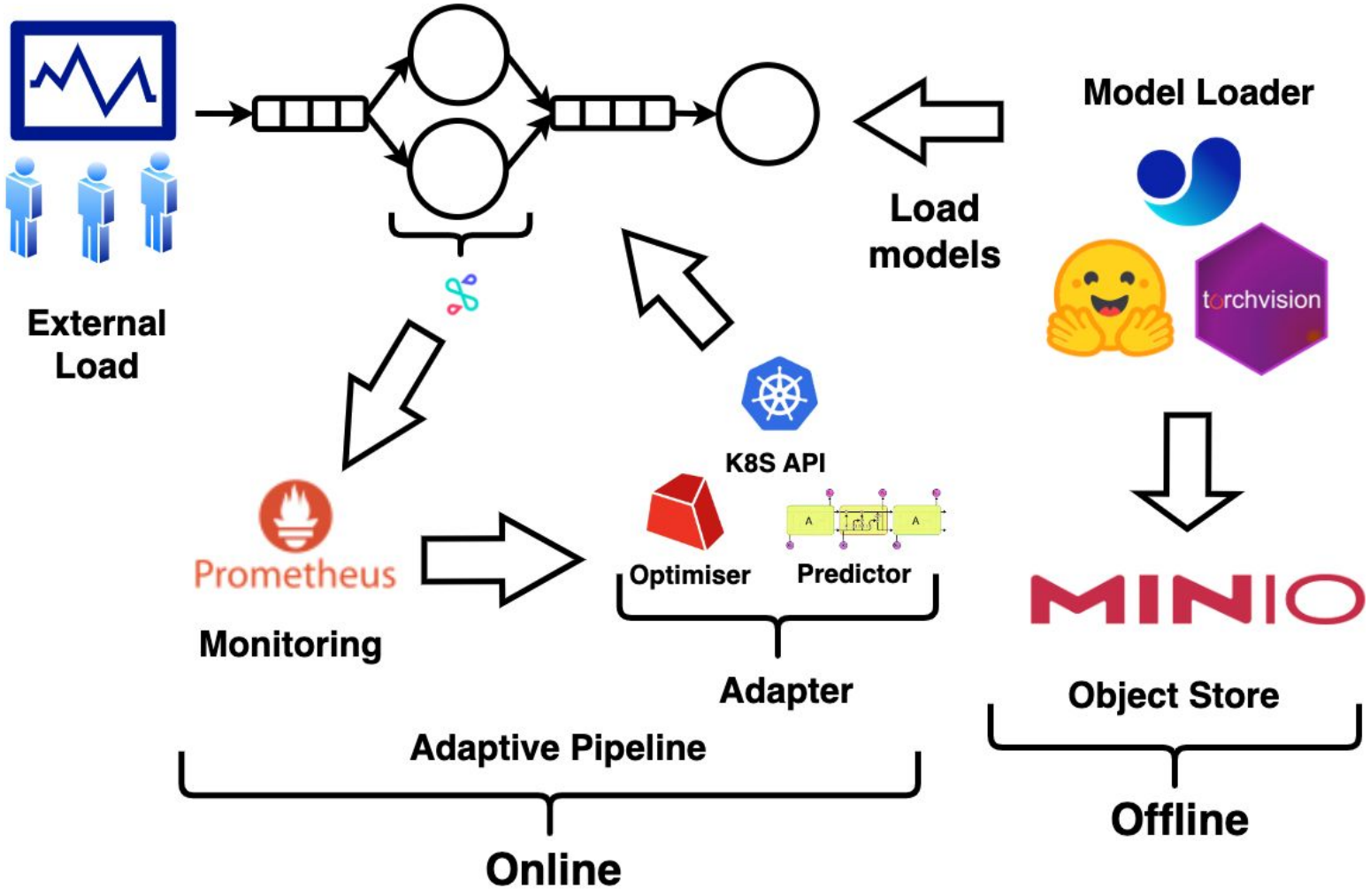
Goal: Providing a flexible inference pipeline



Snapshot of the System



System Design



Problem Formulation

$$f(n,s,I) = \alpha \sum_{s \in P} \Big(\sum_{m \in M_s} a_{s,m}.I_{s,m} \Big) - \beta \sum_{s \in P} n_s.R_s - \delta \sum_{s \in P} b_s$$

Accuracy Objective

Resource Objective

Batch Control

Objective function

$$\begin{aligned} \max \quad & f(n,s,I) \\ \text{subject to} \quad & \sum_{s \in P} l_s(b_s) + q_s(b_s) \le SLA_P, \\ & \text{if } I_{s,m} = 1, \text{ then} \\ & \quad n_s \cdot h_s(b_s) \ge \lambda_p, \quad \forall s \in P \\ & \sum_{m \in M_s} I_{s,m} = 1, \quad \forall s \in P \\ & n_s, b_s \in \mathbb{Z}^+, \quad I_{s,m} \in \{0,1\}, \end{aligned}$$

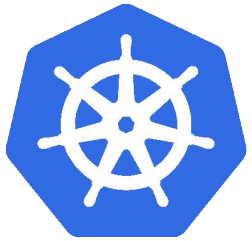
Latency SLA

Throughput Constraint

One active model per node

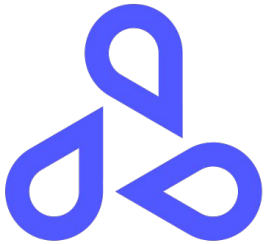
Implementation and Experimental Setup

How to navigate Model Variants



kubernetes

1. Industry standard
2. Used in recent research
3. Complete set of autoscaling, scheduling, observability tools (e.g. CPU usage)
4. APIs for changing the current AutoScaling algorithms

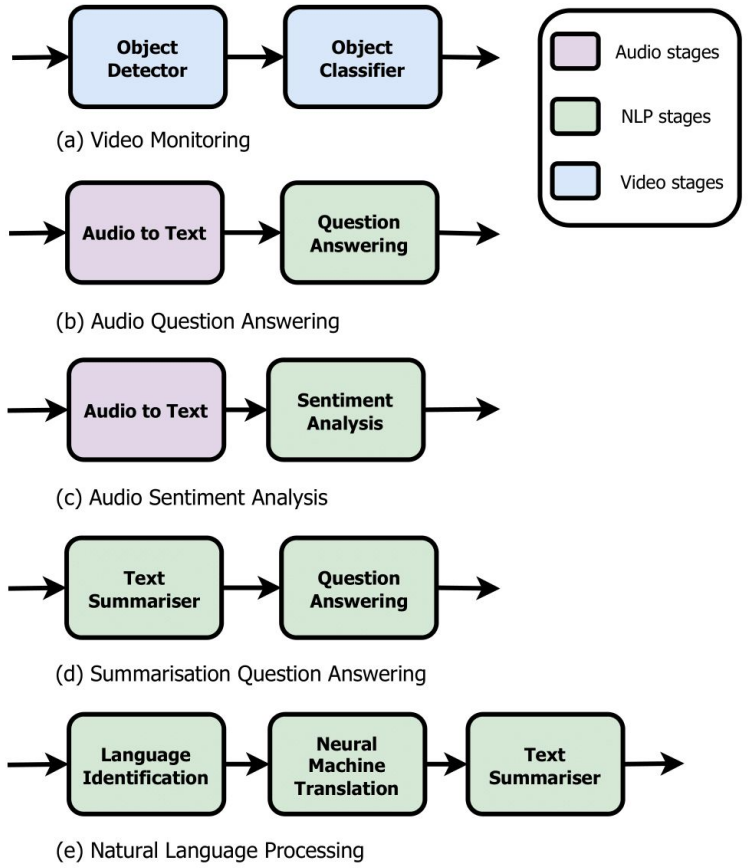


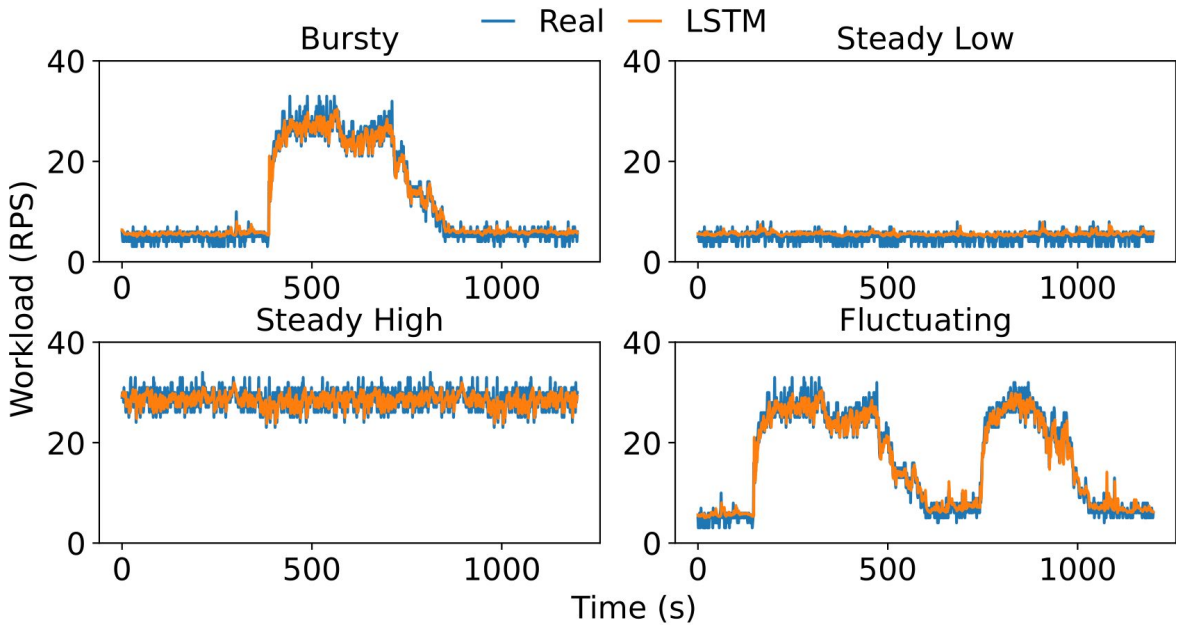
CORE

1. Industry standard ML server
2. Have the ability make inference graph
3. Rest and GRPC endpoints
4. Have many of the features we need like monitoring stack out of the box

Experimental Setup

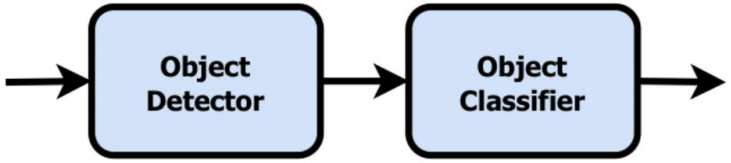
- A six node Kubernetes cluster

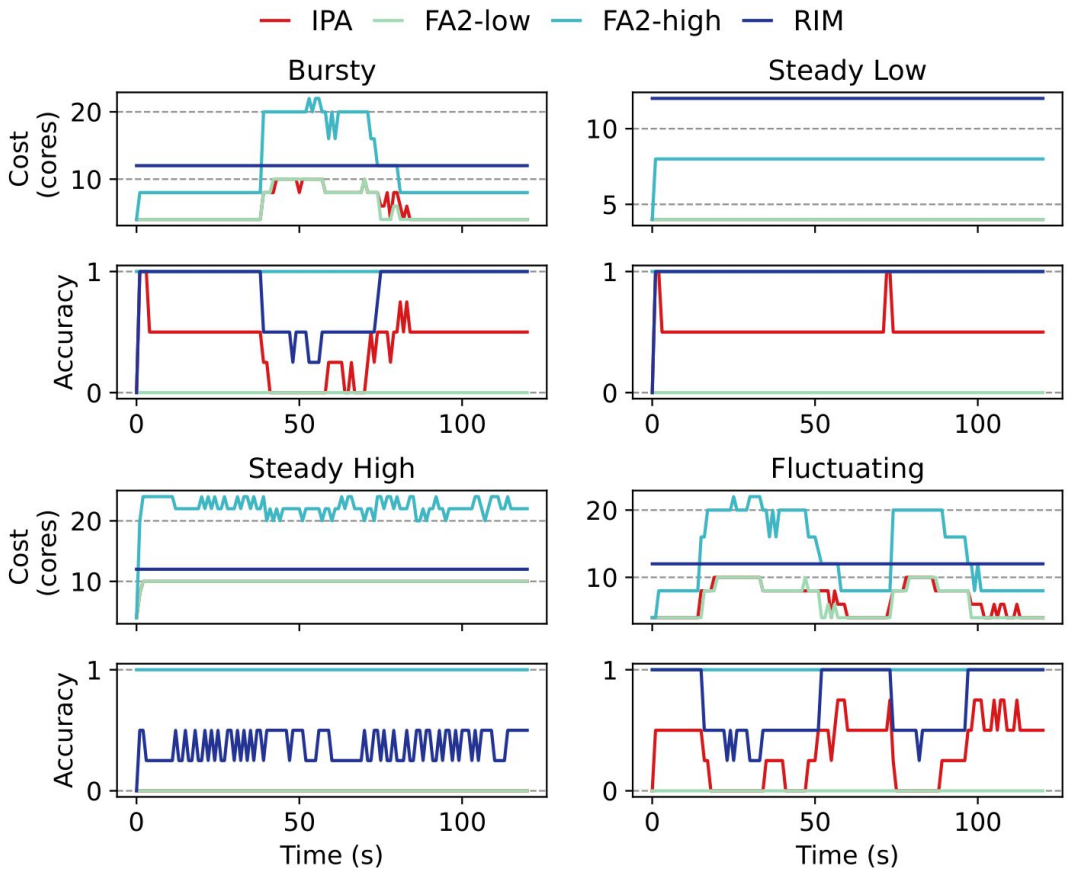


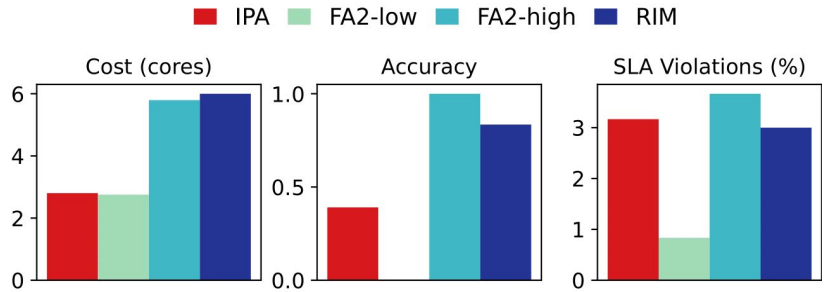


Experimental Results

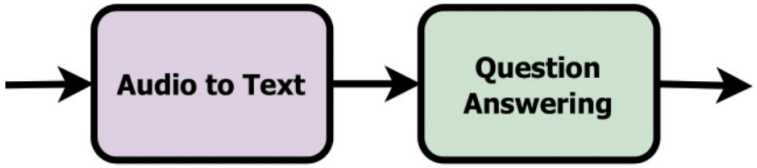
Video Pipeline

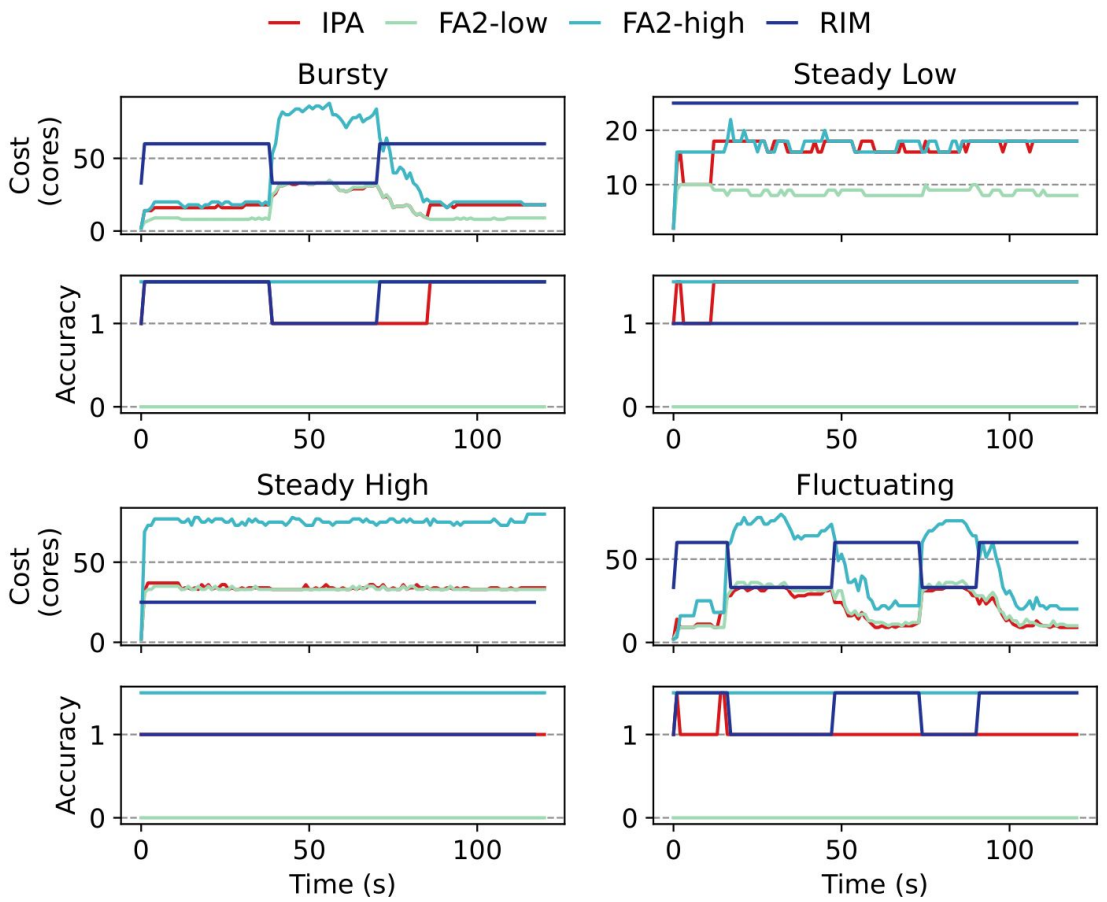


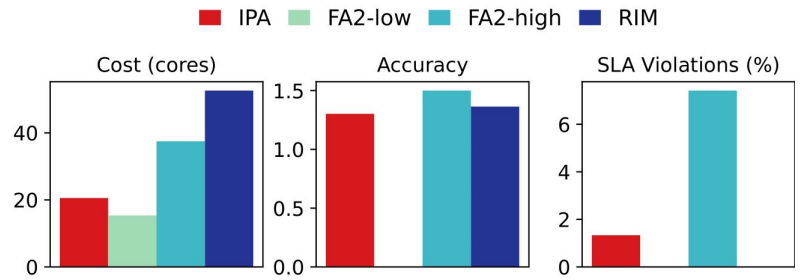




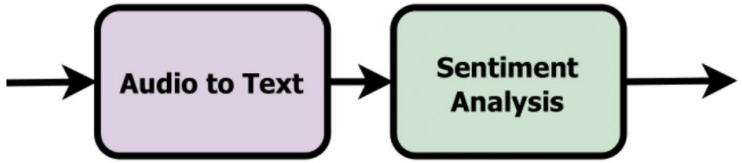
Audio + QA Pipeline

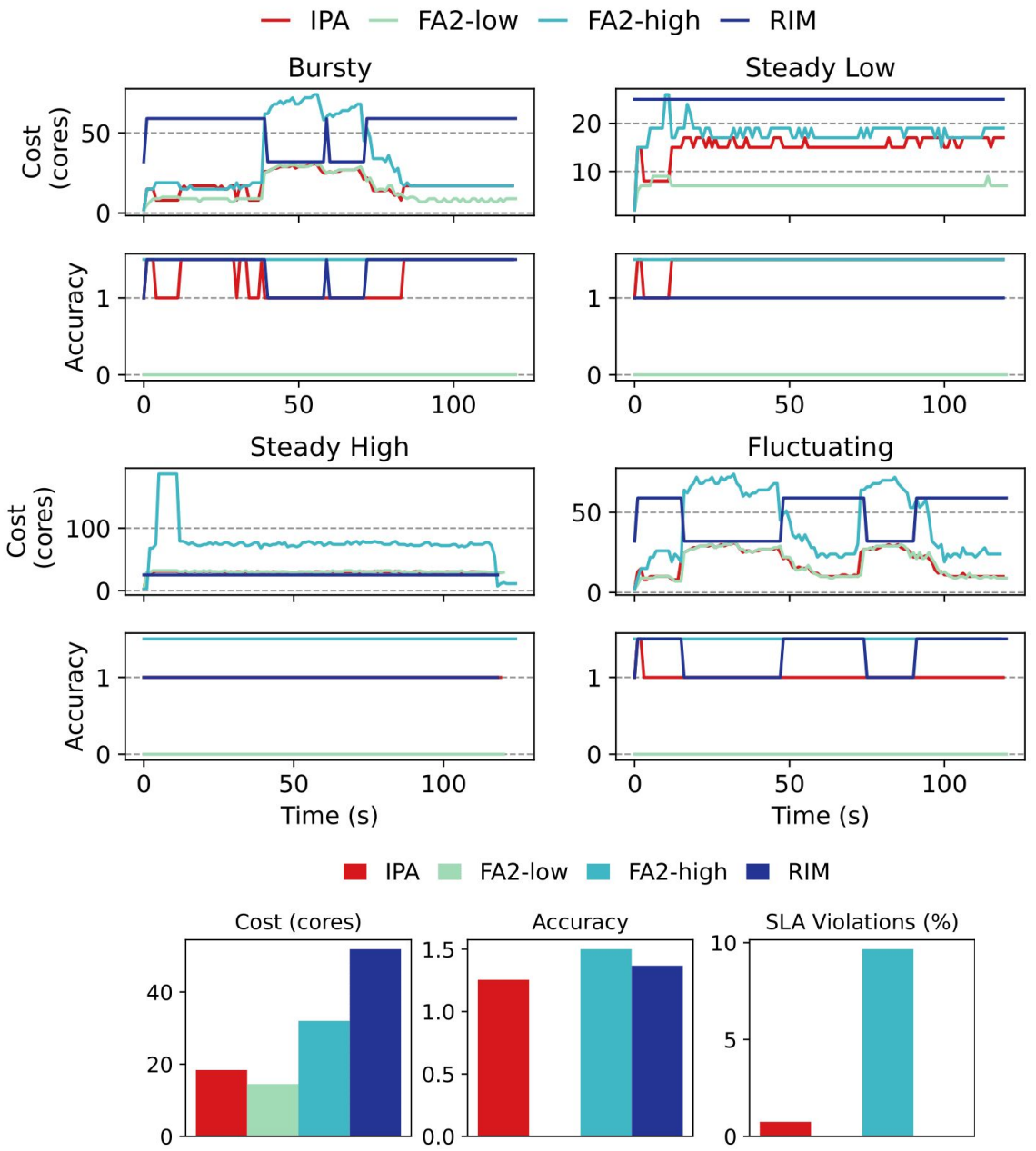




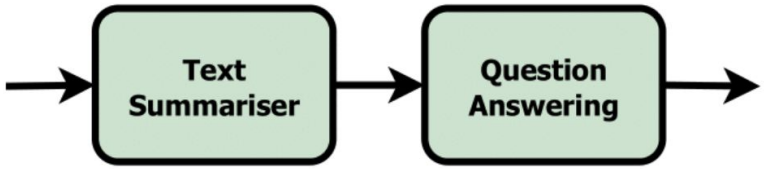


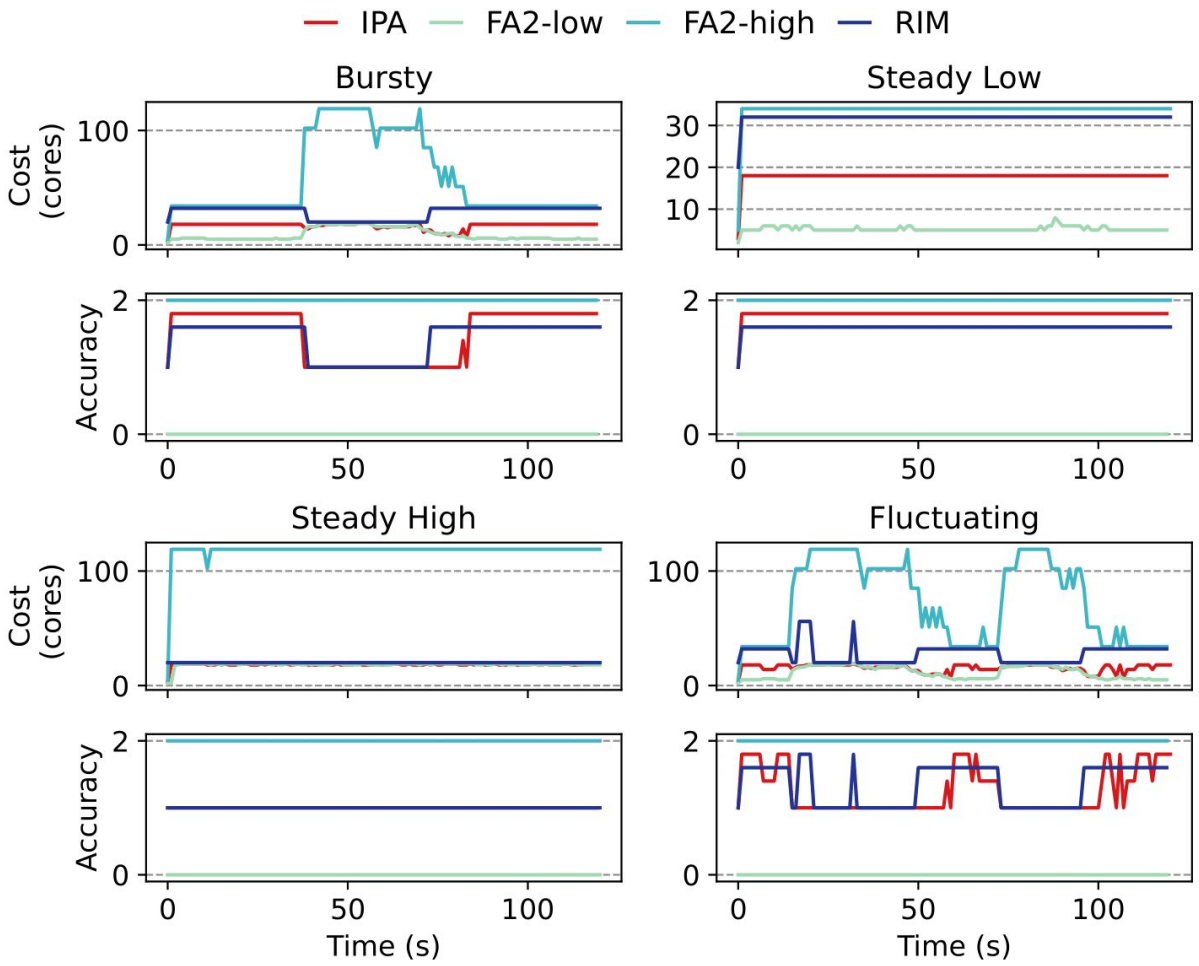
Summarization + QA Pipeline

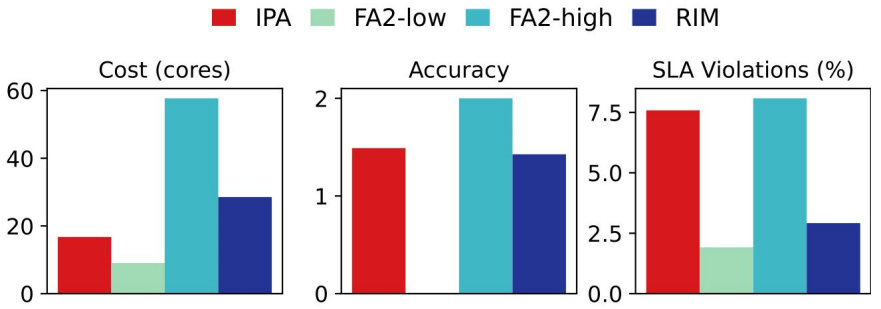




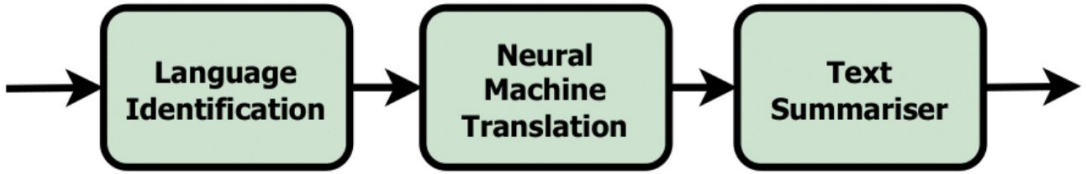
Summarization + QA Pipeline

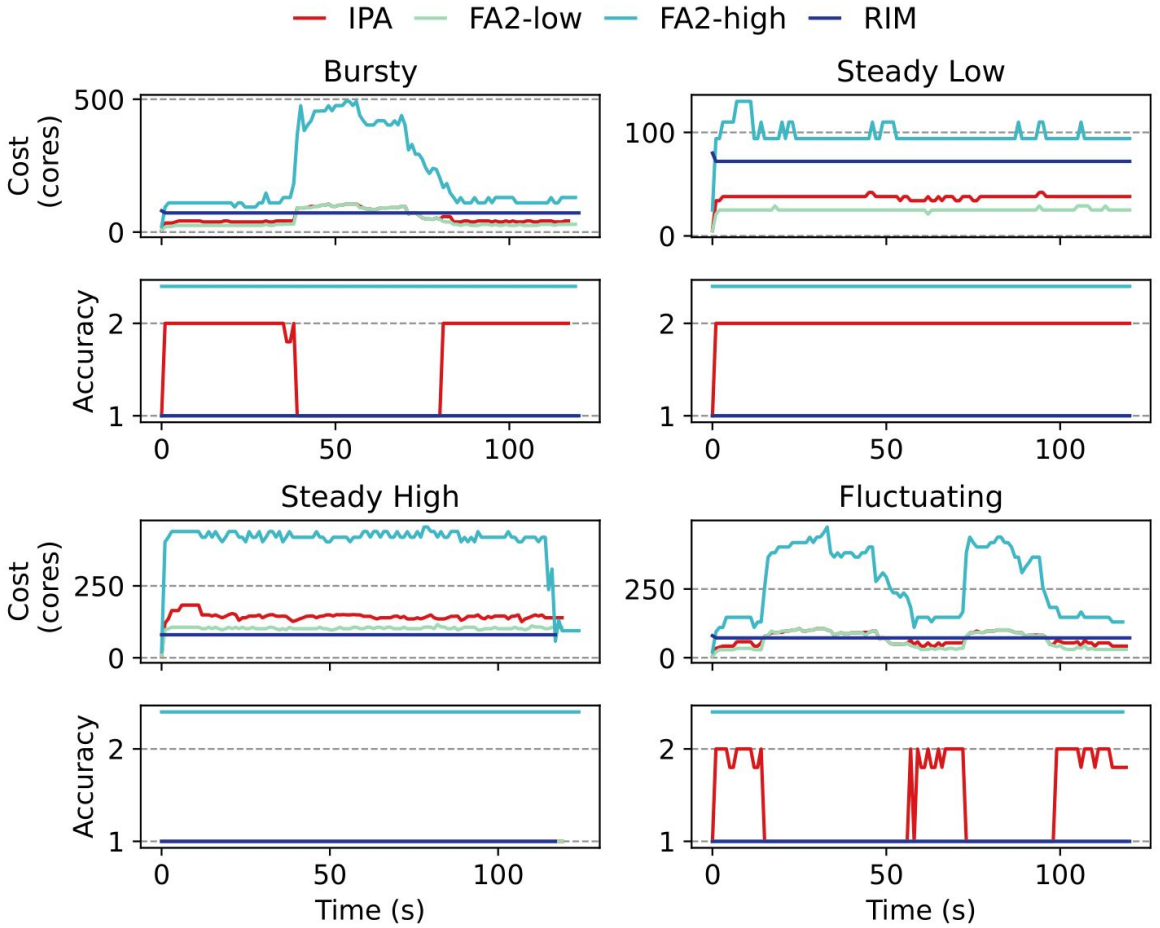


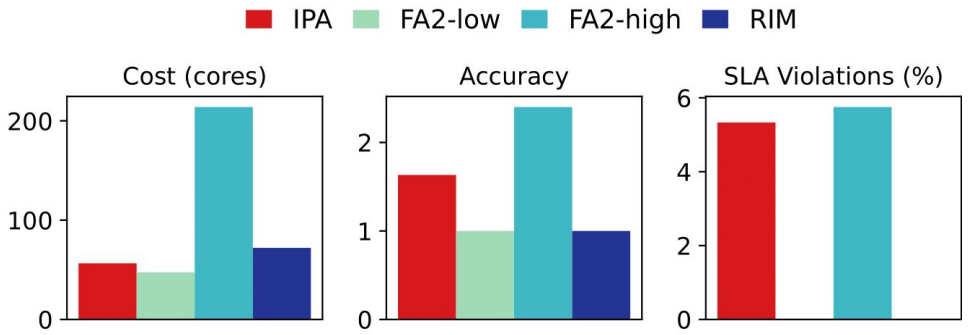




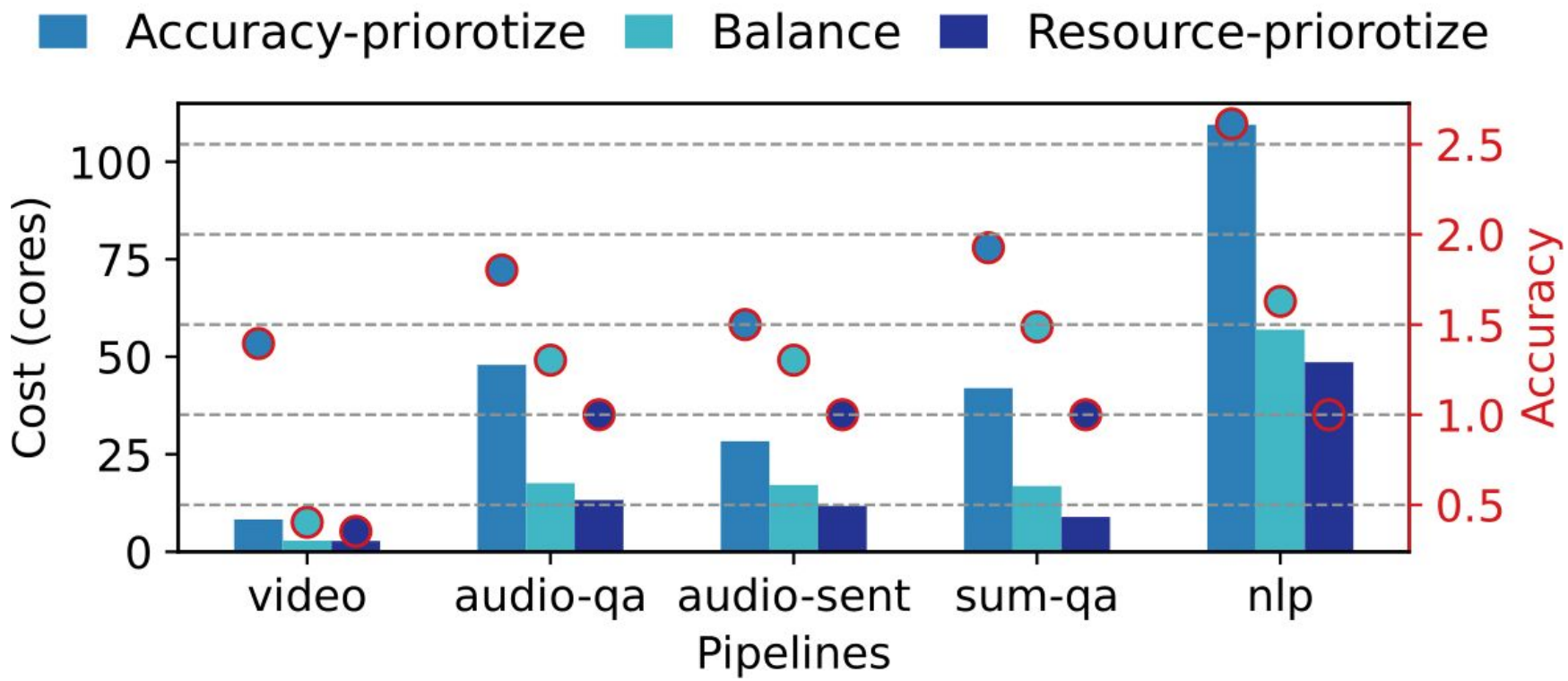
NLP Pipeline



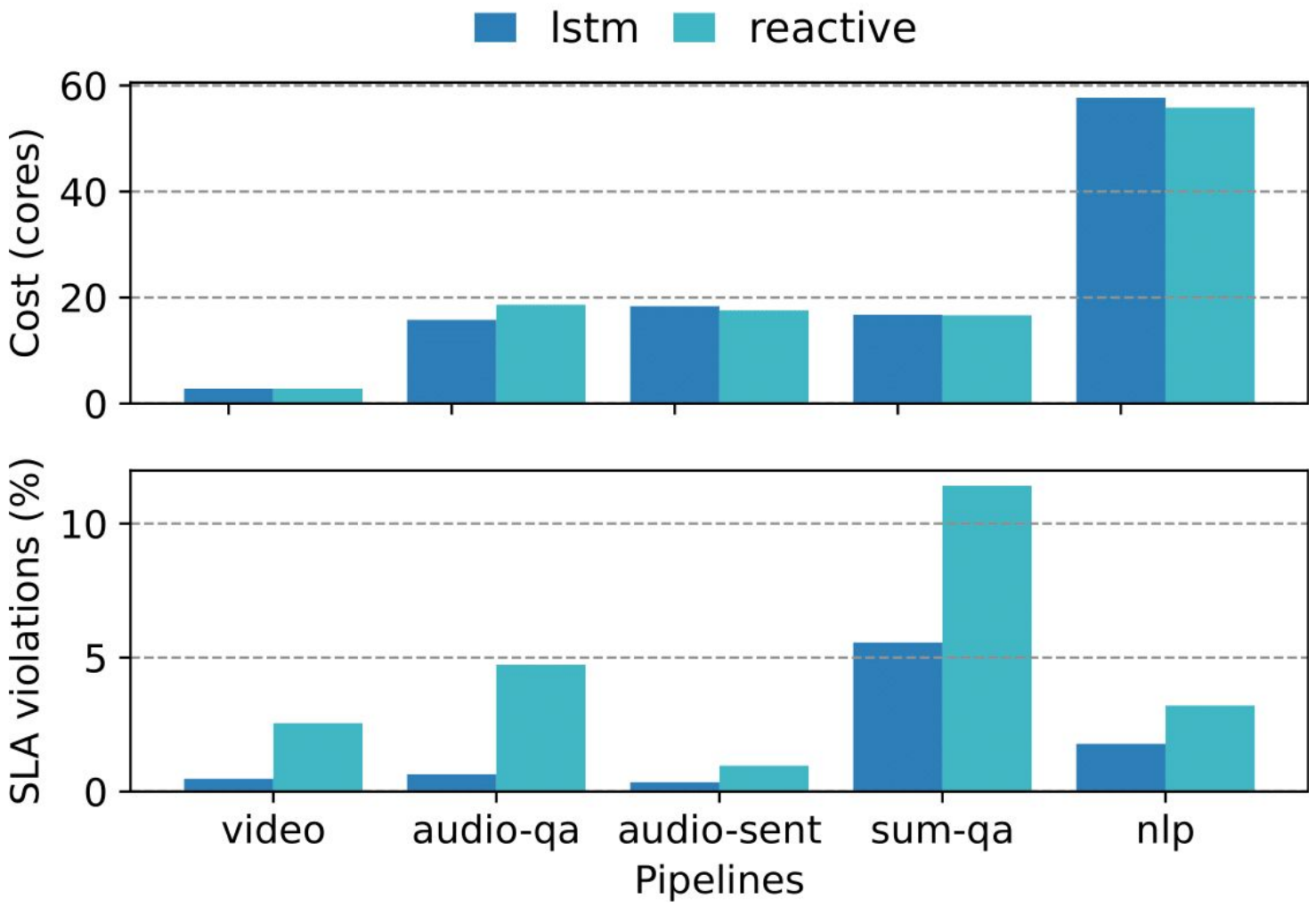




Adaptivity to multiple objectives



Effect of predictor



Gurobi solver scalability