Reconciling Accuracy, Cost, and Latency of Inference Serving Systems

Pooyan Jamshidi
University of South Carolina

https://pooyanjamshidi.github.io/

EuroMLSys

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

Mehran Salmani*, Saeid Ghafouri§‡, Alireza Sanaee§, Kamran Razavi†, Max Mühlhäuser†, Joseph Doyle§, Pooyan Jamshidi‡, Mohsen Sharifi*

Iran University of Science and Technology*, Queen Mary University of London§, Technical University of Darmstadt†, University of South Carolina‡

JSys Journal of Systems Research Volume 4, Issue 1, April 2024

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

Saeid Ghafouri
University of South Carolina & Queen Mary University of London

Kamran Razavi
Technical University of Darmstadt

Mehran Salmani
Technical University of Ilmenau

Alireza Sanaee
Queen Mary University of London

Tania Lorido Botran
Roblox

Lin Wang
Paderborn University

Joseph Doyle
Queen Mary University of London

Pooyan Jamshidi
University of South Carolina

EuroMLSys

Sponge: Inference Serving with Dynamic SLOs Using In-Place Vertical Scaling

Kamran Razavi*
Technical University of Darmstadt

Saeid Ghafouri*
Queen Mary University of London

Max Mühlhäuser
Technical University of Darmstadt

Pooyan Jamshidi
University of South Carolina

Lin Wang
Paderborn University

Problem:

Multi-Objective Optimization with Known Constraints under Uncertainty

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

Solutions:

Different Assumptions

InfAdapter [2023]:
Autoscaling for
ML Inference

IPA [2024]:
Autoscaling for
ML Inference Pipeline

Sponge [2024]:
Autoscaling for
ML Inference Pipeline
Dynamic SLO

EuroMLSys

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

Mehran Salmani*, Saeid Ghafouri§‡, Alireza Sanaee§, Kamran Razavi†, Max Mühlhäuser†, Joseph Doyle§, Pooyan Jamshidi‡, Mohsen Sharifi*

Iran University of Science and Technology*, Queen Mary University of London§, Technical University of Darmstadt†, University of South Carolina‡

InfAdapter [2023]:
Autoscaling for
ML **Model Inference**

JSys Journal of Systems Research Volume 4, Issue 1, April 2024

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

Saeid Ghafouri
University of South Carolina & Queen Mary University of London
Kamran Razavi
Technical University of Darmstadt
Mehran Salmani
Technical University of Ilmenau
Alireza Sanaee
Queen Mary University of London
Tania Lorido Botran
Roblox
Lin Wang
Paderborn University
Joseph Doyle
Queen Mary University of London
Pooyan Jamshidi
University of South Carolina

IPA [2024]:
Autoscaling for
ML Inference Pipeline

EuroMLSys

Sponge: Inference Serving with Dynamic SLOs Using In-Place Vertical Scaling

Kamran Razavi*
Technical University of Darmstadt
Saeid Ghafouri*
Queen Mary University of London
Max Mühlhäuser
Technical University of Darmstadt
Pooyan Jamshidi
University of South Carolina
Lin Wang
Paderborn University

Sponge [2024]:
Autoscaling for
ML Inference Pipeline with
Dynamic SLO

“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!

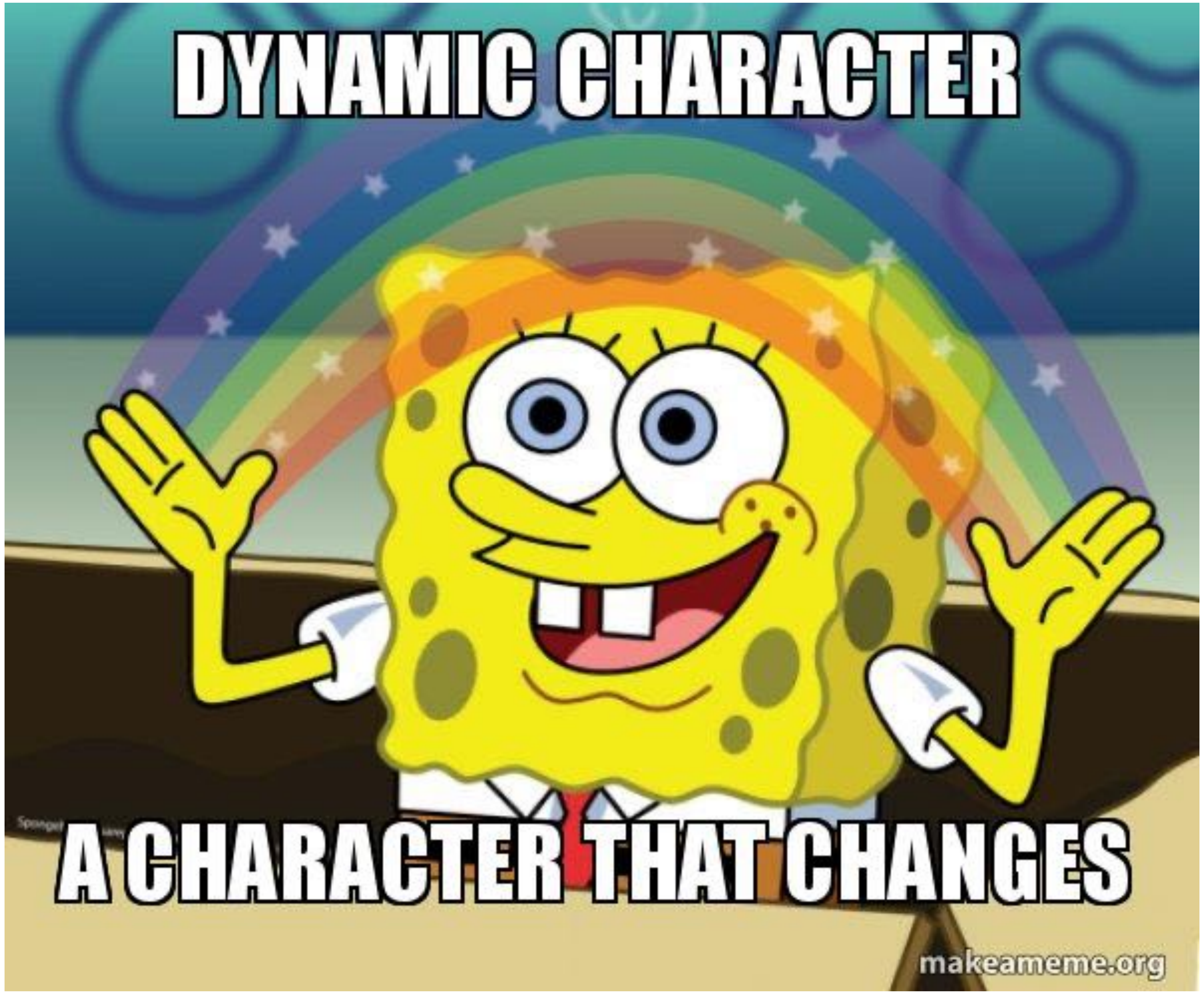
Cost-Efficient!

Highly Accurate!

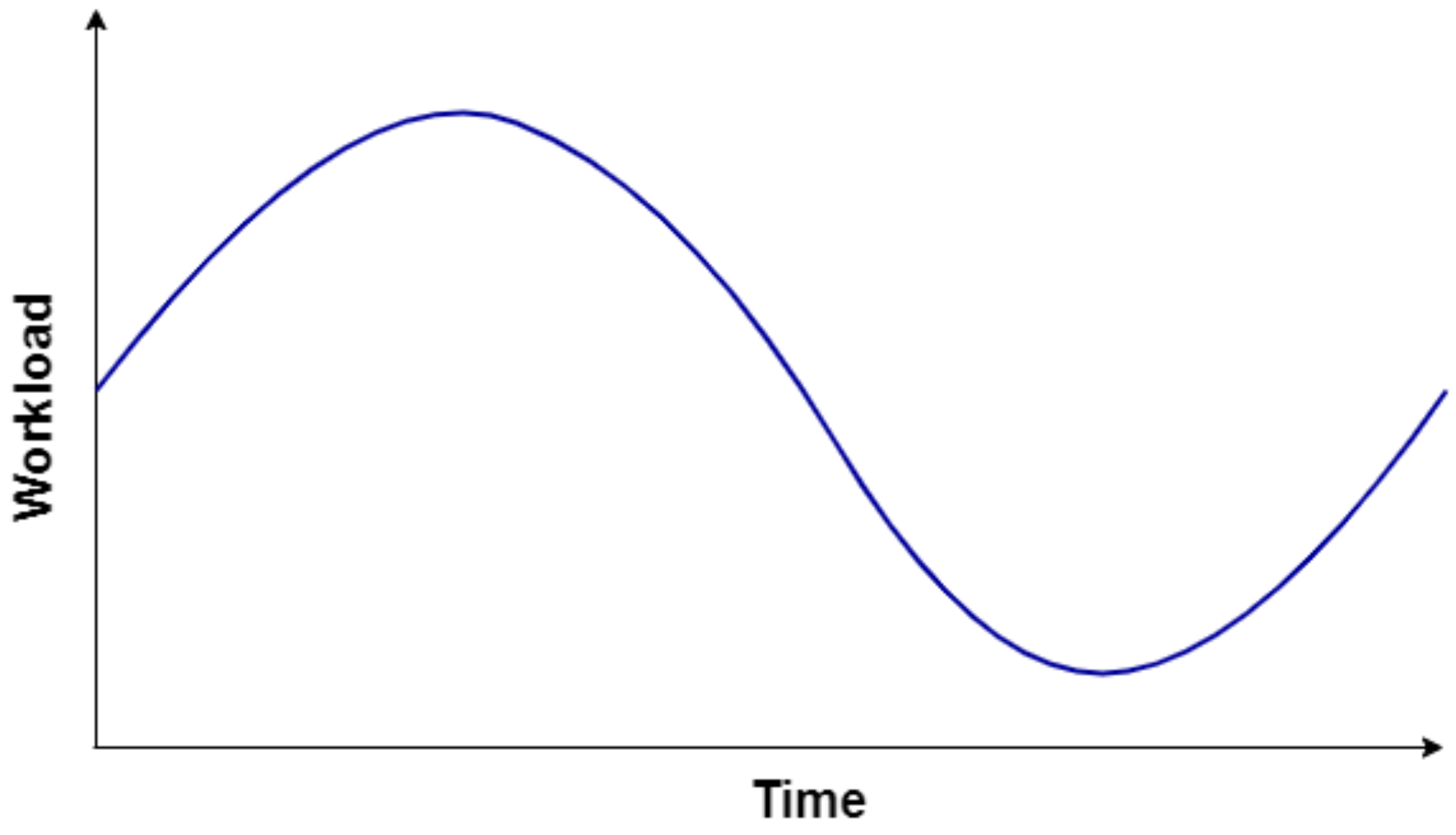
ML inference services have strict & **conflicting** requirements

Highly Responsive!	Cost-Efficient!	Highly Accurate!
		

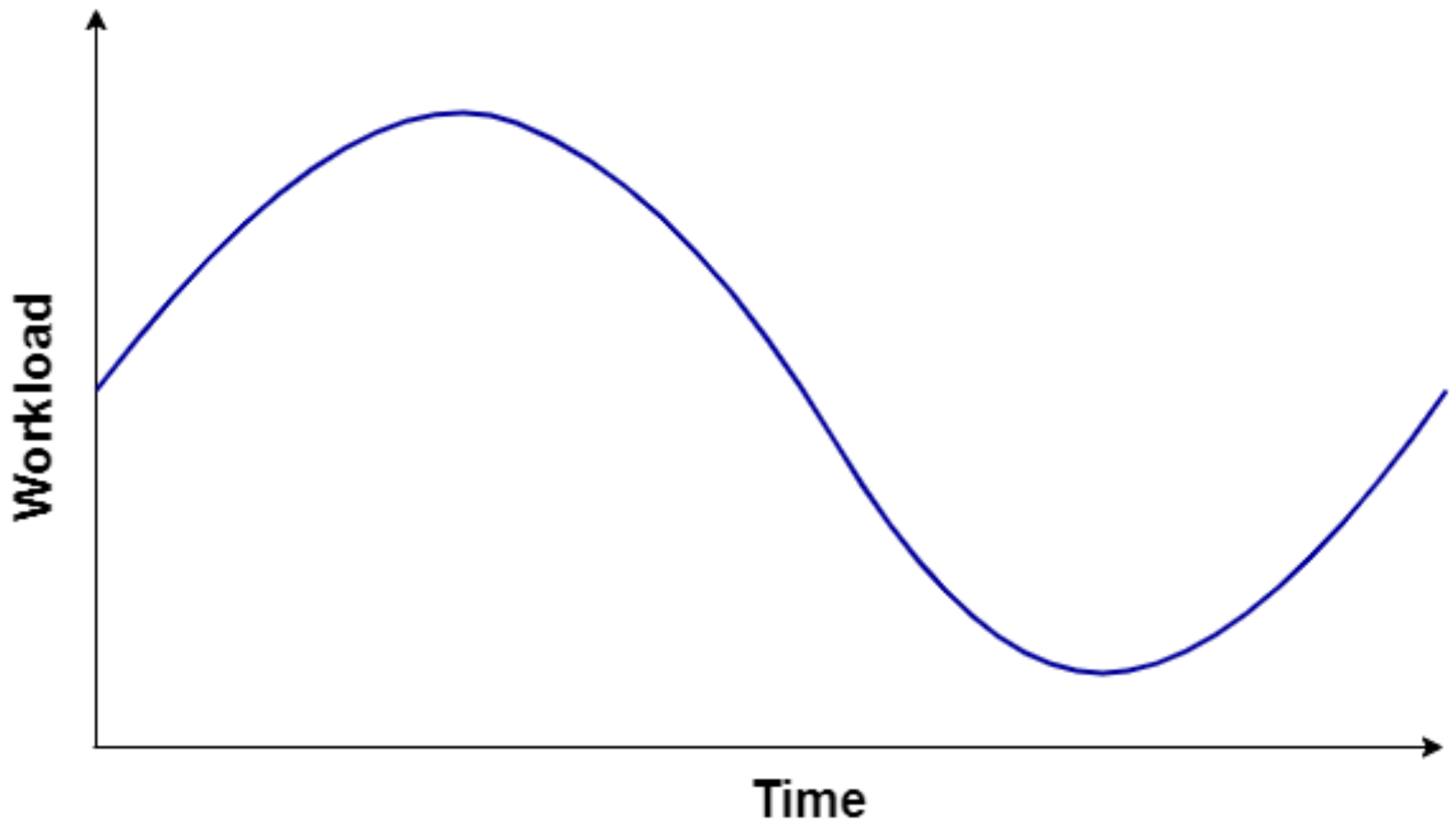
More challenge: Dynamic workload

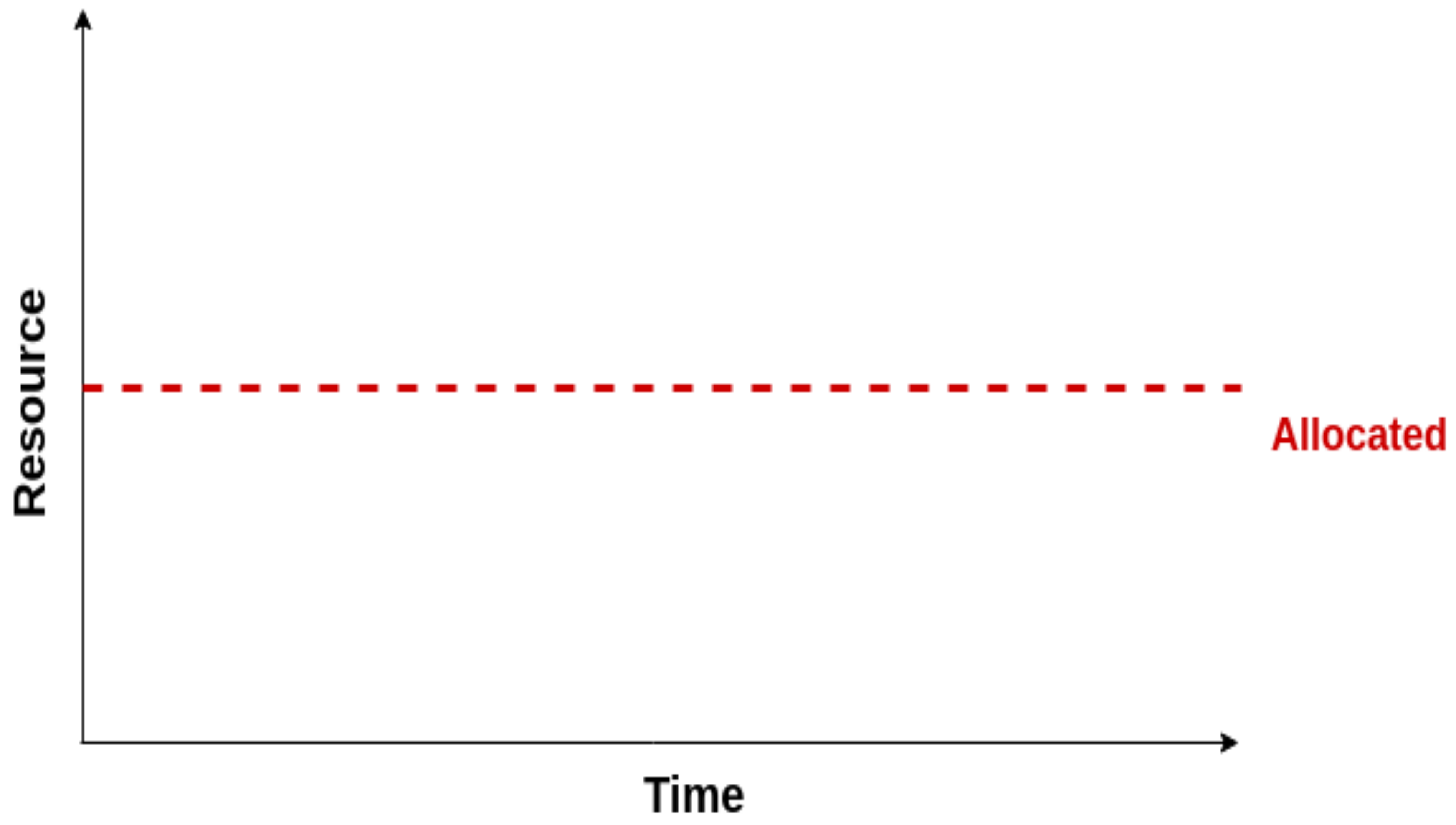


Resource allocation

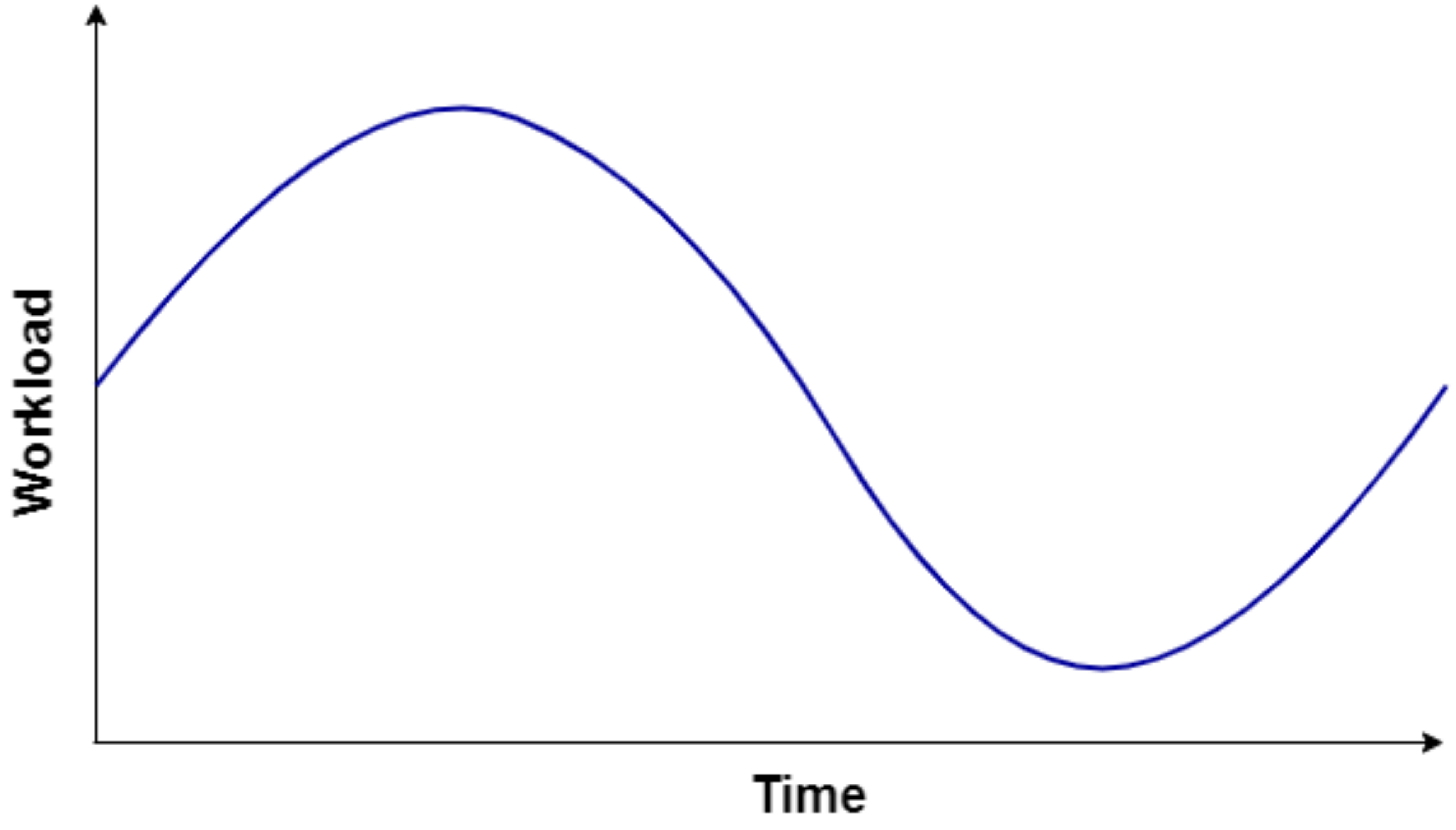


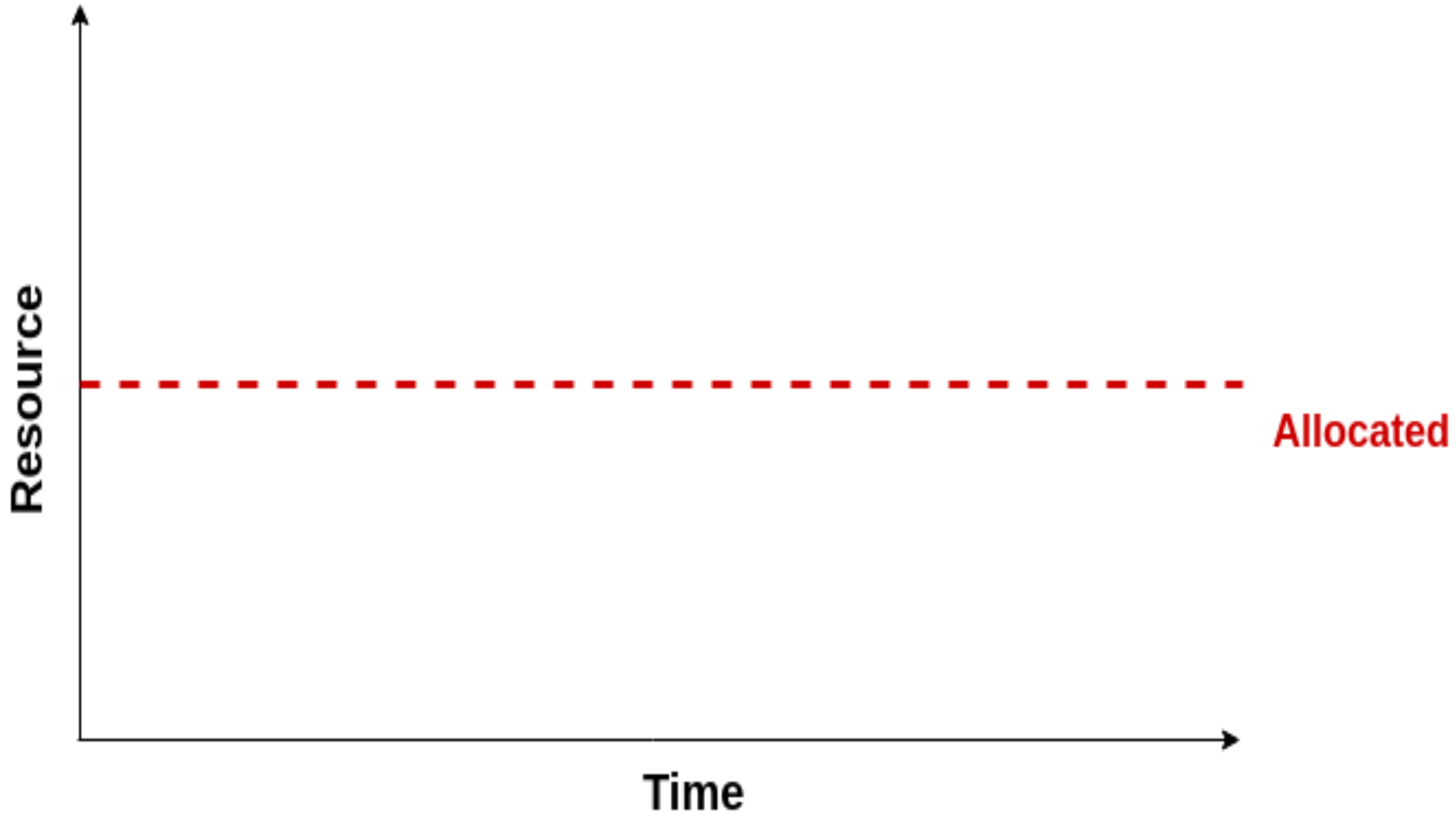
Resource allocation

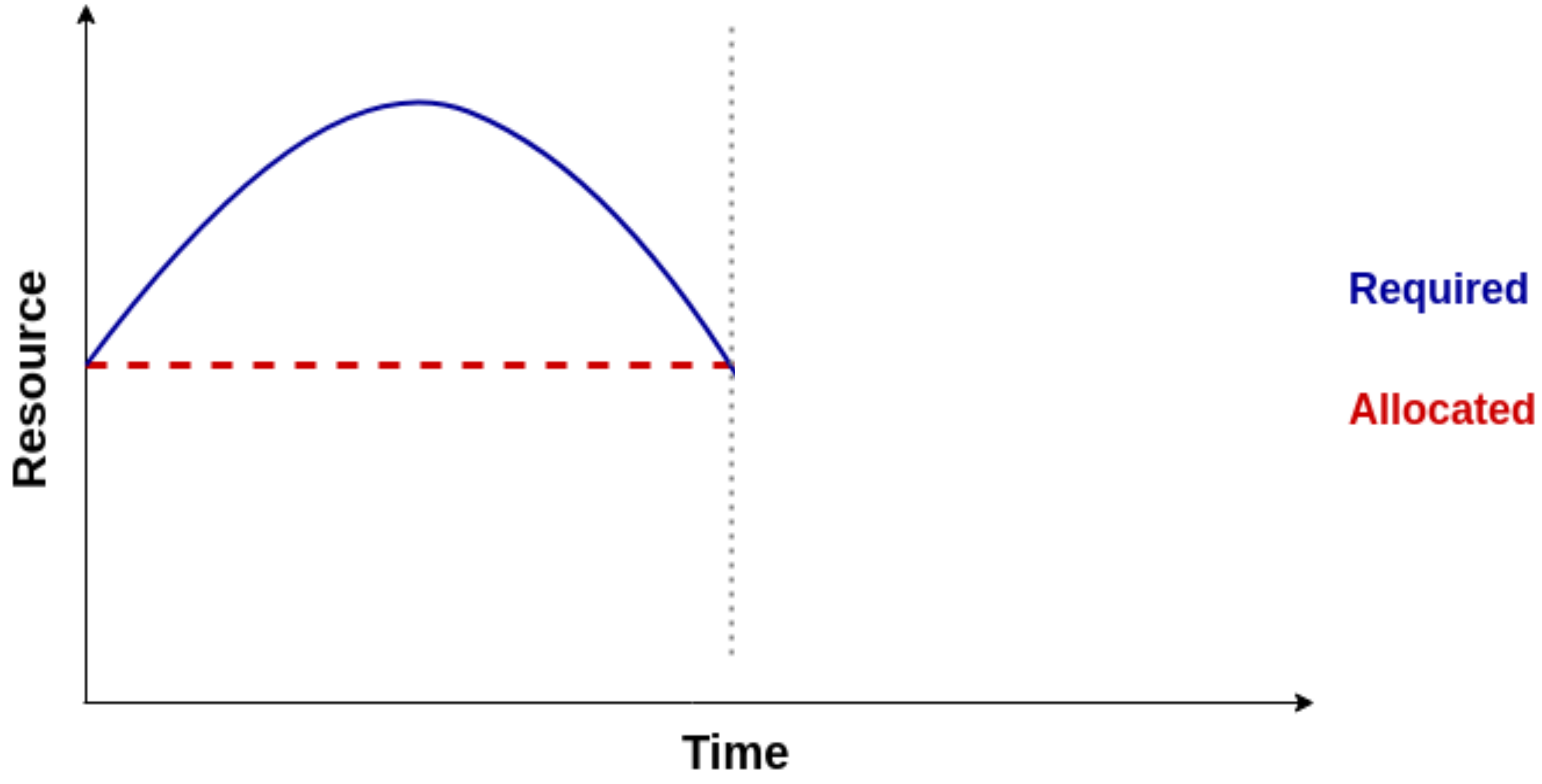




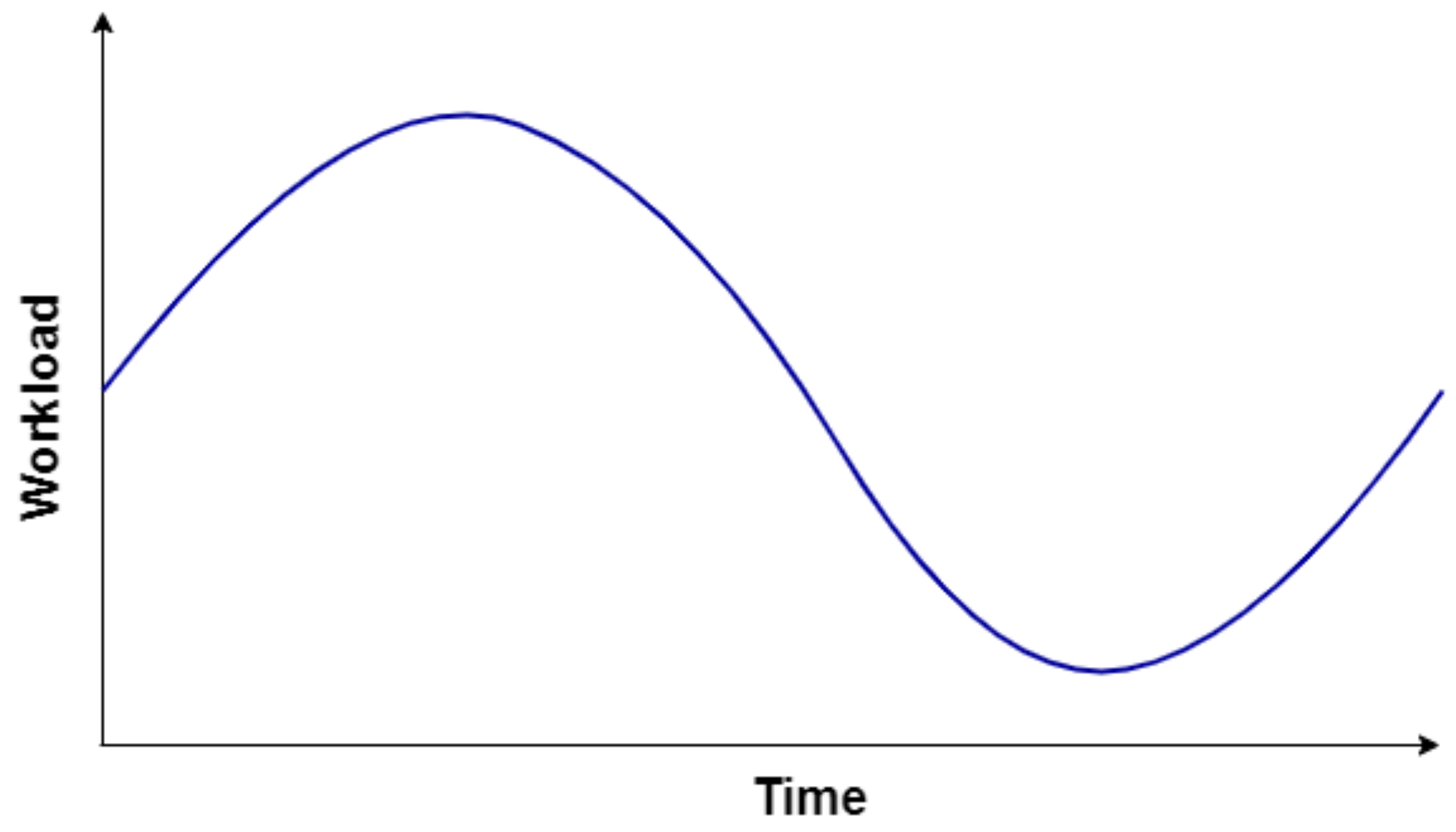
Resource allocation

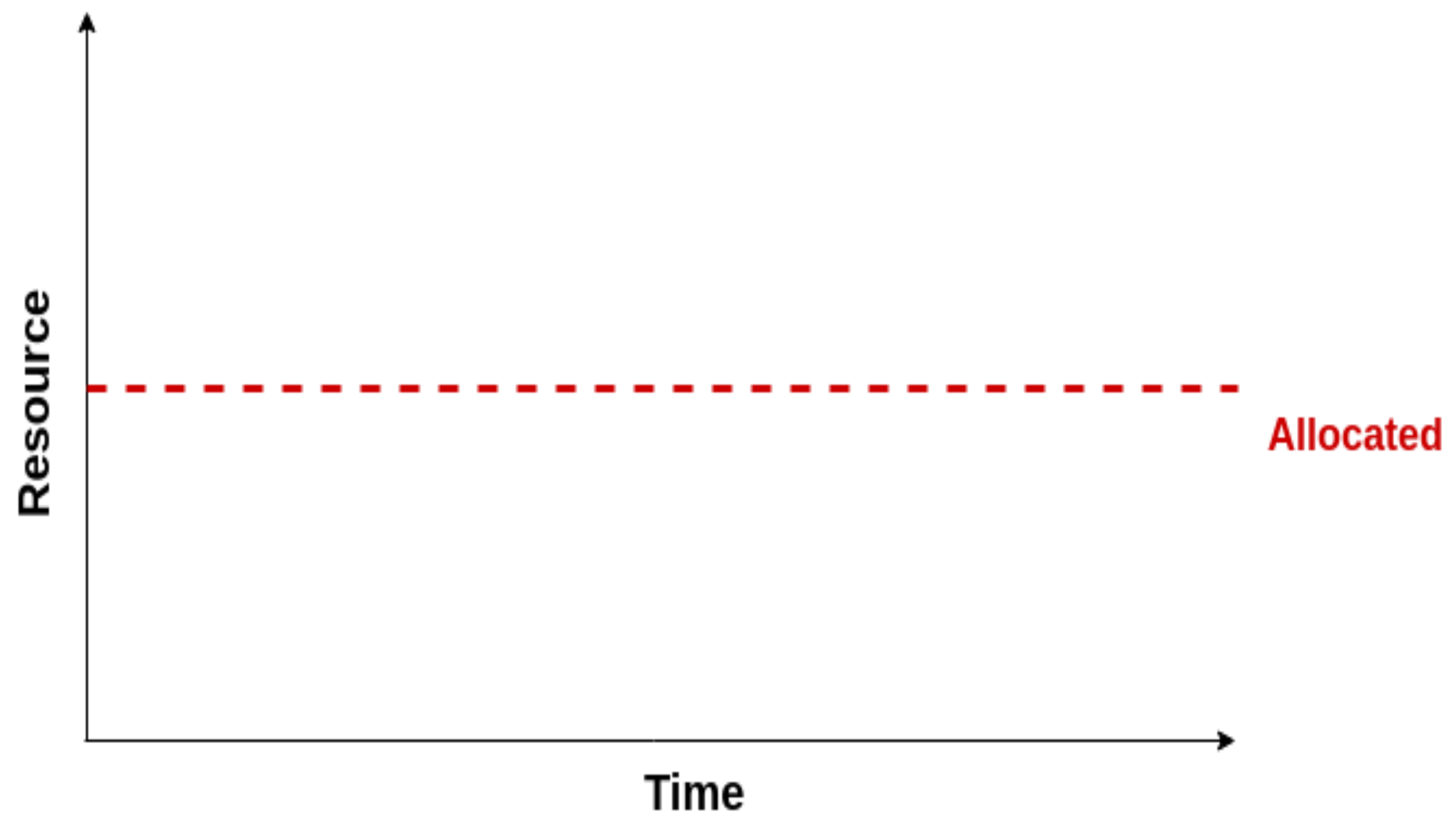


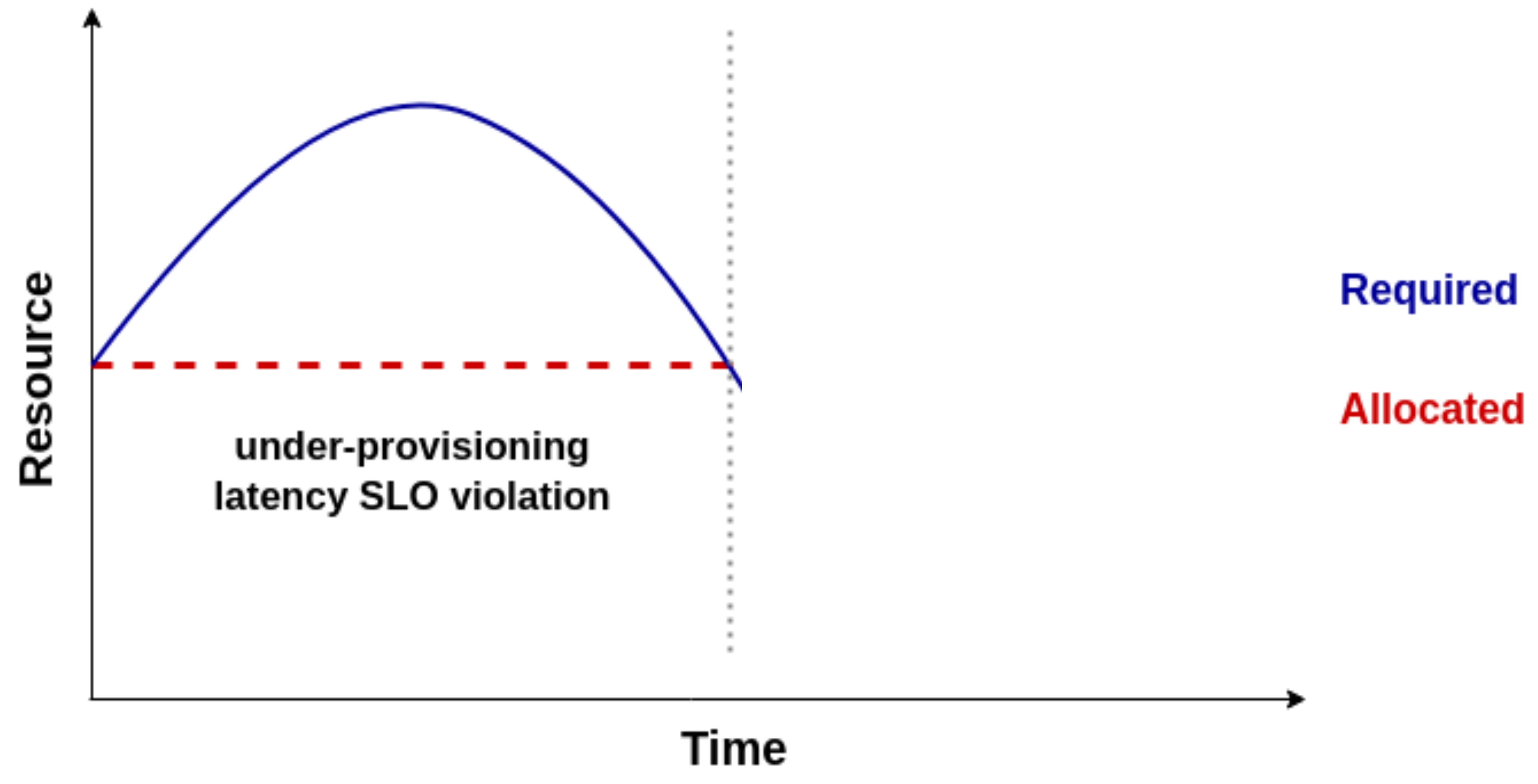




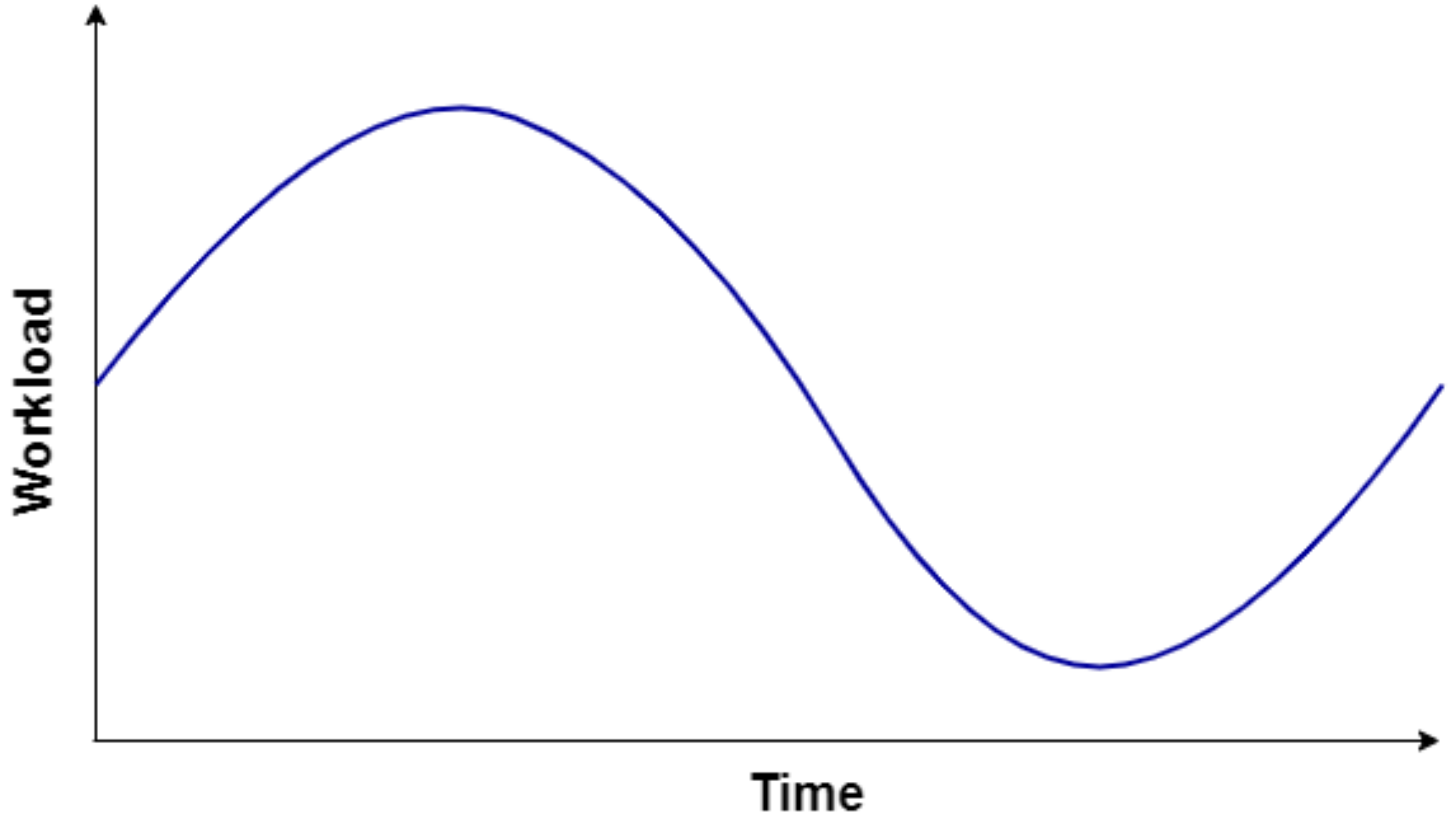
Resource allocation

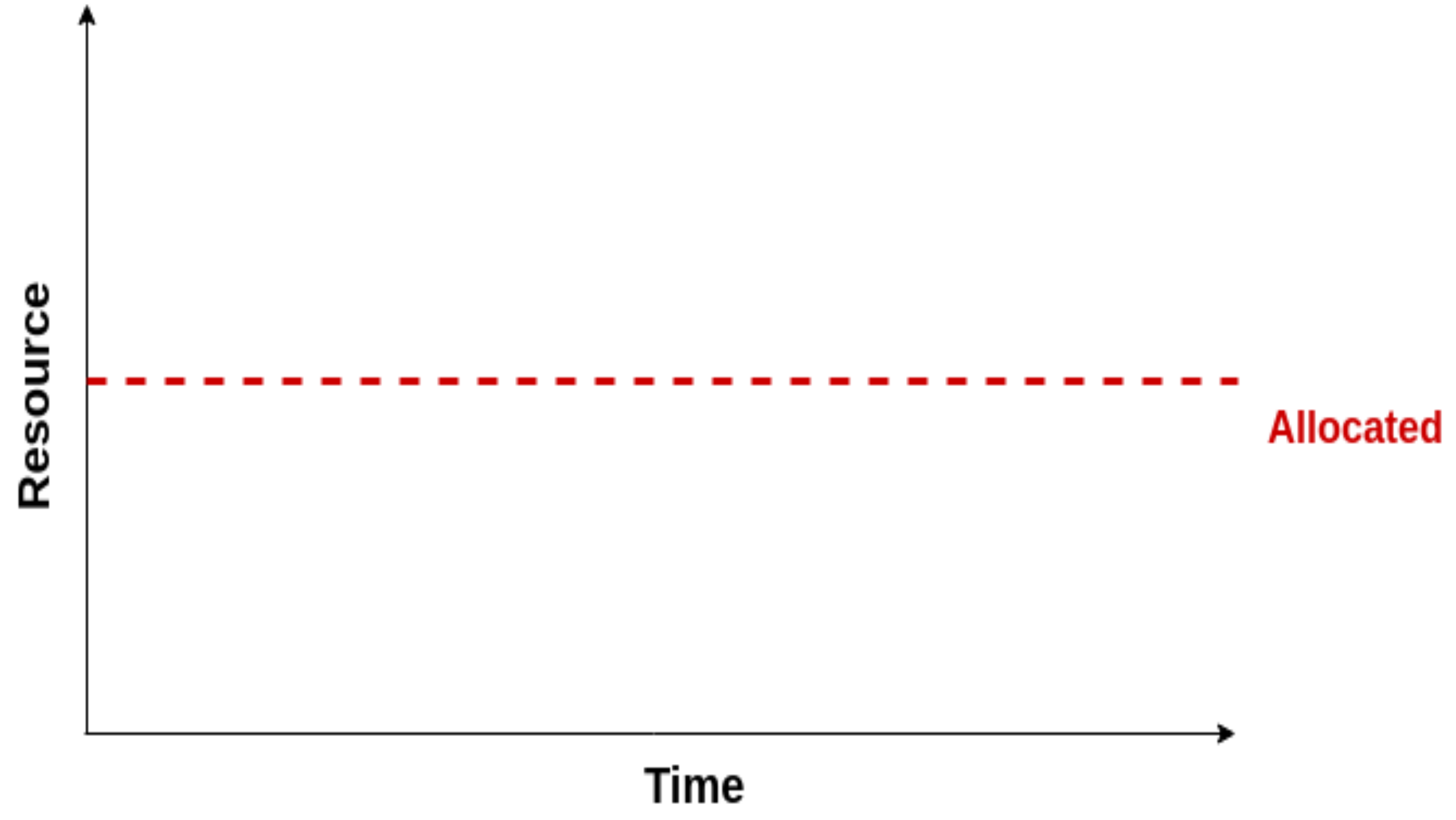


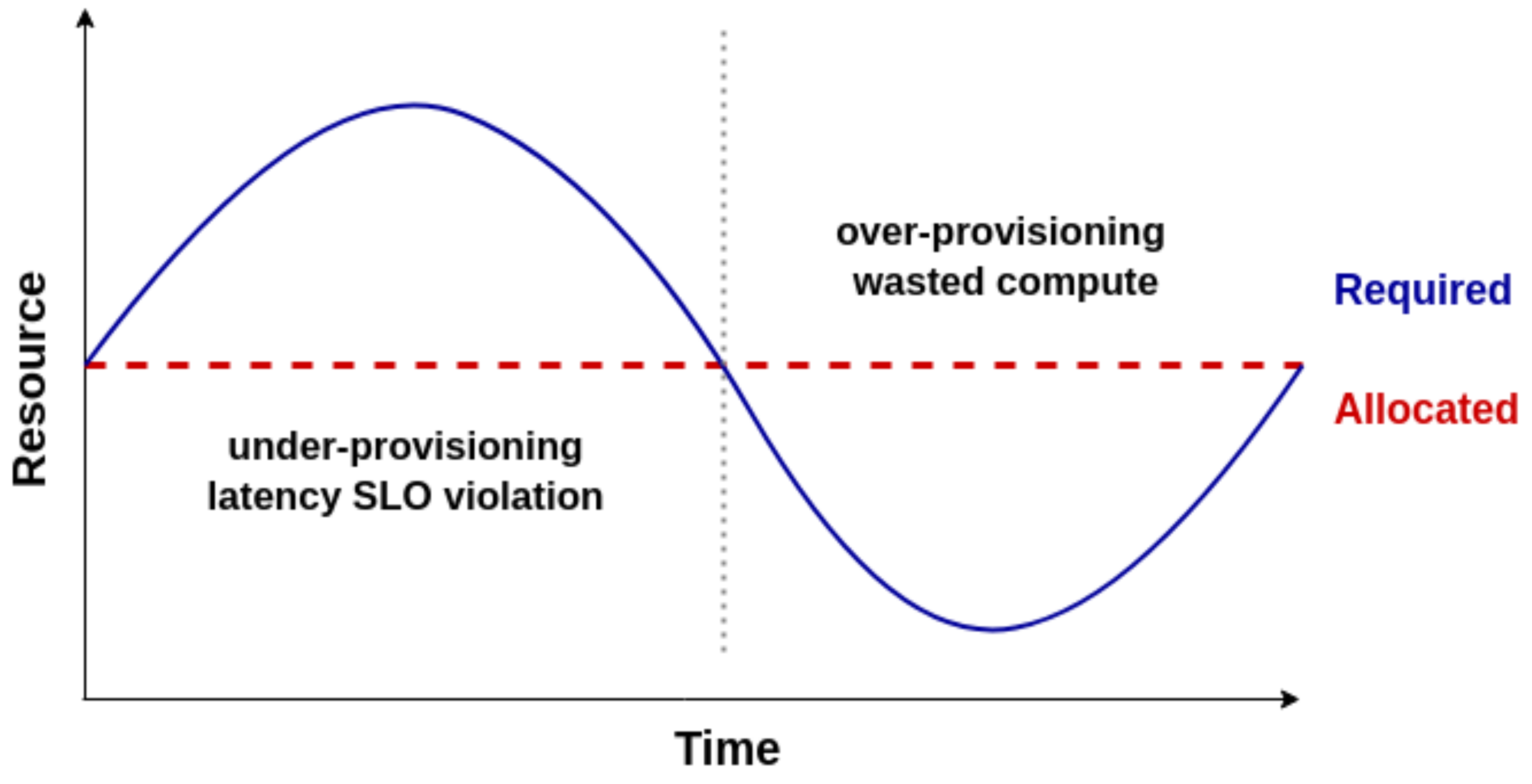




Resource allocation







Resource allocation

Over Provisioning

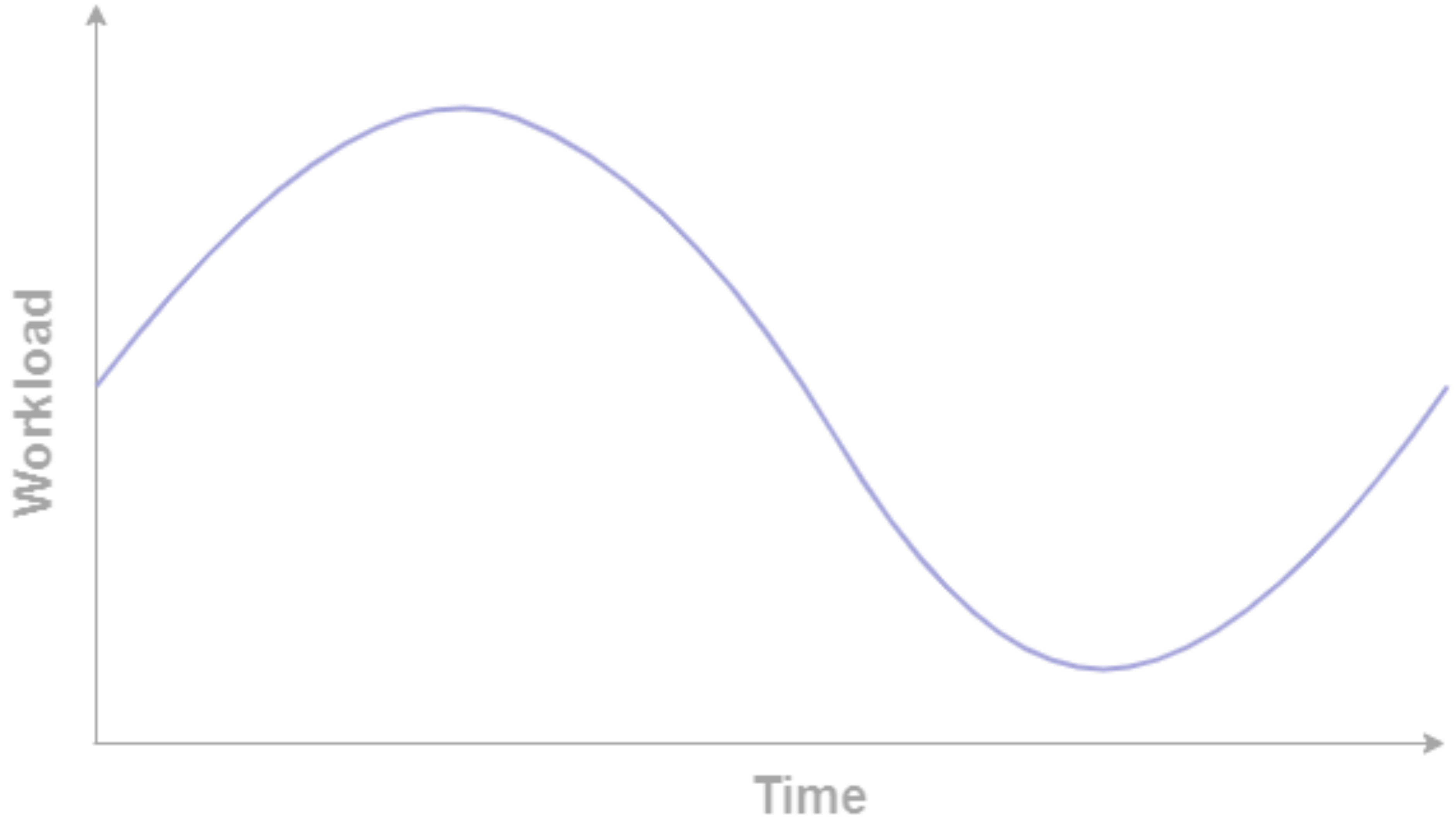
Under Provisioning

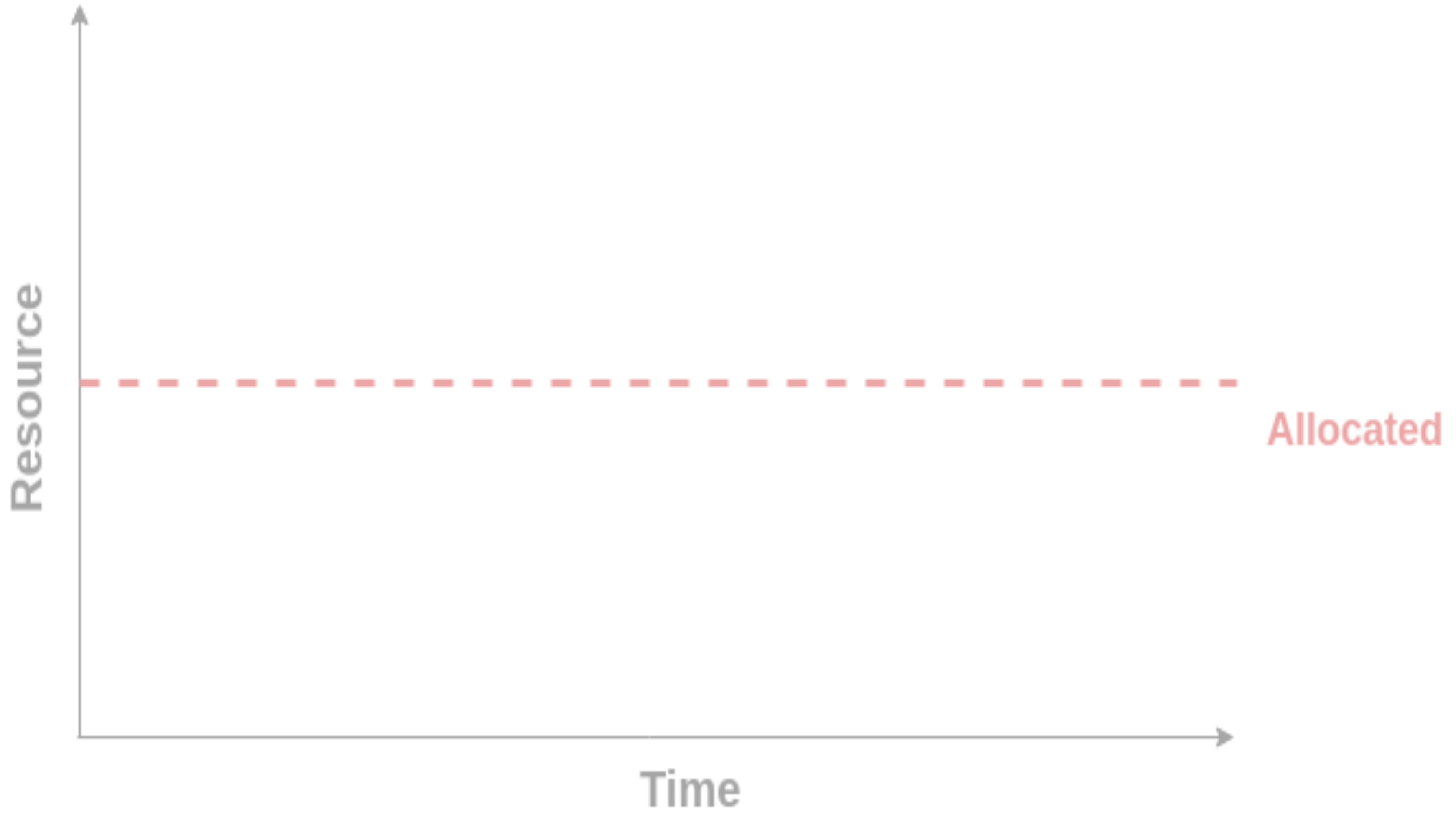
Quality adaptation

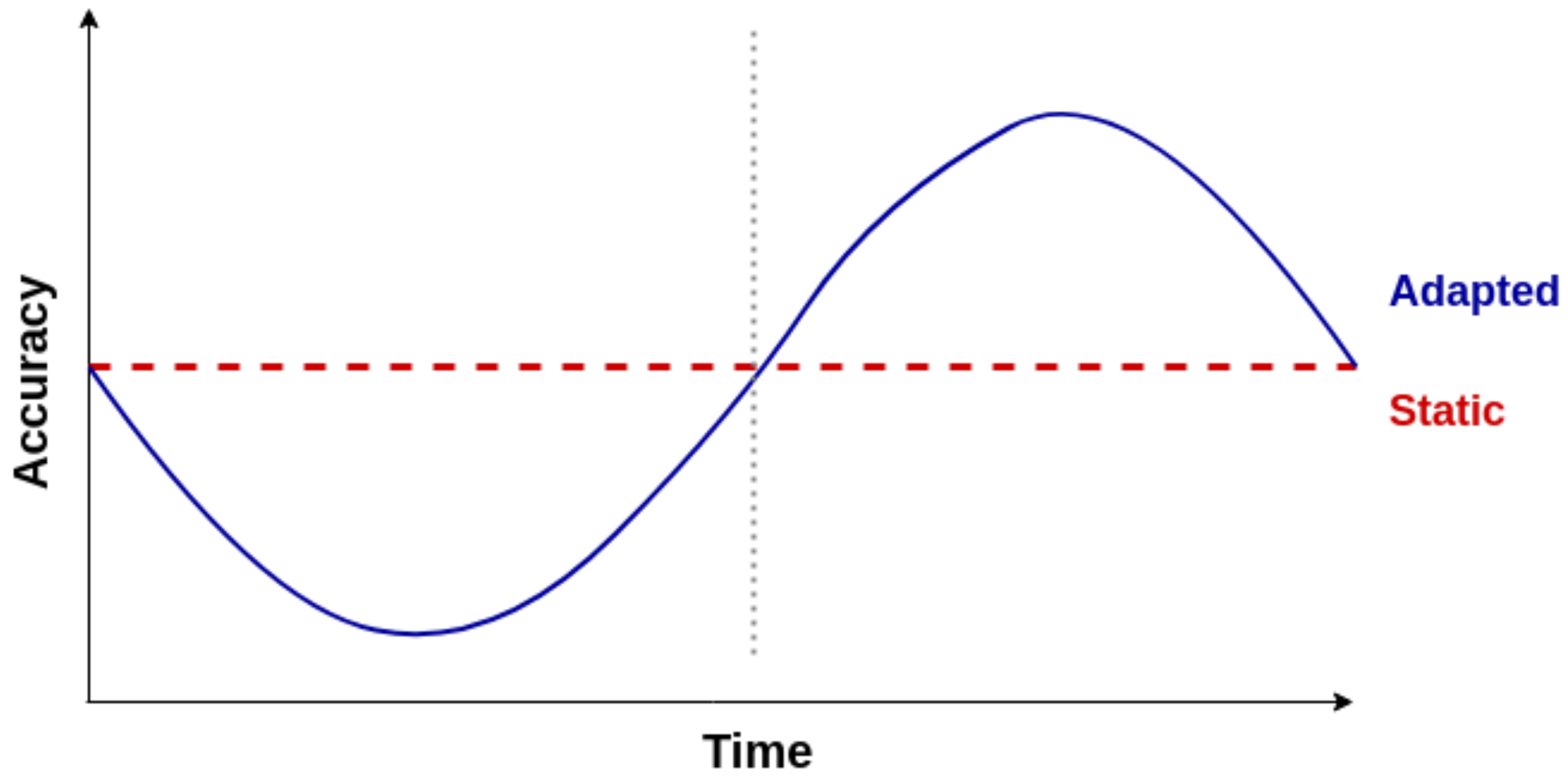
ResNet18: Tiger

ResNet152: Dog

Quality adaptation





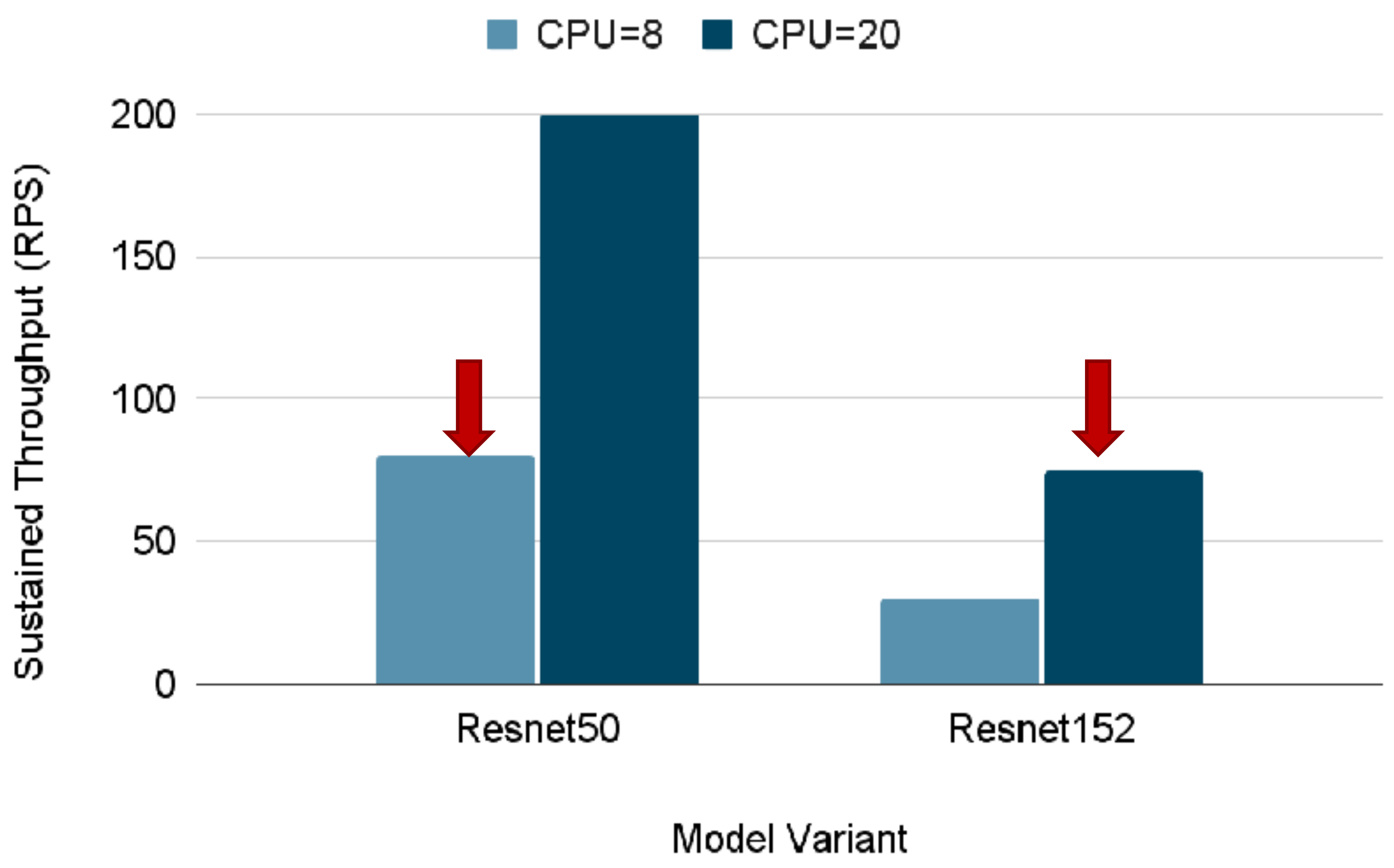


Solution: InfAdapter

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

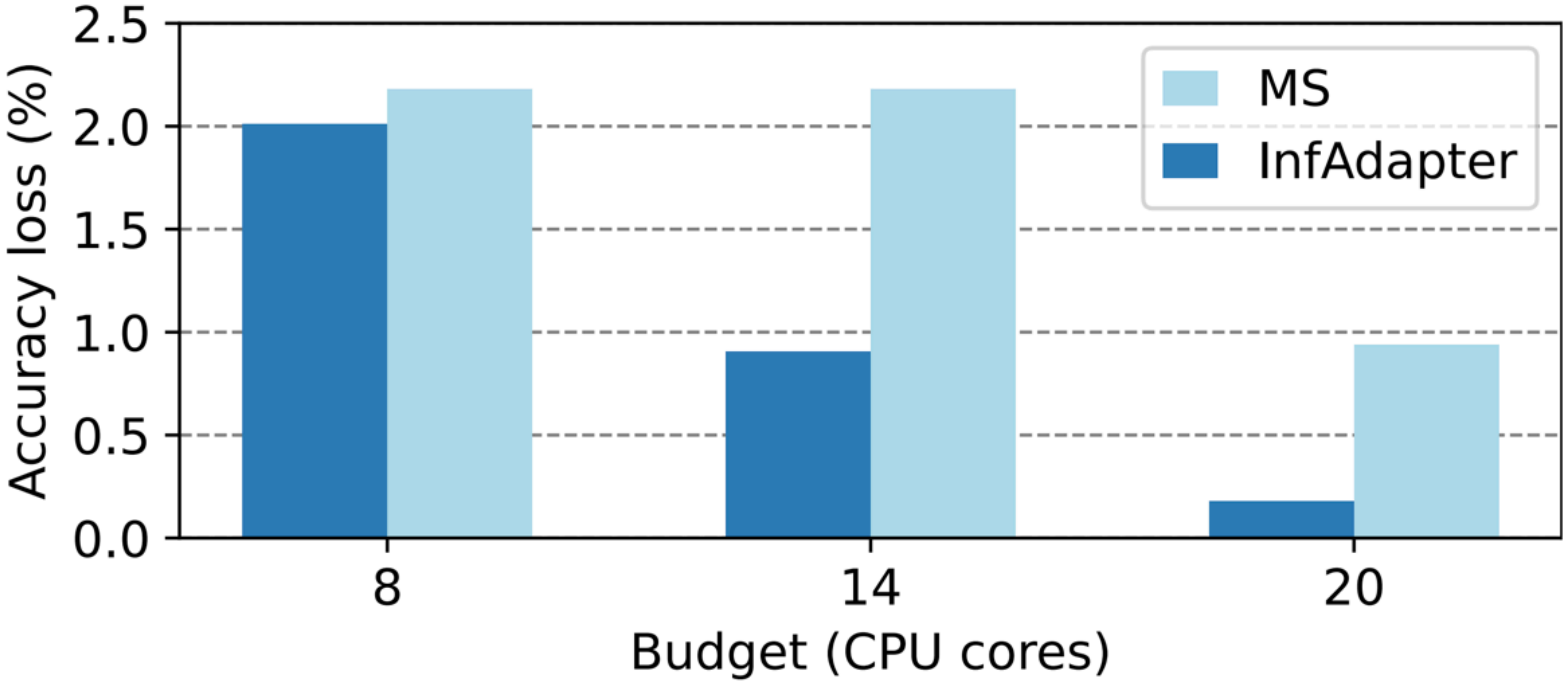
InfAdapter: Why?

Different throughputs with different model variants

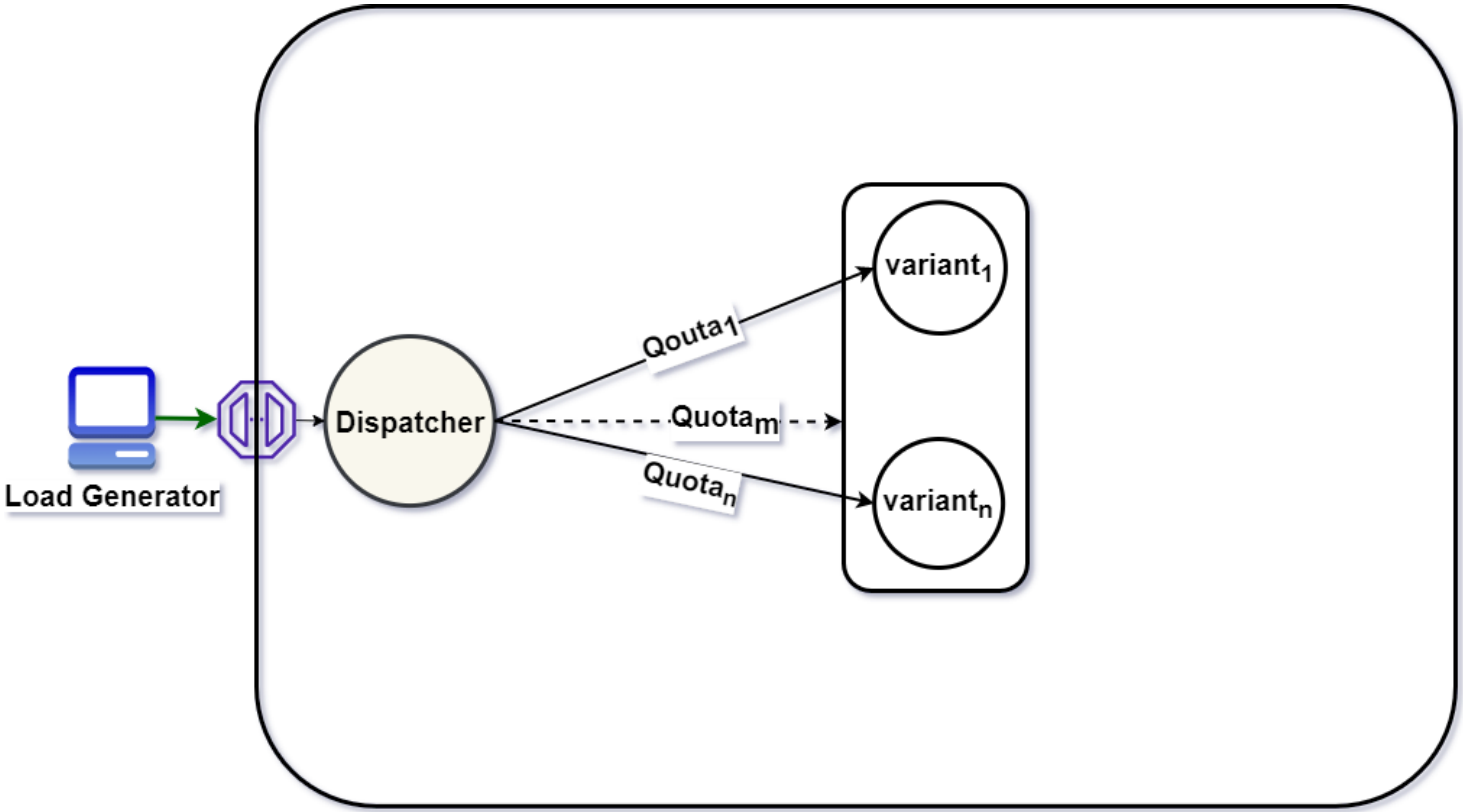


InfAdapter: Why?

Higher average accuracy by using multiple model variants



InfAdapter: How?



Selecting a **subset of model variants**, each having its size meeting latency requirements for the predicted workload while **maximizing accuracy and minimizing resource cost**

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

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

Supporting incoming workload

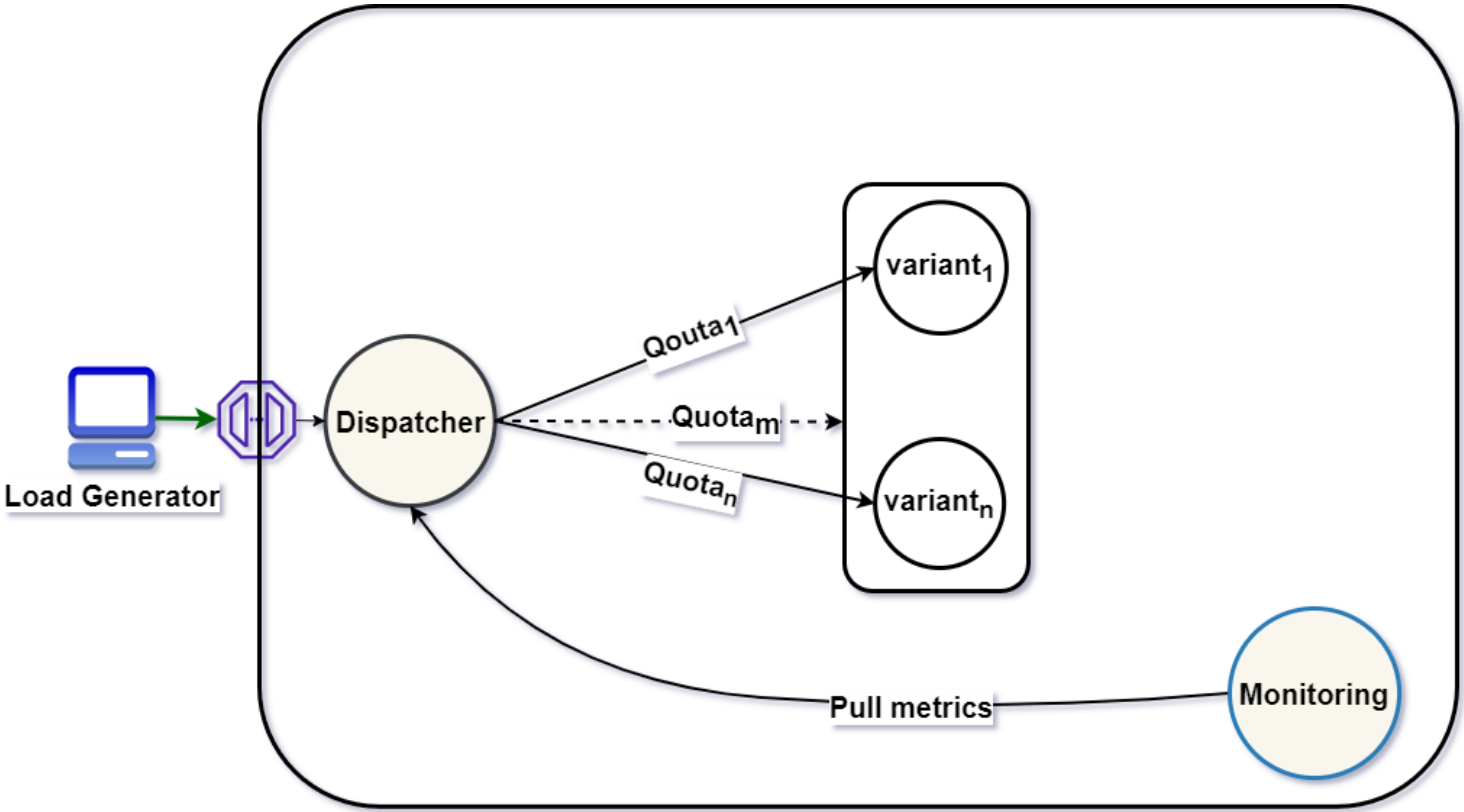
InfAdapter: Formulation

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

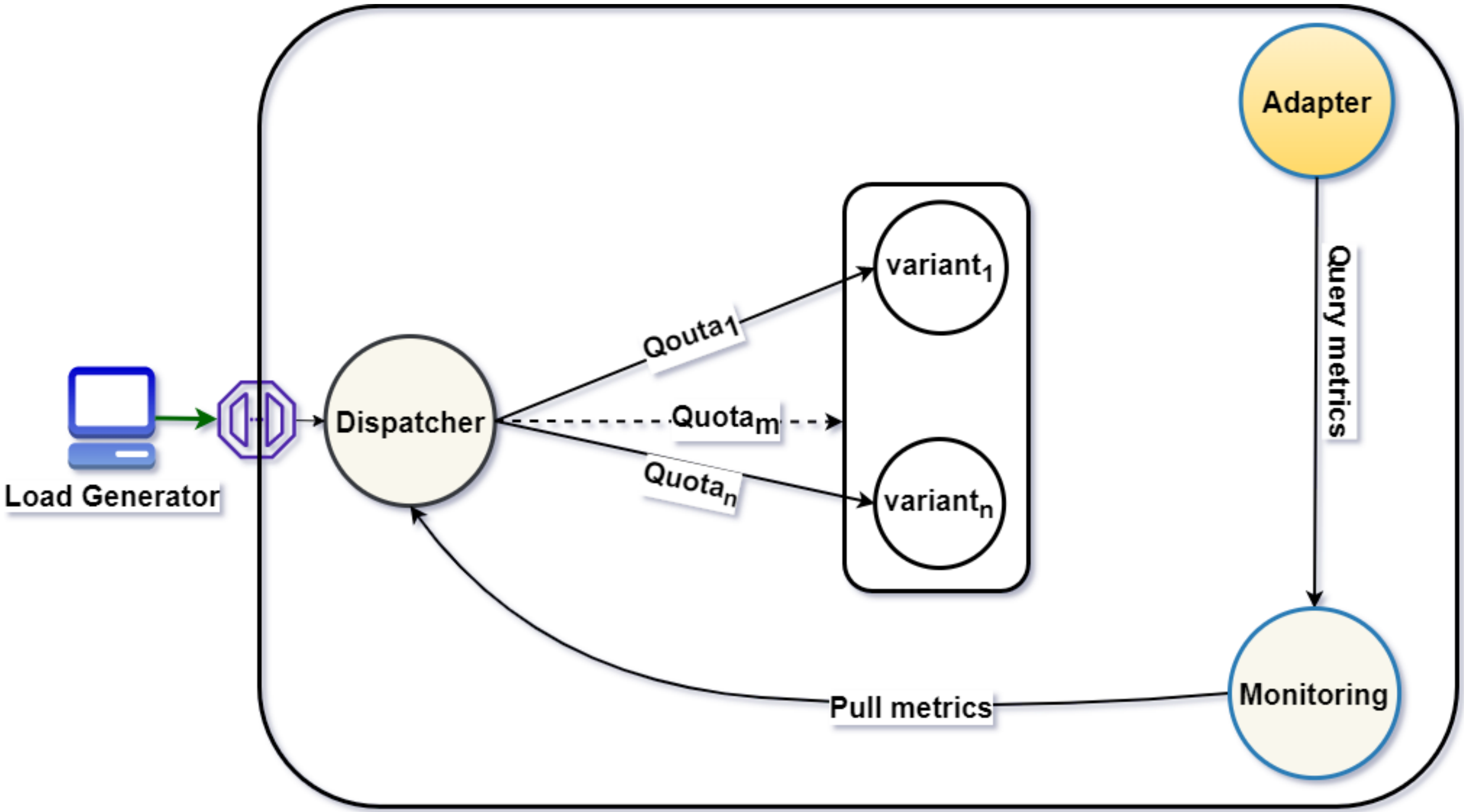
Supporting incoming workload

Guaranteeing end-to-end latency

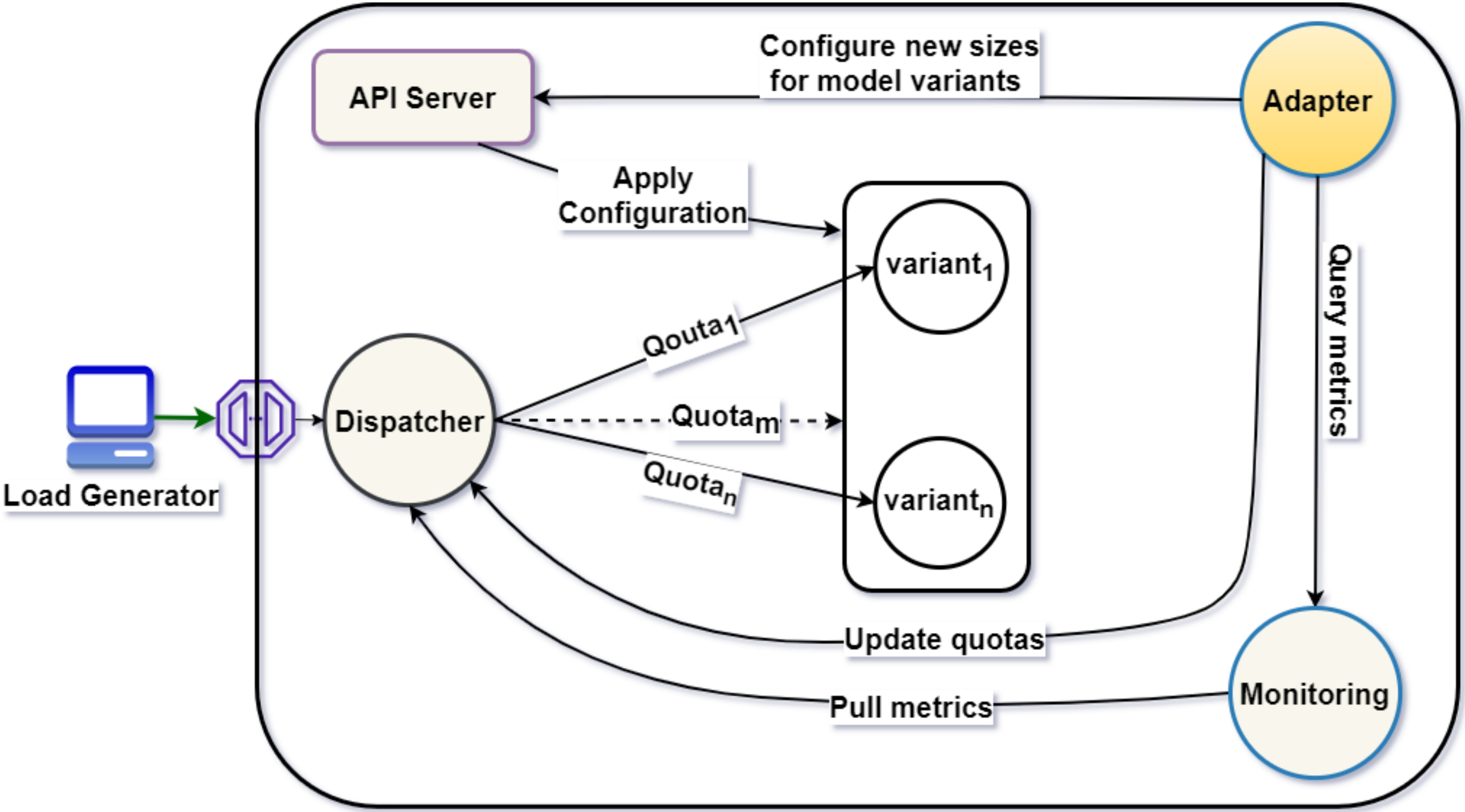
InfAdapter: Design



InfAdapter: Design



InfAdapter: Design



InfAdapter: Experimental evaluation setup

Workload: **Twitter-trace** sample (2022-08)

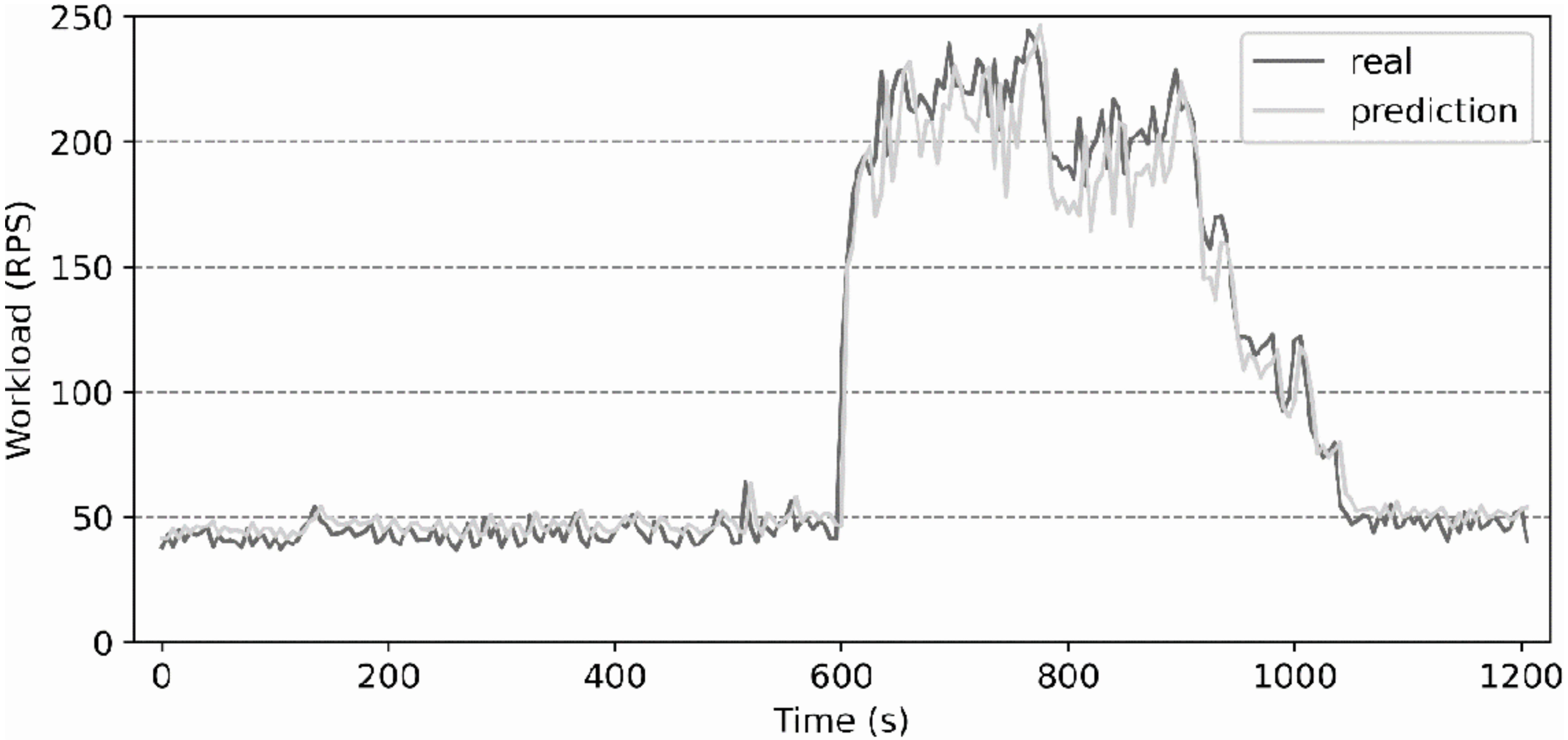
Baselines: **Kubernetes VPA** and **Model-Switching**

Used models: Resnet18, Resnet34, Resnet50, Resnet101, Resnet152

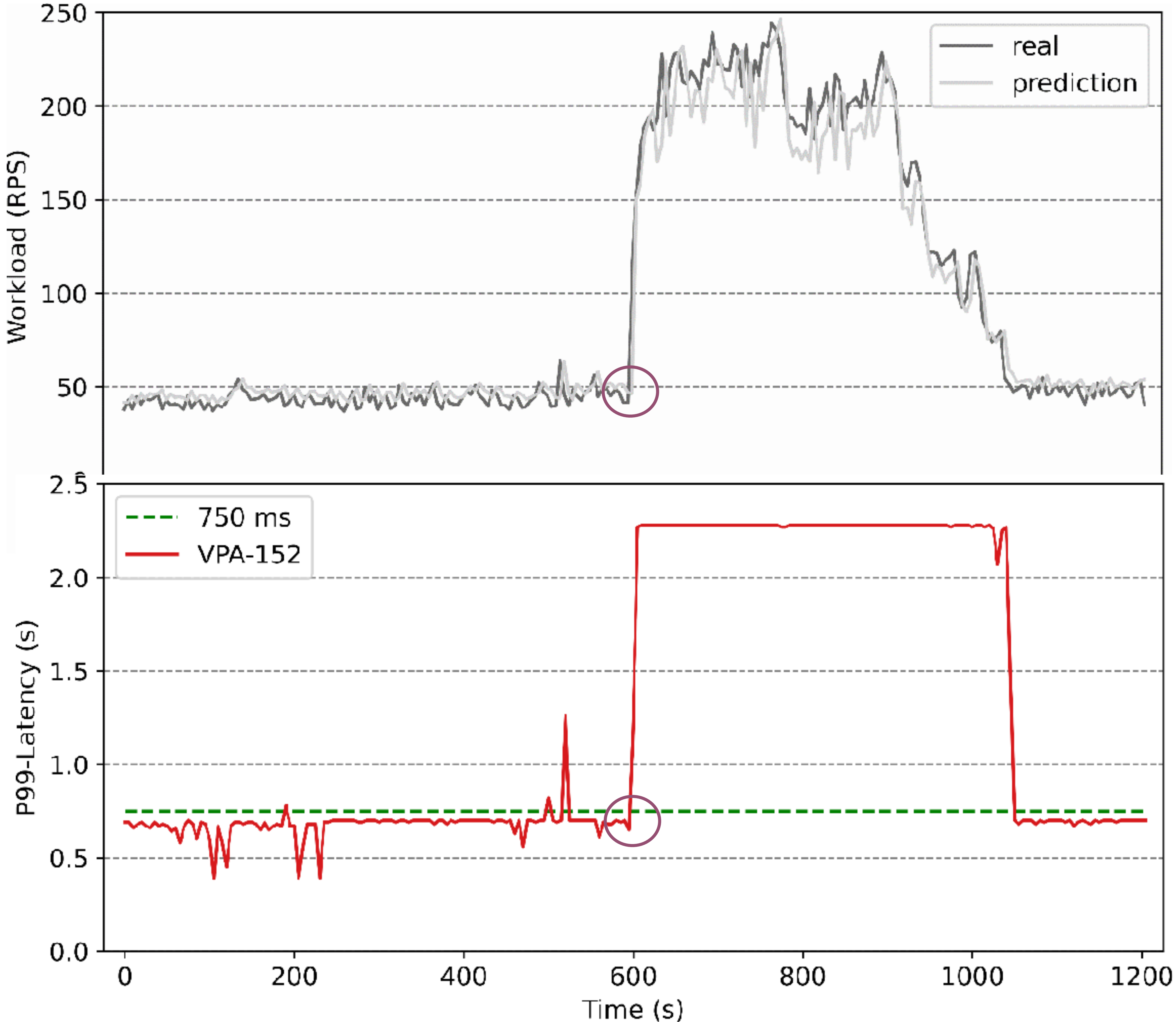
Interval adaptation: 30 seconds

Kubernetes cluster: 48 Cores, 192 GiB RAM

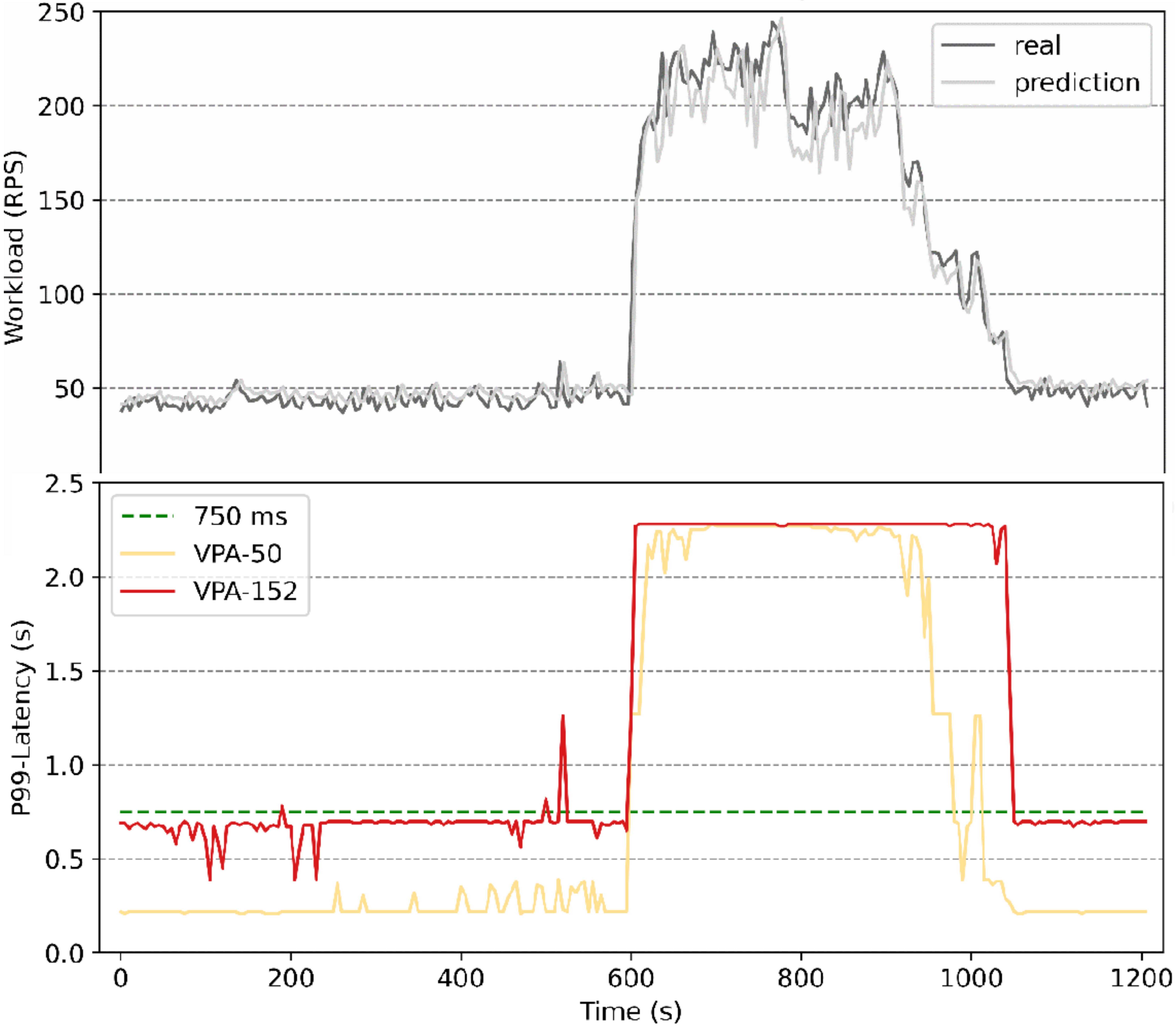
Workload Pattern



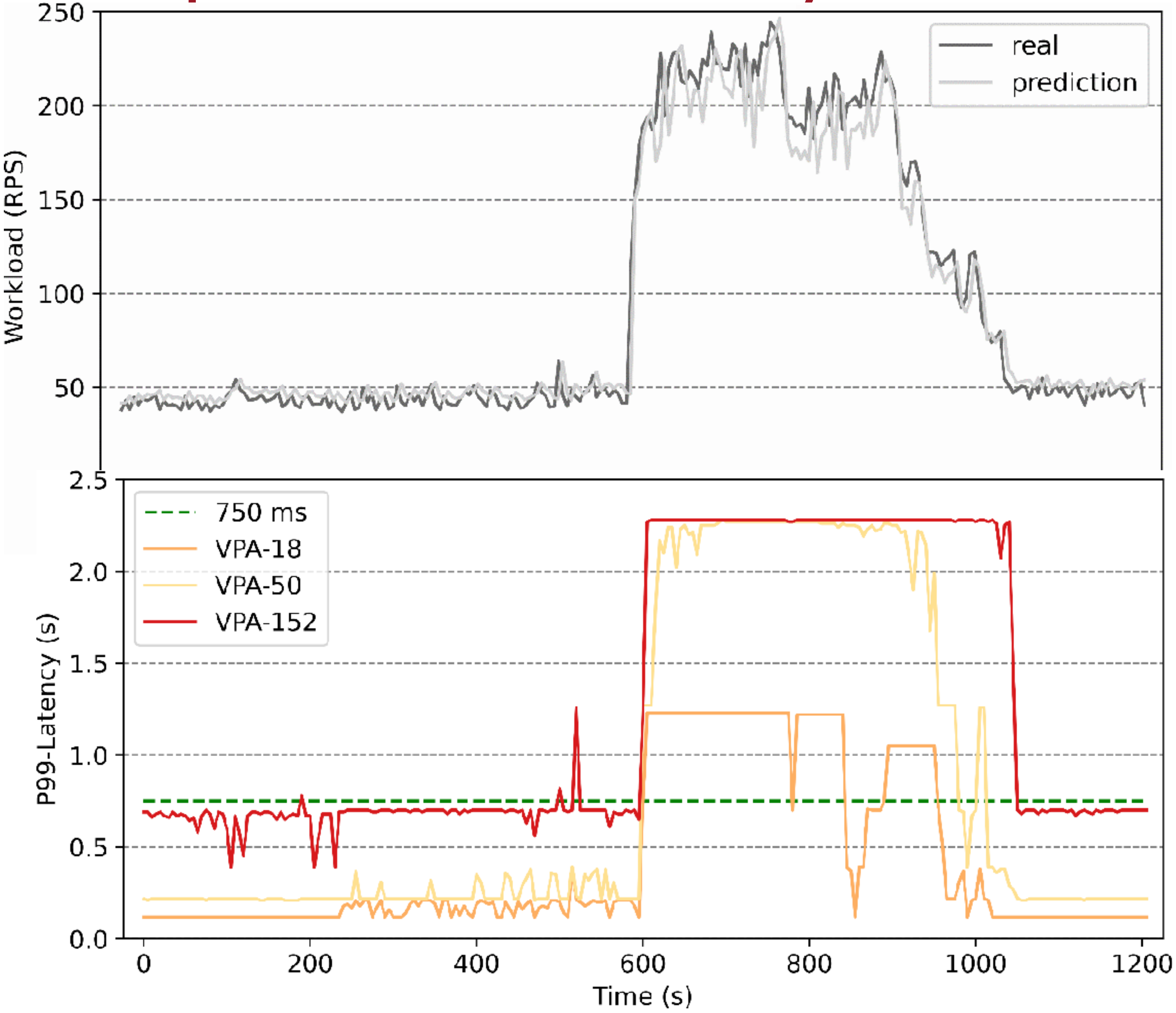
InfAdapter: P99-Latency evaluation



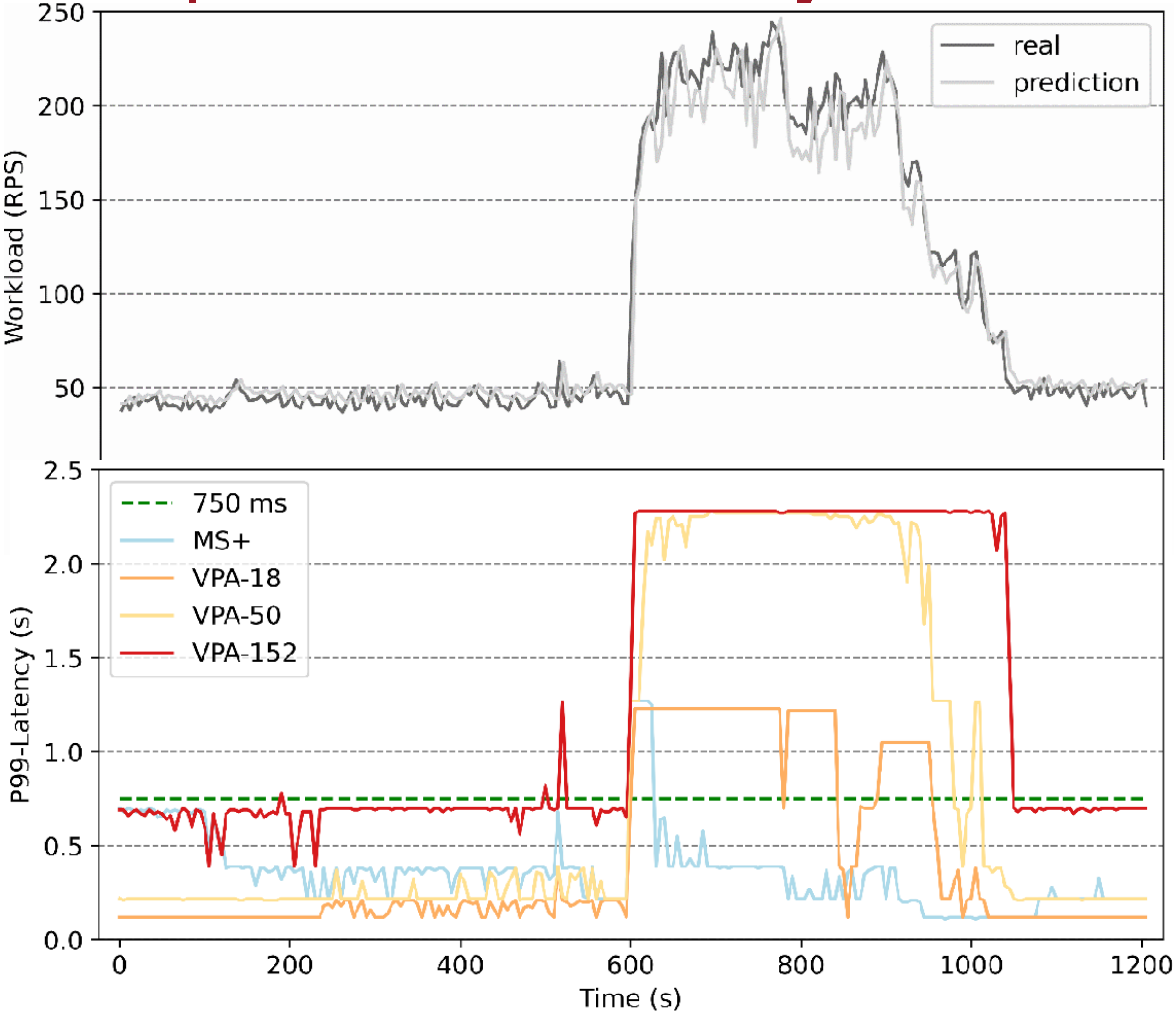
InfAdapter: P99-Latency evaluation



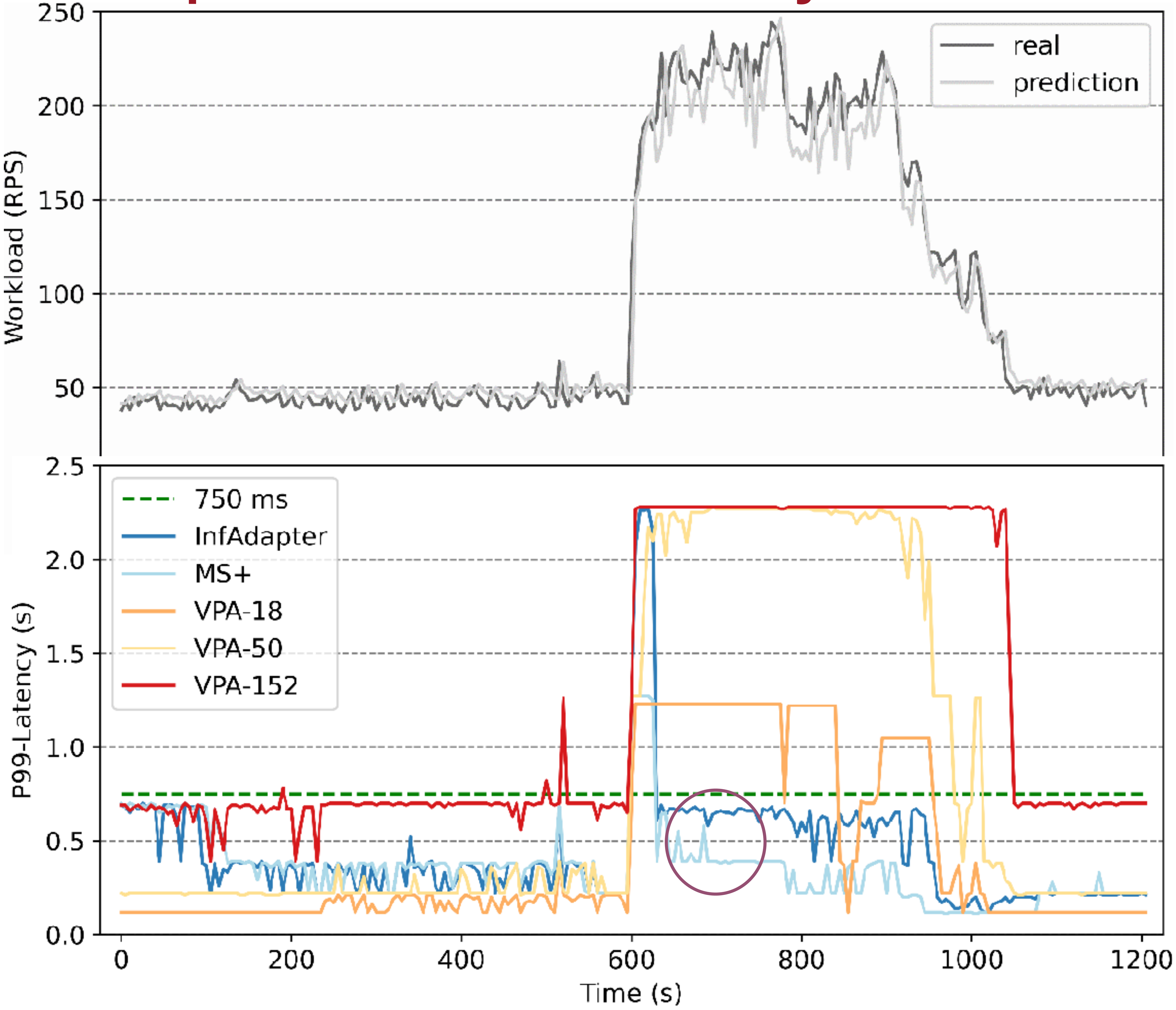
InfAdapter: P99-Latency evaluation



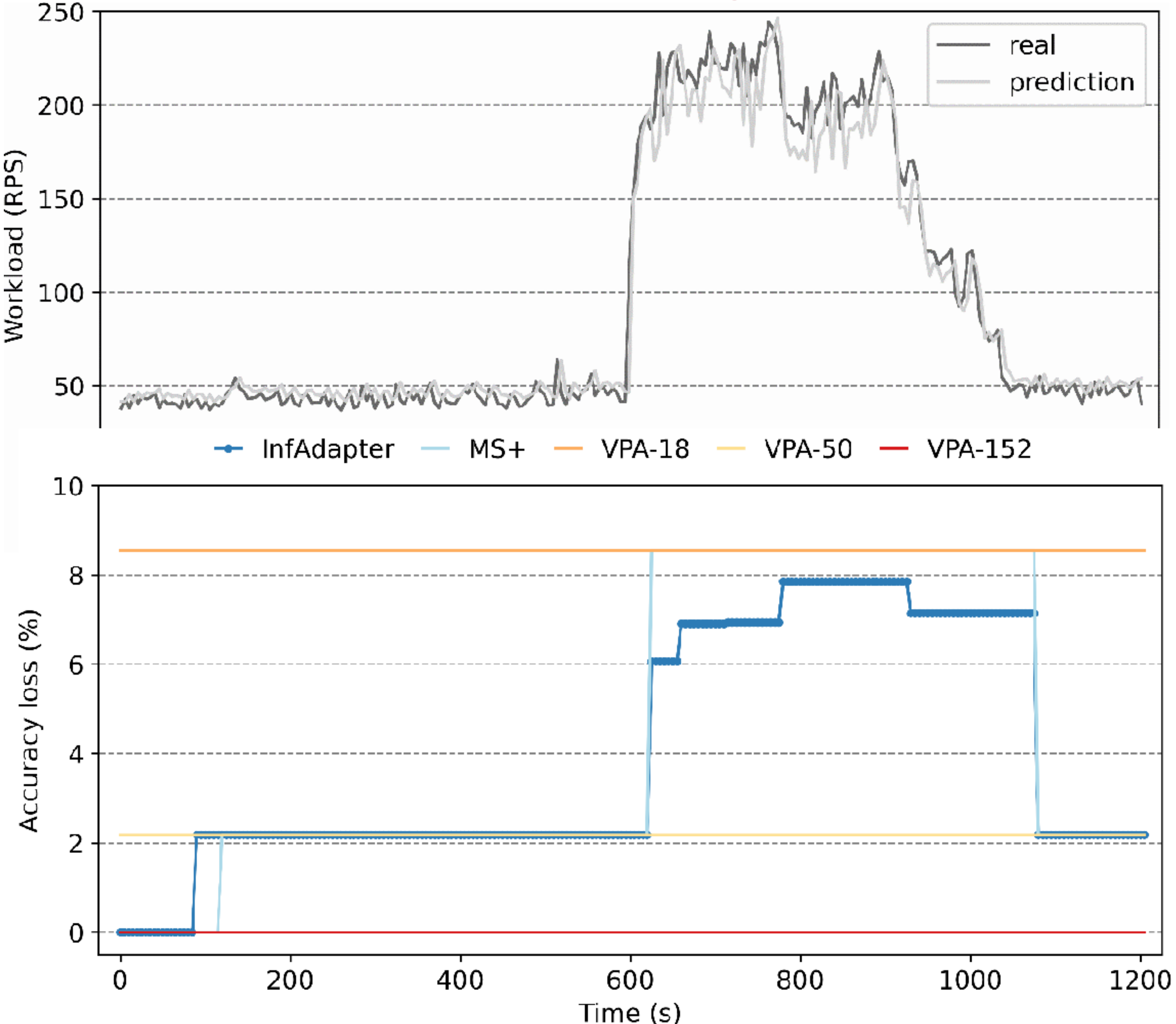
InfAdapter: P99-Latency evaluation



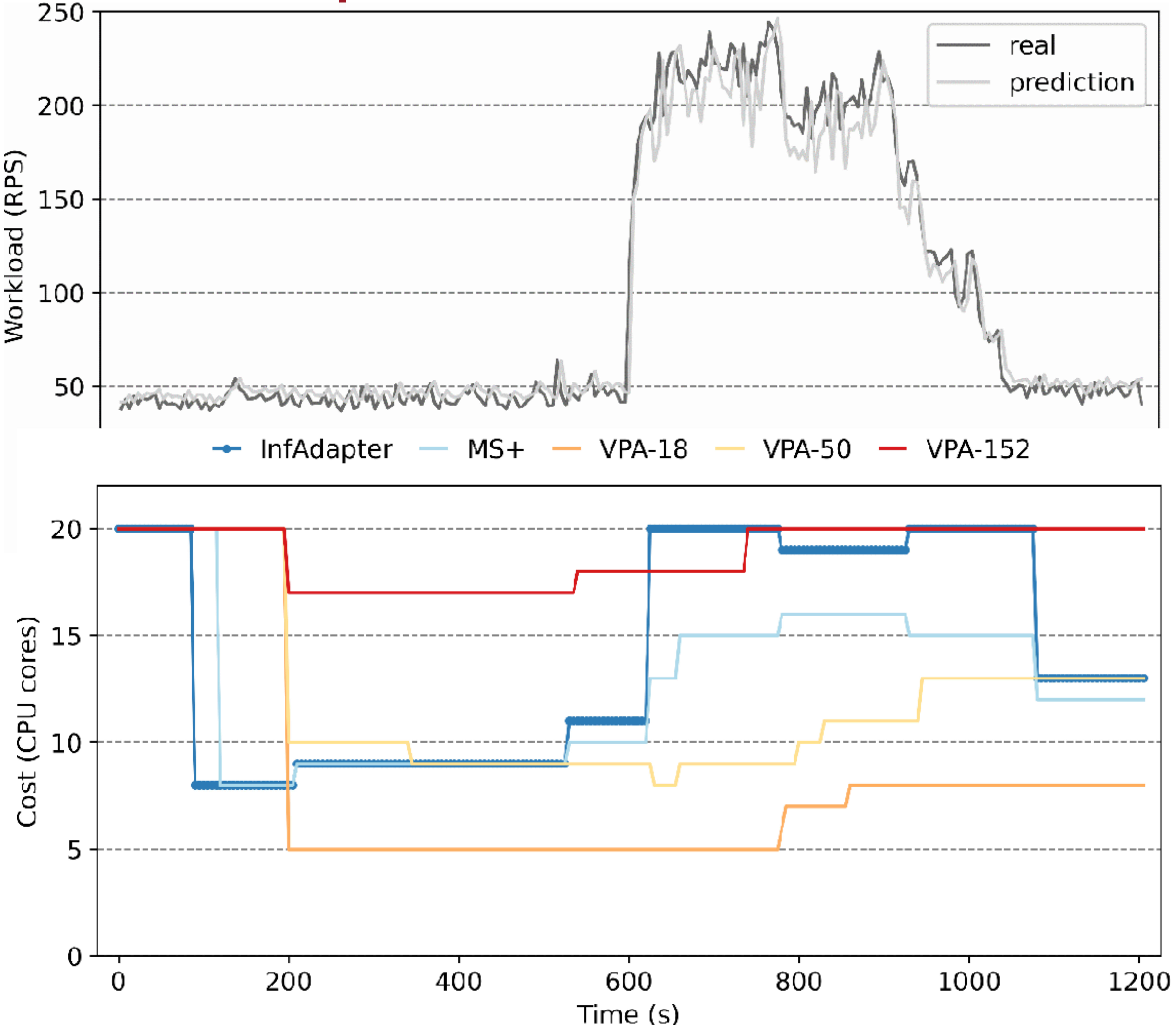
InfAdapter: P99-Latency evaluation



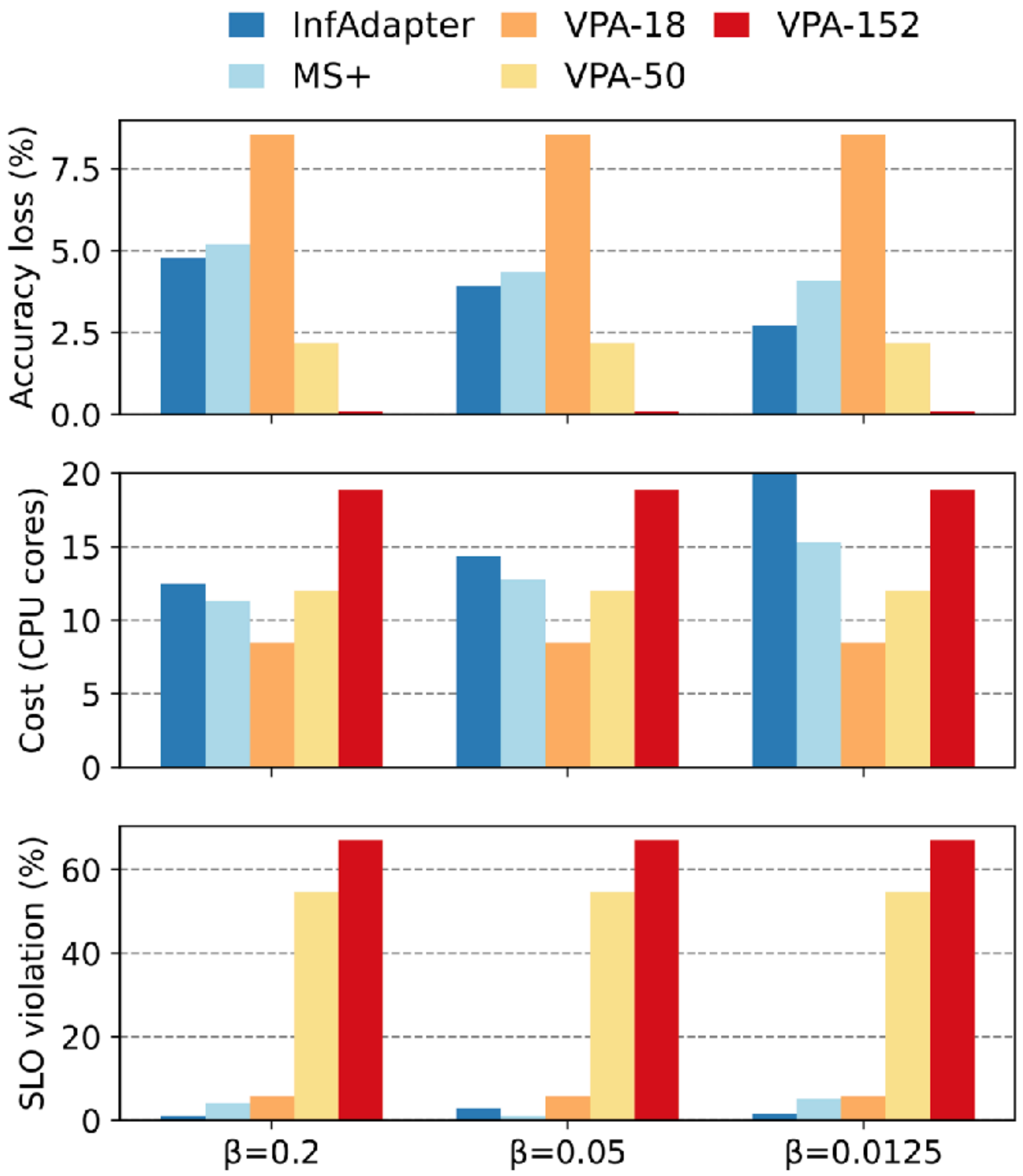
InfAdapter: Accuracy evaluation



InfAdapter: Cost evaluation



InfAdapter: Tradeoff Space



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

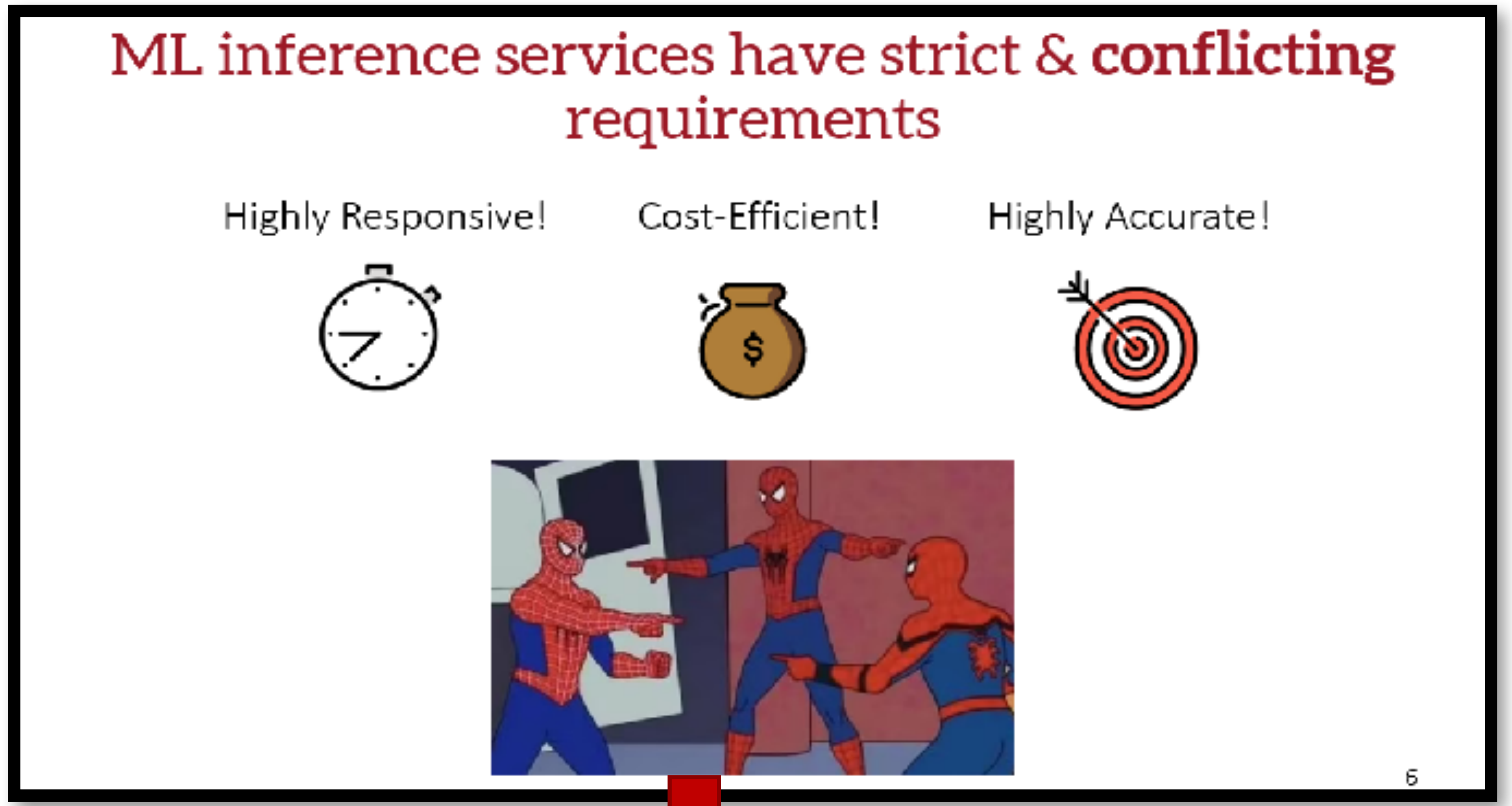
Inference Serving Systems should consider accuracy, latency, and cost at the same time.

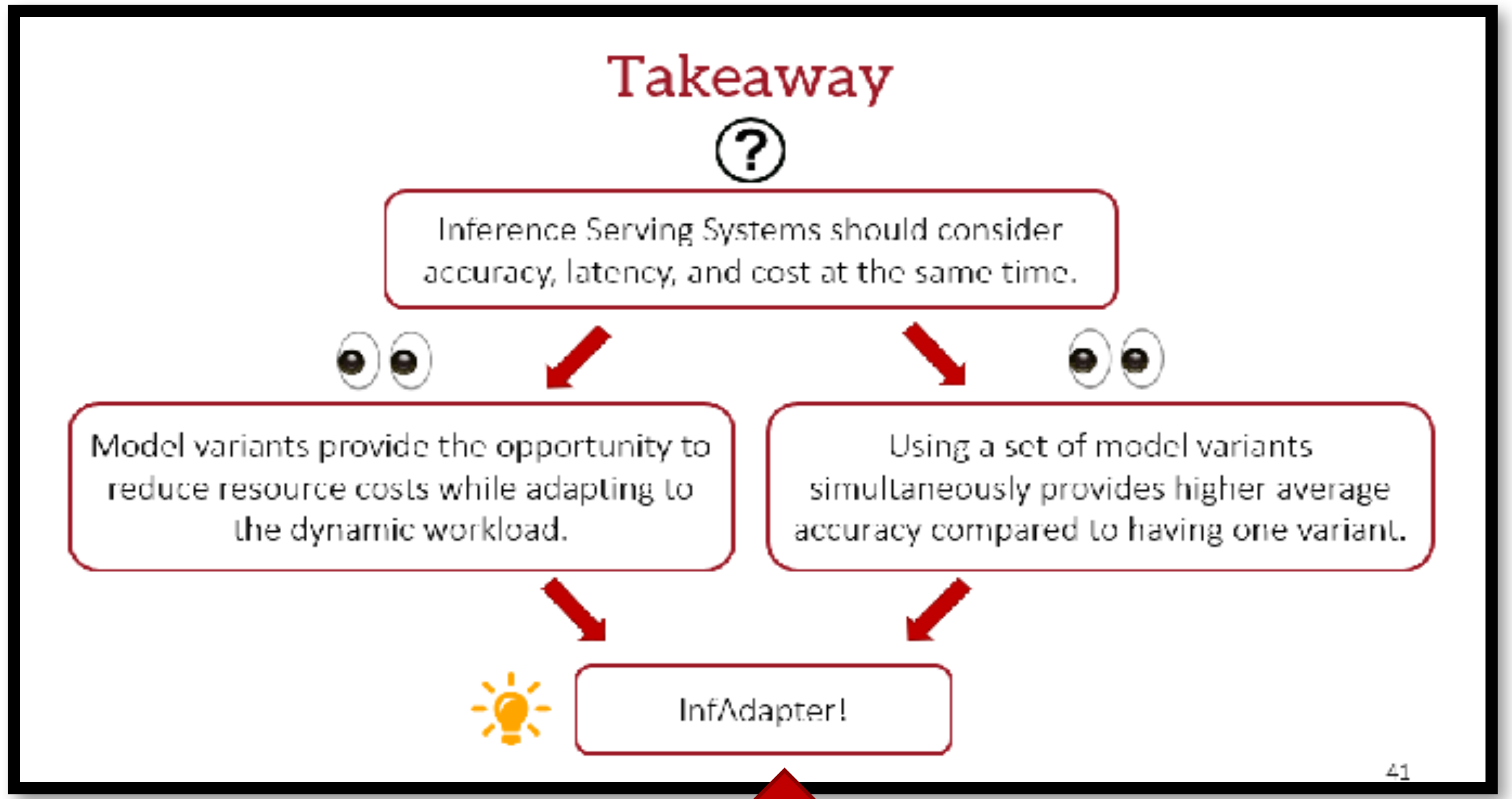
Model variants provide the opportunity to reduce resource costs while adapting to the dynamic workload.

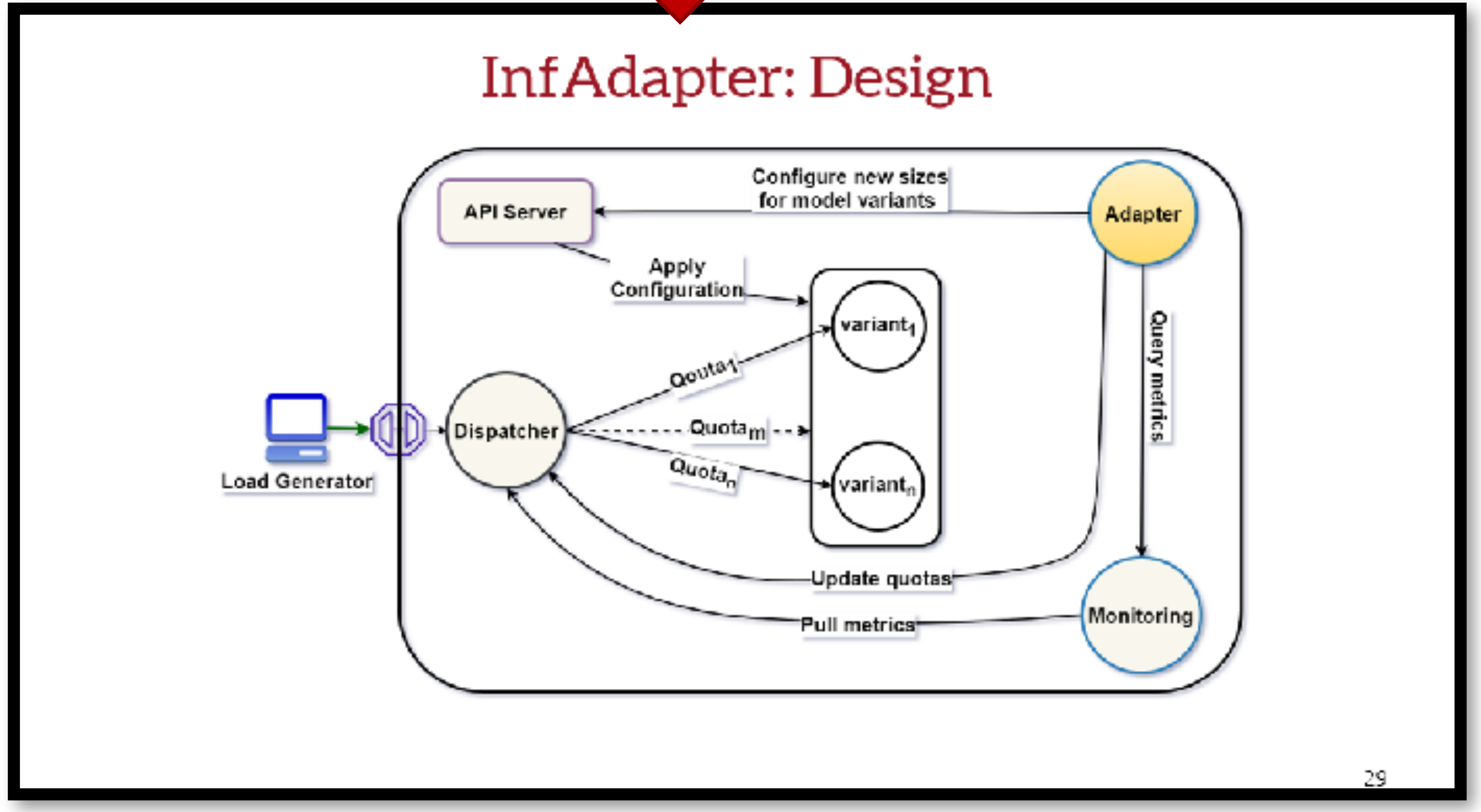
Using a set of model variants simultaneously provides higher average accuracy compared to having one variant.

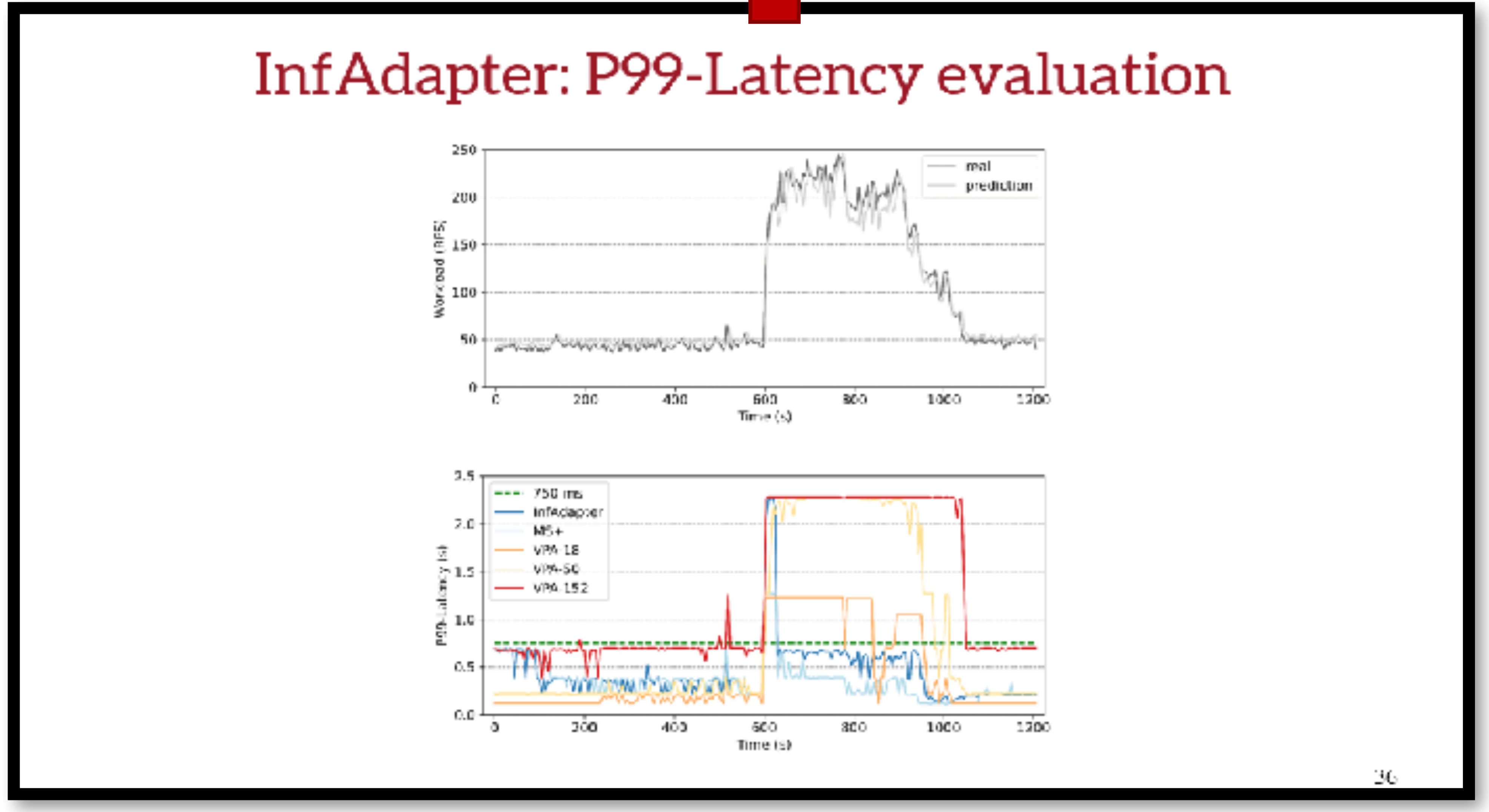
InfAdapter!

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









EuroMLSys

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

Mehran Salmani[*], Saeid Ghafouri[§‡], Alireza Sanaee[§], Kamran Razavi[†], Max Mühlhäuser[†], Joseph Doyle[§], Pooyan Jamshidi[‡], Mohsen Sharifi[*]

Iran University of Science and Technology[*], Queen Mary University of London[§], Technical University of Darmstadt[†], University of South Carolina[‡]

InfAdapter [2023]: Autoscaling for ML **Model Inference**

JSys Journal of Systems Research Volume 4, Issue 1, April 2024

[SOLUTION] IPA: INFERENCE PIPELINE ADAPTATION TO ACHIEVE HIGH ACCURACY AND COST-EFFICIENCY

Saeid Ghafouri
University of South Carolina & Queen Mary University of London

Kamran Razavi
Technical University of Darmstadt

Mehran Salmani
Technical University of Ilmenau

Alireza Sanaee
Queen Mary University of London

Tania Lorido Botran
Roblox

Lin Wang
Paderborn University

Joseph Doyle
Queen Mary University of London

Pooyan Jamshidi
University of South Carolina

IPA [2024]: Autoscaling for **ML Inference Pipeline**

EuroMLSys

Sponge: Inference Serving with Dynamic SLOs Using In-Place Vertical Scaling

Kamran Razavi[*]
Technical University of Darmstadt

Saeid Ghafouri[*]
Queen Mary University of London

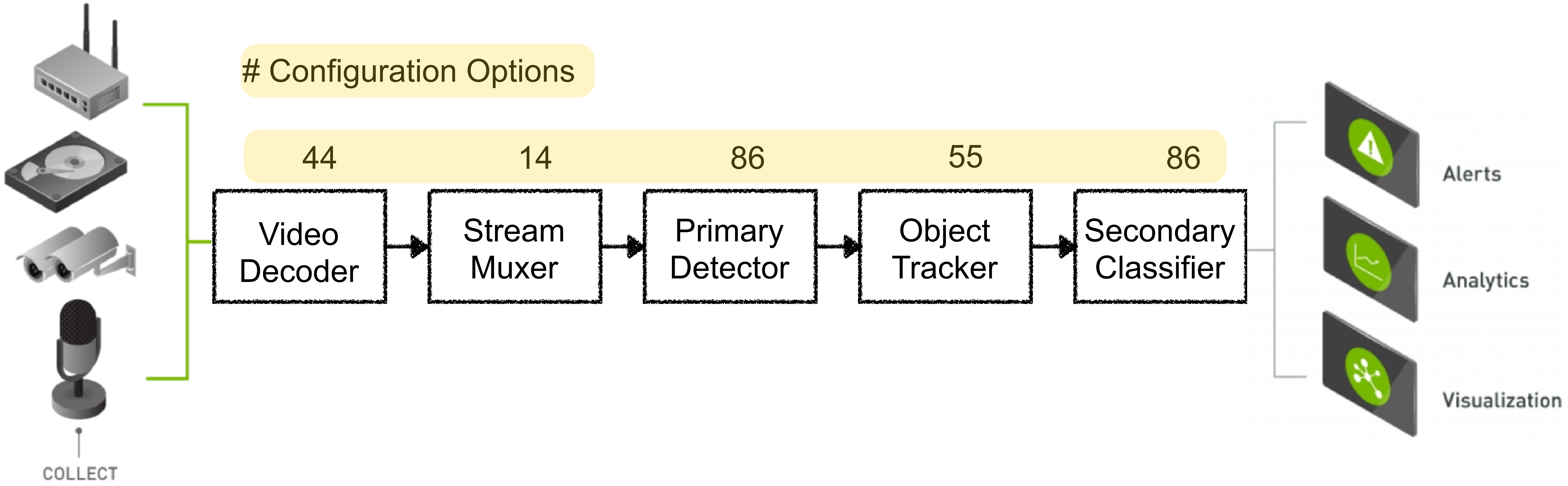
Max Mühlhäuser
Technical University of Darmstadt

Pooyan Jamshidi
University of South Carolina

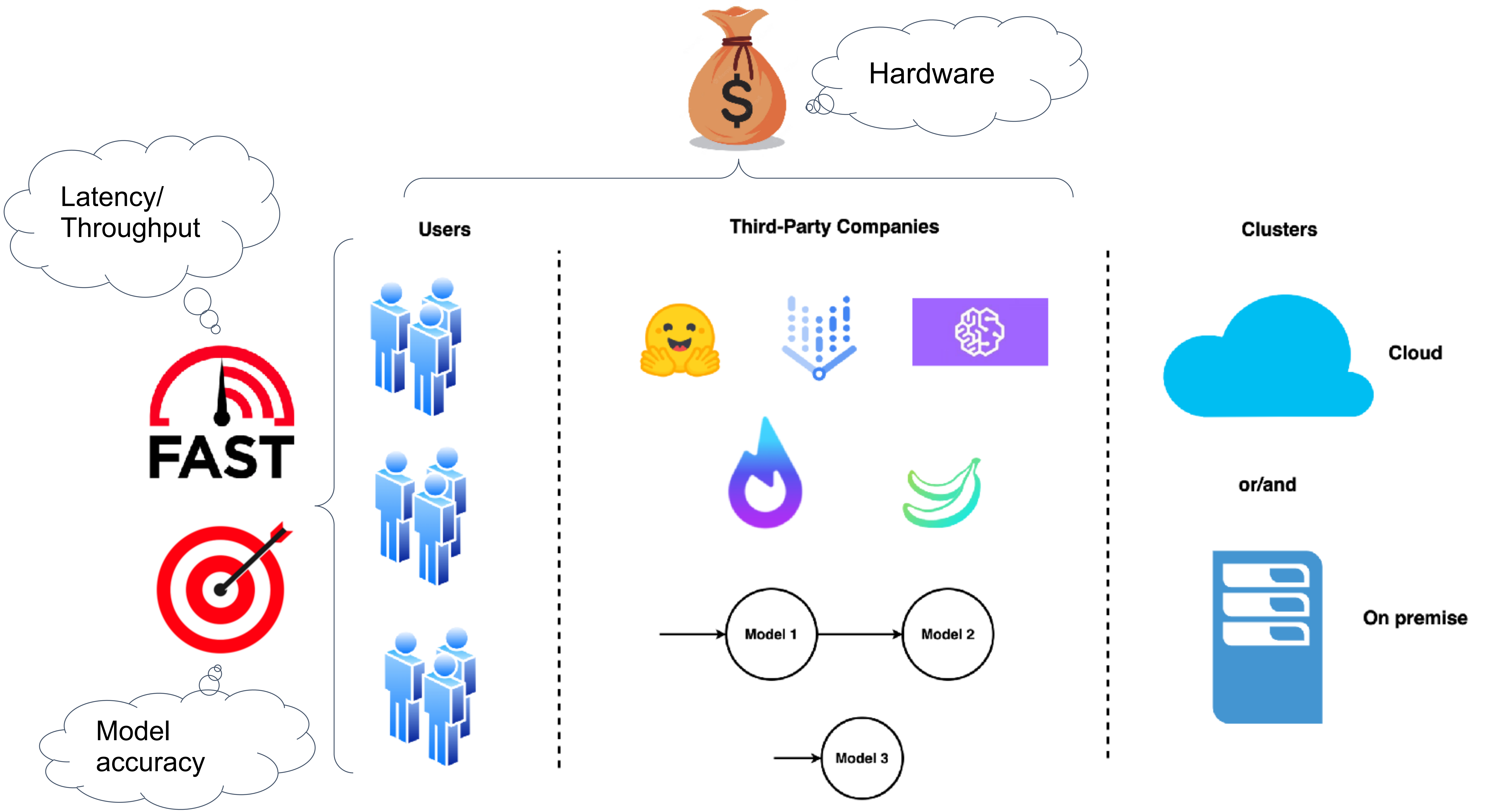
Lin Wang
Paderborn University

Sponge [2024]: Autoscaling for ML Inference Pipeline with **Dynamic SLO**

Inference Pipeline



What should be characteristic of an inference pipeline?



What should be characteristic of an inference pipeline?

- **Scalability**: The pipeline should be able to handle large volumes of data and scale horizontally to accommodate increases in input size or request frequency.
- **Low Latency**: Inference should be fast, especially in real-time or near-real-time applications. The pipeline should minimize processing time to deliver quick predictions.
- **Reproducibility**: The pipeline should consistently produce the same results for the same input, ensuring that predictions are reproducible across different environments.
- **Robustness and Fault Tolerance**: The pipeline should be resilient to failures, with mechanisms to handle errors gracefully, such as retry logic, circuit breakers, or fallback models.

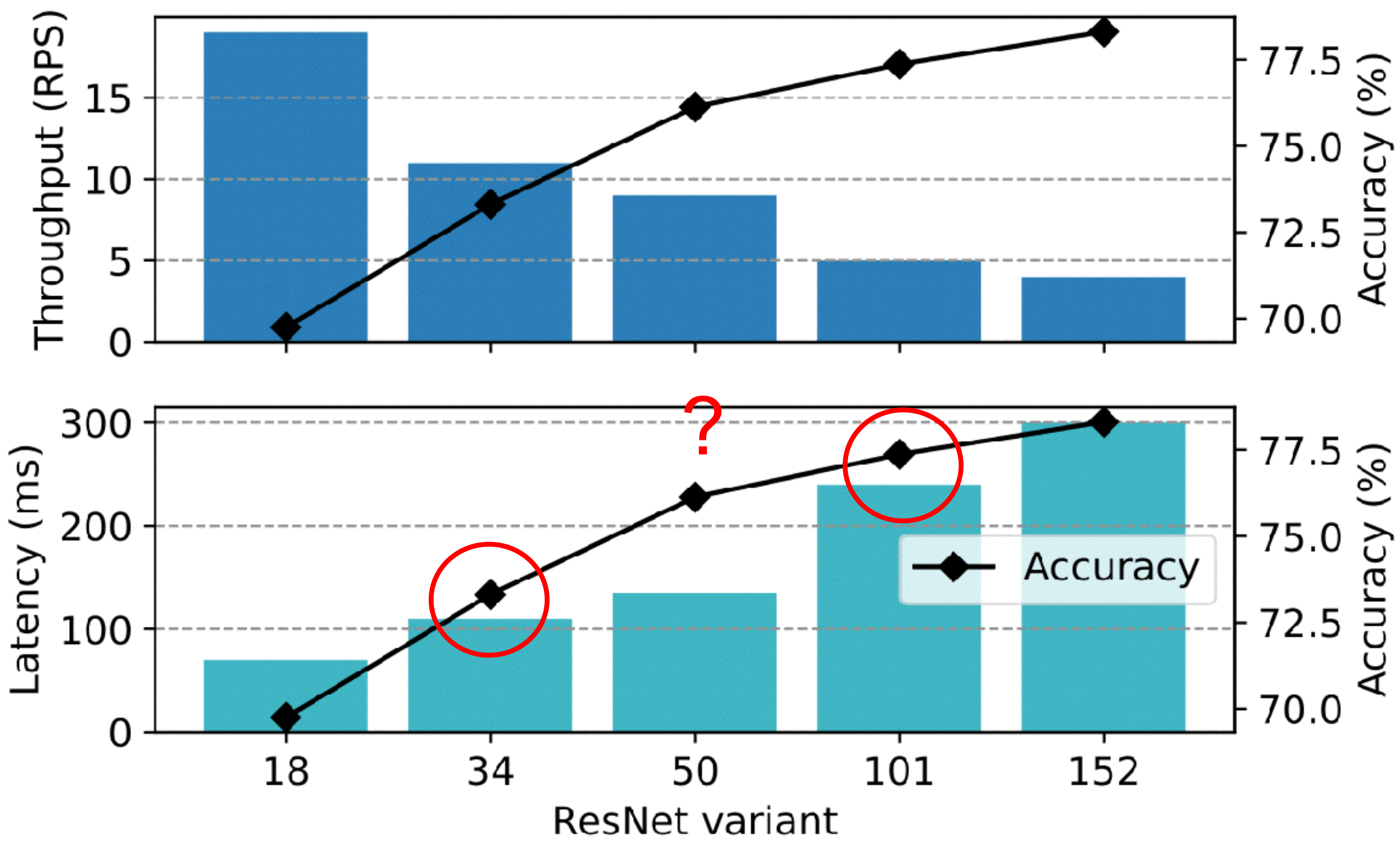
What should be characteristic of an inference pipeline?

- **Model Management**: The pipeline should allow for easy integration, updating, and switching of models. This includes versioning, rollback capabilities, and support for different model formats (e.g., TensorFlow, PyTorch, ONNX).

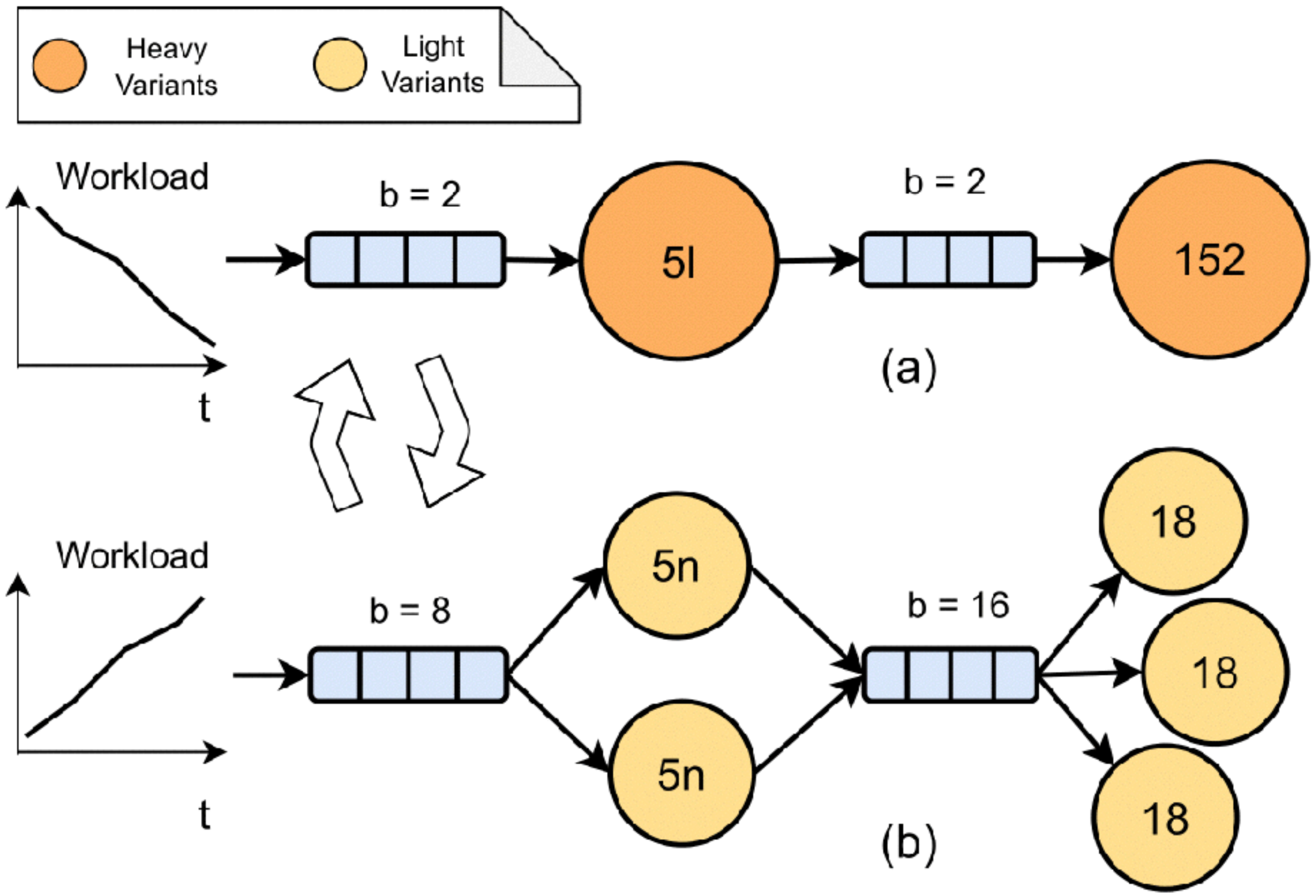
- **Resource Efficiency**: The pipeline should make optimal use of computational resources, balancing the trade-offs between cost, speed, and accuracy. This includes utilizing CPU/GPU resources effectively and managing memory usage.

- **Adaptability**: The pipeline should be flexible enough to adapt to new types of data, different model architectures, or changes in the environment (e.g., hardware upgrades or cloud migration).

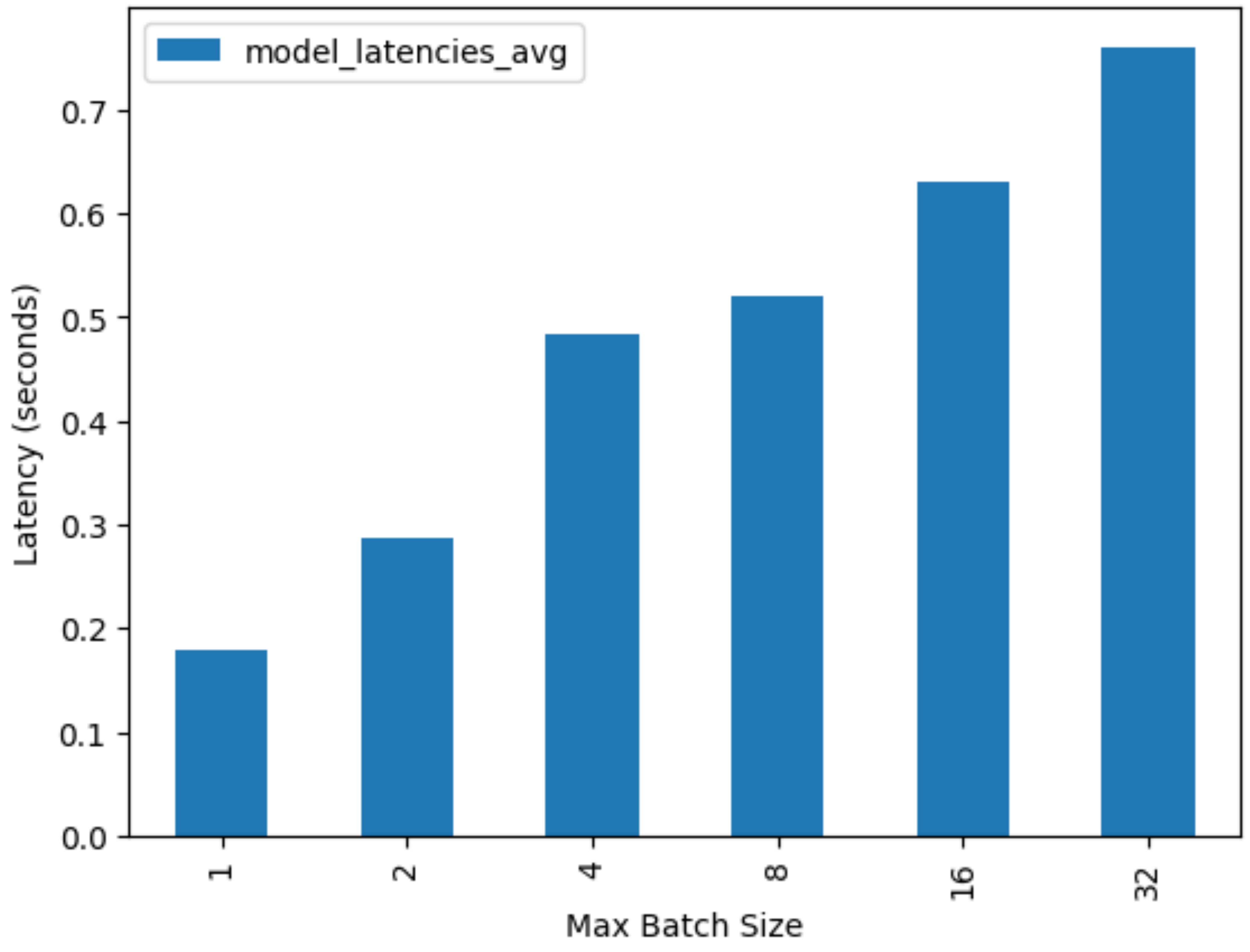
Is only scaling enough?

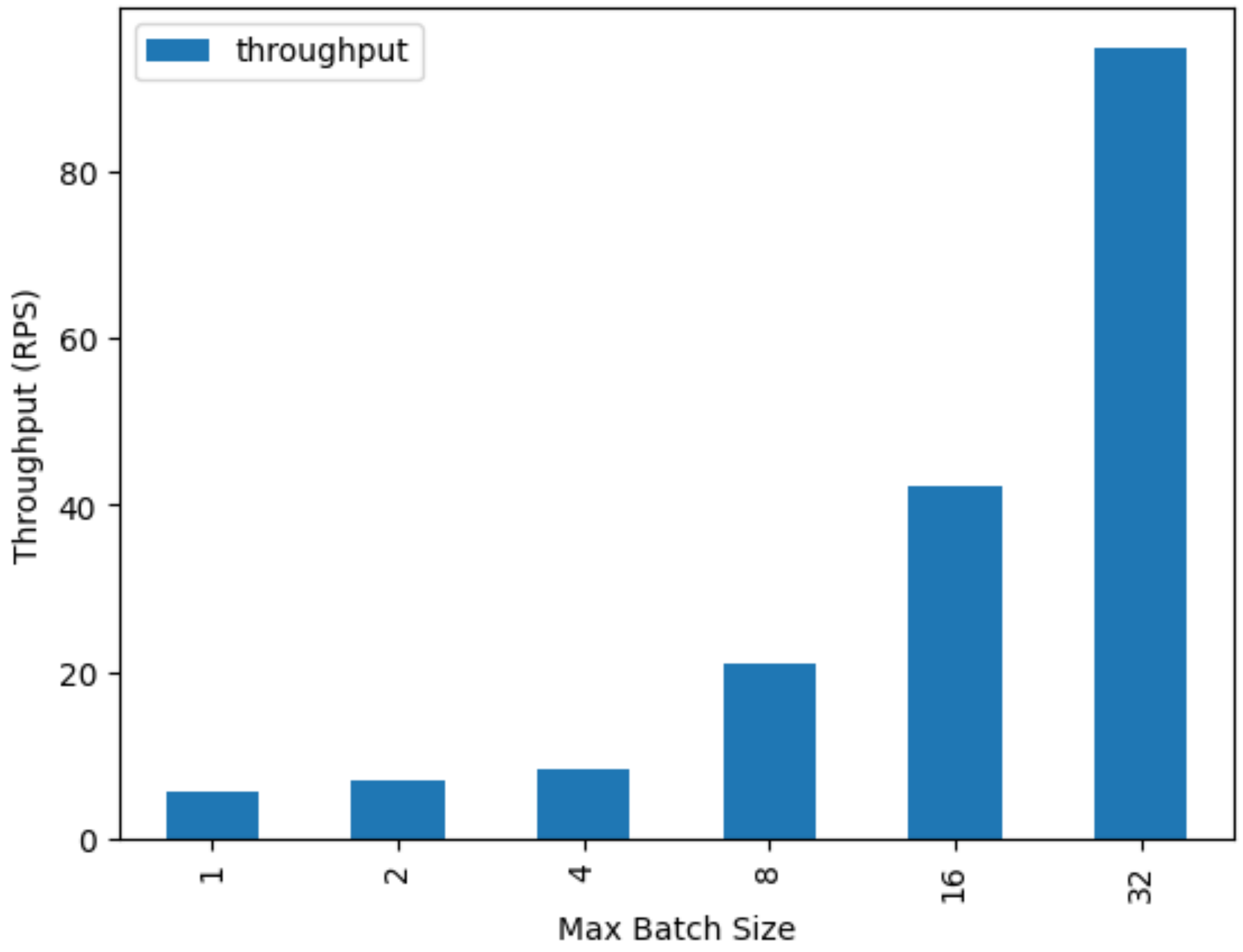


The Variabilities ML Pipelines



Effect of Batching

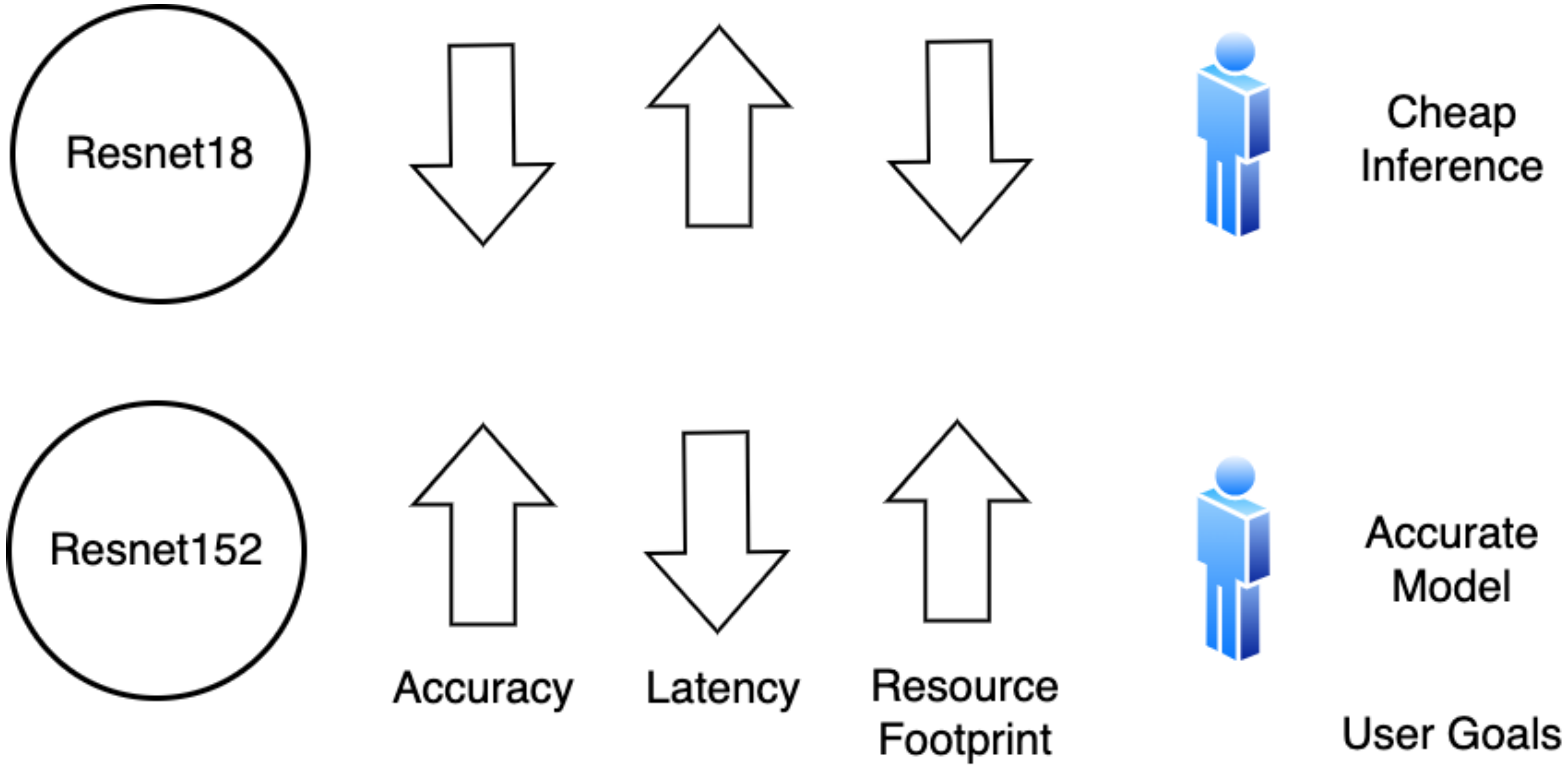




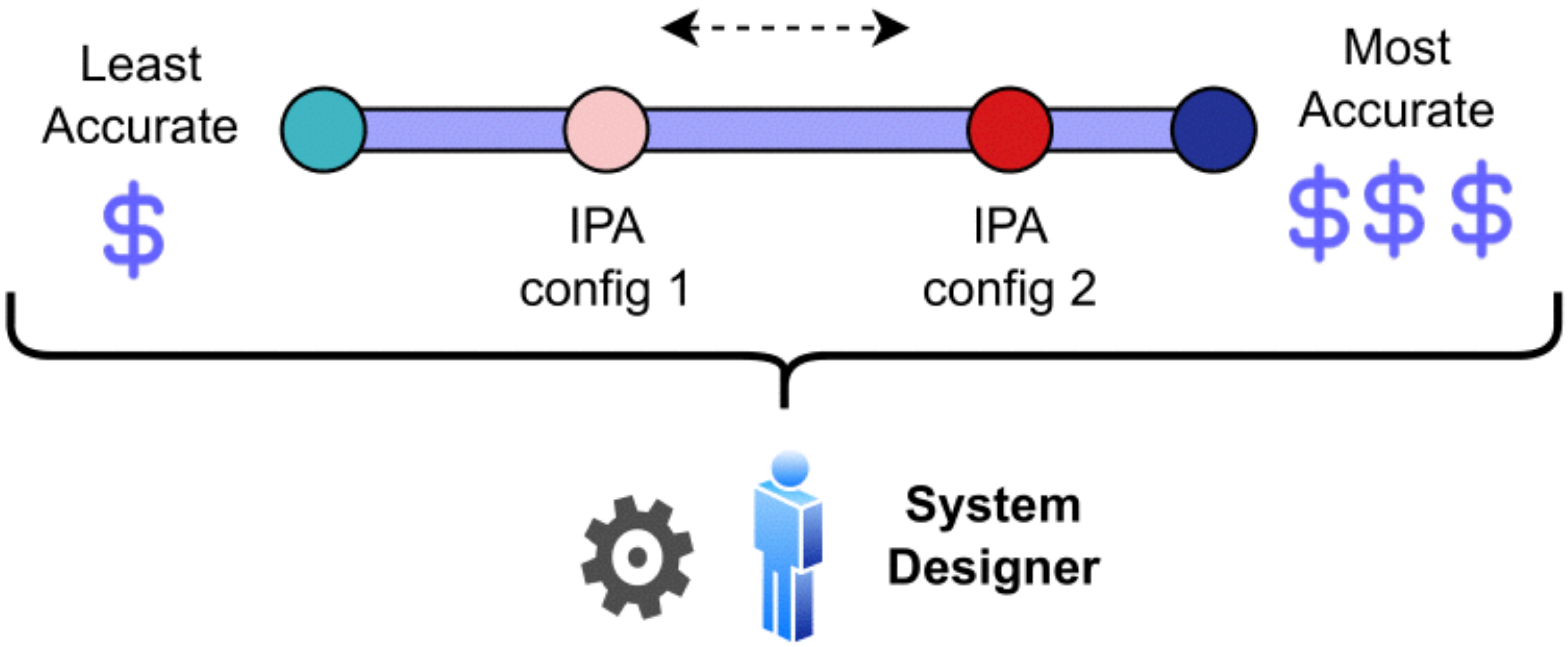
How to navigate the Accuracy/Latency trade-off space?

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

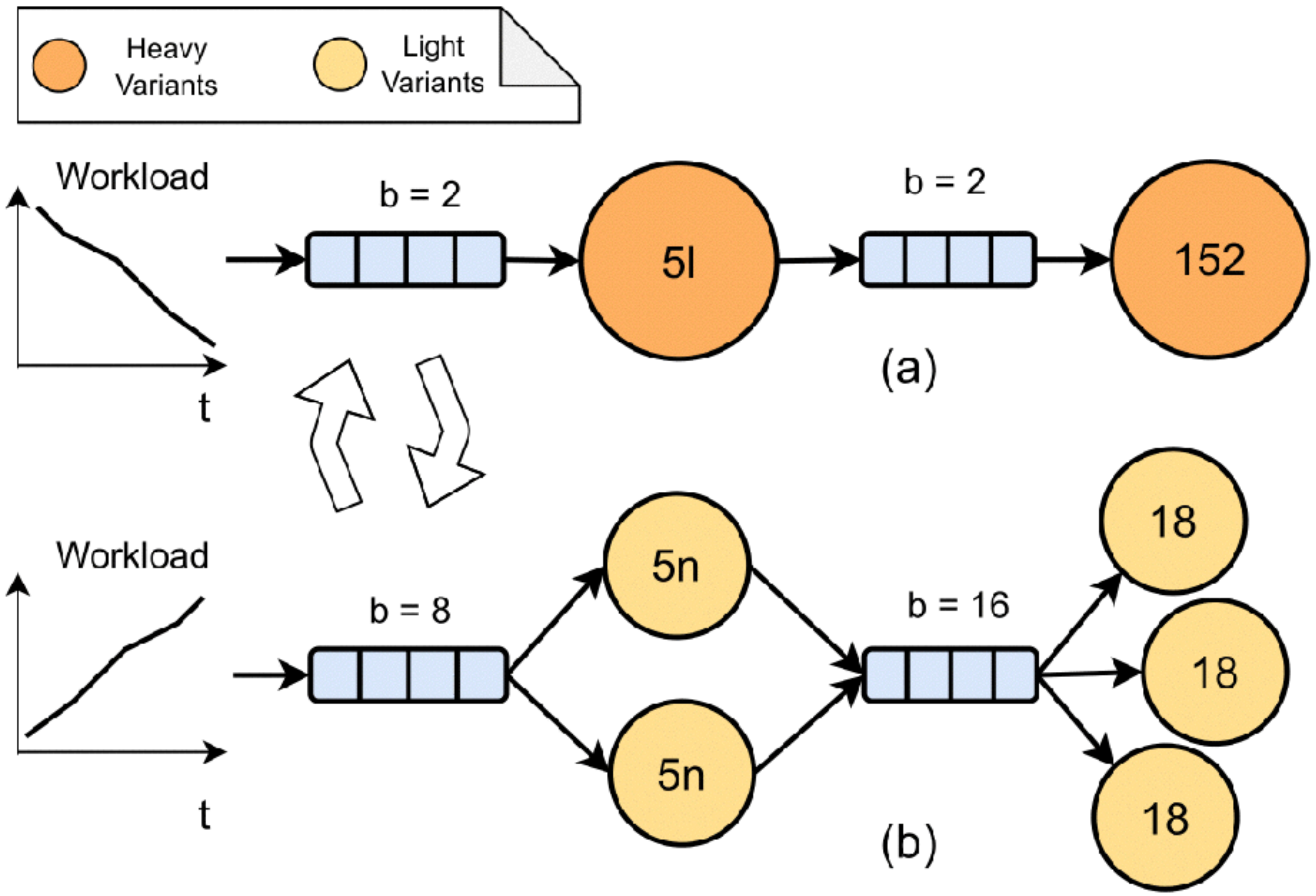
infaas



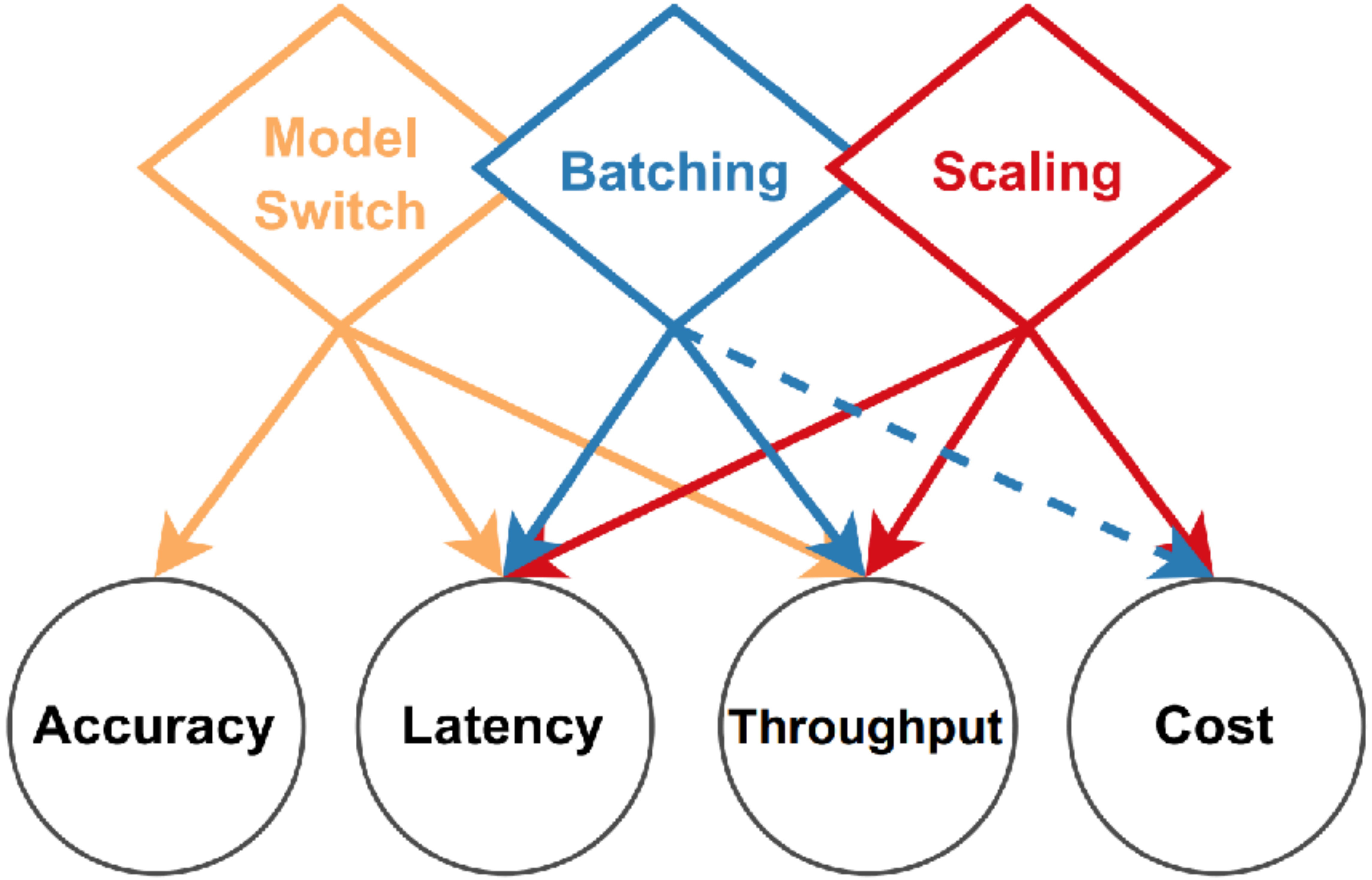
Goal: Providing a flexible inference pipeline

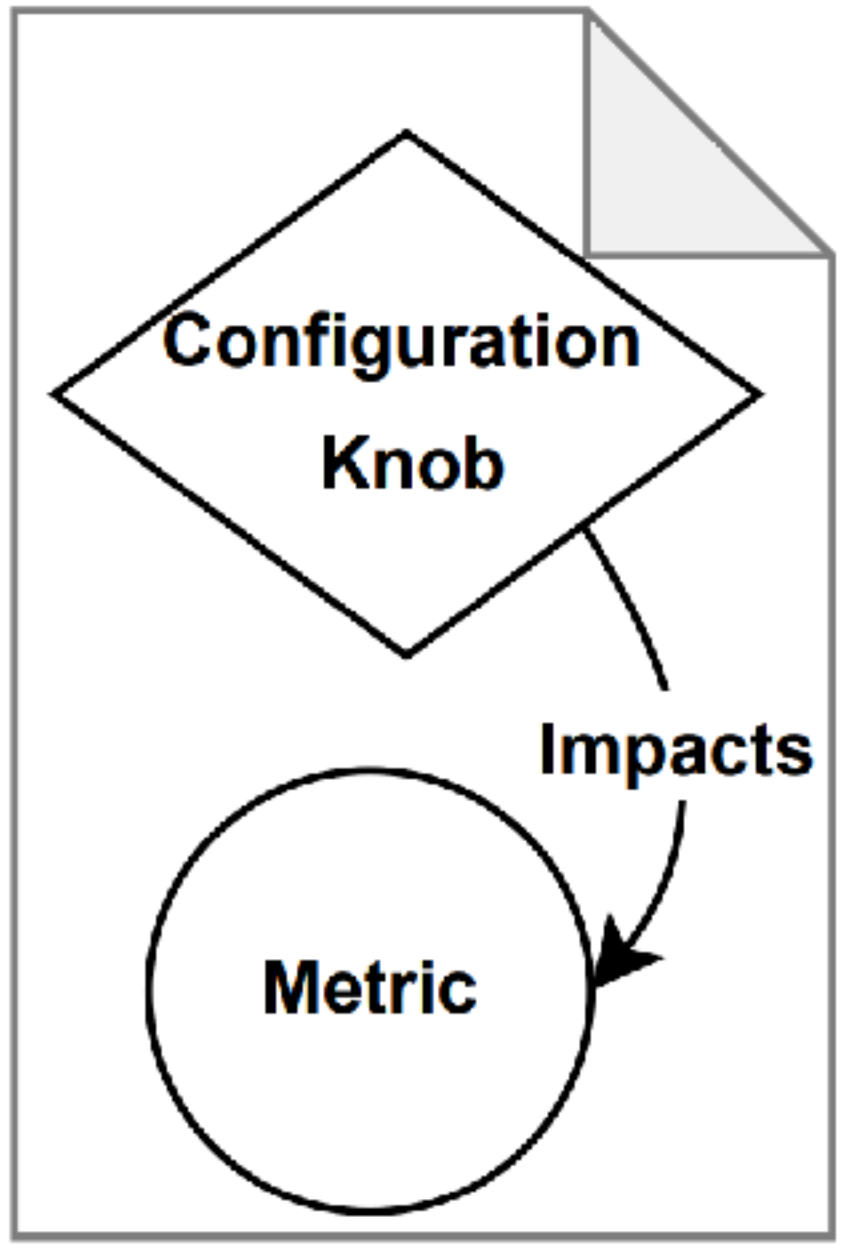


Snapshot of the System



Search Space





Problem Formulation

$$f(n,s,I) = \alpha \sum_{s \in P} \Big(\sum_{m \in M_s} a_{s,m} . I_{s,m} \Big)$$

Accuracy Objective

$$- \beta \sum_{s \in P} n_s . R_s$$

Resource Objective

$$- \delta \sum_{s \in P} b_s$$

Batch Control

Problem Formulation

Latency SLA

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

Throughput Constraint

One active model per node

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

Implementation and Experimental Setup

How to navigate Model Variants

kubernetes

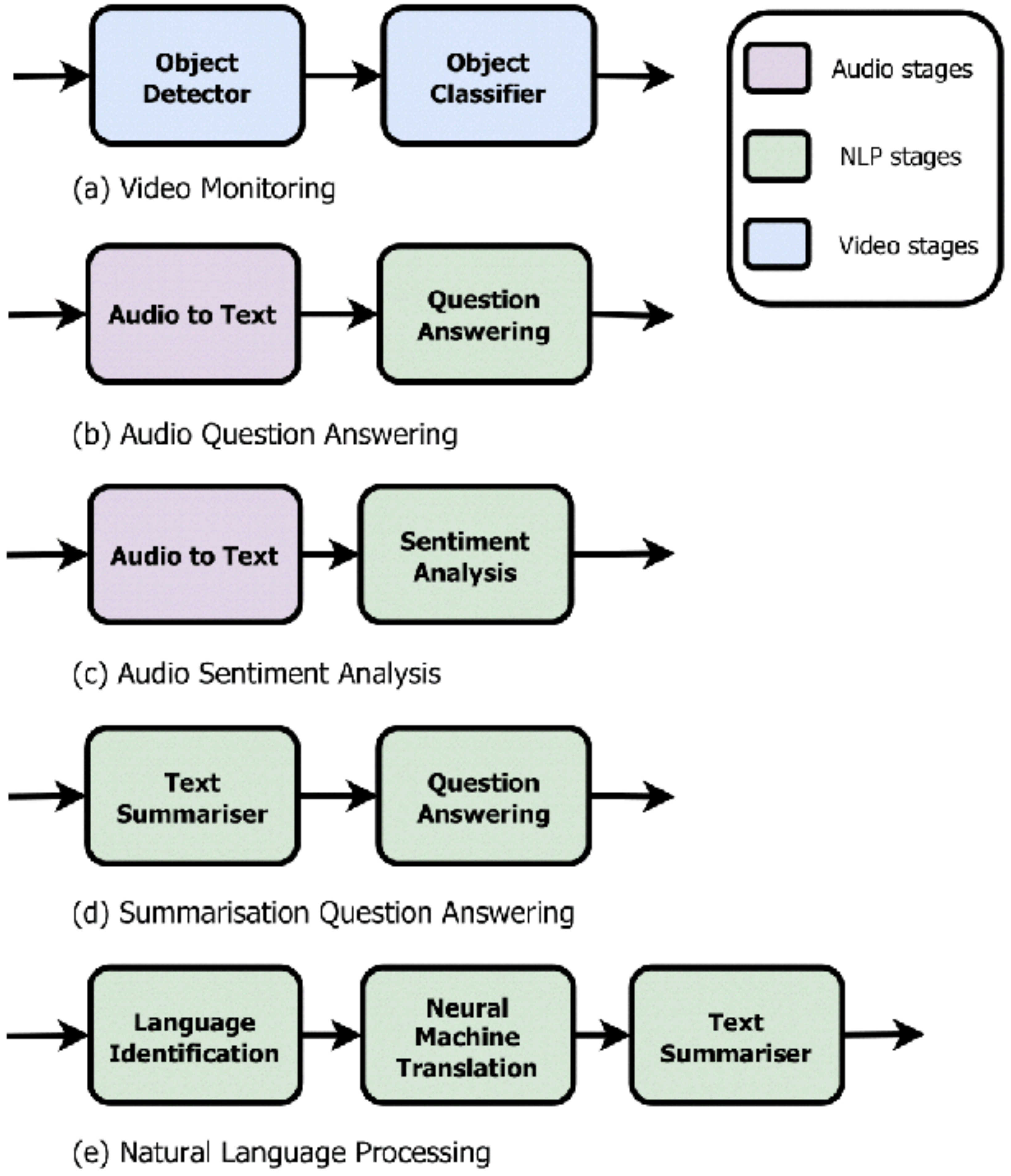
1. Industry standard
2. Used in recent research
3. Complete set of autoscaling, scheduling, observability tools (e.g. CPU usage)
4. APIs for changing the current AutoScaling algorithms

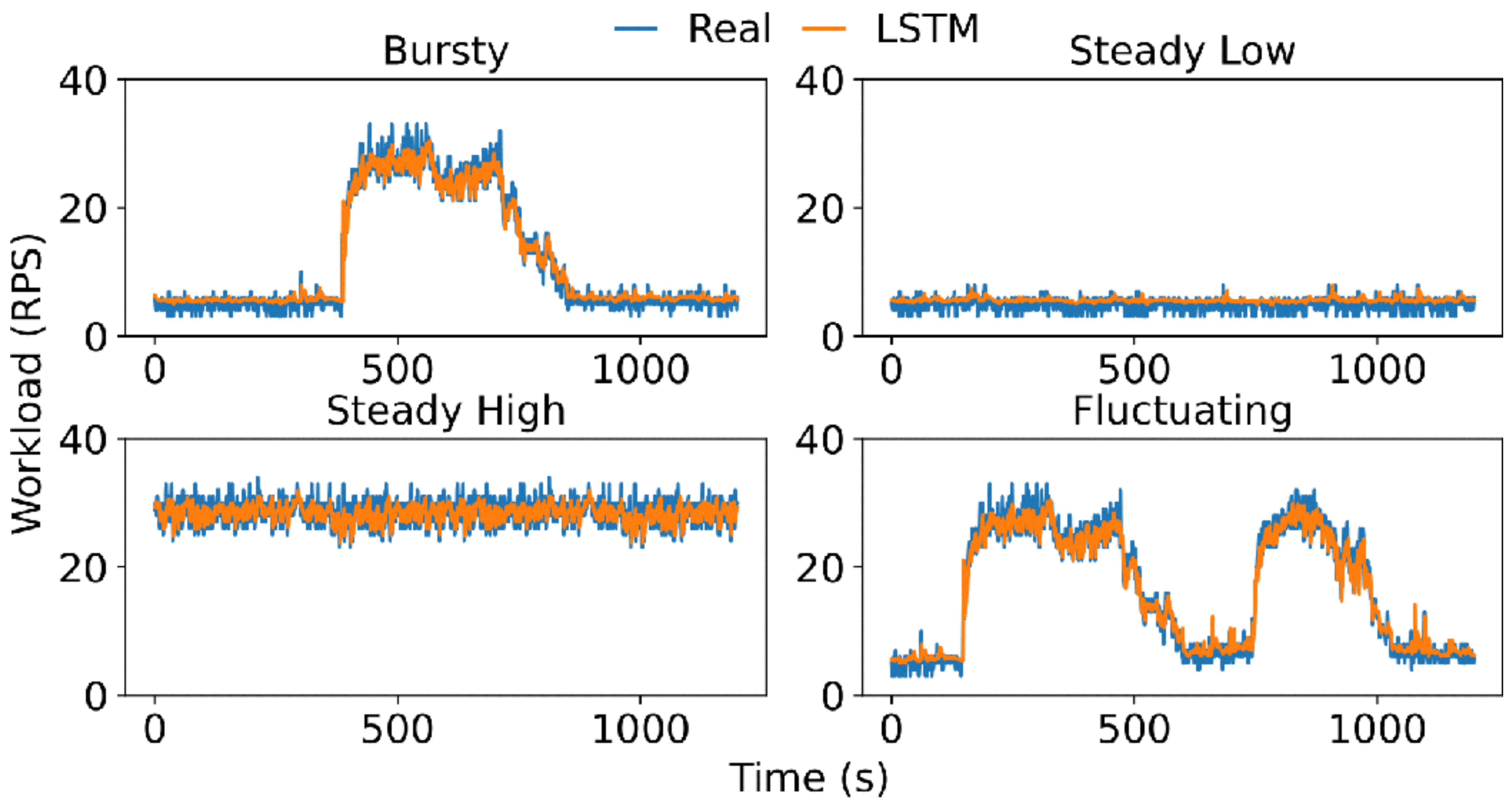
CORE

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

Evaluation

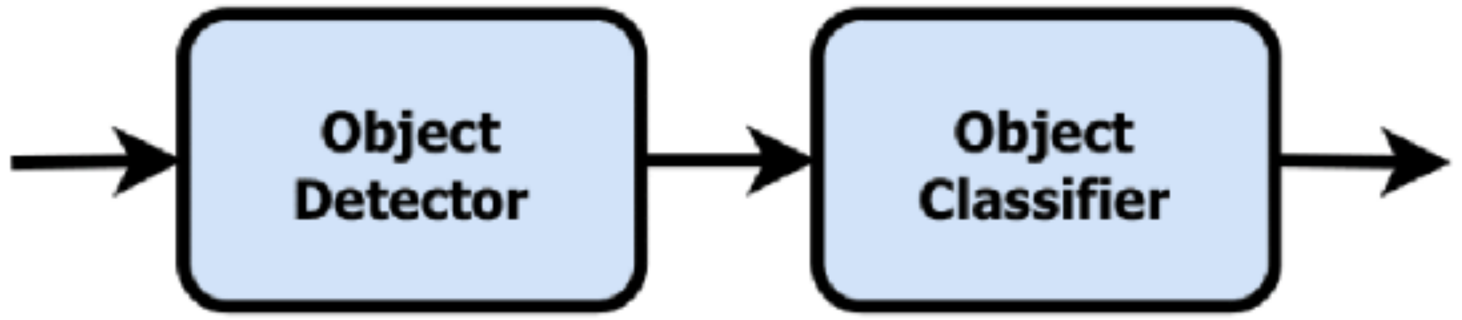
https://github.com/reconfigurable-ml-pipeline/ipa

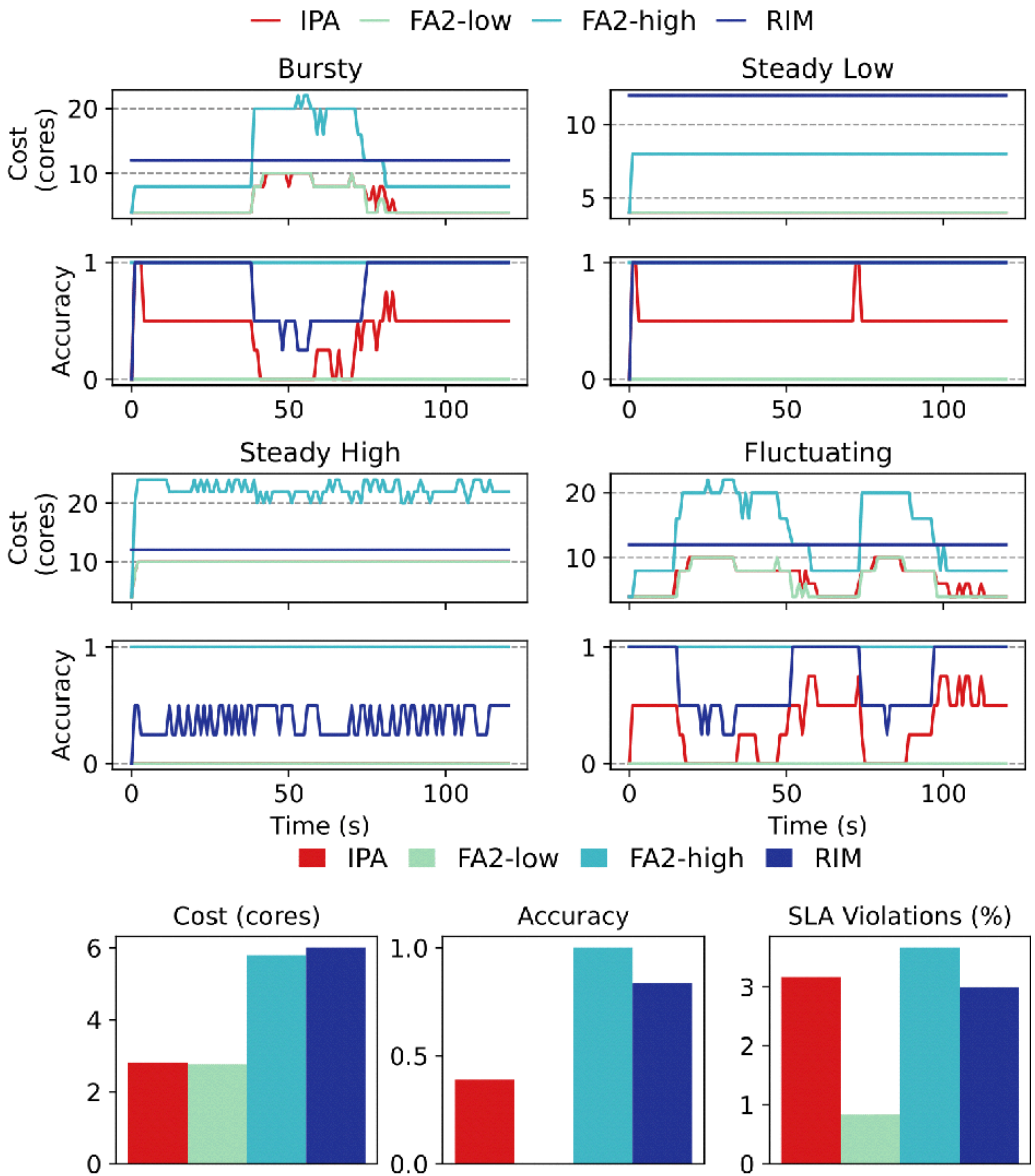




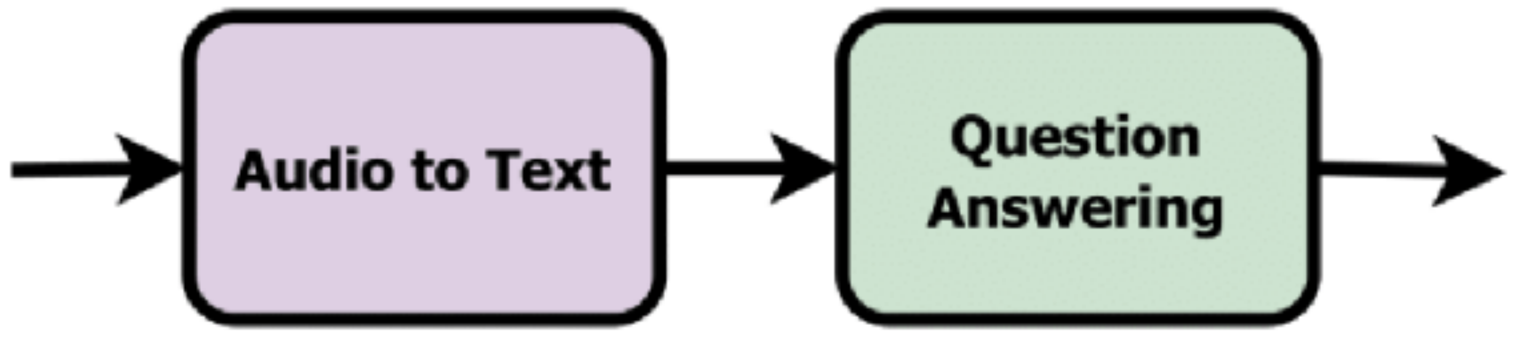
Experimental Results

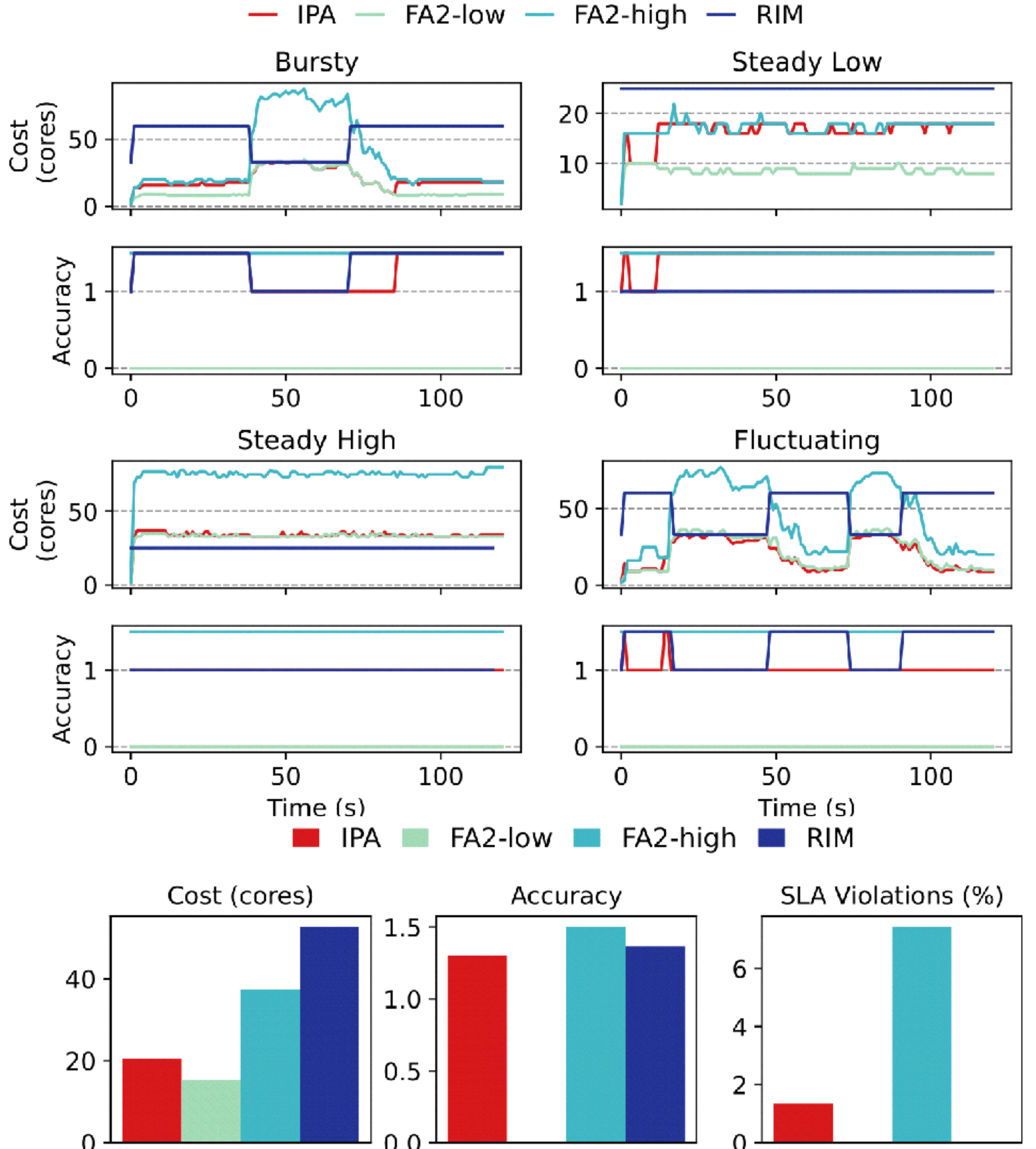
Video Pipeline



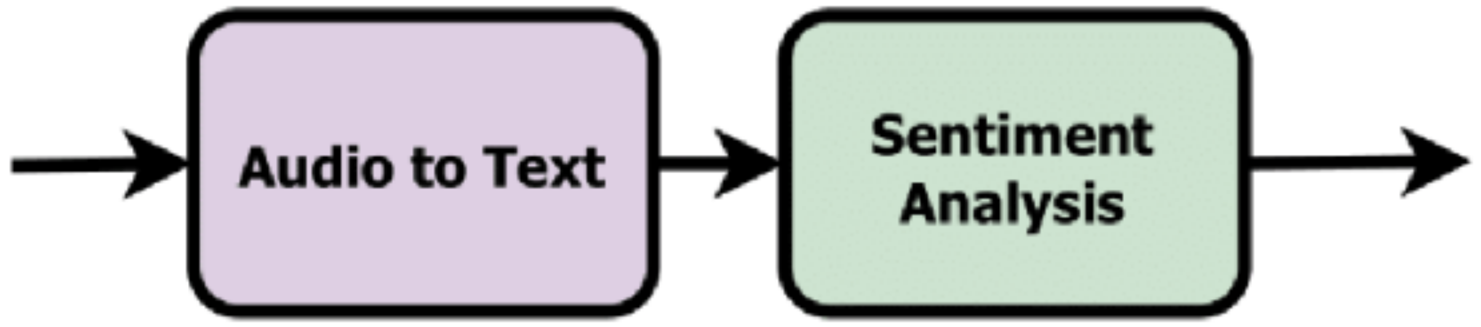


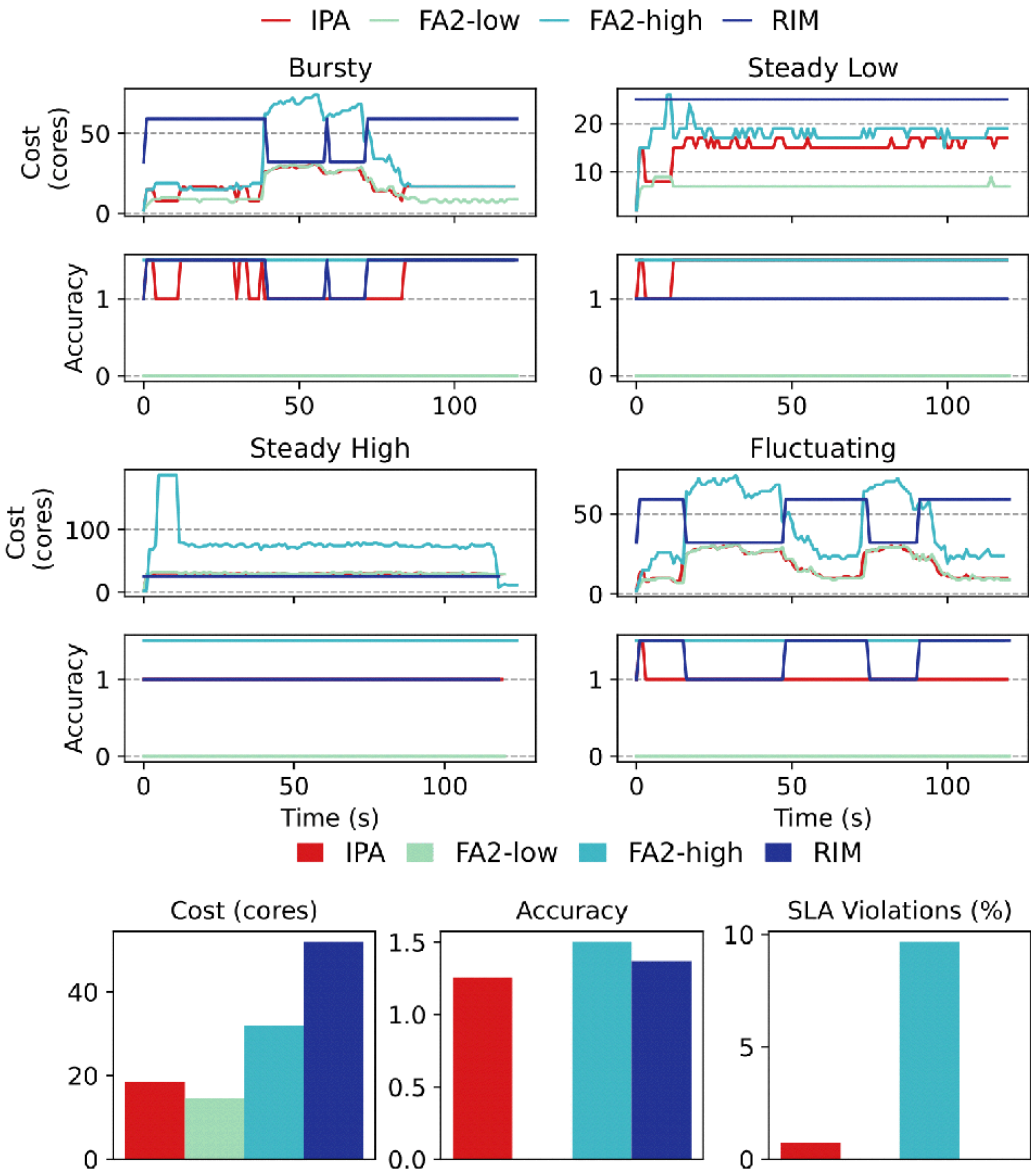
Audio + QA Pipeline



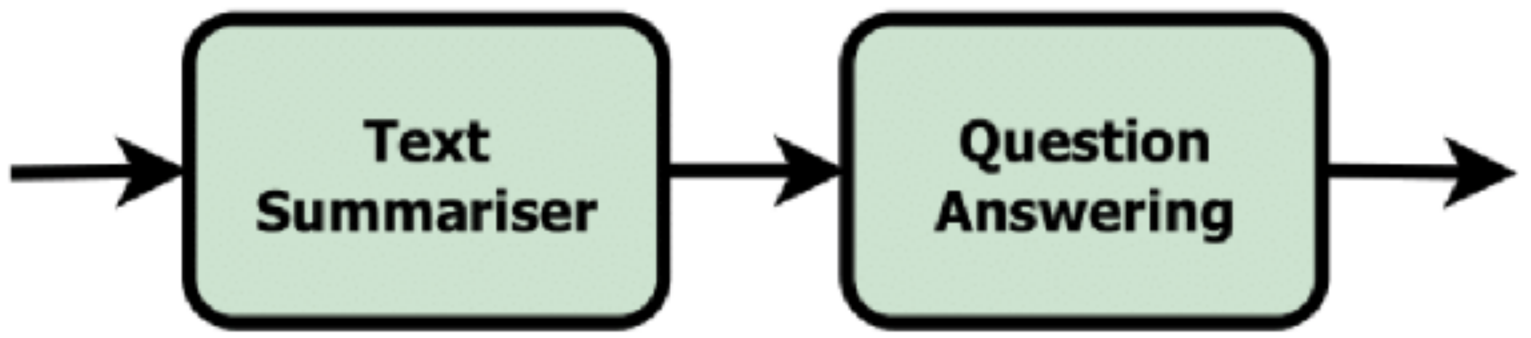


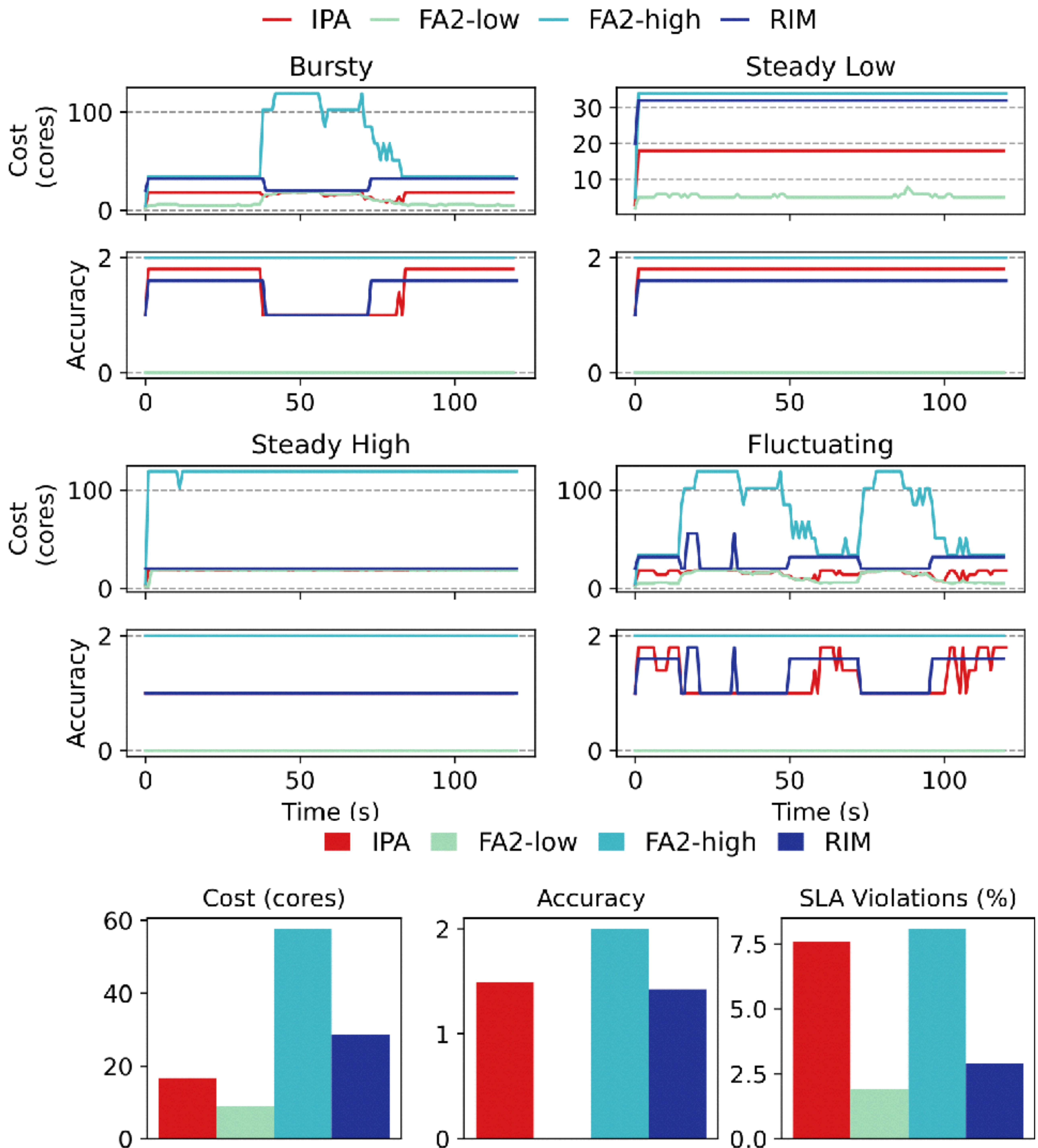
Summarization + QA Pipeline



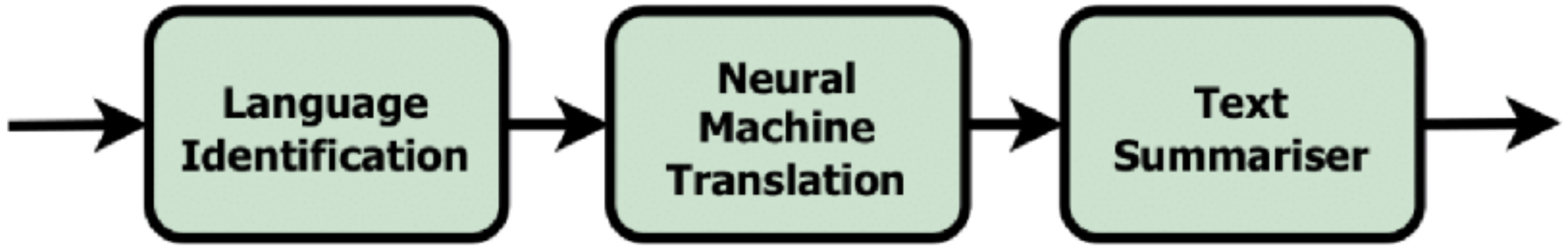


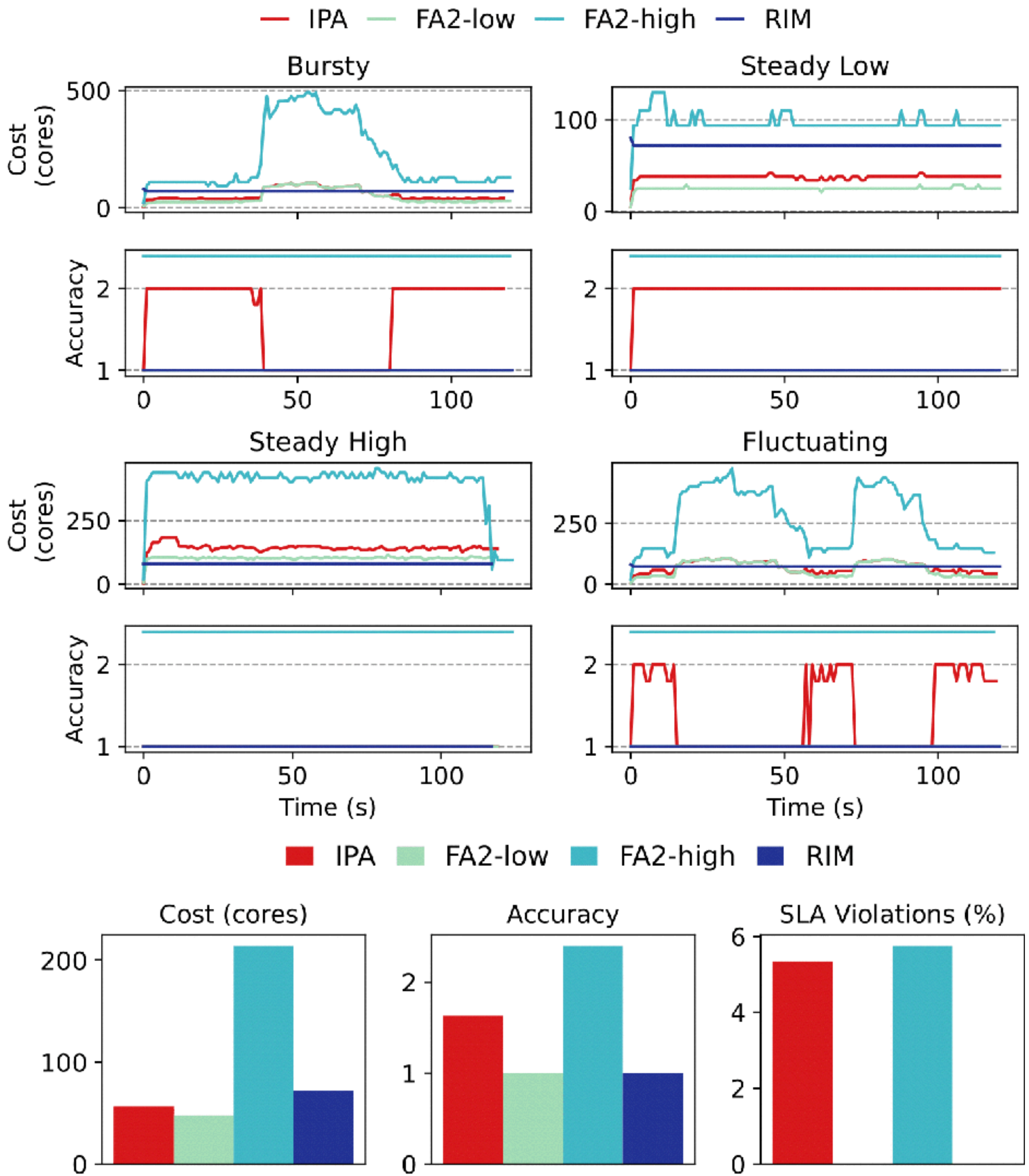
Summarization + QA Pipeline



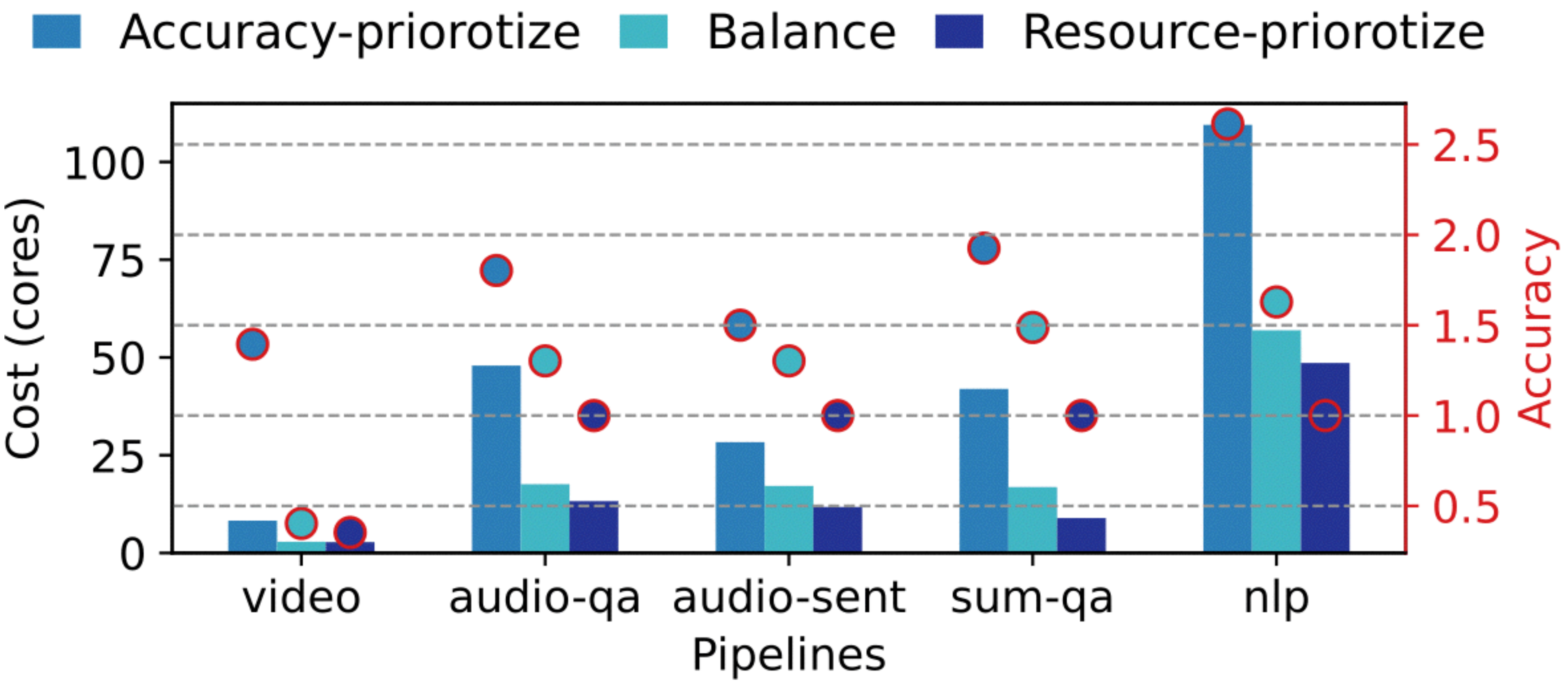


NLP Pipeline

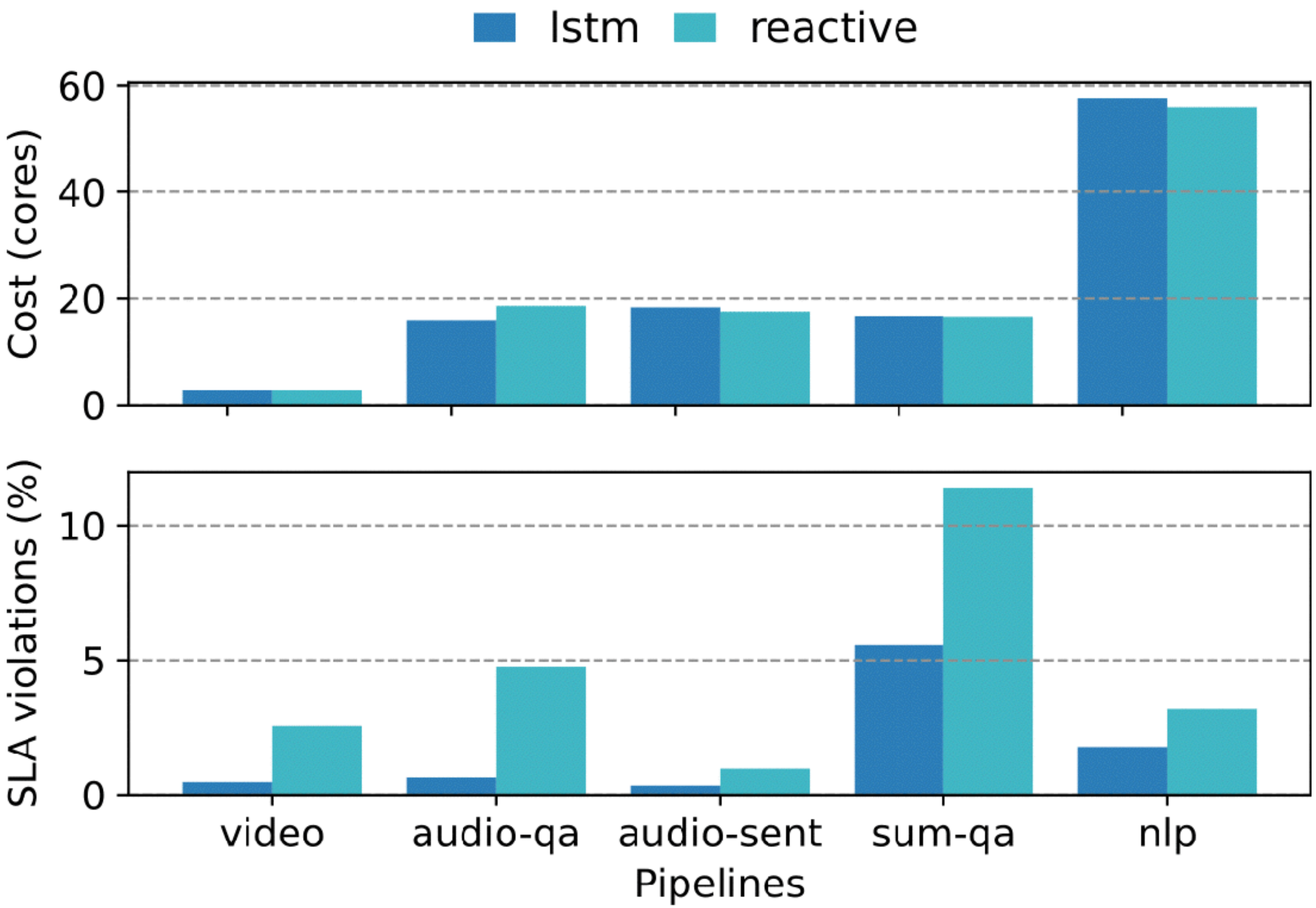




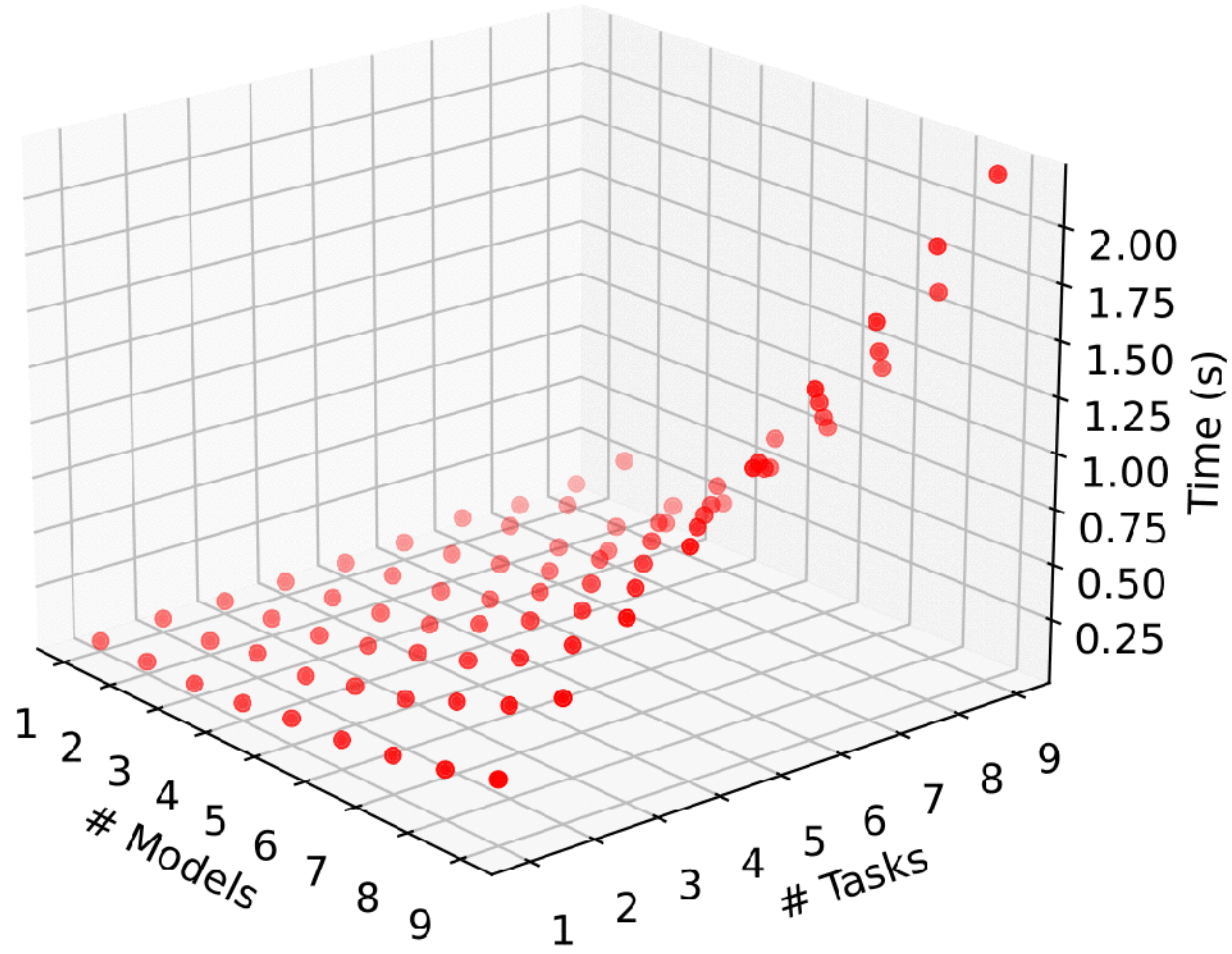
Adaptivity to multiple objectives



Effect of predictor

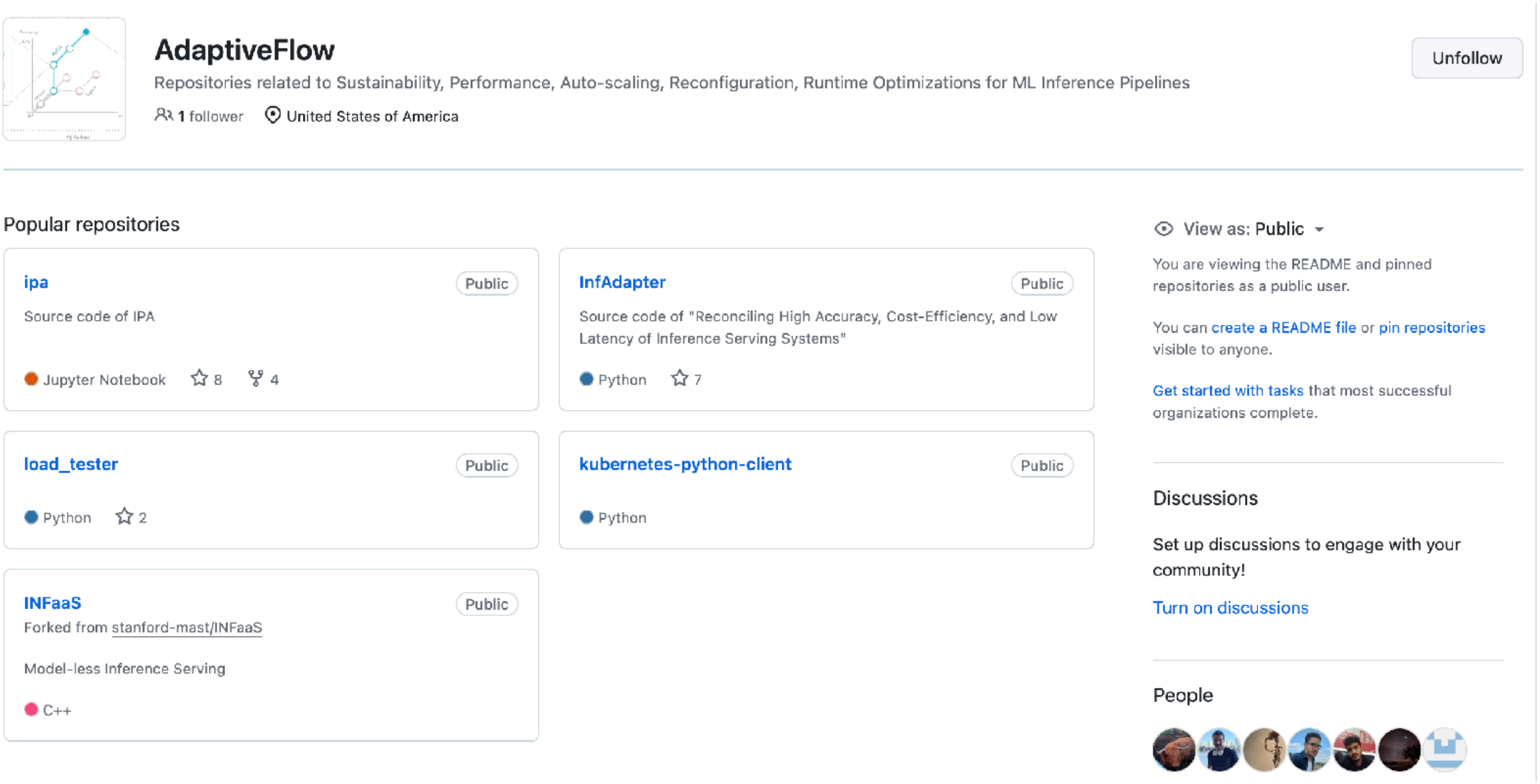


Gurobi solver scalability



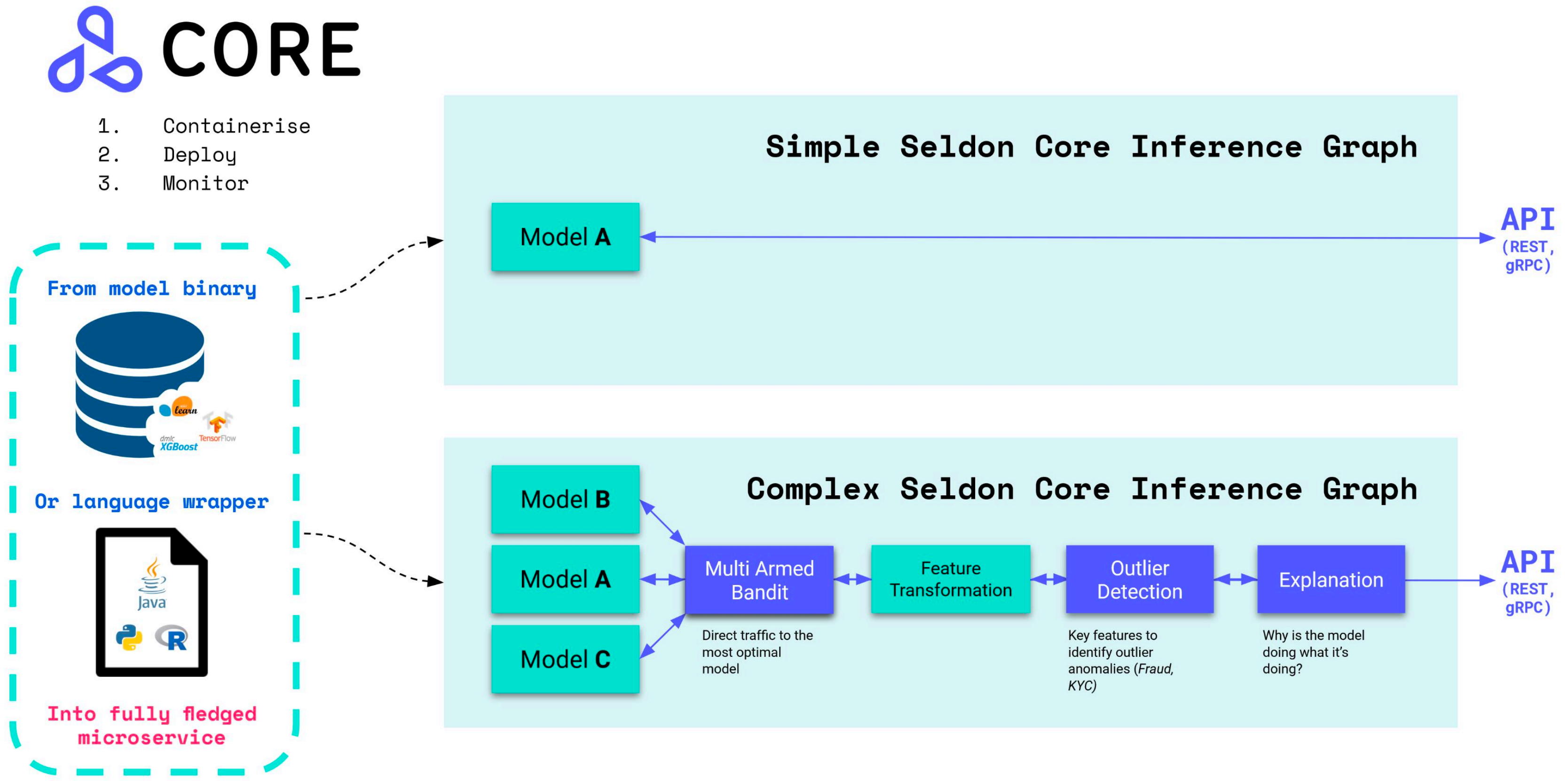
Full replication package is available

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

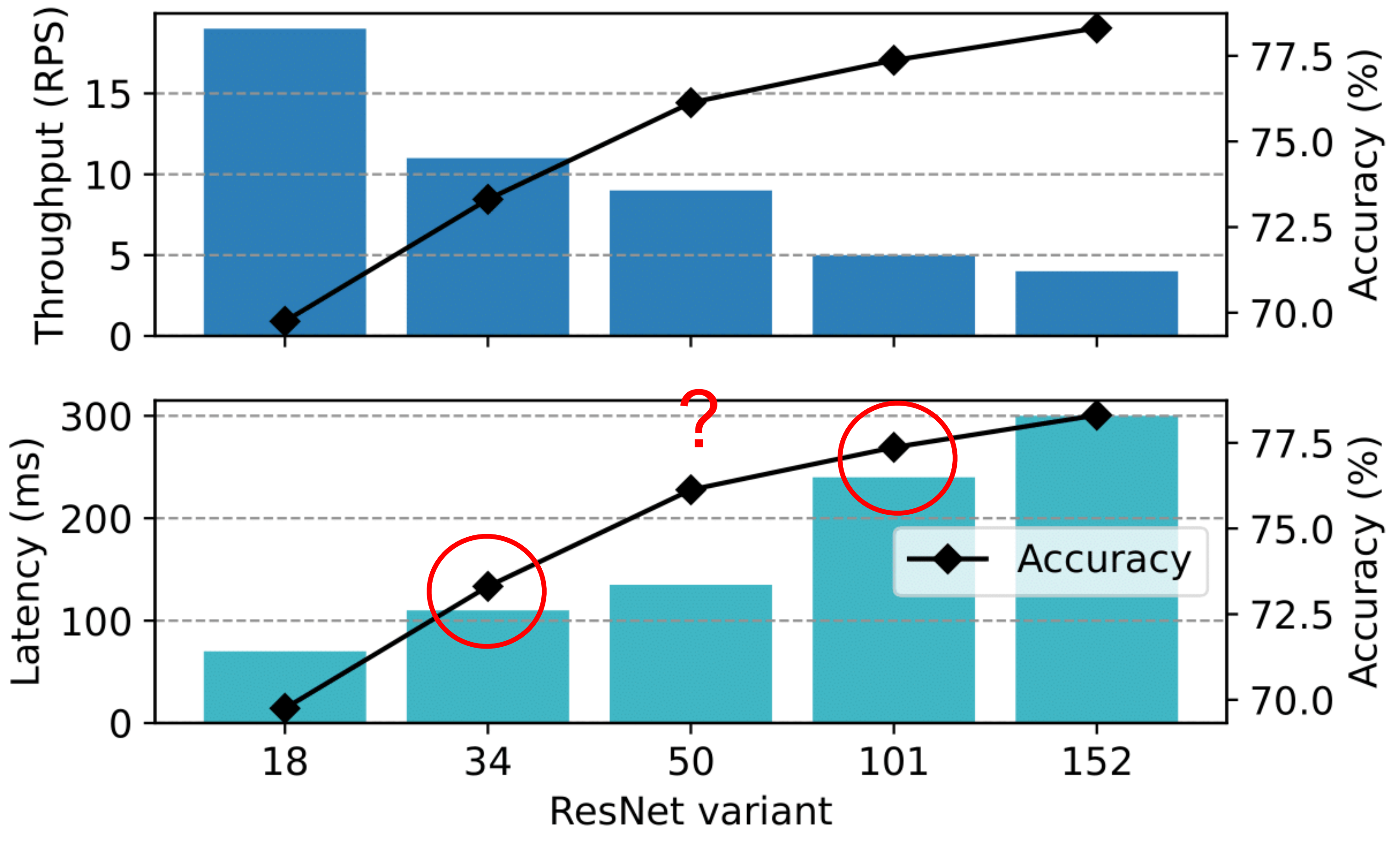


Model Serving

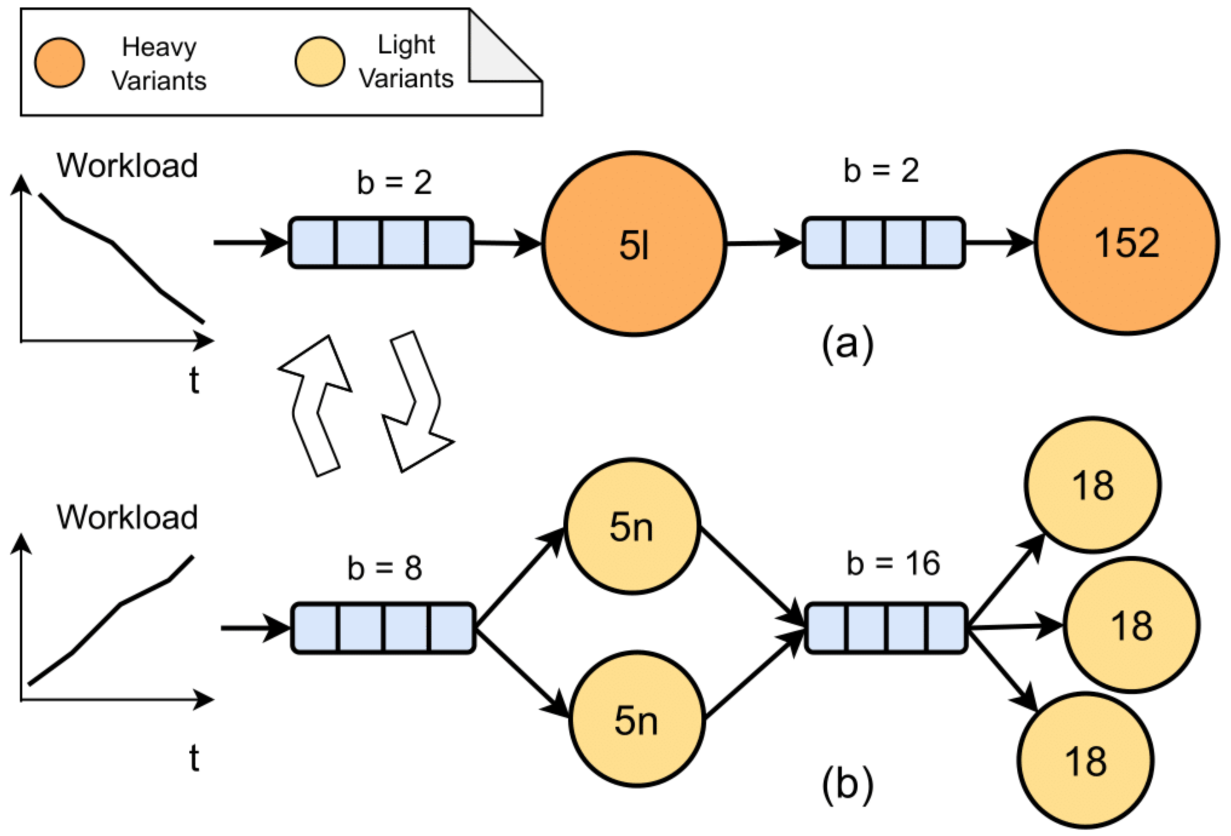
Pipeline



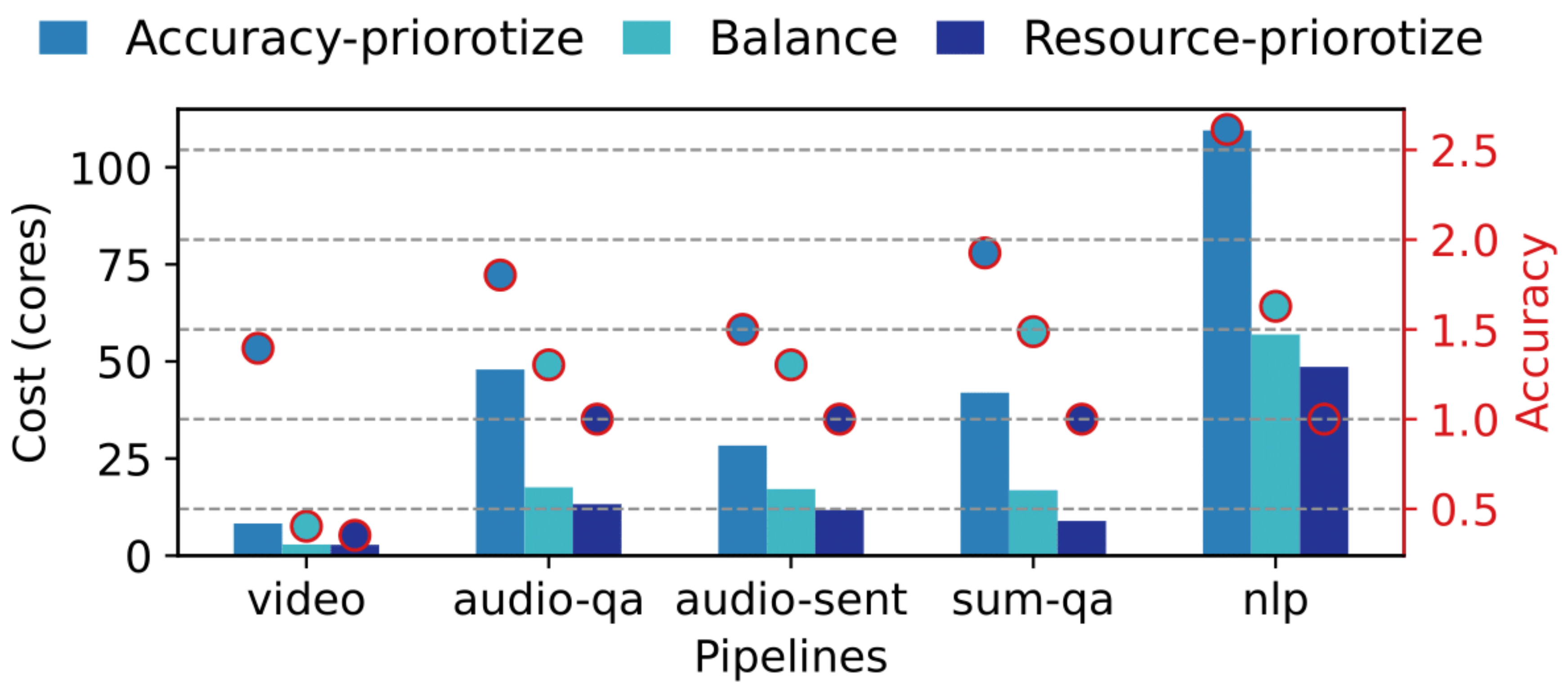
Is only scaling enough?



Snapshot of the System



Adaptivity to multiple objectives



EuroMLSys

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

Mehran Salmani[*], Saeid Ghafouri[§‡], Alireza Sanaee[§], Kamran Razavi[†], Max Mühlhäuser[†], Joseph Doyle[§], Pooyan Jamshidi[‡], Mohsen Sharifi[*]

Iran University of Science and Technology[*], Queen Mary University of London[§], Technical University of Darmstadt[†], University of South Carolina[‡]

InfAdapter [2023]: Autoscaling for ML **Model Inference**

JSys Journal of Systems Research Volume 4, Issue 1, April 2024

[SOLUTION] IPA: INFERENCE PIPELINE ADAPTATION TO ACHIEVE HIGH ACCURACY AND COST-EFFICIENCY

Saeid Ghafouri
University of South Carolina & Queen Mary University of London

Kamran Razavi
Technical University of Darmstadt

Mehran Salmani
Technical University of Ilmenau

Alireza Sanaee
Queen Mary University of London

Tania Lorido Botran
Roblox

Lin Wang
Paderborn University

Joseph Doyle
Queen Mary University of London

Pooyan Jamshidi
University of South Carolina

IPA [2024]: Autoscaling for **ML Inference Pipeline**

EuroMLSys

Sponge: Inference Serving with Dynamic SLOs Using In-Place Vertical Scaling

Kamran Razavi[*]
Technical University of Darmstadt

Saeid Ghafouri[*]
Queen Mary University of London

Max Mühlhäuser
Technical University of Darmstadt

Pooyan Jamshidi
University of South Carolina

Lin Wang
Paderborn University

Sponge [2024]: Autoscaling for ML Inference Pipeline with **Dynamic SLO**

Dynamic User -> Dynamic Network Bandwidths

- Users move
  - Fluctuations in the network bandwidths
    - Reduced time-budget for processing requests

SLO

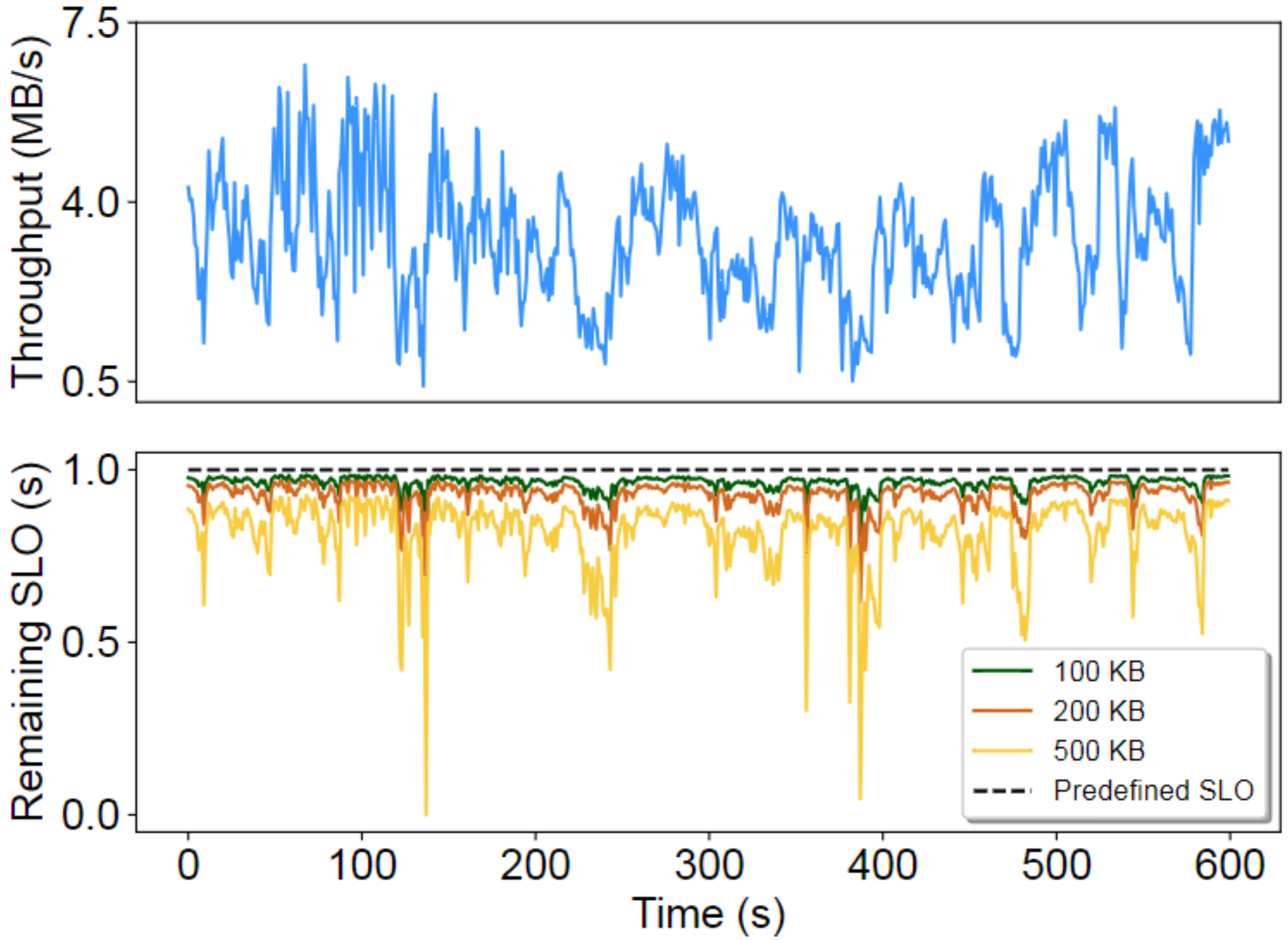
network latency processing latency

Dynamic User -> Dynamic Network Bandwidths

- Users move
  - Fluctuations in the network bandwidths
    - Reduced time-budget for processing requests

SLO

network latency processing latency



Inference Serving Requirements

Highly Responsive!
(end-to-end latency guarantee)

Cost-Efficient!
(least resource consumption)

Resource Scaling

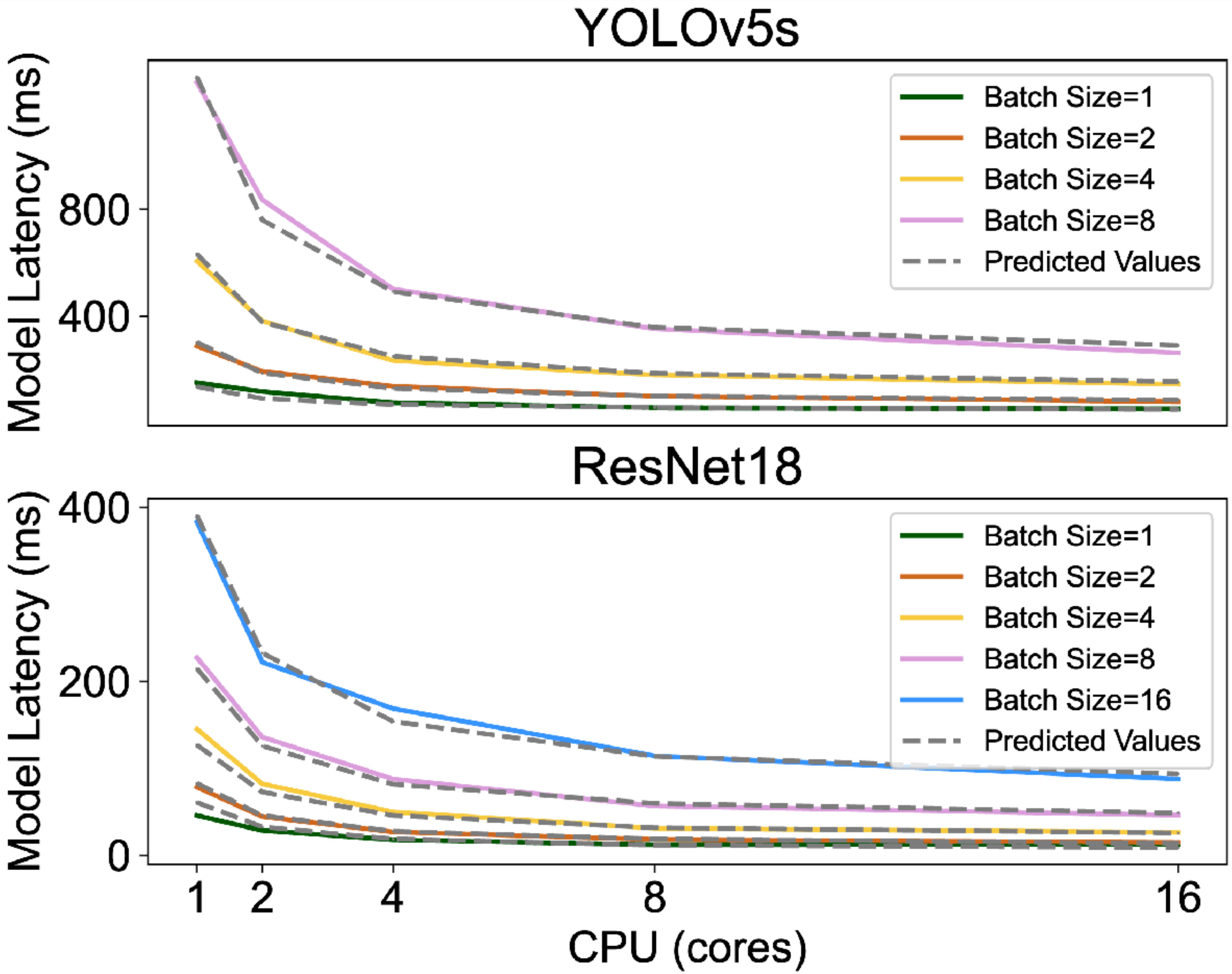
Sponge!

In-place Vertical Scaling
(more responsive)

Horizontal Scaling
(more cost efficient)

Vertical Scaling DL Model Profiling

- How much resource should be allocated to a DL model?
  - Latency/batch size → linear relationship
  - Latency/CPU allocation → inverse relationship



Problem Formulation

$$
\begin{aligned}
\text{Minimize} \quad & c + \delta \times b \\
\text{subject to} \quad & l(b, c) + q_r(b, c) + \mathrm{cl}_{max} \leq SLO, \quad \forall r \in R \\
& h(b, c) \geq \lambda \\
& b, c \in \mathbb{Z}^+
\end{aligned}
$$

Problem Formulation

Minimize resource costs

$$
\begin{aligned}
\text{Minimize} \quad & c + \delta \times b \\
\text{subject to} \quad & l(b, c) + q_r(b, c) + \mathrm{cl}_{max} \leq SLO, \quad \forall r \in R \\
& h(b, c) \geq \lambda \\
& b, c \in \mathbb{Z}^+
\end{aligned}
$$

Problem Formulation

Minimize resource costs

$$
\begin{aligned}
\text{Minimize} \quad & c + \delta \times b \\
\text{subject to} \quad & l(b,c) + q_r(b,c) + \mathrm{cl}_{max} \le SLO, \quad \forall r \in R \\
& h(b,c) \ge \lambda \\
& b, c \in \mathbb{Z}^{+}
\end{aligned}
$$

Limit the batch size to grow infinitely!

Problem Formulation

Minimize resource costs

$$\text{Minimize} \quad c + \delta \times b$$

Limit the batch size to grow infinitely!

$$\text{subject to} \quad l(b,c) + q_r(b,c) + \mathrm{cl}_{max} \le SLO, \quad \forall r \in R$$

$$h(b,c) \ge \lambda$$

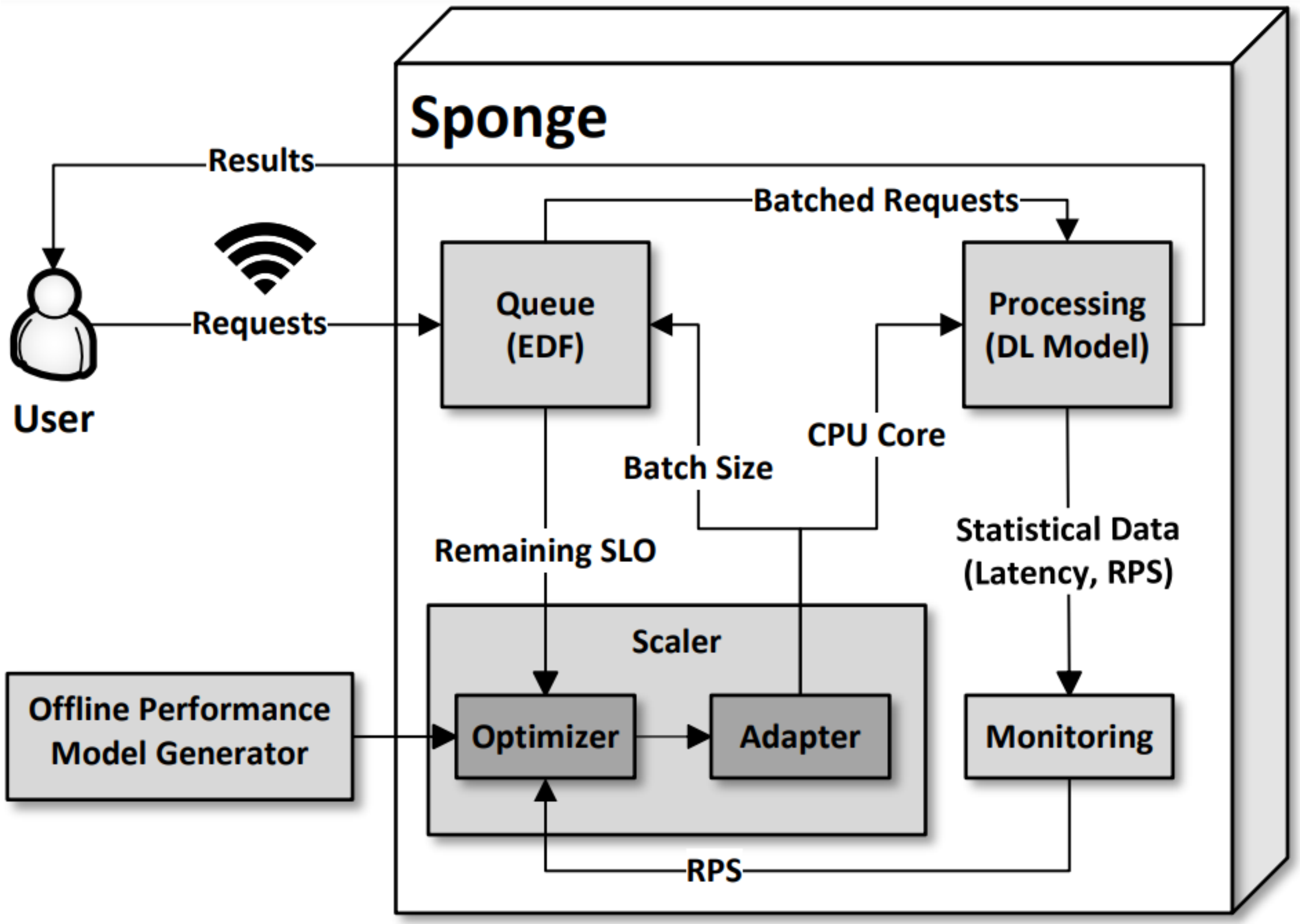
$$b, c \in \mathbb{Z}^+$$

R	Set of all requests
b	Model's batch size
c	Model's CPU allocation
cl_r	Communication latency associated with $r \in R$
cl_{max}	Highest cl_r in R
SLO	Pre-defined SLO for R
$l(b,c)$	Processing time of a model with allocation core c and batch size b
$q_r(b,c)$	Queuing time of $r \in R$ with allocation core c and batch size b
$h(b,c)$	Throughput of a model with allocation core c and batch size b
λ	Request arrival rate

System Design

3 design choices:

1. **In-place vertical scaling**
   - Fast response time
2. **Request reordering**
   - High priority requests
3. **Dynamic batching**
   - Increase system utilization

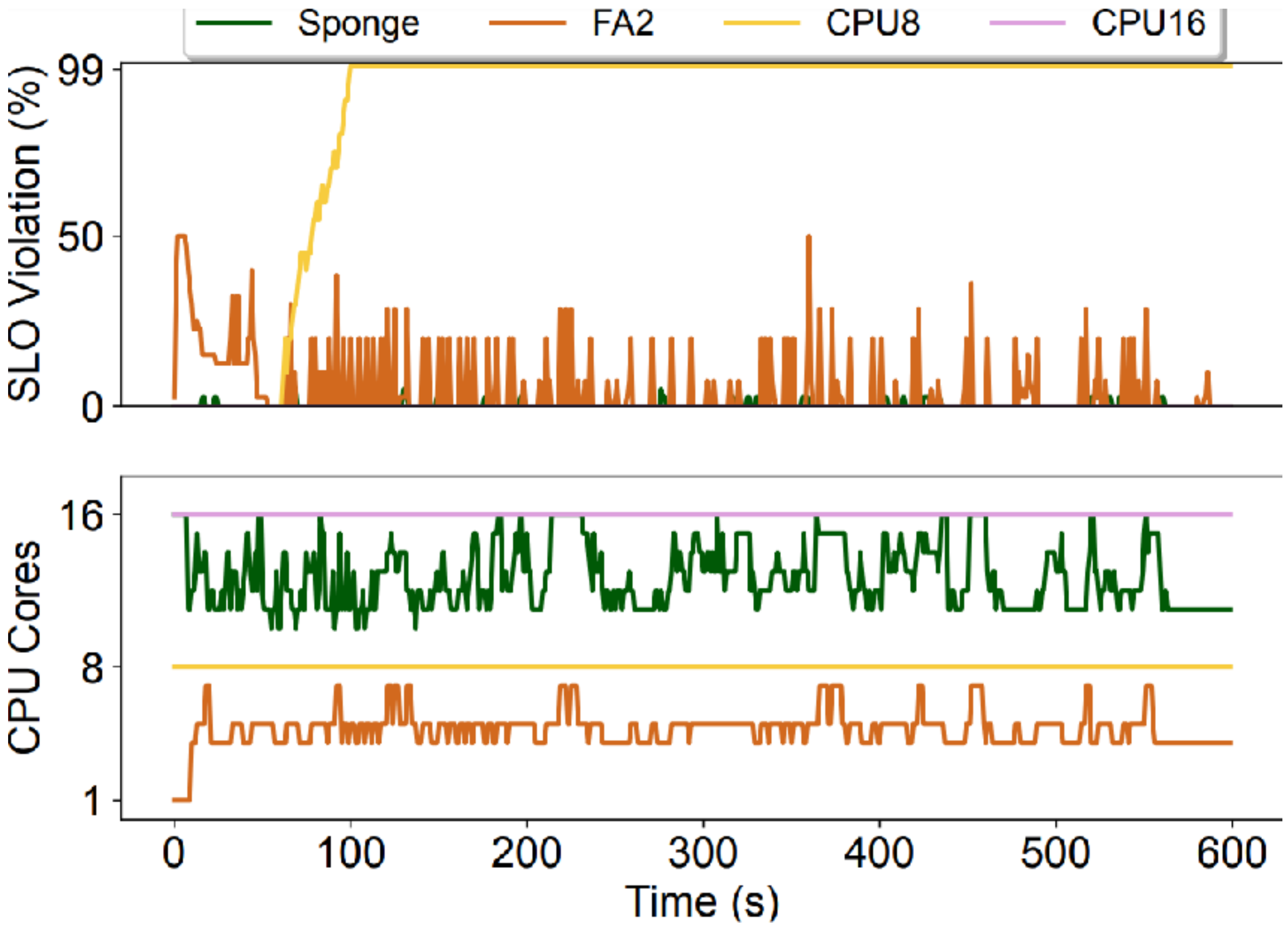


Evaluation

SLO guarantees (99th percentile) with up to 20% resource save up compared to static resource allocation.

Sponge source code:

https://github.com/saeid93/sponge



Future Directions

Resource Scaling

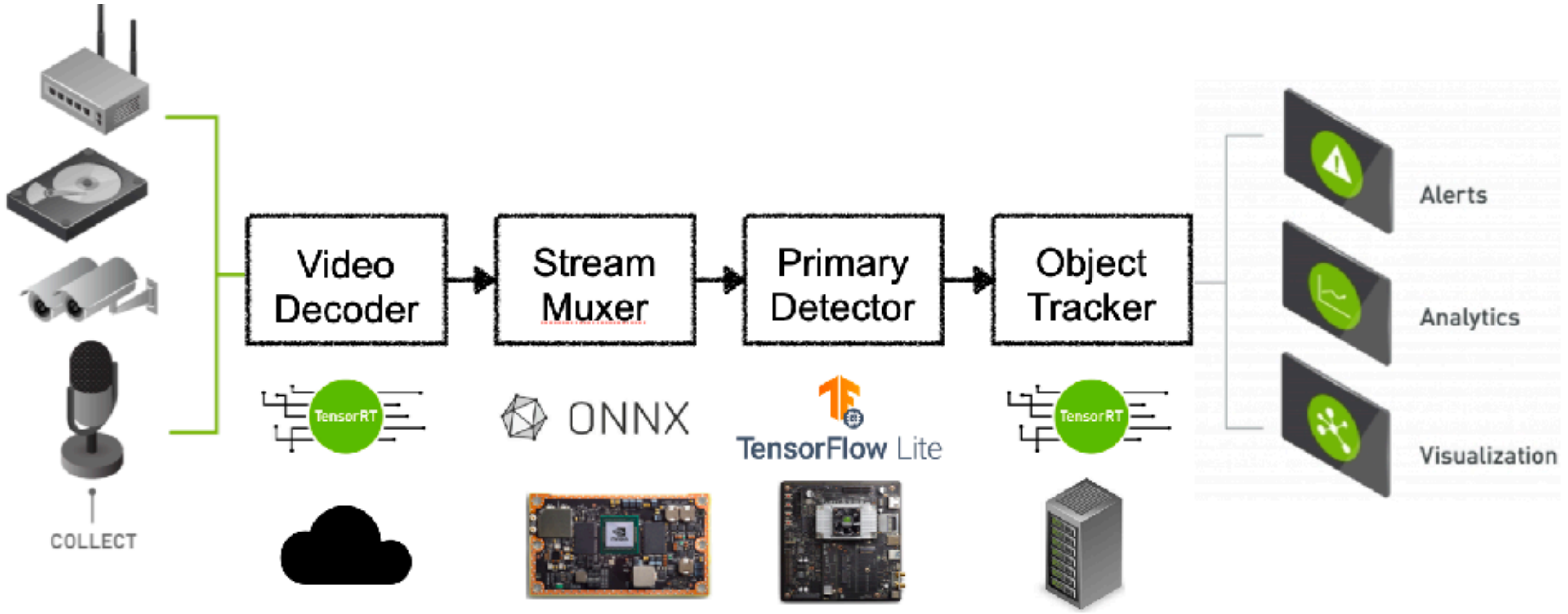
Sponge!

In-place Vertical Scaling
(more responsive)

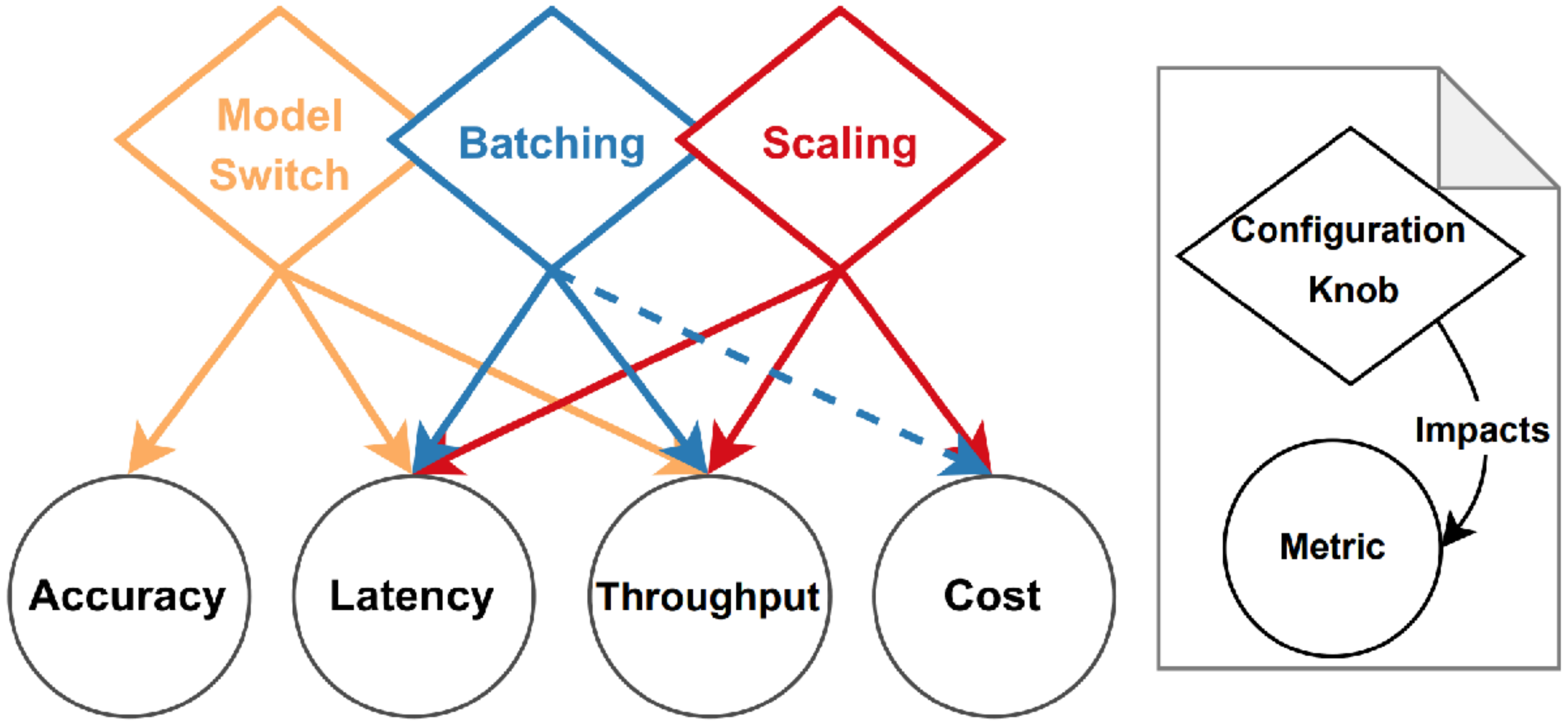
Horizontal Scaling
(more cost efficient)

How can both scaling mechanisms be used jointly under a dynamic workload to be responsive and cost efficient while guaranteeing SLOs?

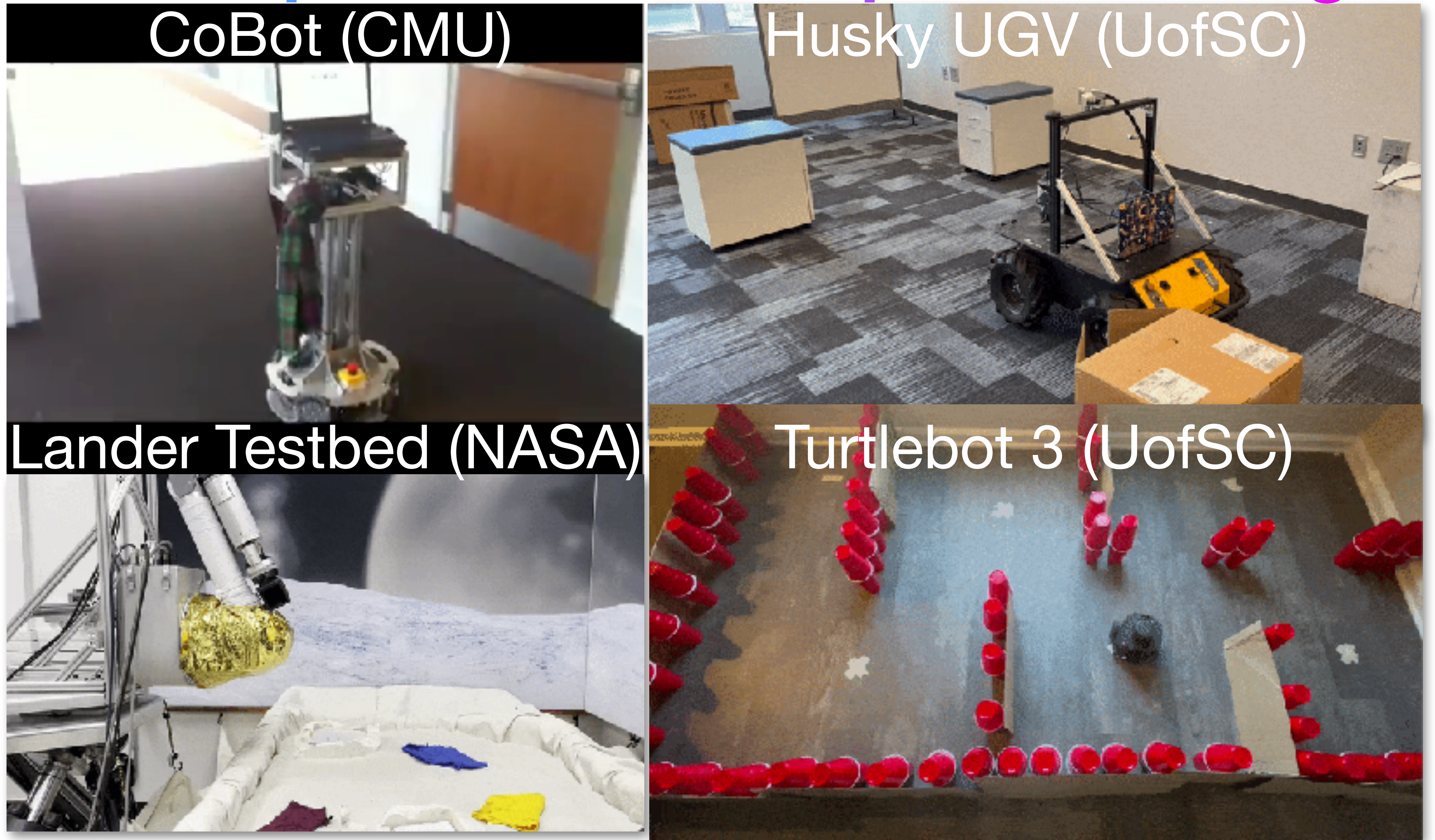
The variability space (design space) of (composed) systems is exponentially increasing



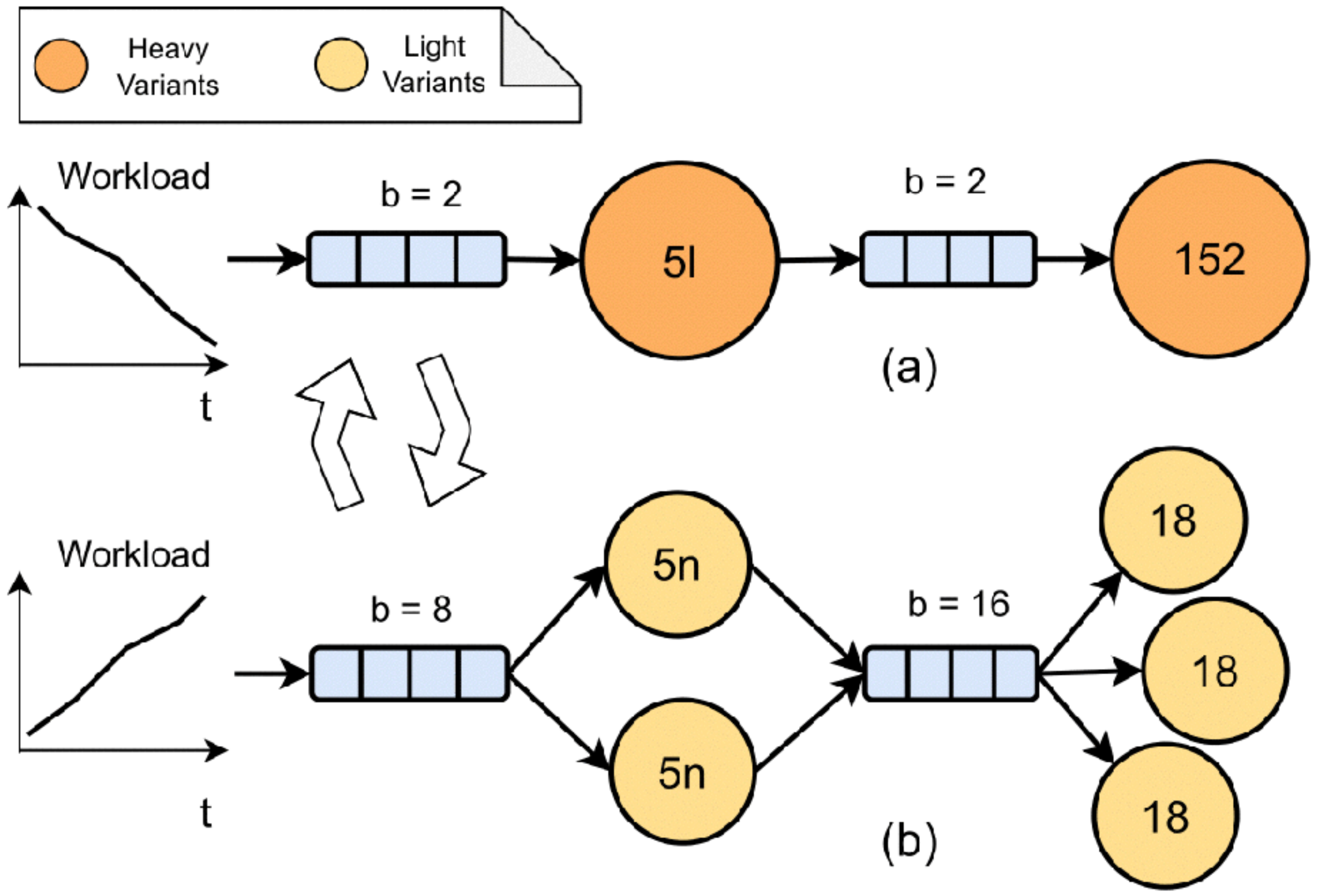
Performance goals are competing and users have preferences over these goals



Systems operate in uncertain environments with imperfect and incomplete knowledge



Goal: Enabling users to find the right quality tradeoff



Thank you, Saeid Ghafouri!

Optimizing Production ML Inference for Accuracy and Cost Efficiency

Pushing the Boundaries of Cost-Effective ML Inference on Chameleon Testbed

May 28, 2024 by Saeid Ghafouri

User Experiments, Featured

Over the past decade, advancements in machine learning (ML) have paved the way for numerous real-world use cases such as chatbots, self-driving cars, and recommender systems. Traditional ML applications typically use a single deep neural network (DNN) to perform inference tasks, such as object recognition or natural language understanding. In contrast, modern ML systems – those used in sophisticated systems (think digital assistant services such as Amazon Alexa) – are very complex. These systems employ a series of interconnected DNNs, often structured as directed acyclic graphs (DAGs), to handle a variety of inference tasks, including speech recognition, question interpretation, question answering, and text-to-speech conversion, all working together to meet user queries and requirements.