



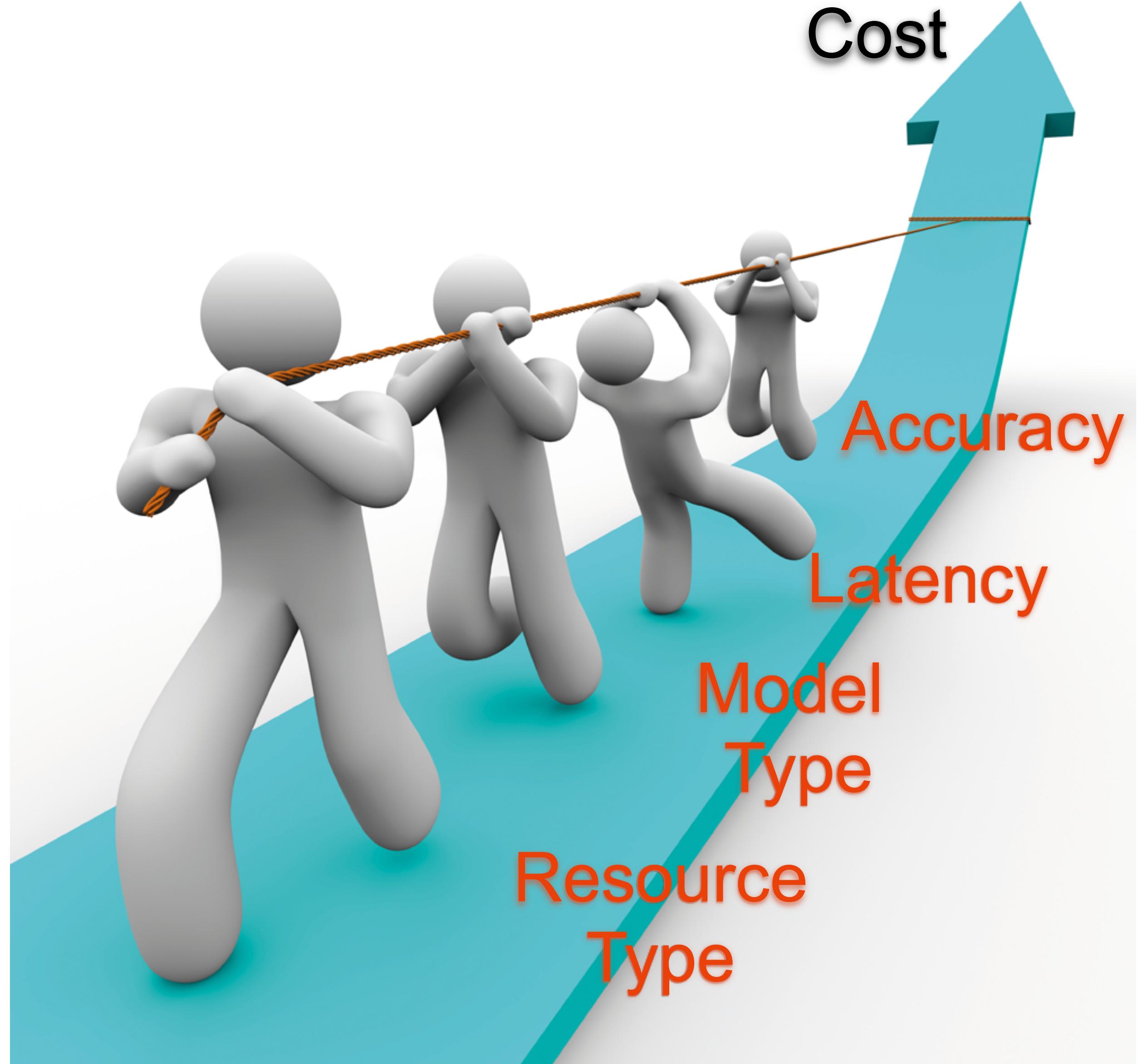
PennState

# Implications of Public Cloud Resource Heterogeneity for Inference Serving

Jashwant Raj Gunasekaran, Cyan Subhra Mishra, Prashanth Thinakaran,  
Mahmut Kandemir, Chita Das

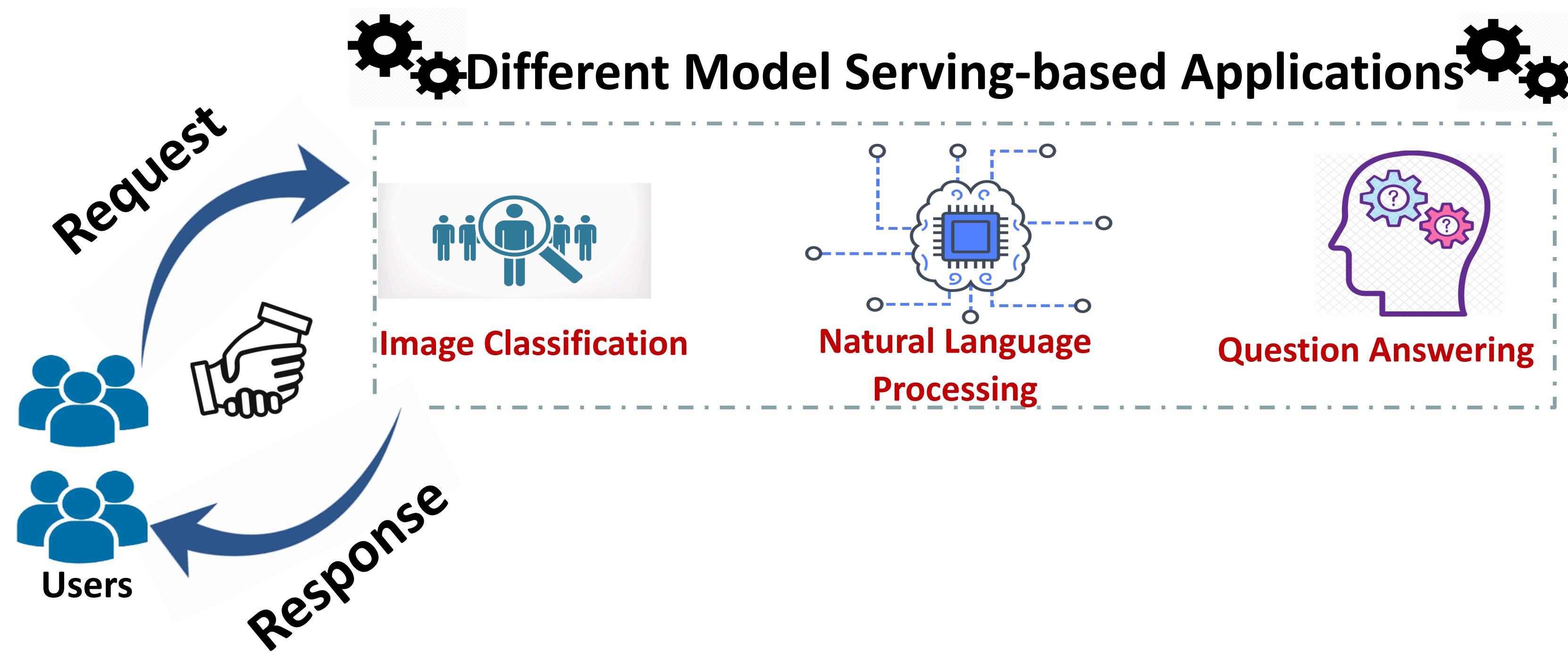
*Sixth International Workshop on Serverless Computing (WoSC6)*  
*Dec 8, 2020*

# Model Serving in Public Cloud

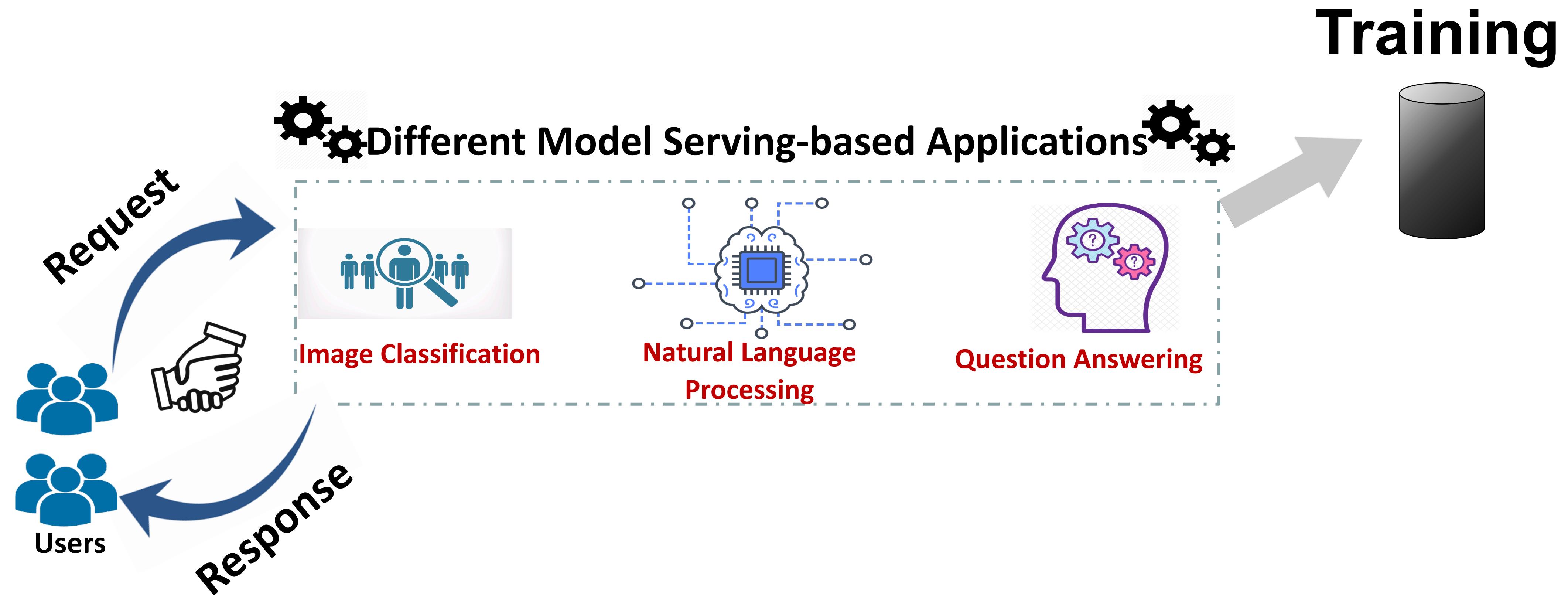


# Model Serving Hosted on Cloud

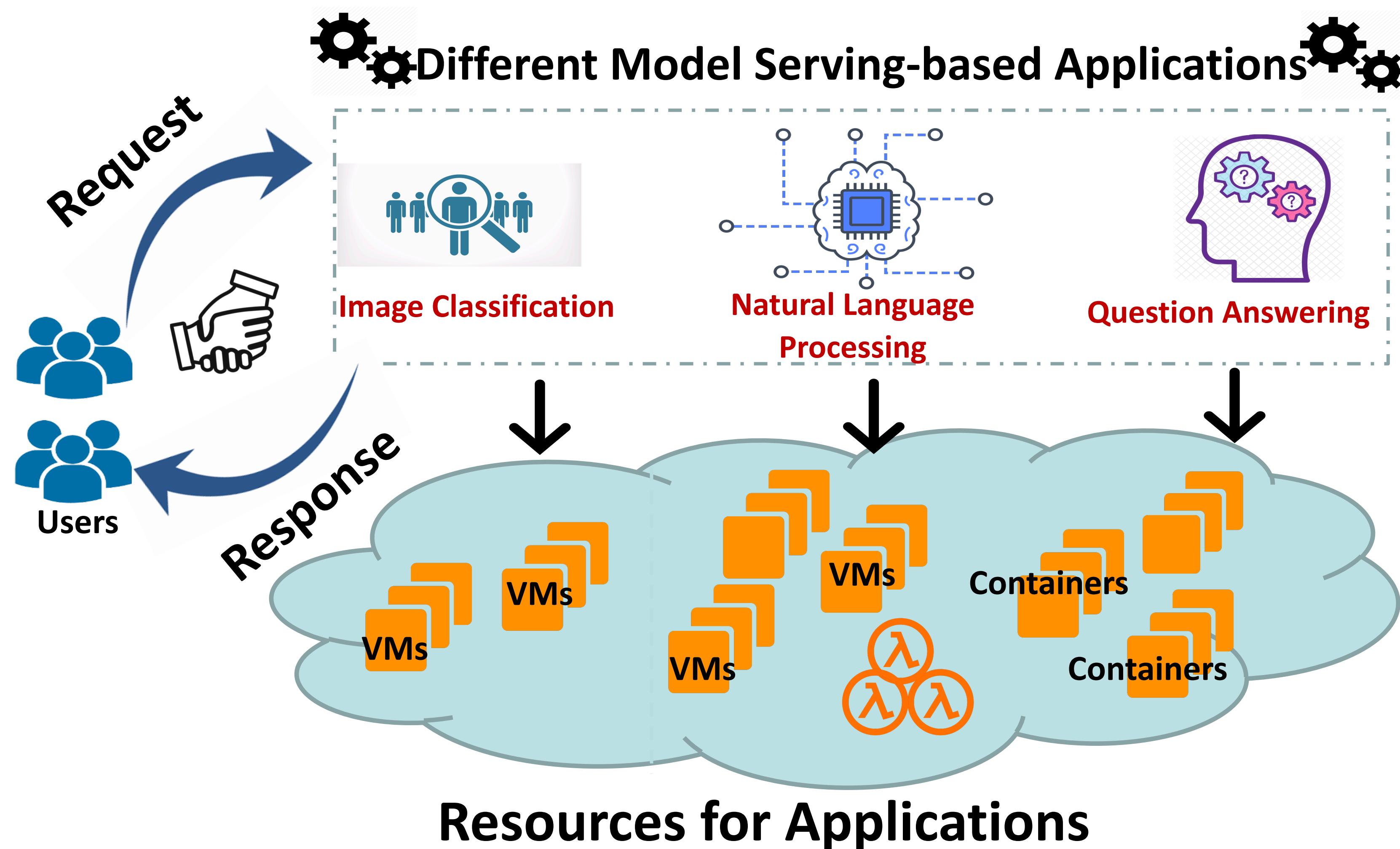
# Model Serving Hosted on Cloud



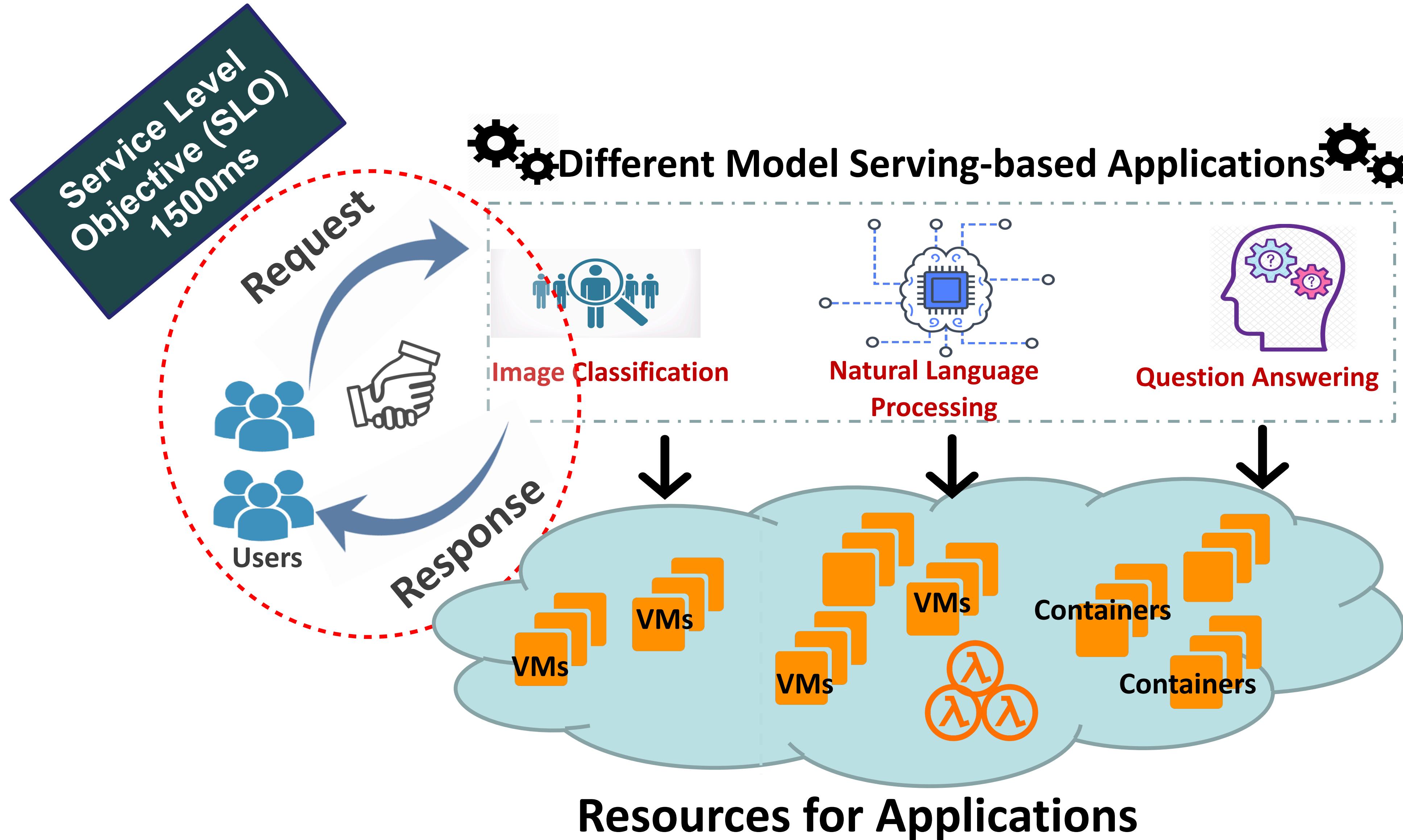
# Model Serving Hosted on Cloud



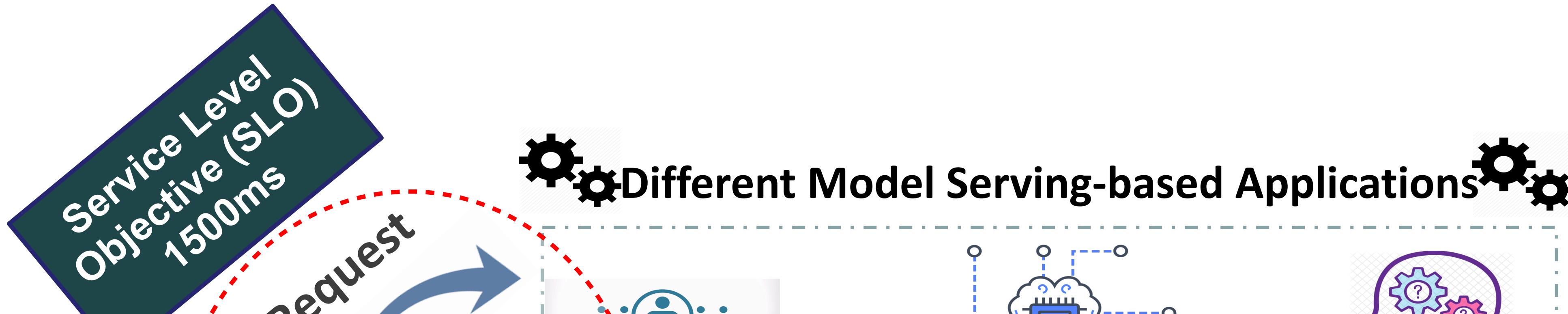
# Model Serving Hosted on Cloud



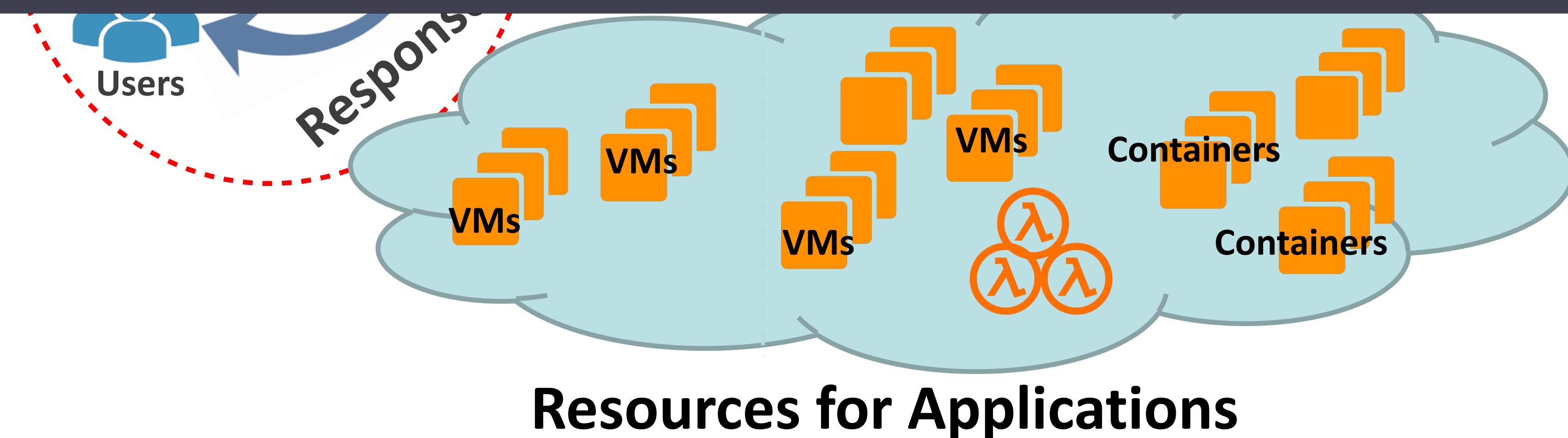
# Model Serving Hosted on Cloud



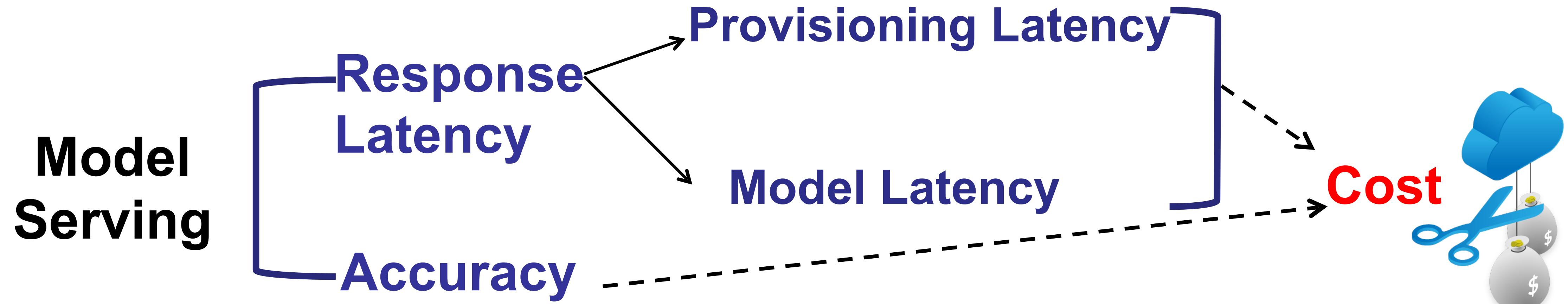
# Model Serving Hosted on Cloud



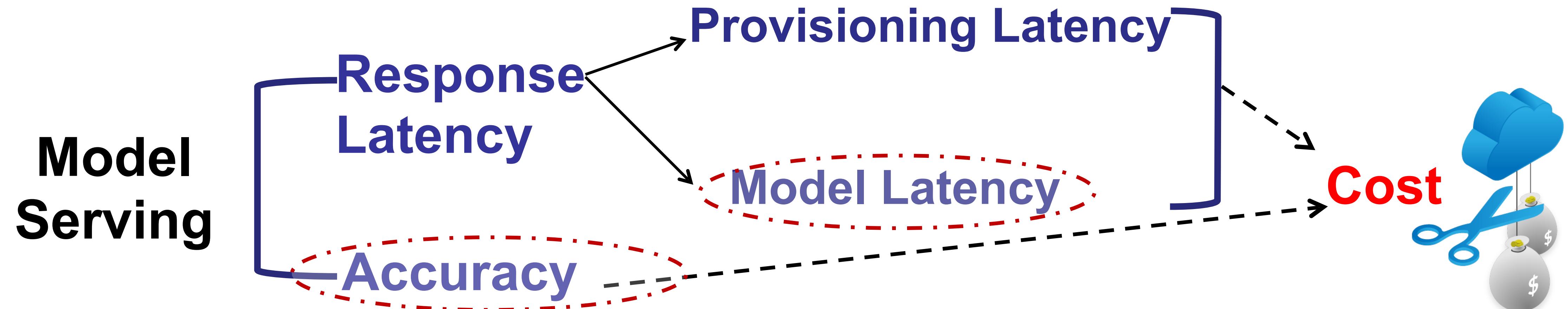
How to optimize both model selection and resource selection?



# Model Serving Requirements



# Model Serving Requirements



# Model Serving Requirements

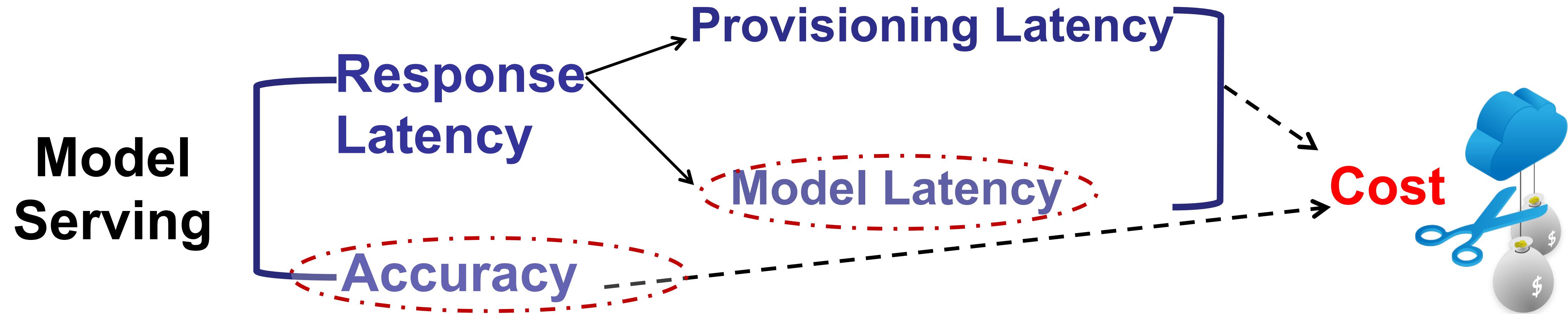
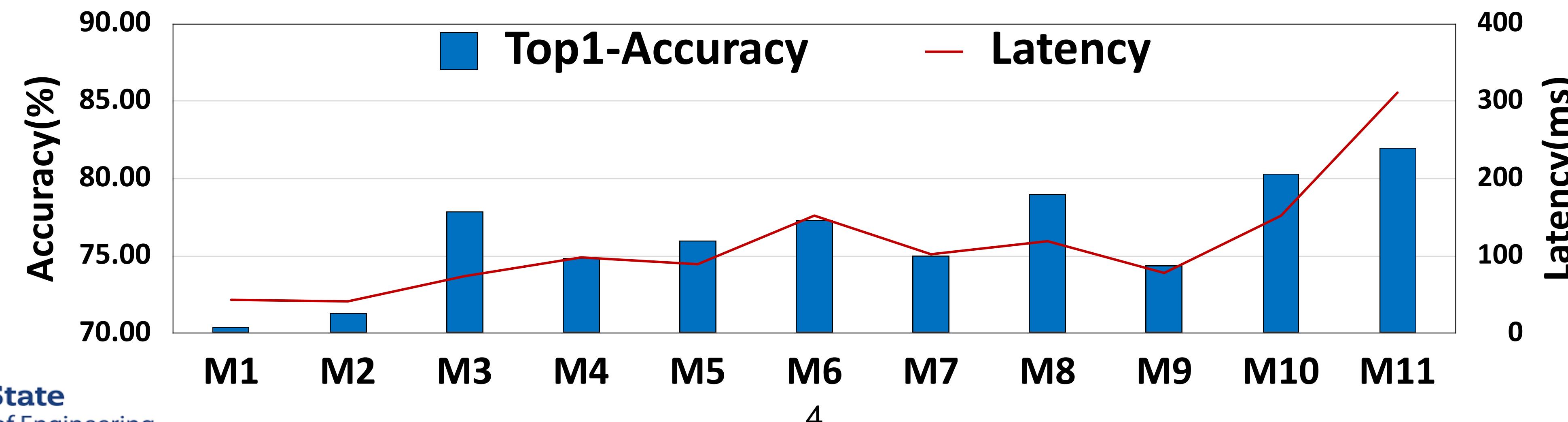
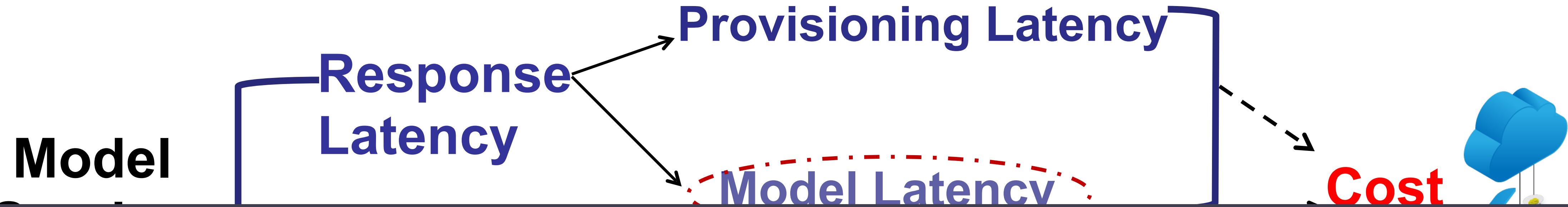


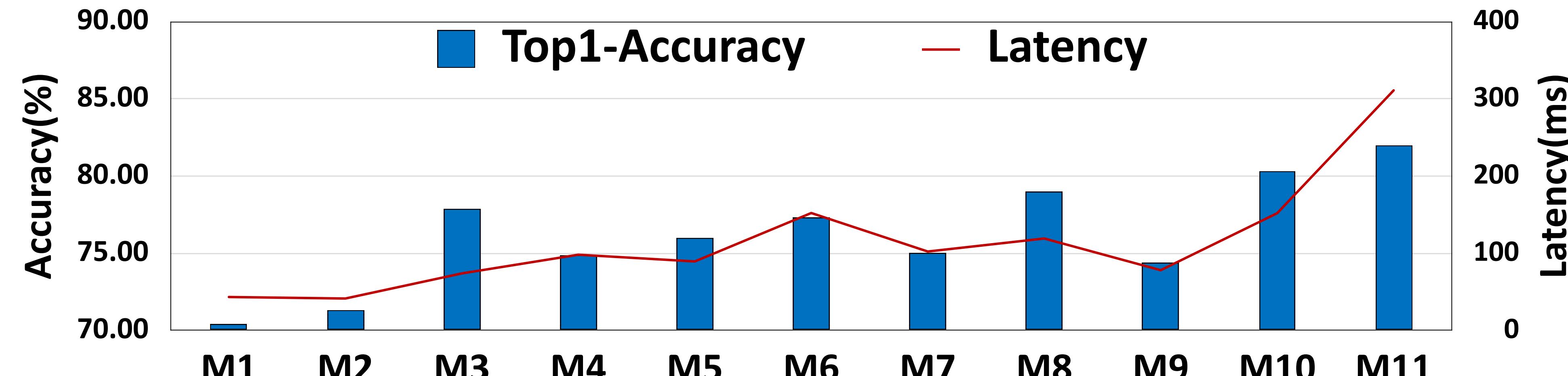
Image Classification using pretrained keras models



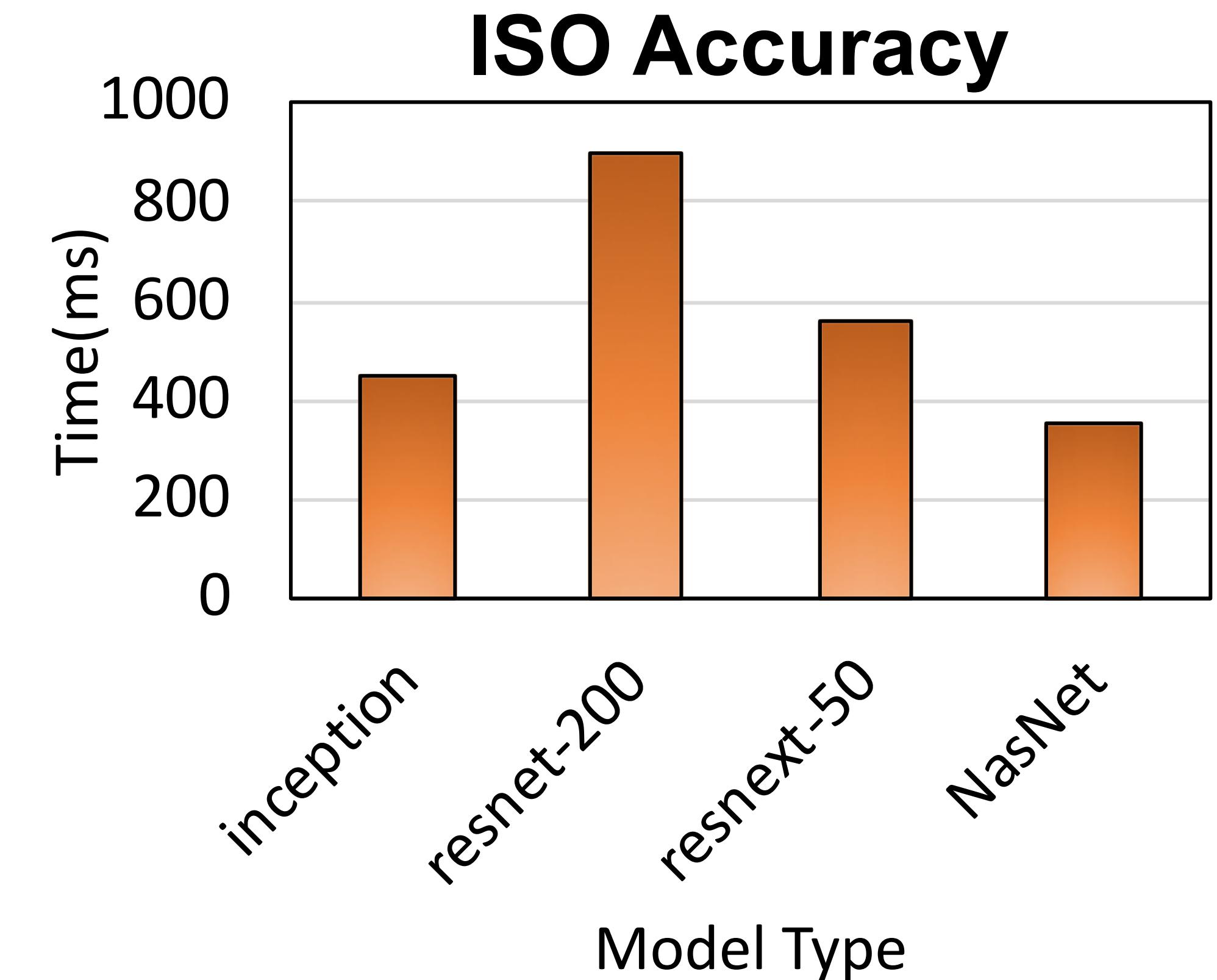
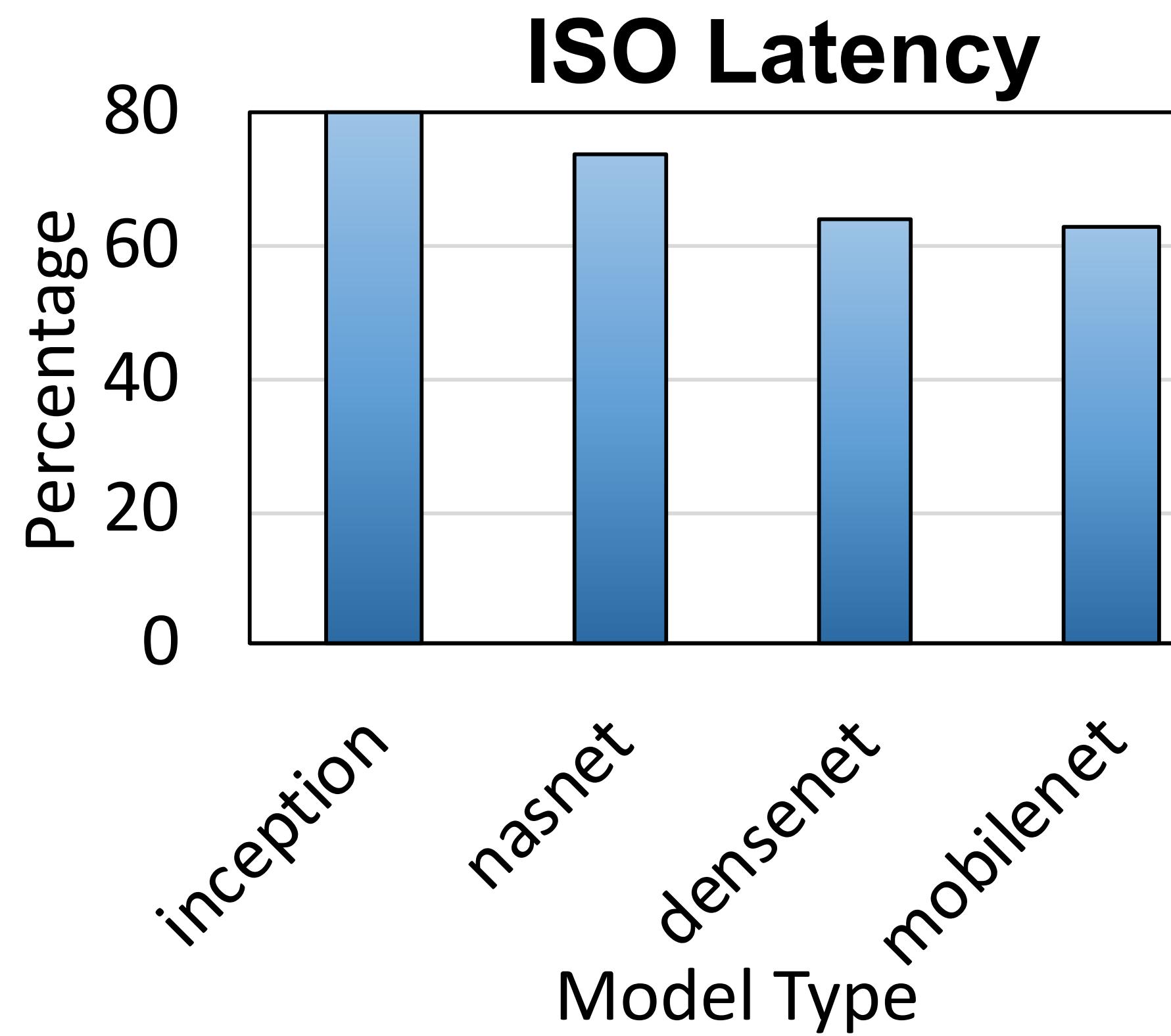
# Model Serving Requirements



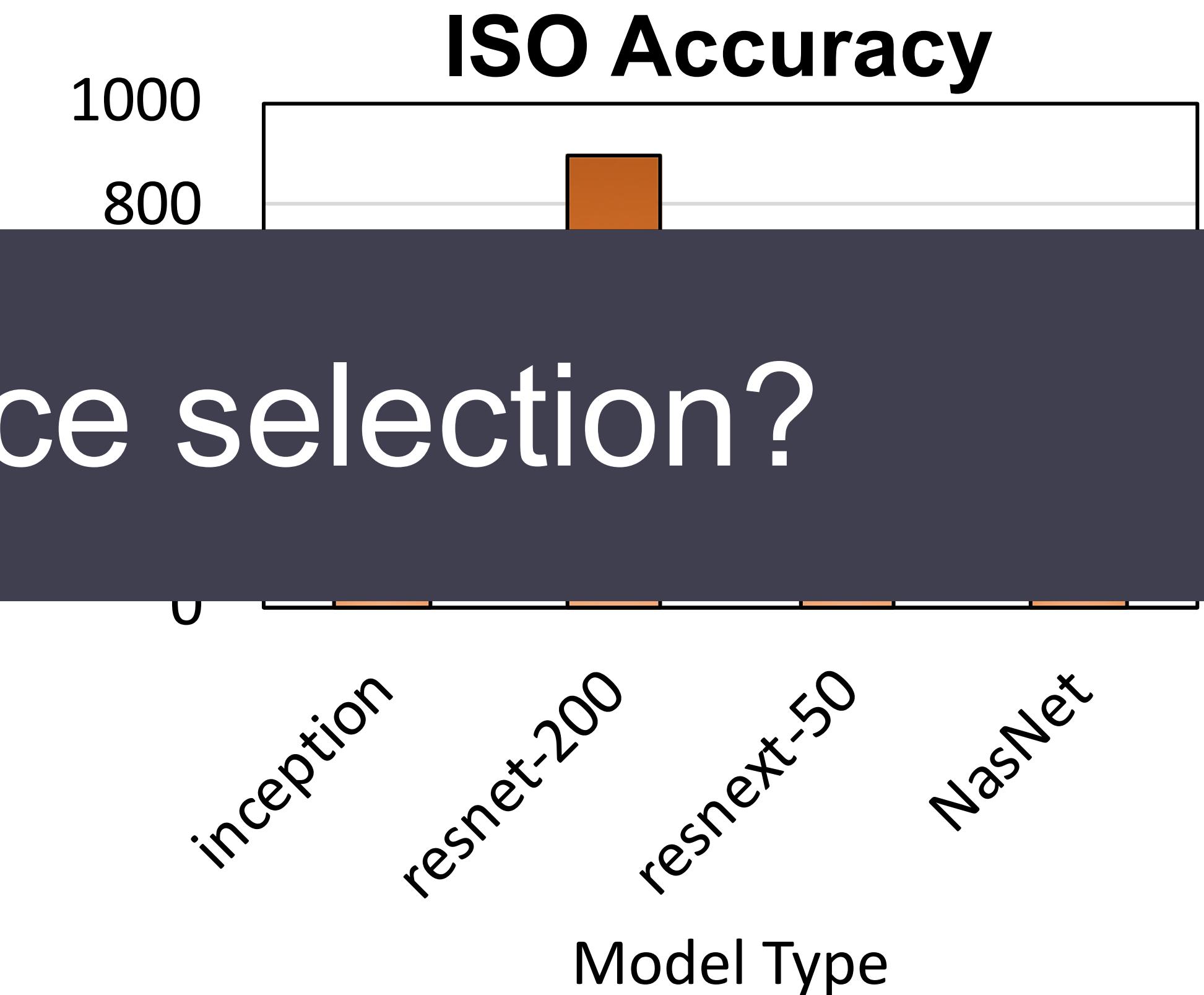
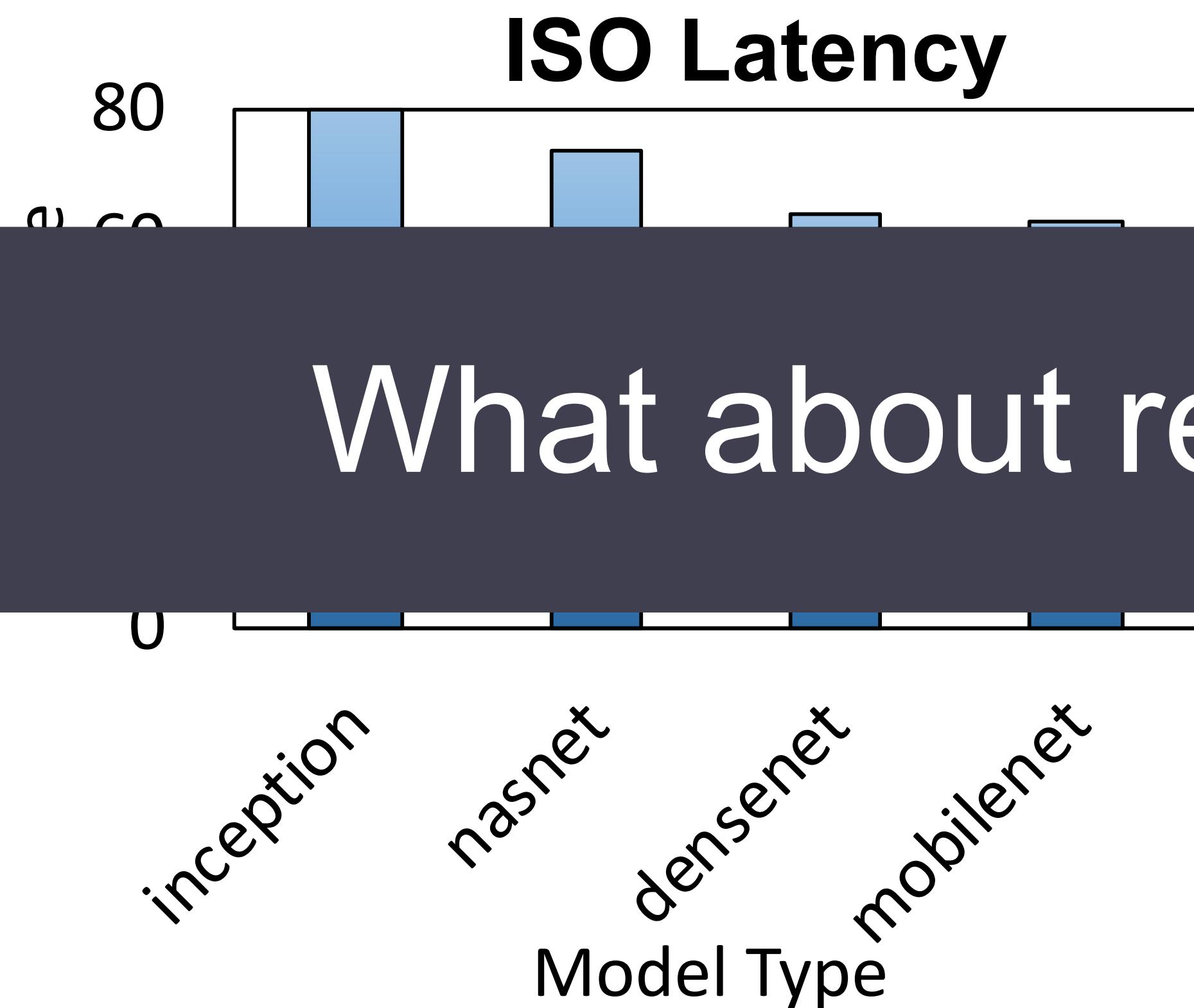
## Model Serving Challenges?



# Model Selection



# Model Selection



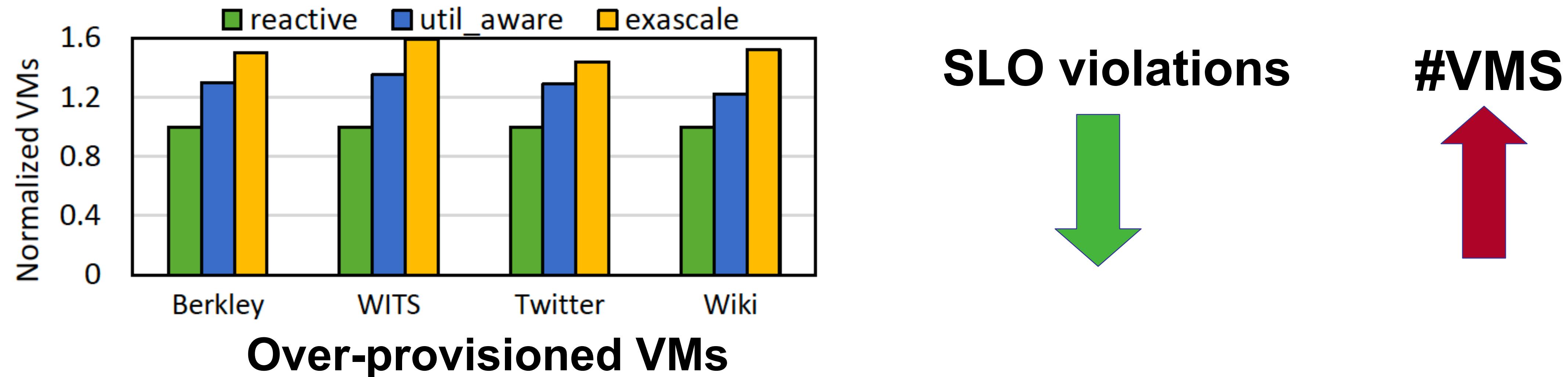
What about resource selection?

# Analyzing Prior Works

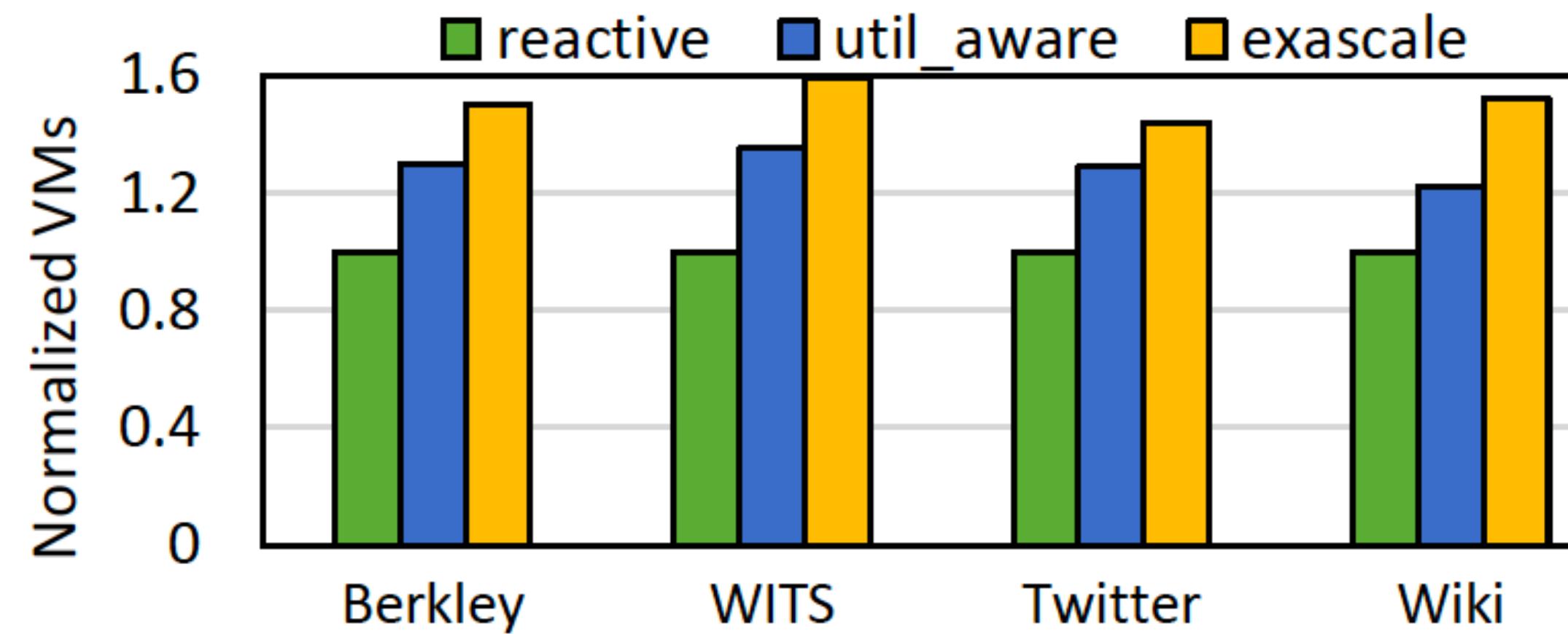
# Analyzing Prior Works



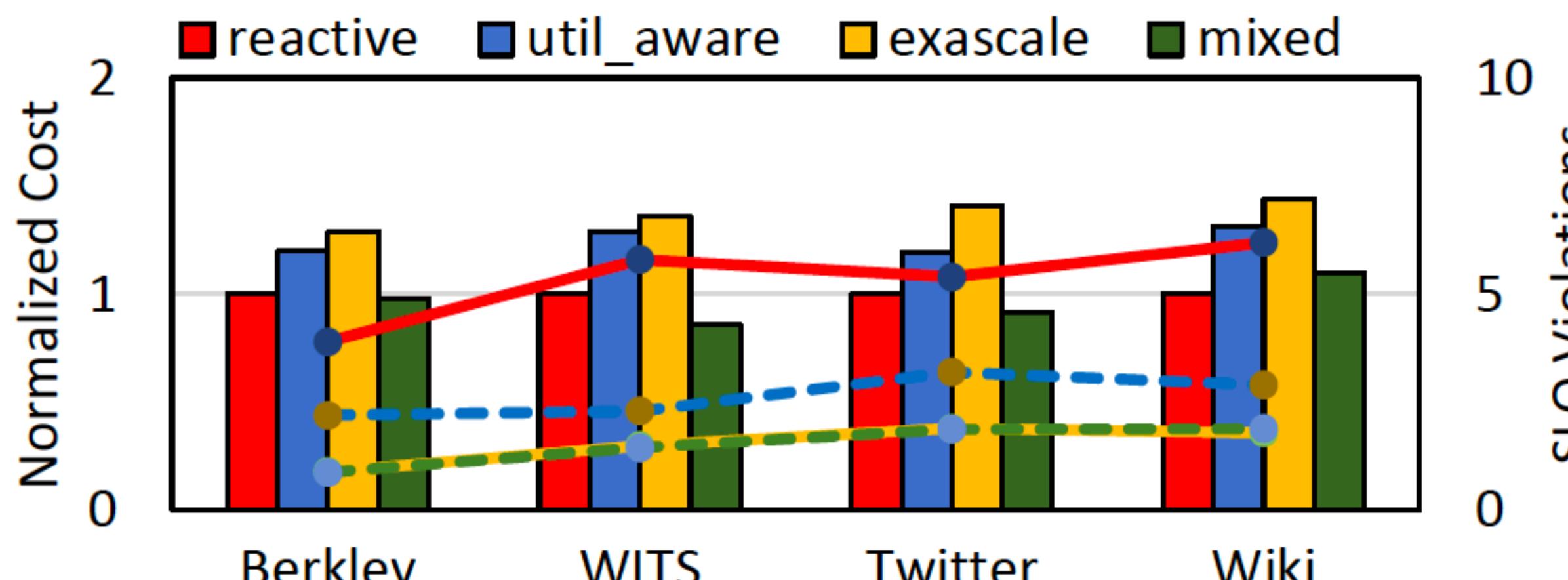
# Analyzing Prior Works



# Analyzing Prior Works

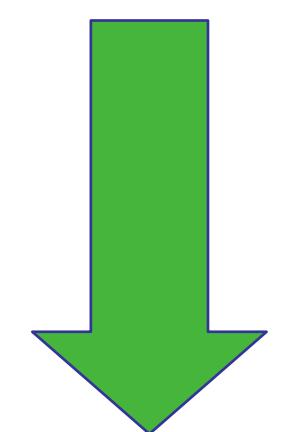


**Over-provisioned VMs**

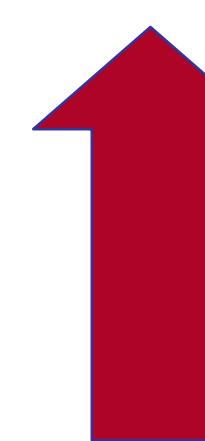


**Cost of Different Policies**

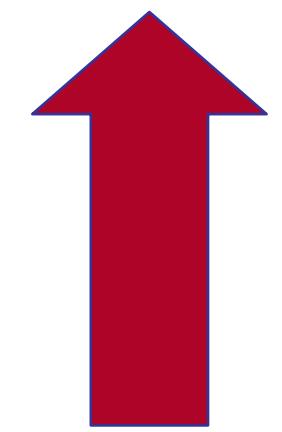
**SLO violations**



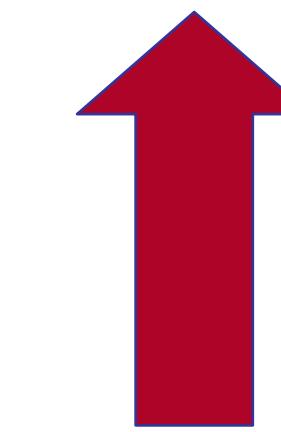
**#VMS**



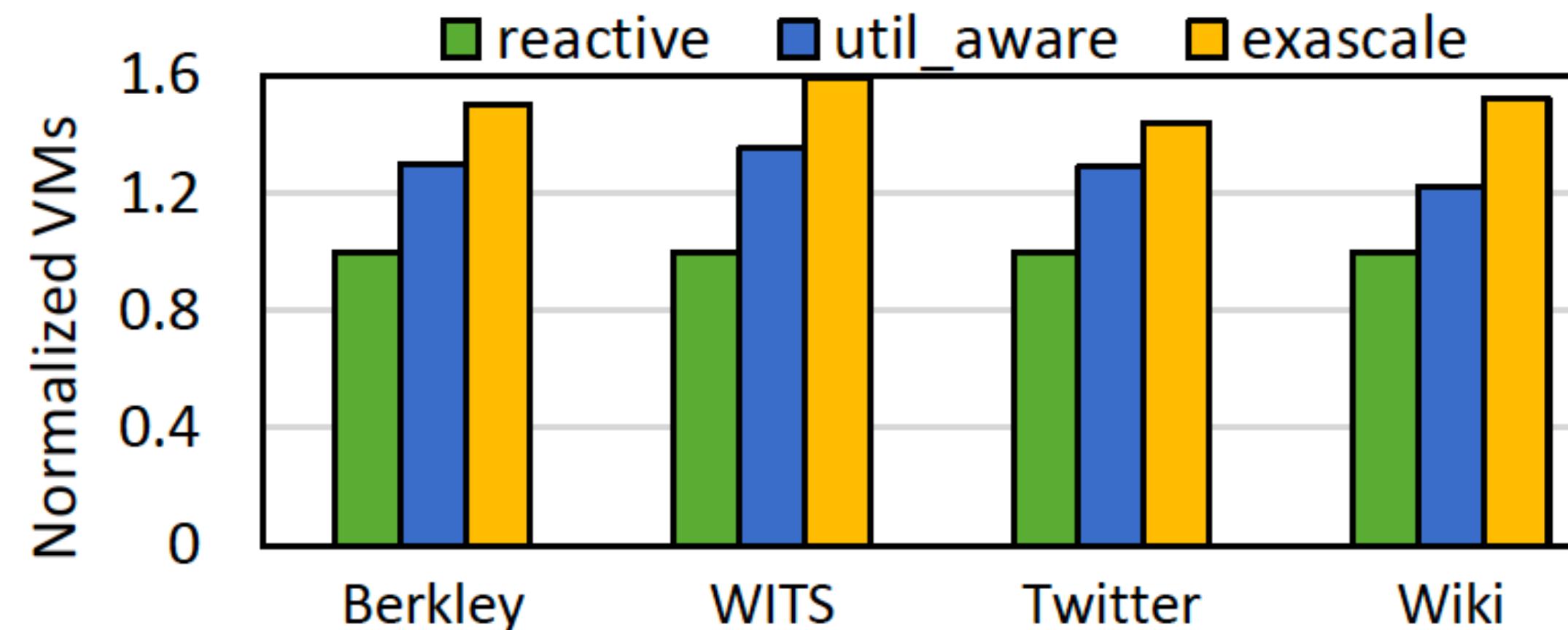
**VM-Cost**



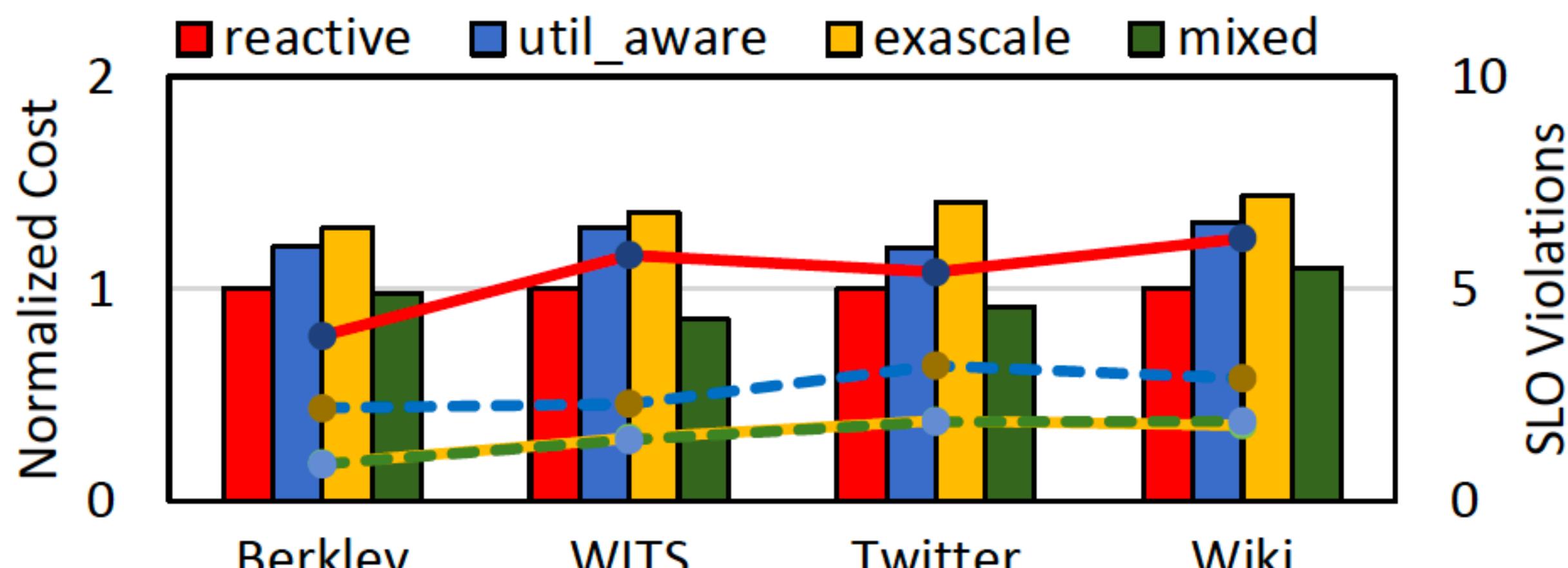
**#VMS**



# Analyzing Prior Works

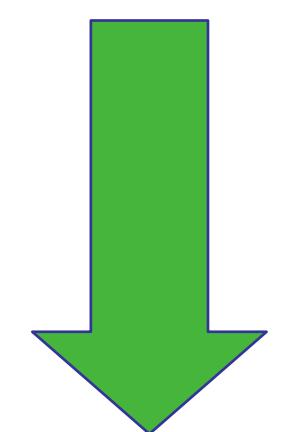


Over-provisioned VMs

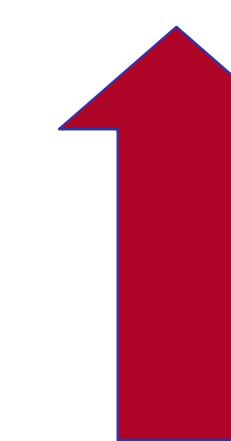


Cost of Different Policies

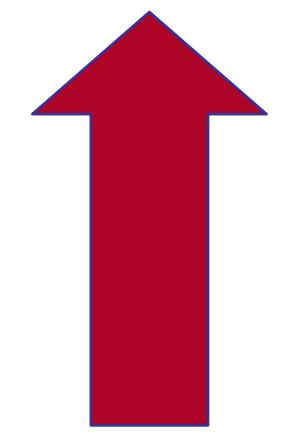
SLO violations



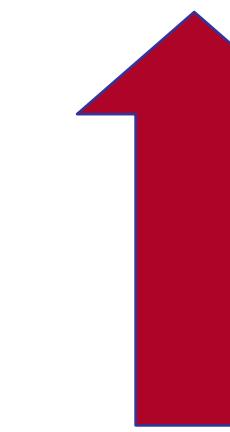
#VMS



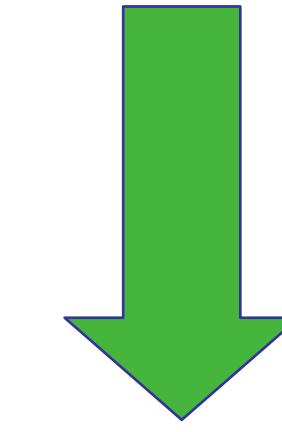
VM-Cost



#VMS

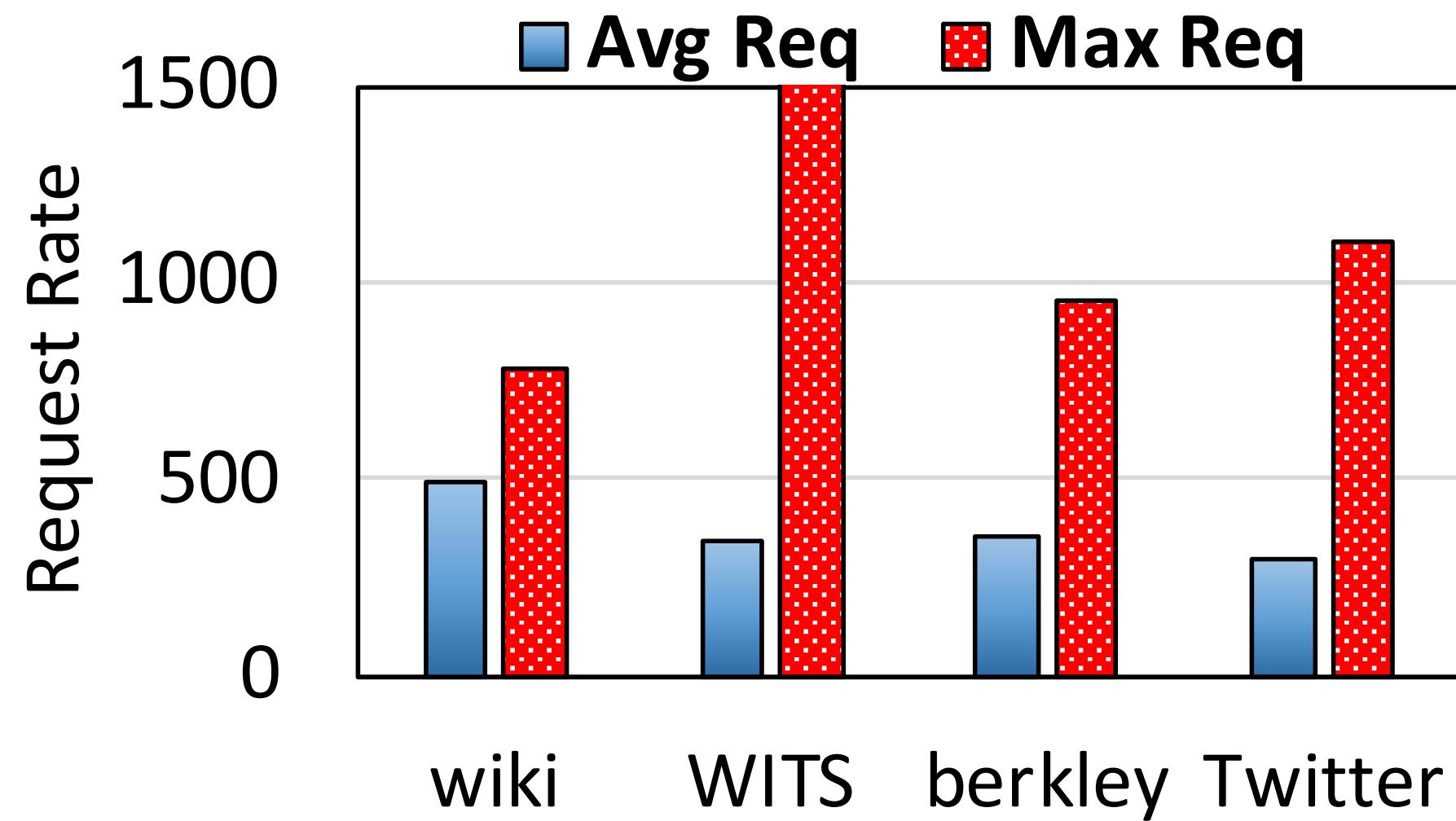


Mixed



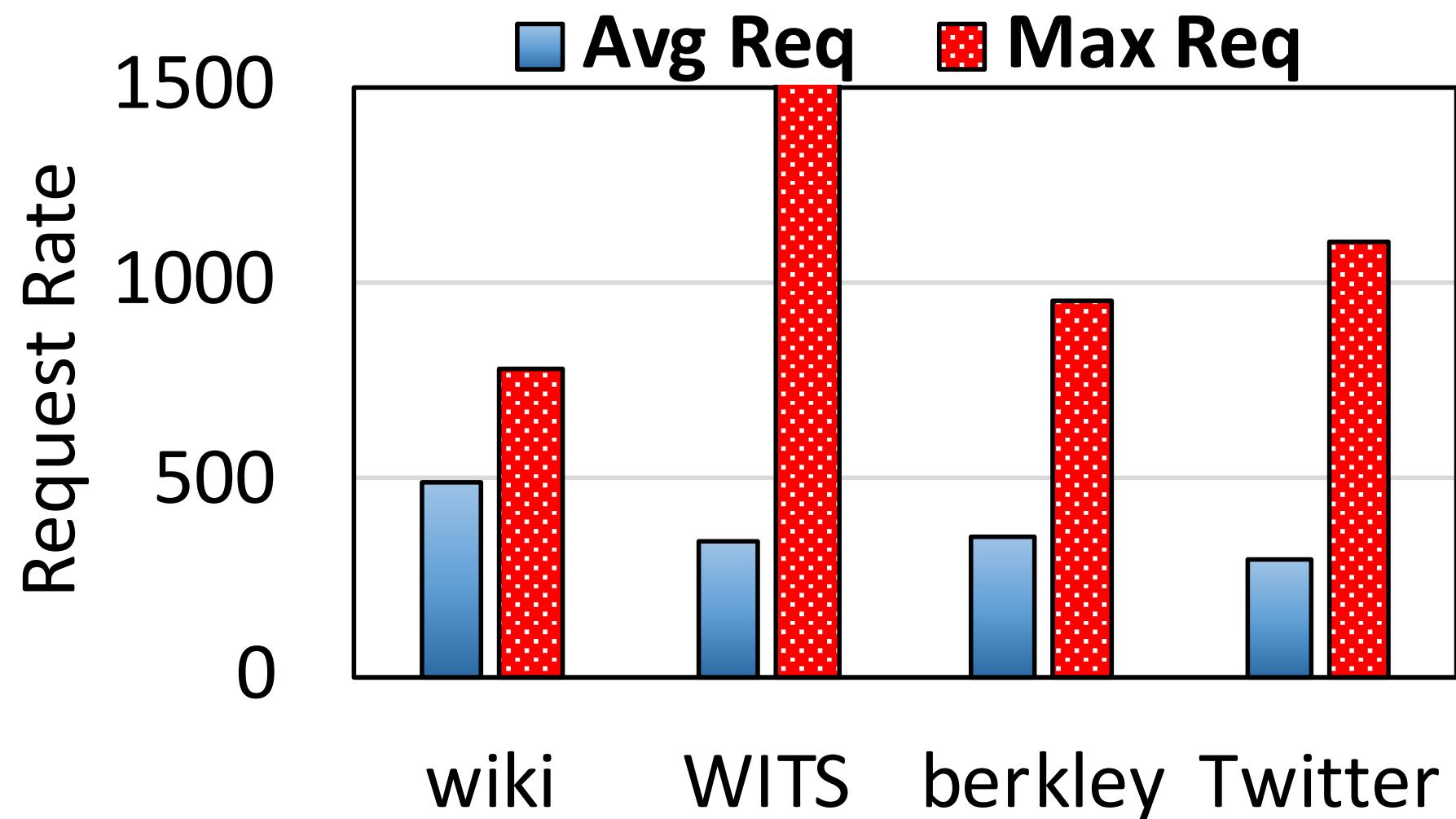
# Challenges with Serverless

## Arrival rate variability



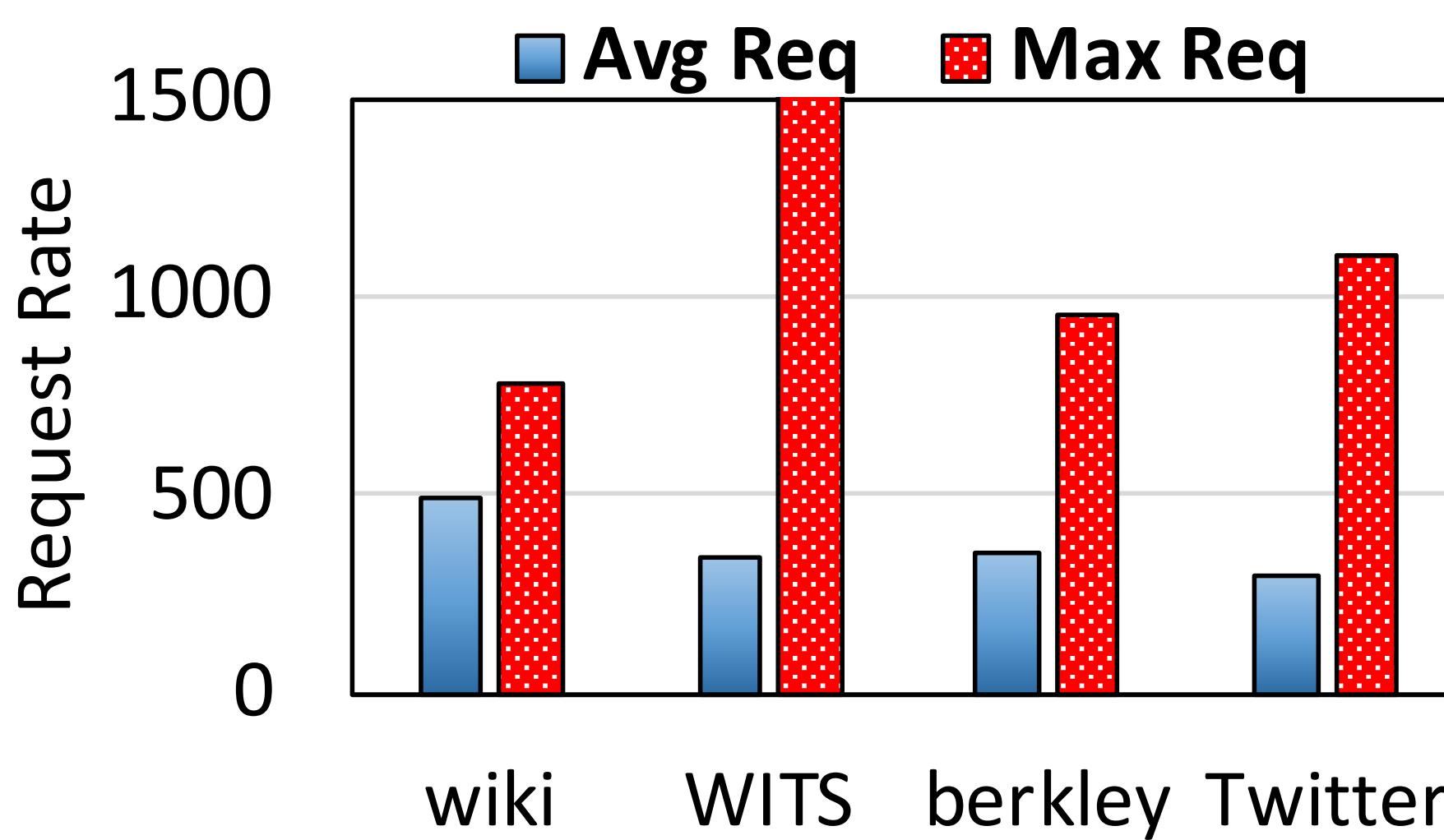
# Challenges with Serverless

## Arrival rate variability



# Challenges with Serverless

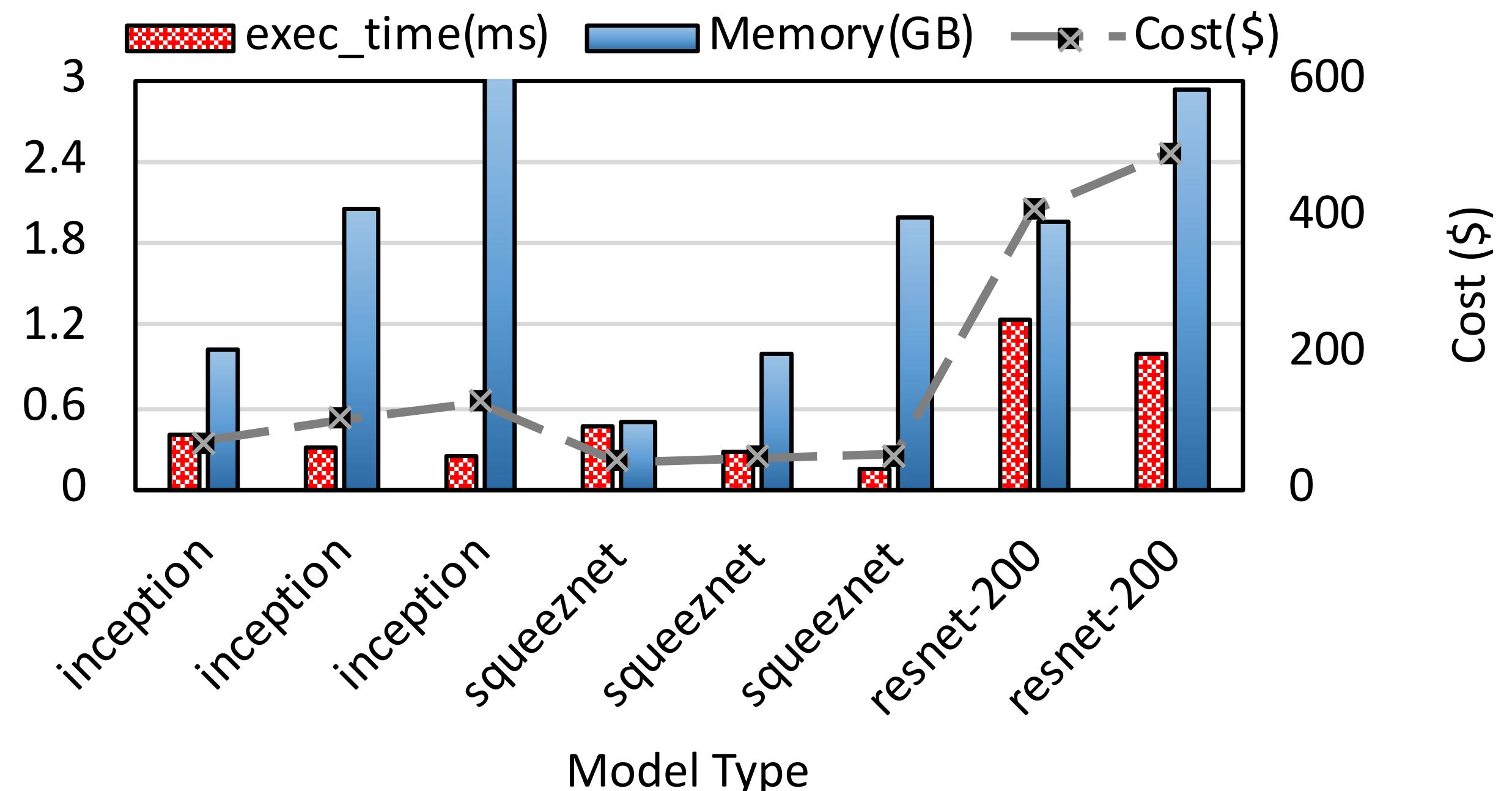
## Arrival rate variability



**Wiki**  
**Twitter**

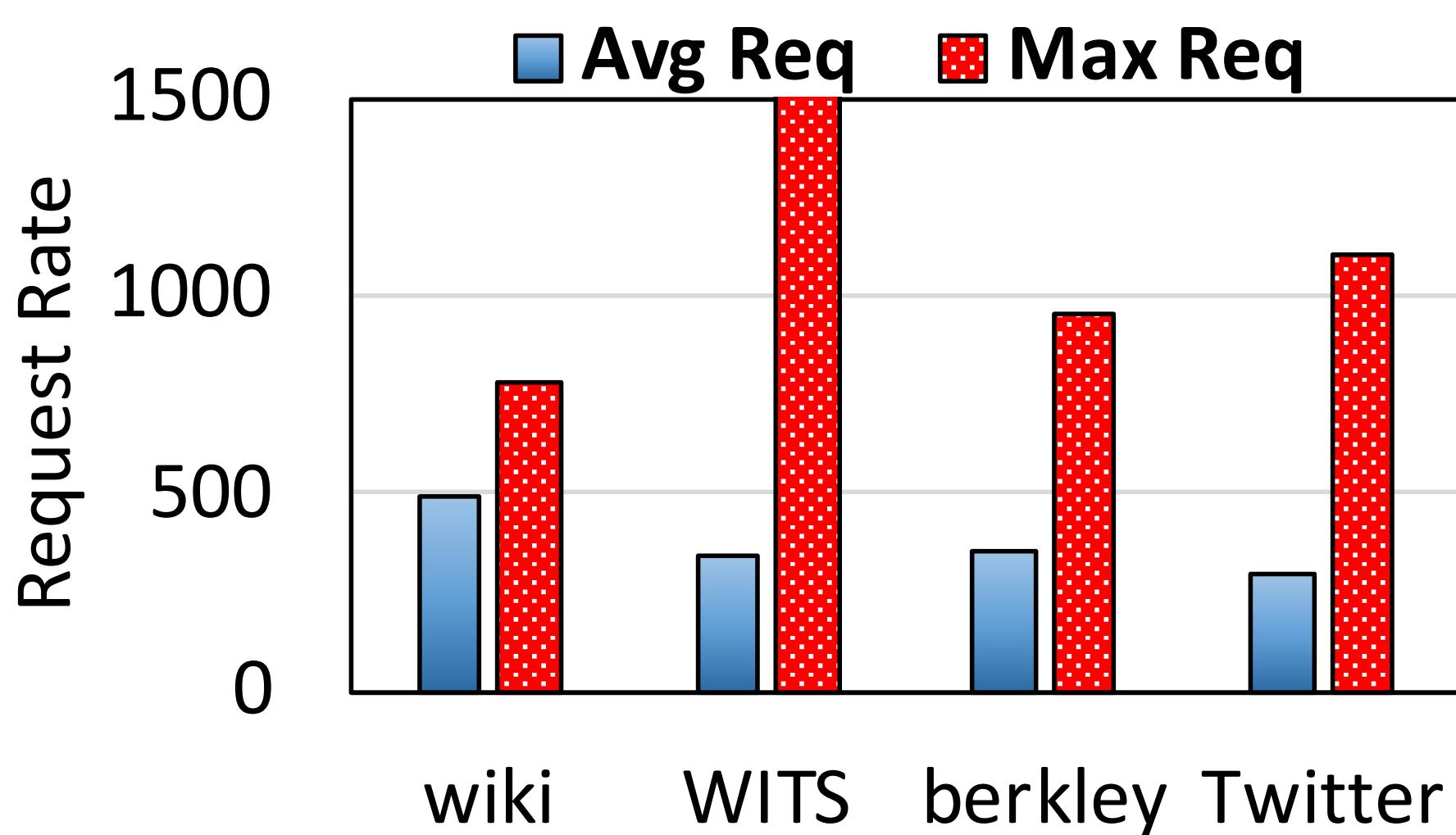


## Serverless Function Configuration



# Challenges with Serverless

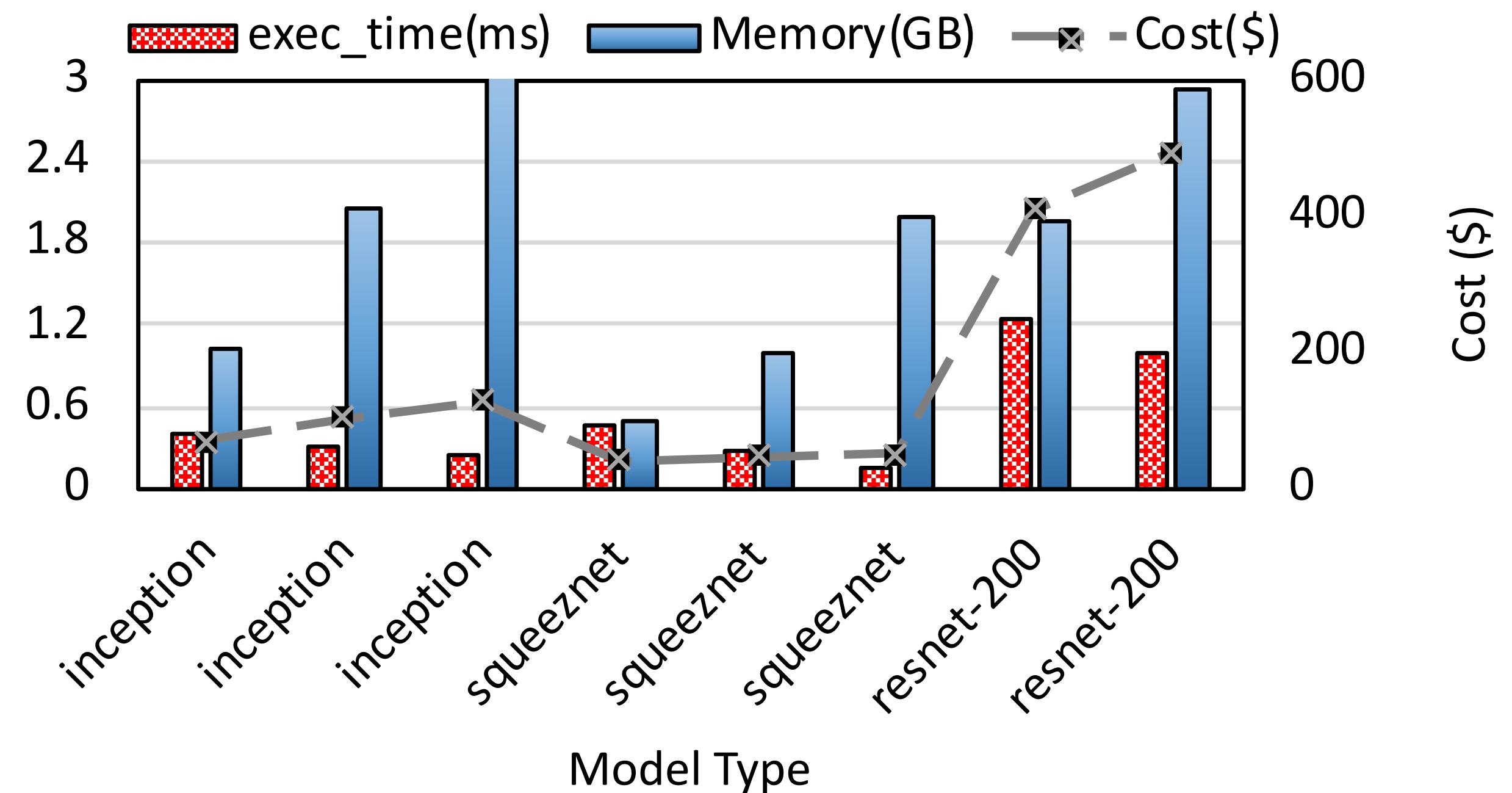
## Arrival rate variability



**Wiki**  
**Twitter**



## Serverless Function Configuration



Cost is **1.5x** higher for 0.2x lower latency

# What we need?

# What we need?

- How to make the users oblivious of model selection from the extensive pool of models?

# What we need?

- How to make the users oblivious of model selection from the extensive pool of models?
- How to right-size VMs and appropriately configure the serverless functions?

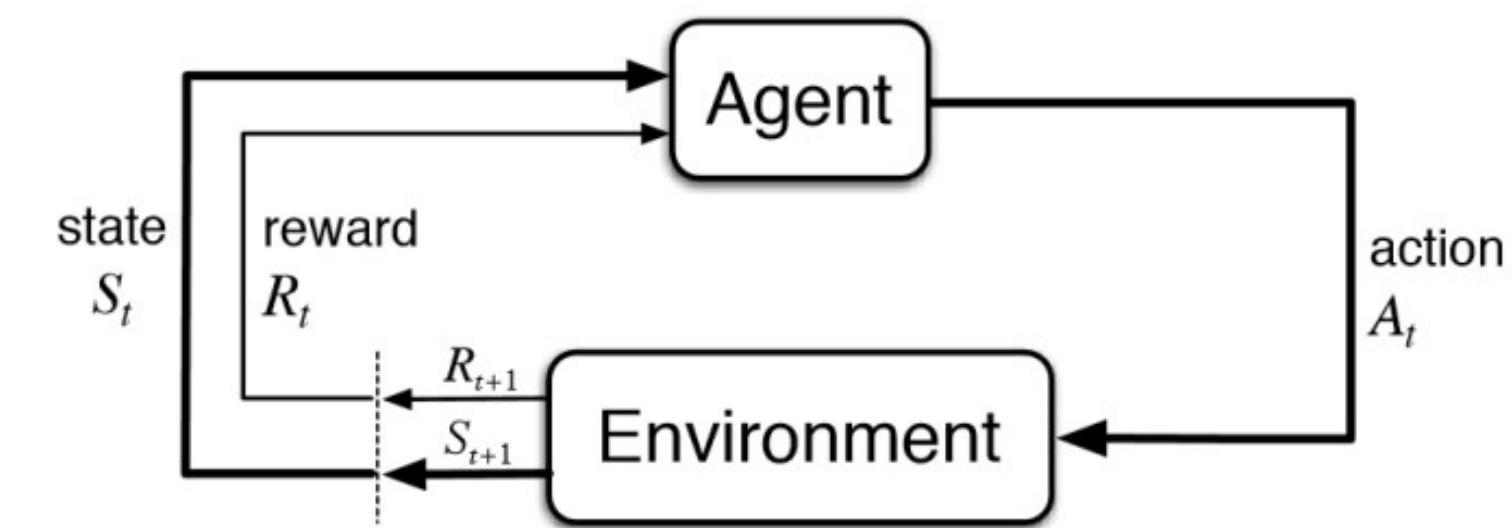
# What we need?

- How to make the users oblivious of model selection from the extensive pool of models?
- How to right-size VMs and appropriately configure the serverless functions?
- What is the right degree to combine serverless functions along with VMs for dynamic load?

# Proposed Solutions

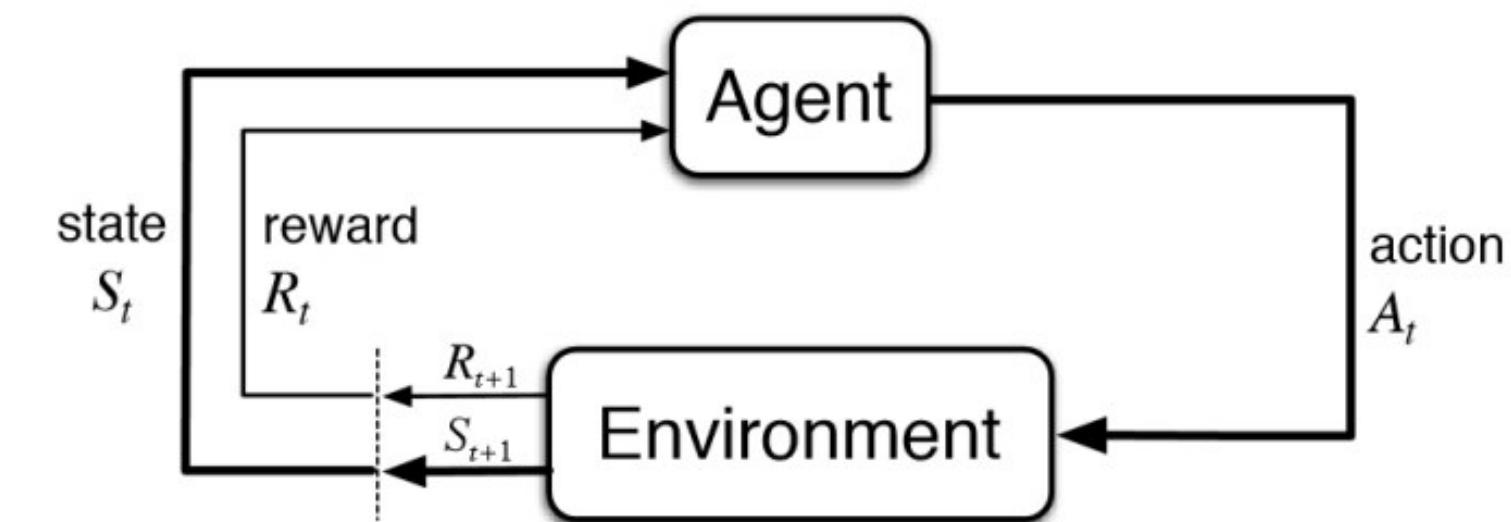
# Proposed Solutions

- Feedback-driven learning based model selection.

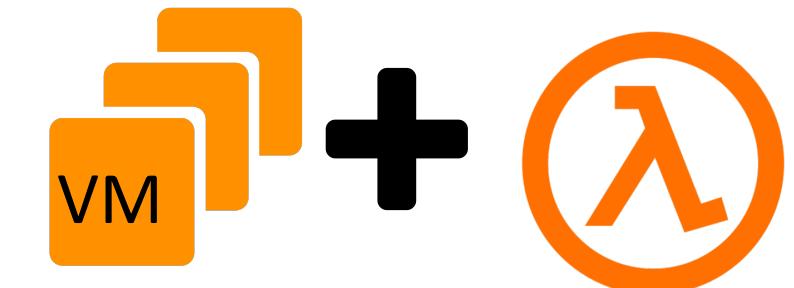


# Proposed Solutions

- Feedback-driven learning based model selection.
- Load-Based Procurement Policies



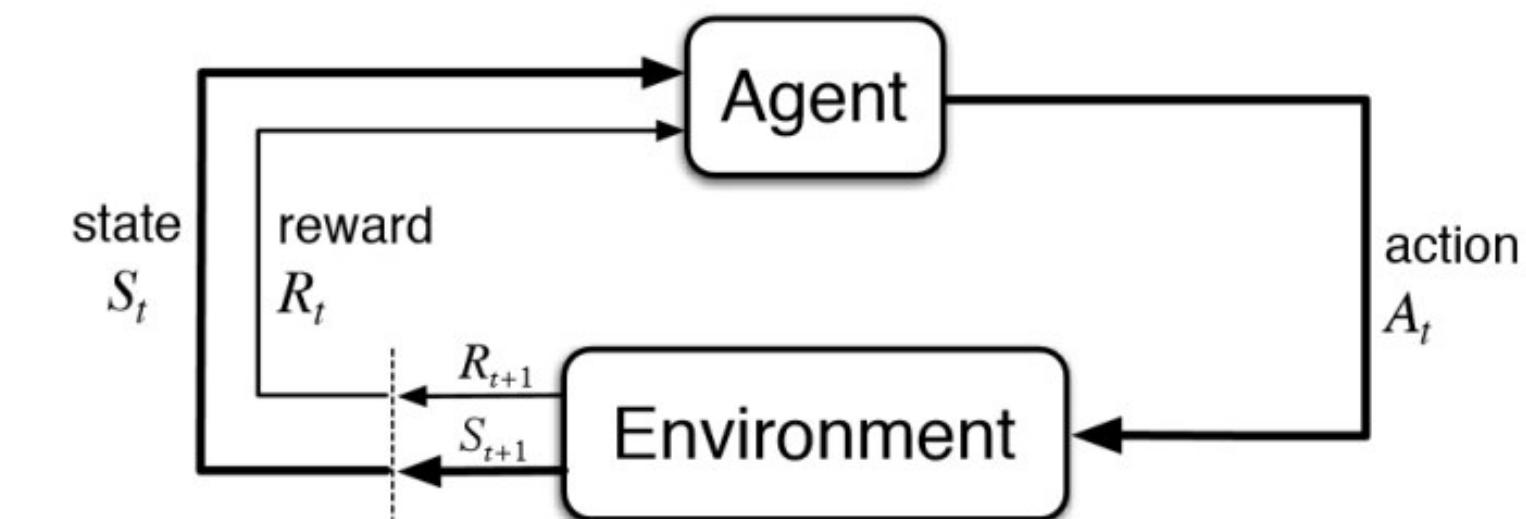
Static Load



Dynamic Load

# Proposed Solutions

- Feedback-driven learning based model selection.
- Load-Based Procurement Policies
- Provisioning latency and SLO aware resource selection



Static Load

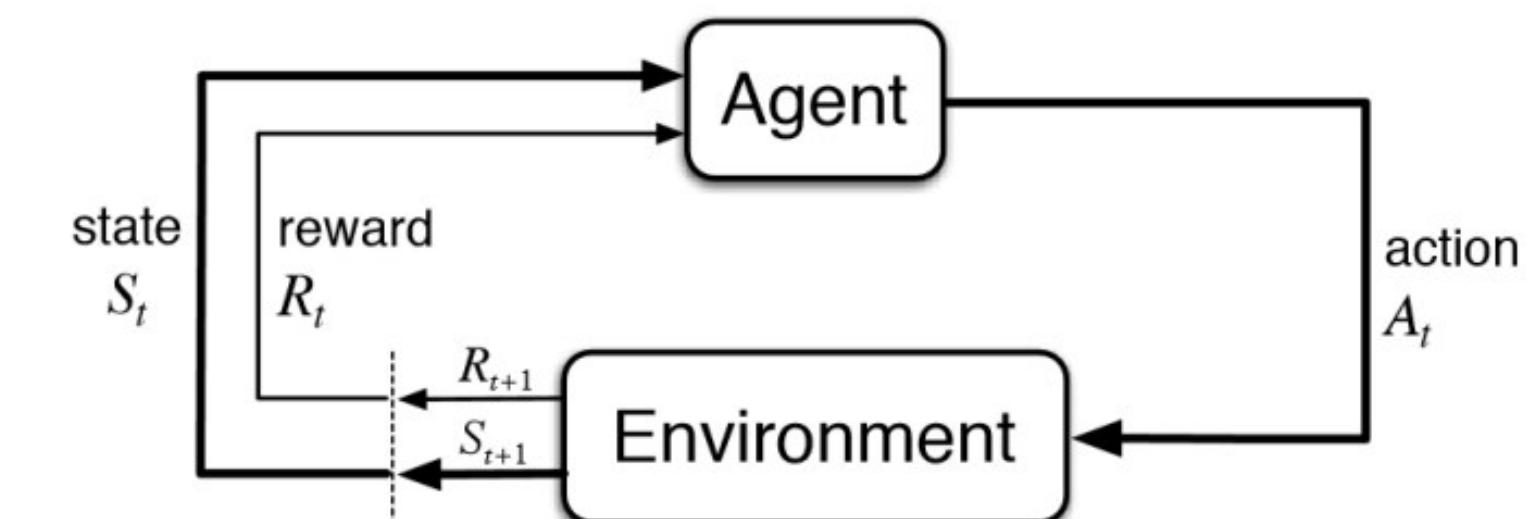


Dynamic Load

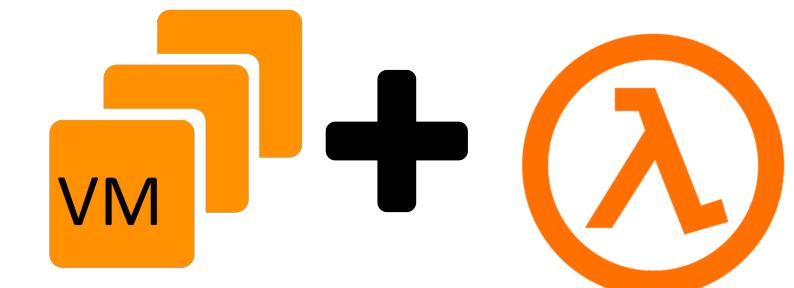


# Proposed Solutions

- Feedback-driven learning based model selection.
- Load-Based Procurement Policies
- Provisioning latency and SLO aware resource selection
- Dynamic serverless configurations.



Static Load

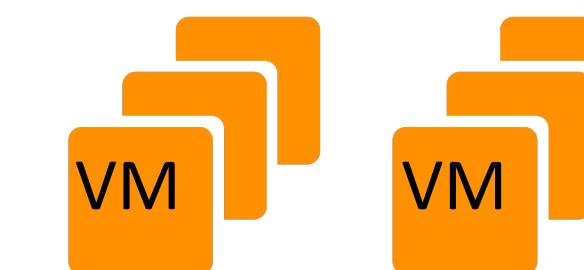
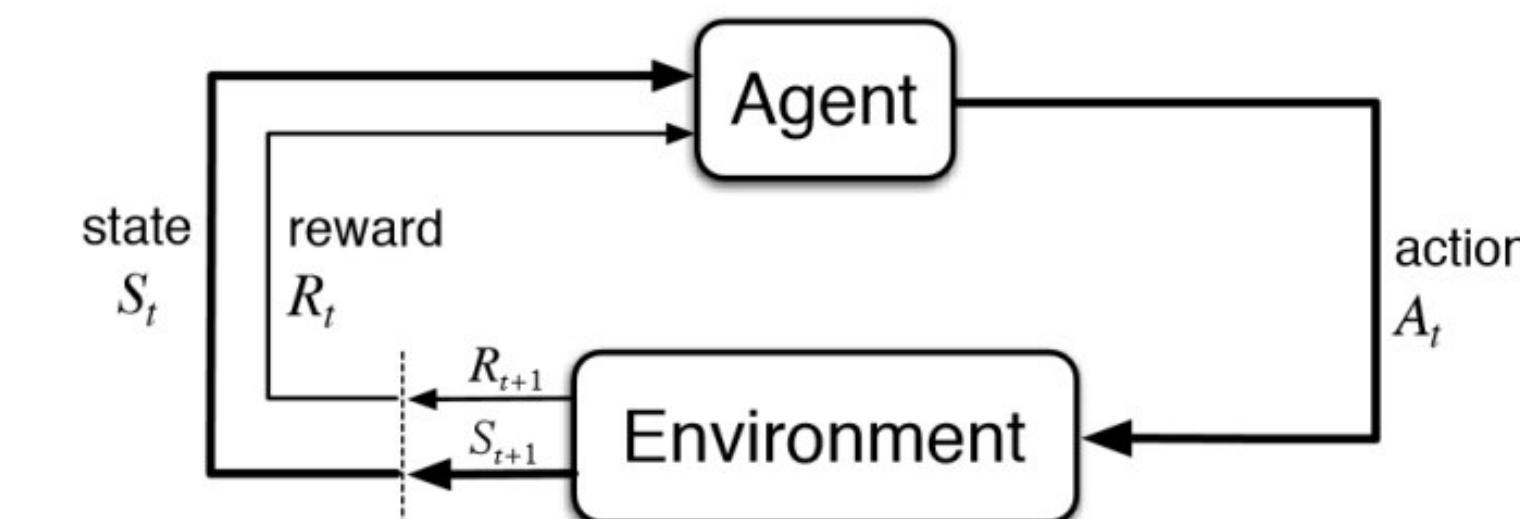


Dynamic Load

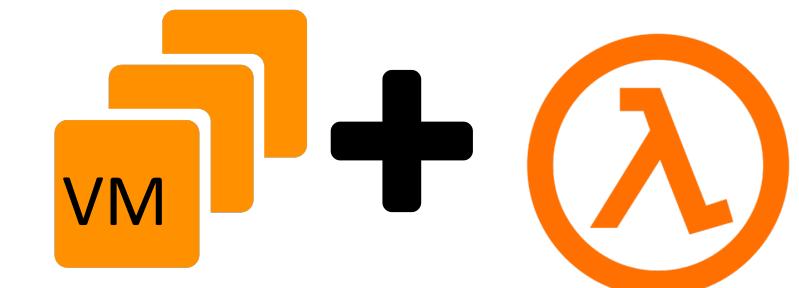


# Proposed Solutions

- Feedback-driven learning based model selection.
- Load-Based Procurement Policies
- Provisioning latency and SLO aware resource selection
- Dynamic serverless configurations.



Static Load

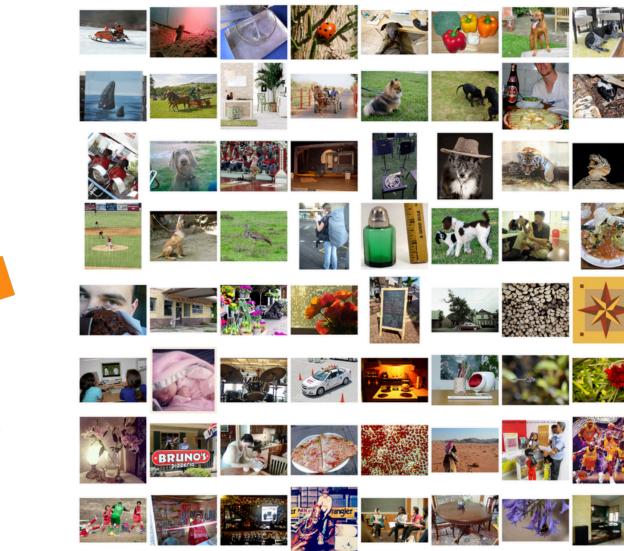


Dynamic Load



# Implementation and Evaluation

- Mxnet Framework.
- AWS resources.
- Pretrained ML models on imagenet dataset.



# Implementation and Evaluation

- Mxnet Framework.
- AWS resources.
- Pretrained ML models on imagenet dataset.



Query Type	Memory Required (GB)	Memory Allocated (GB)	Average Execution (ms)	Requests per vCPU for VMs
Caffenet	1.024	3.072	300	4
Googlenet	0.456	2.048	450	3
Squeezezenet	0.154	2.048	130	6
Resnet-18	0.304	3.072	320	3
Resnet-200	1.024	3.072	956	1
Resnext-50	0.645	3.072	560	2

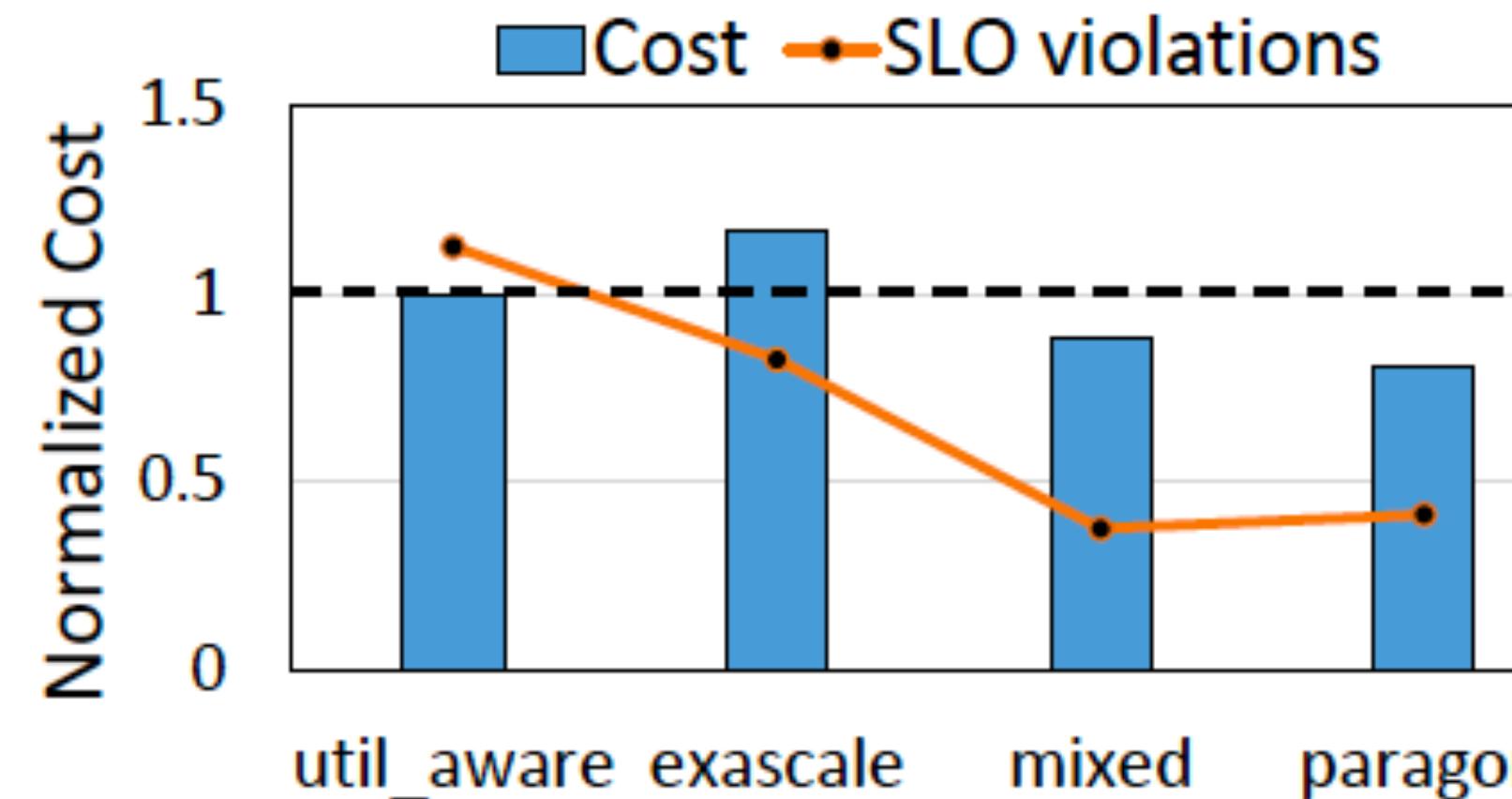
# Implementation and Evaluation

- Mxnet Framework.
- AWS resources.
- Pretrained ML models on imagenet dataset.

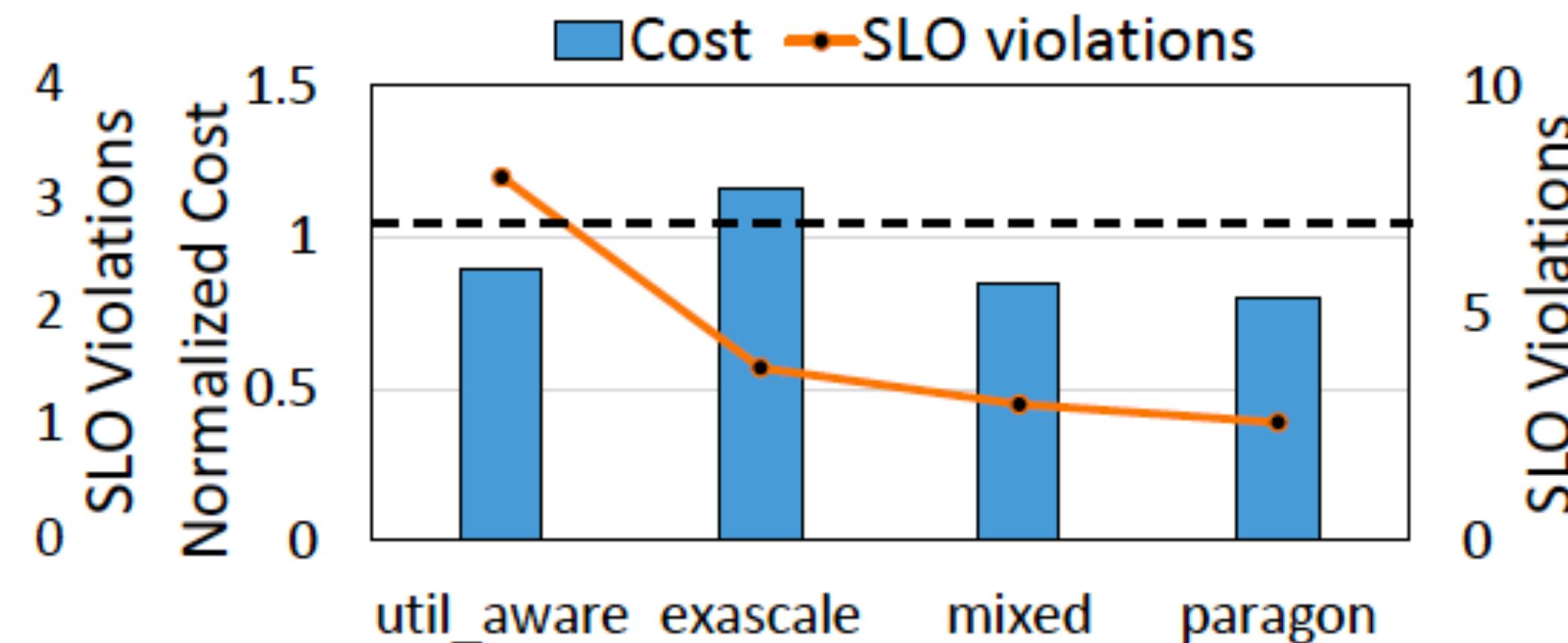


Query Type	Memory Required (GB)	Memory Allocated (GB)	Average Execution (ms)	Requests per vCPU for VMs
Caffenet	1.024	3.072	300	4
Googlenet	0.456	2.048	450	3
Squeezezenet	0.154	2.048	130	6
Resnet-18	0.304	3.072	320	3
Resnet-200	1.024	3.072	956	1
Resnext-50	0.645	3.072	560	2

# Initial Results



(a) Workload-1: Berkeley Trace.



(b) Workload-1: WITS Trace.

**60% less SLO Violations.**  
**10% reduction in deployment costs**

