Maximizing Resource Efficiency for Next Generation Cloud Platforms

Jashwant Raj Gunasekaran

Advisors: Dr. Mahmut T. Kandemir & Dr. Chita R. Das

High Performance Computing Lab

Dissertation Defense

May 6, 2021

PennState
College of Engineering

PennState
High Performance
Computing Lab

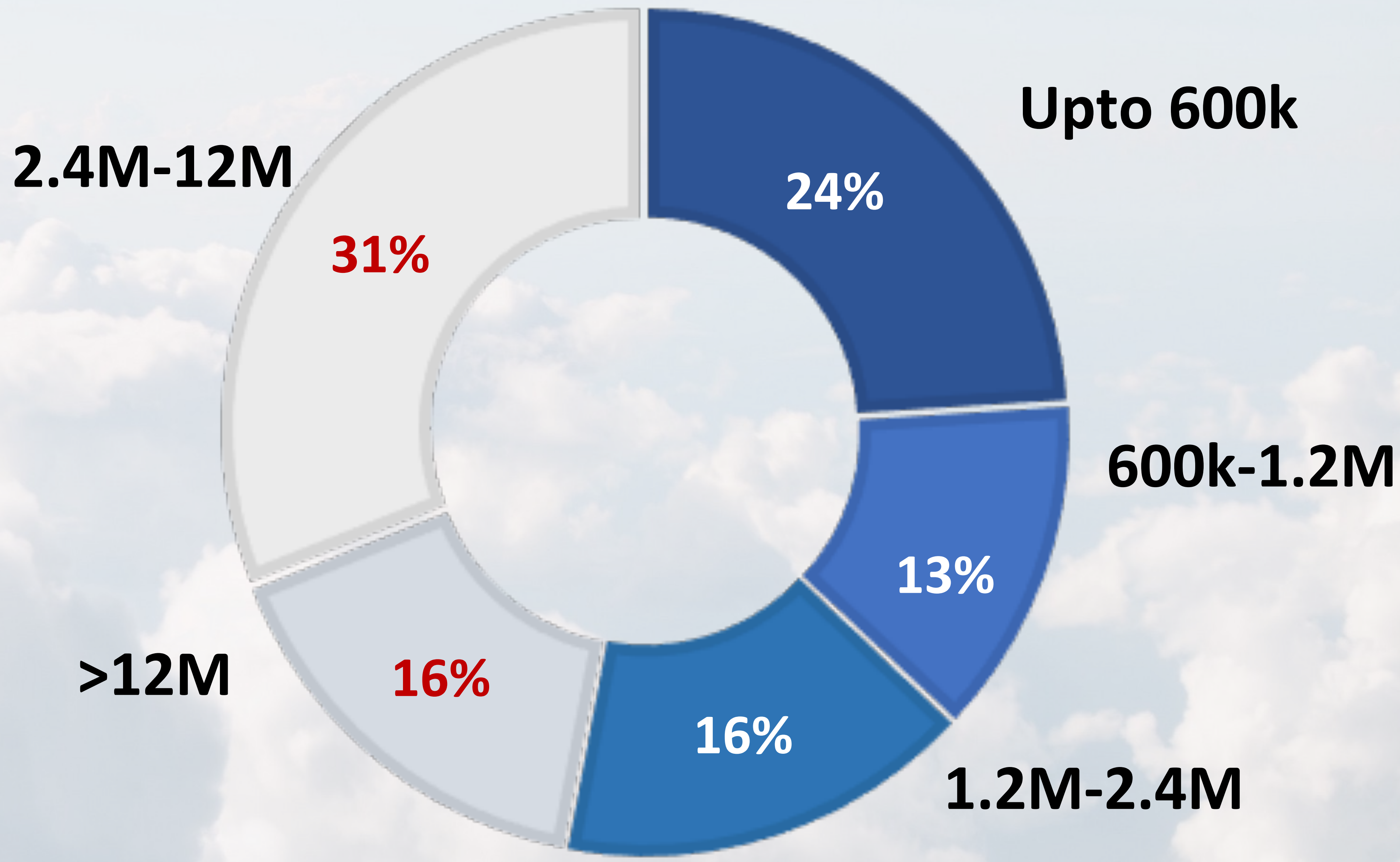
Research Philosophy

Cloud is about *how* you do computing,
not *where* you do computing!

Paul Maritz, Former CEO, Vmware

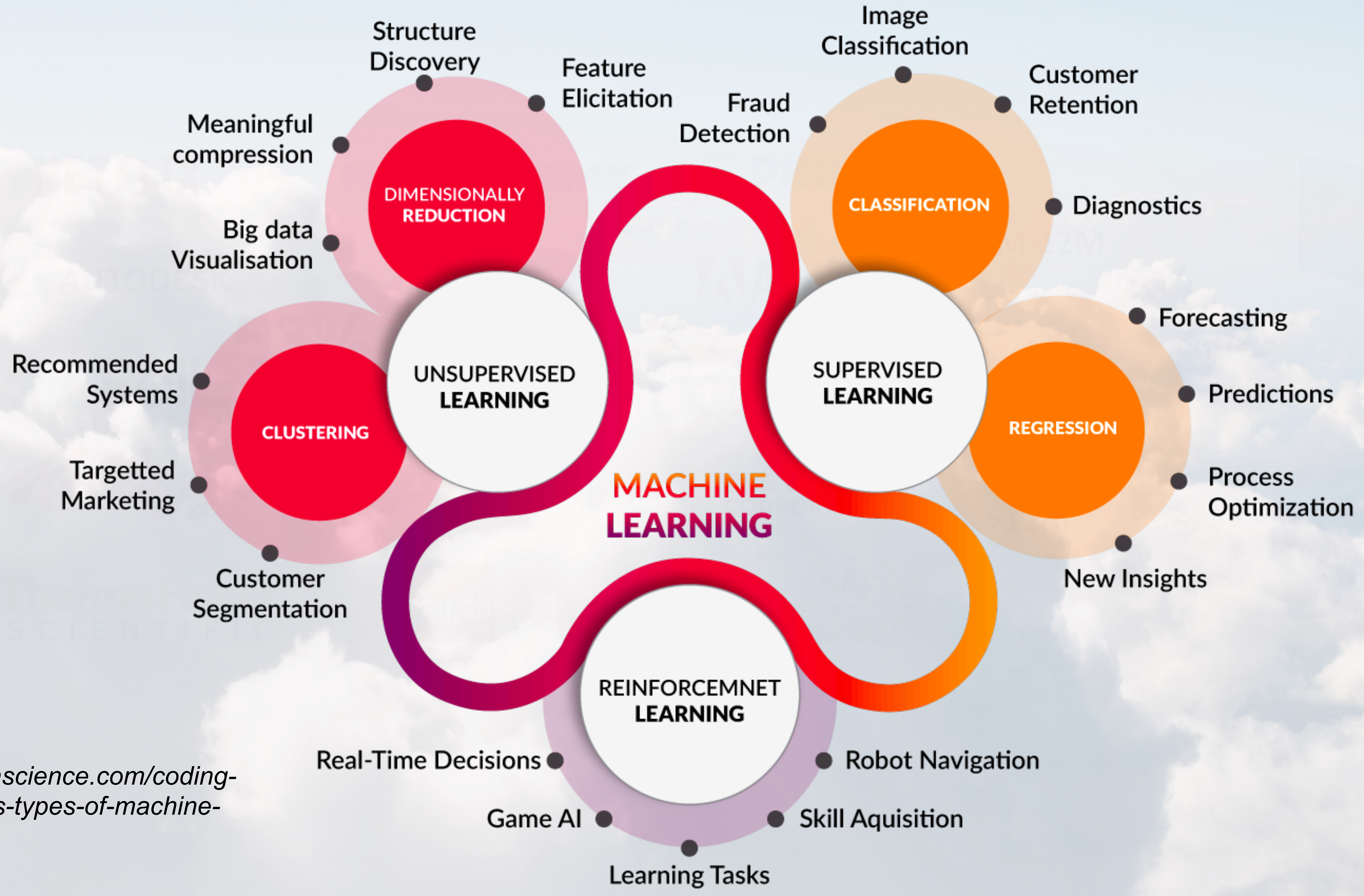
PUSH FOR MORE CLOUD ADOPTION

53%



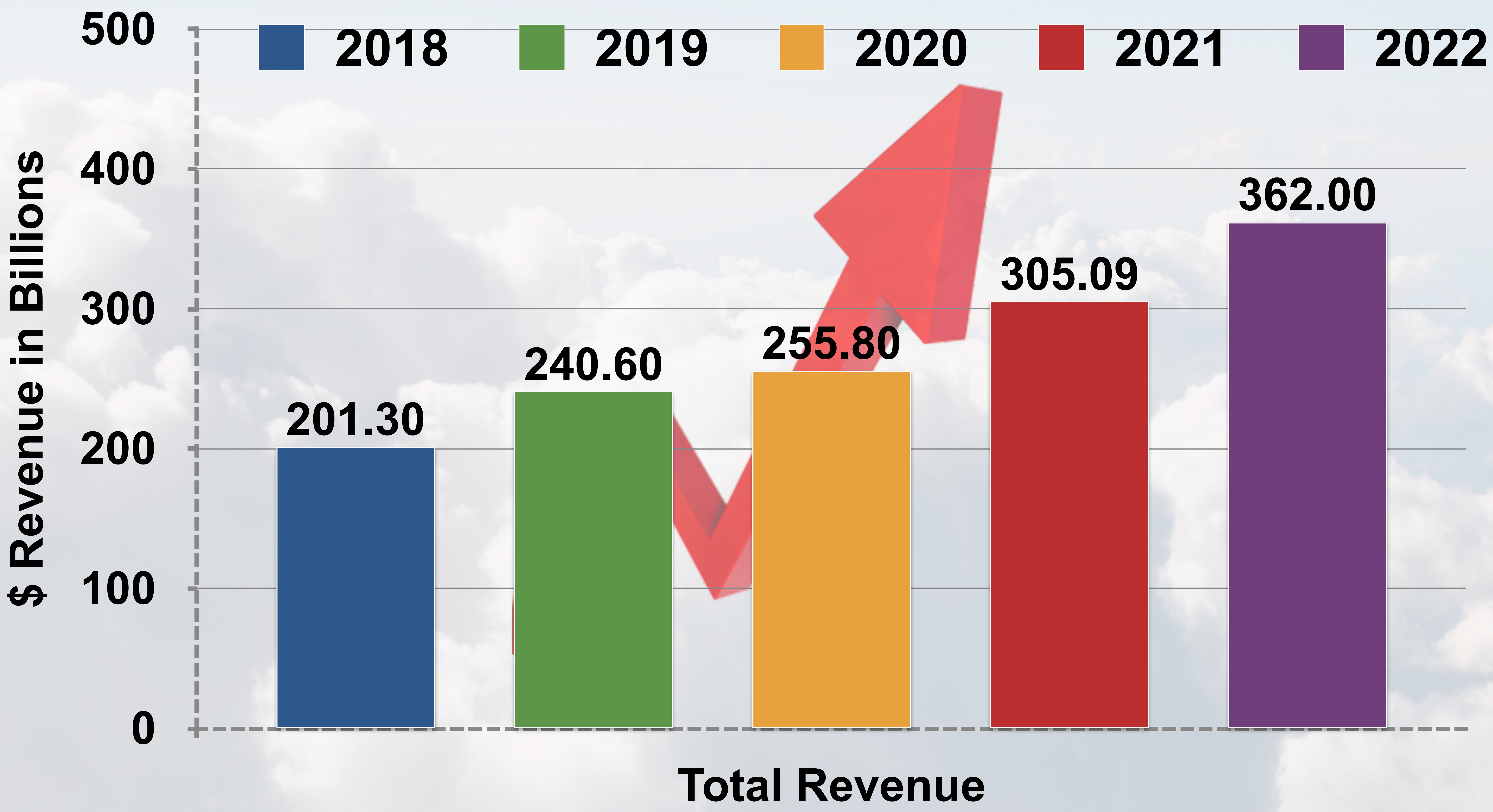
Source: Flexera 2020 Cloud

PUSH FOR MORE CLOUD ADOPTION



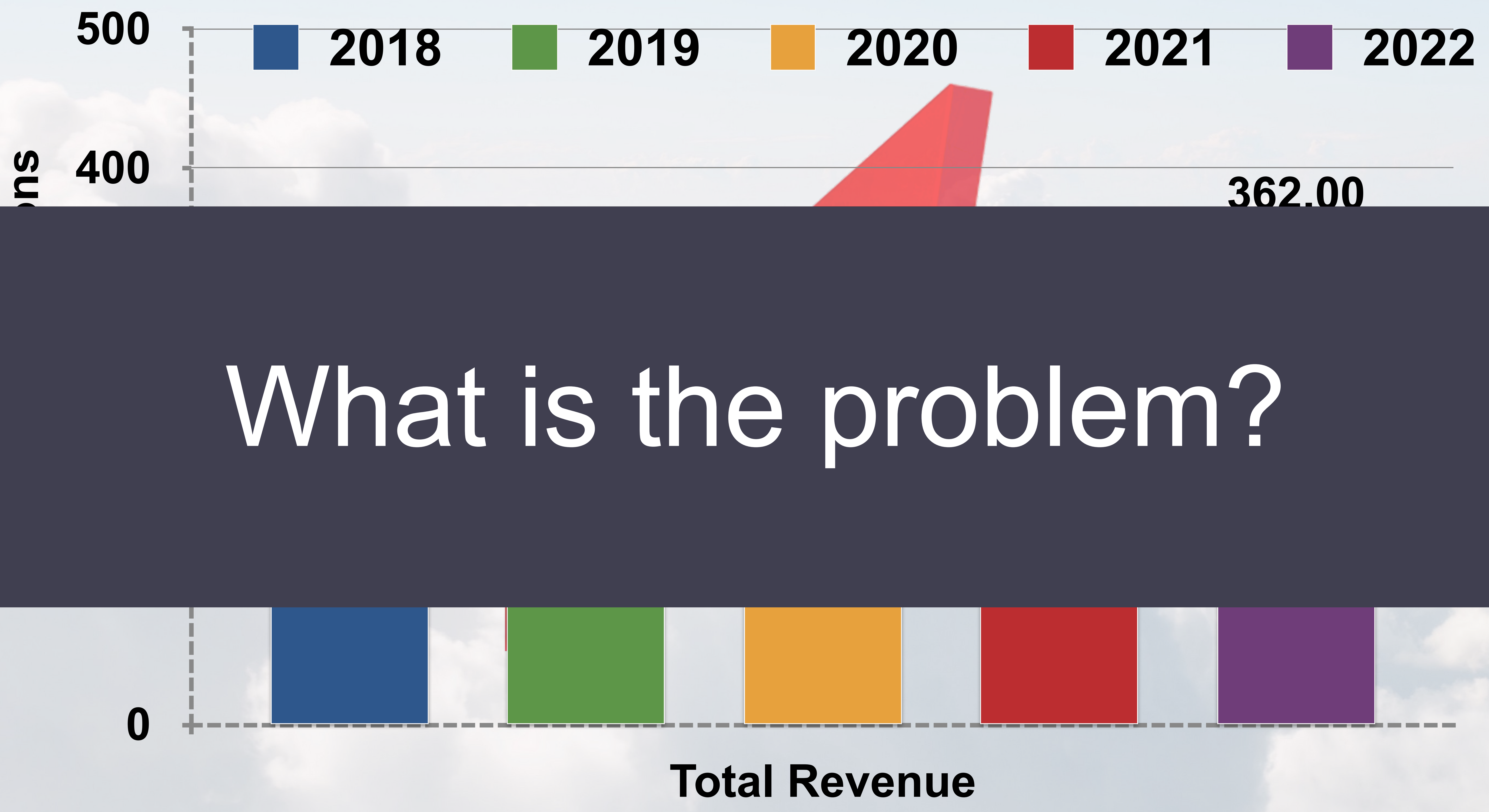
Source: https://towardsdatascience.com/coding-deep-learning-for-beginners-types-of-machine-

Public Cloud Revenue



Source: Gartner

Public Cloud Revenue



What is the problem?

Source: Gartner

We just got our cloud bills this month

I don’t have the money to pay this time! I should ask **Dr Kandemir’s** Pcard!

I forgot to turn off my VMs! **Dr Kesidis** will be furious!

I chose the wrong tier! Wasted **Dr. Bhuvan’s** grant money!

I exceeded my free quota! Will **Dr Das** help me?

Not Only Grad Students

We just got our cloud bills this month

I don’t have the money to pay this time! I should ask **Dr Kandemir’s** Pcard!

I chose the wrong tier! Wasted **Dr. Bhuvan’s** grant money!

Why is cost important?

But Also Cloud Clients

PennState College of Engineering

PennState High Performance Computing Lab

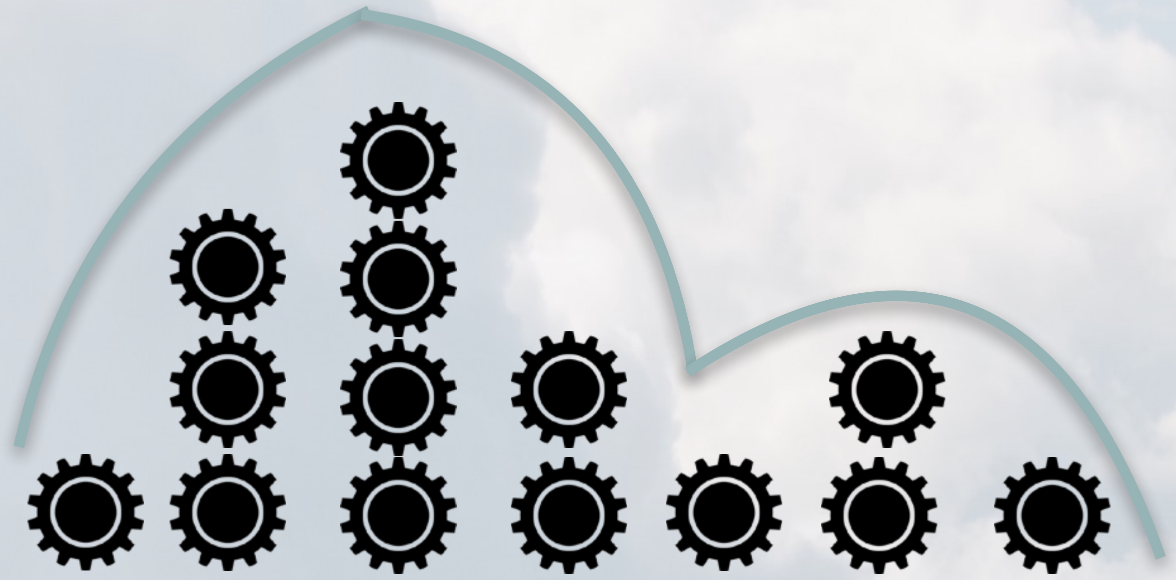
Tenant-side Problems

~35%

~77%

~73%

Resource Selection



AutoScaling

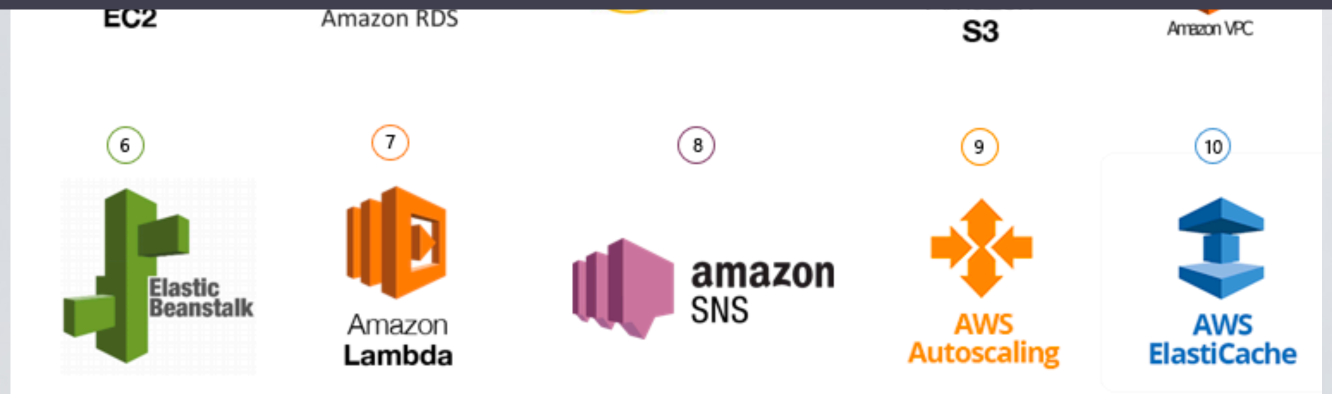
TENANT-SIDE PROBLEMS

~35%

~77%

~73%

What about providers?



Resource Selection

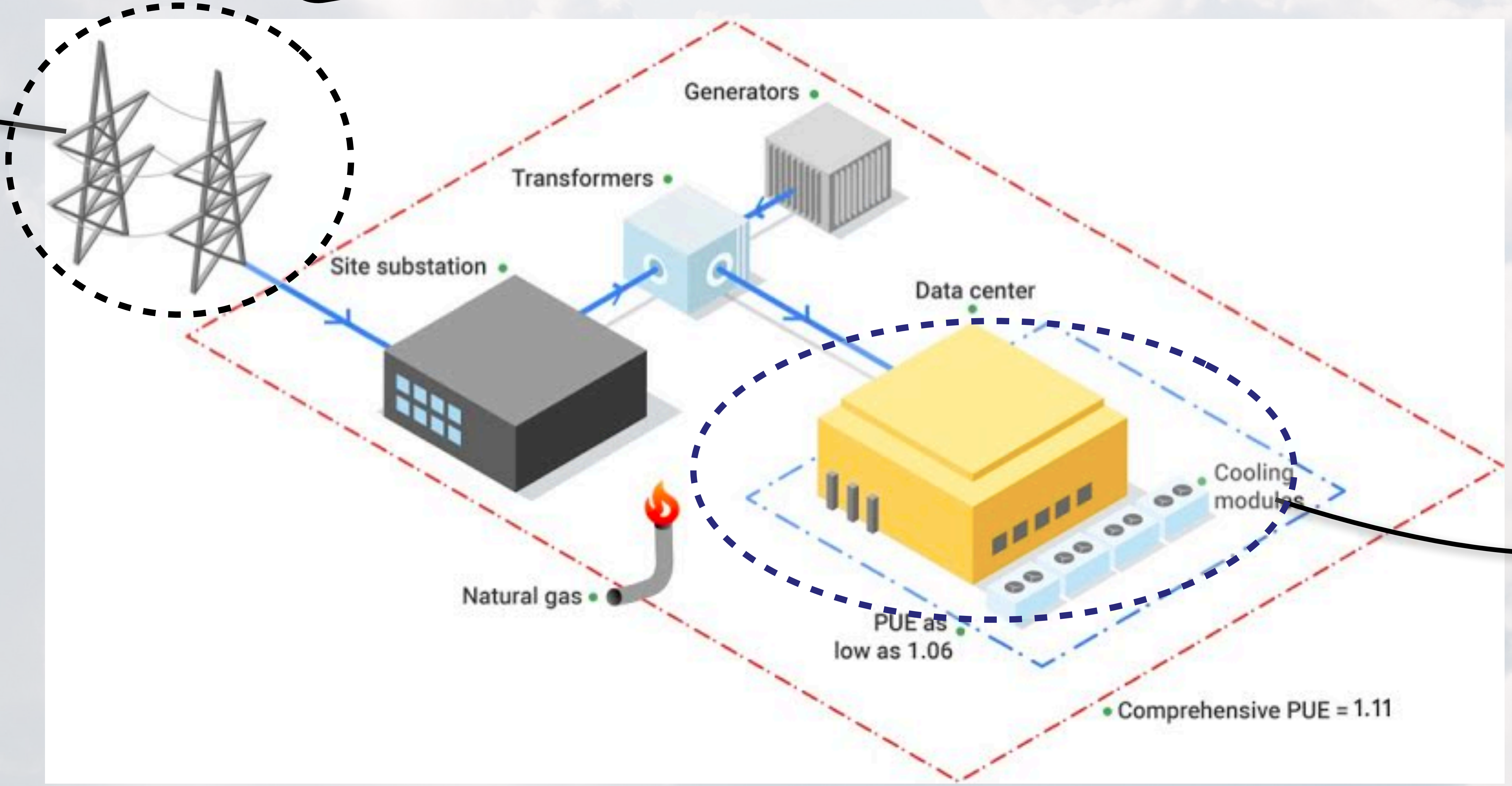
AutoScaling

PROVIDER EXPENDITURE

OpEx

CapEx > ~5x OpEx !

CapEx



Amortized cost of 50000 servers in Microsoft datacenter. Source: The Cost of Cloud, ACM SIGCOMM'09

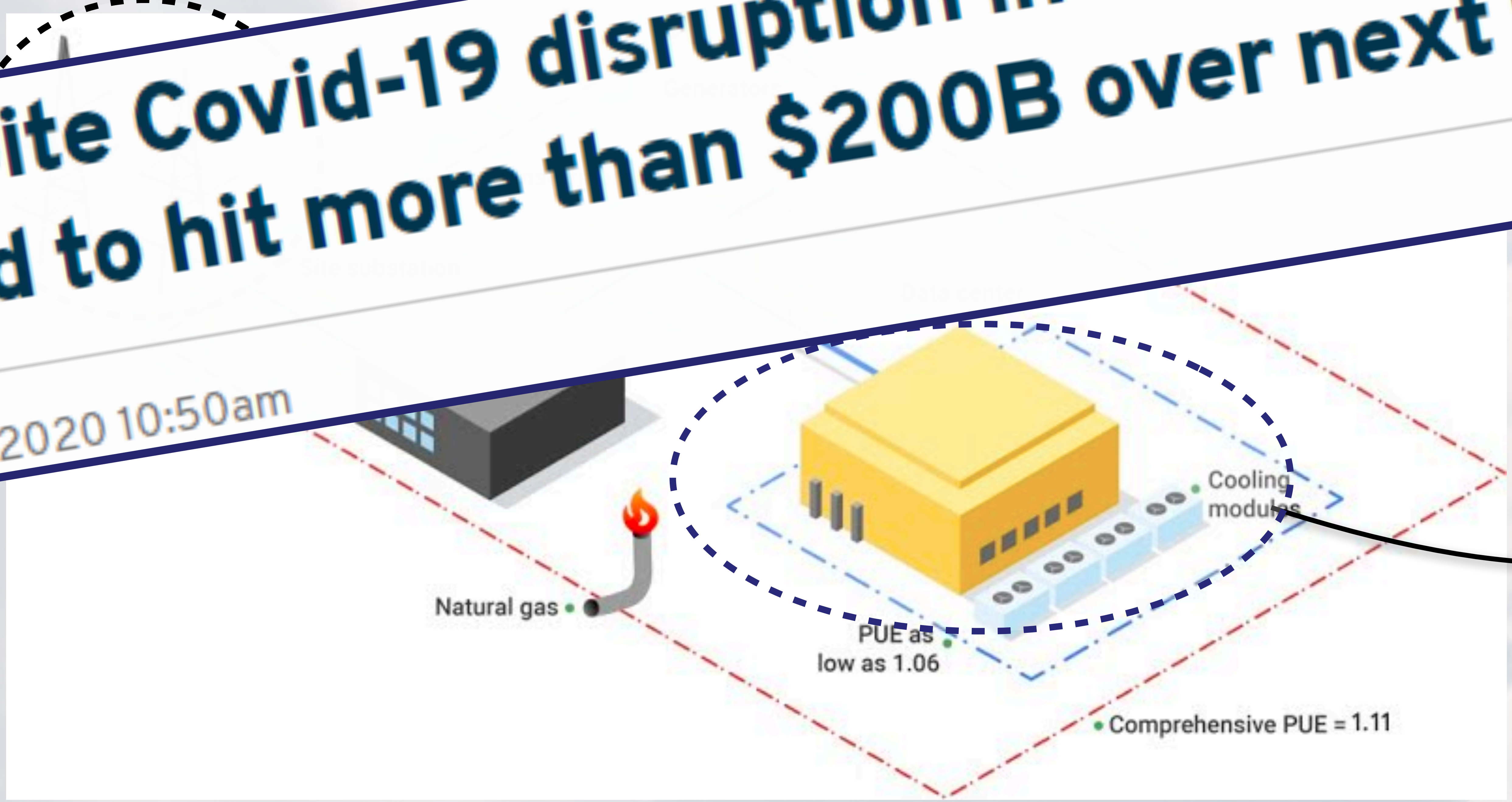
PROVIDER EXPENDITURE

OpEx

CapEx > ~5x OpEx !

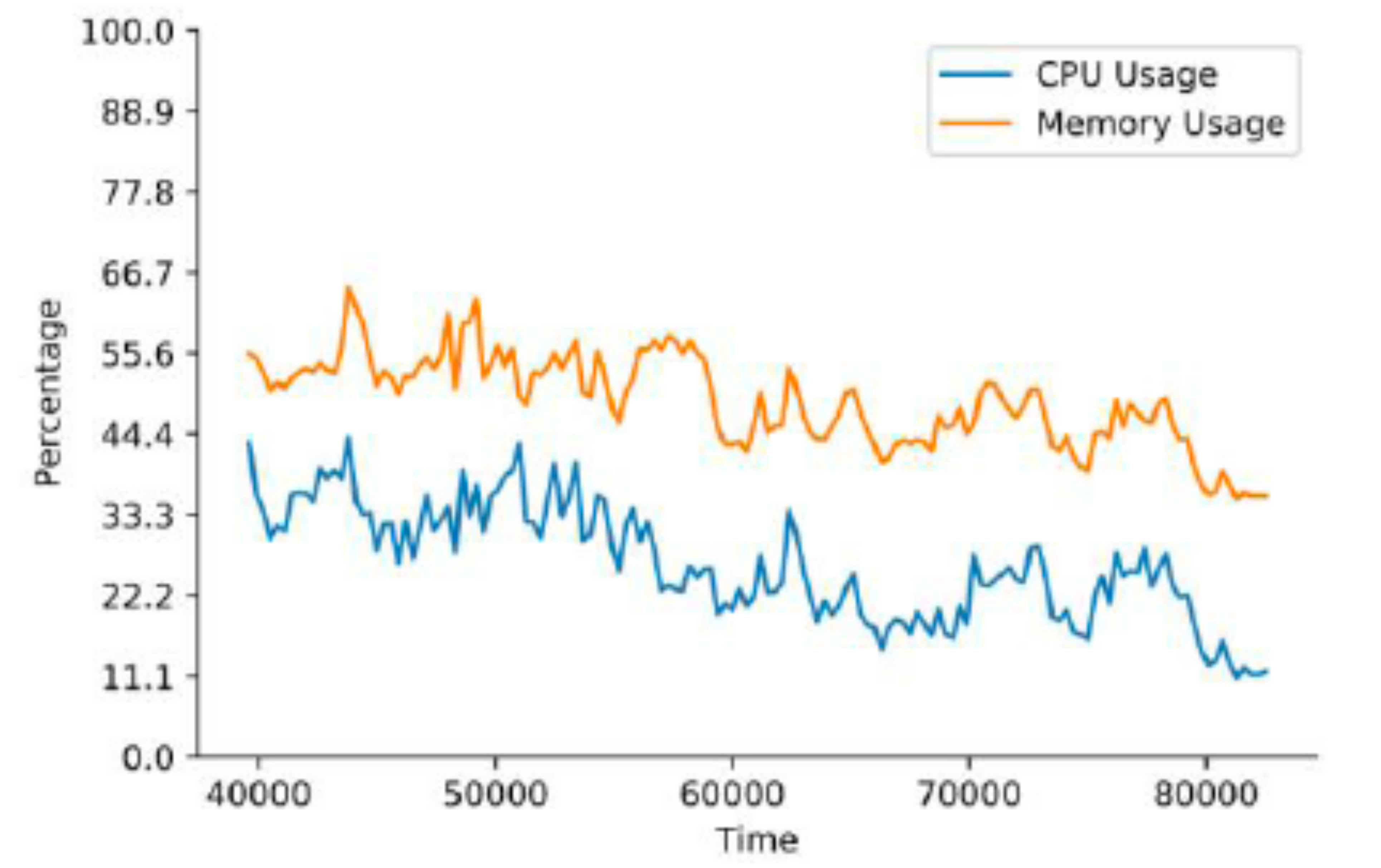
Report: Despite Covid-19 disruption in 2020, data center capex poised to hit more than $200B over next five years

by Mike Robuck | Jul 24, 2020 10:50am

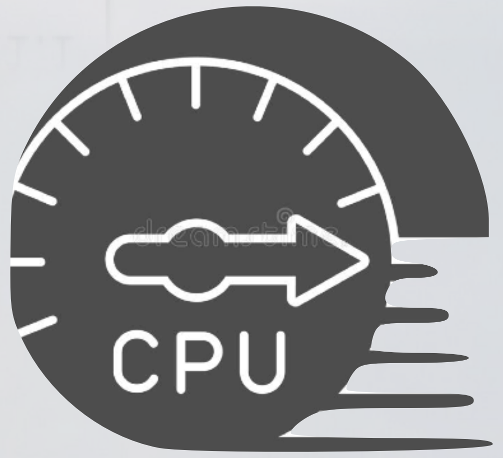


Amortized cost of 50000 servers in Microsoft datacenter. Source: The Cost of Cloud, ACM SIGCOMM'09

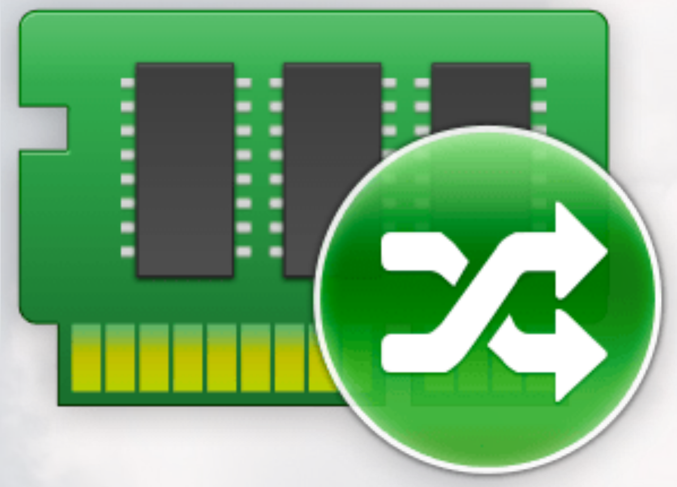
Provider Side Problems



Source:Alibaba Datacenter Case Study, IEEE Access'19



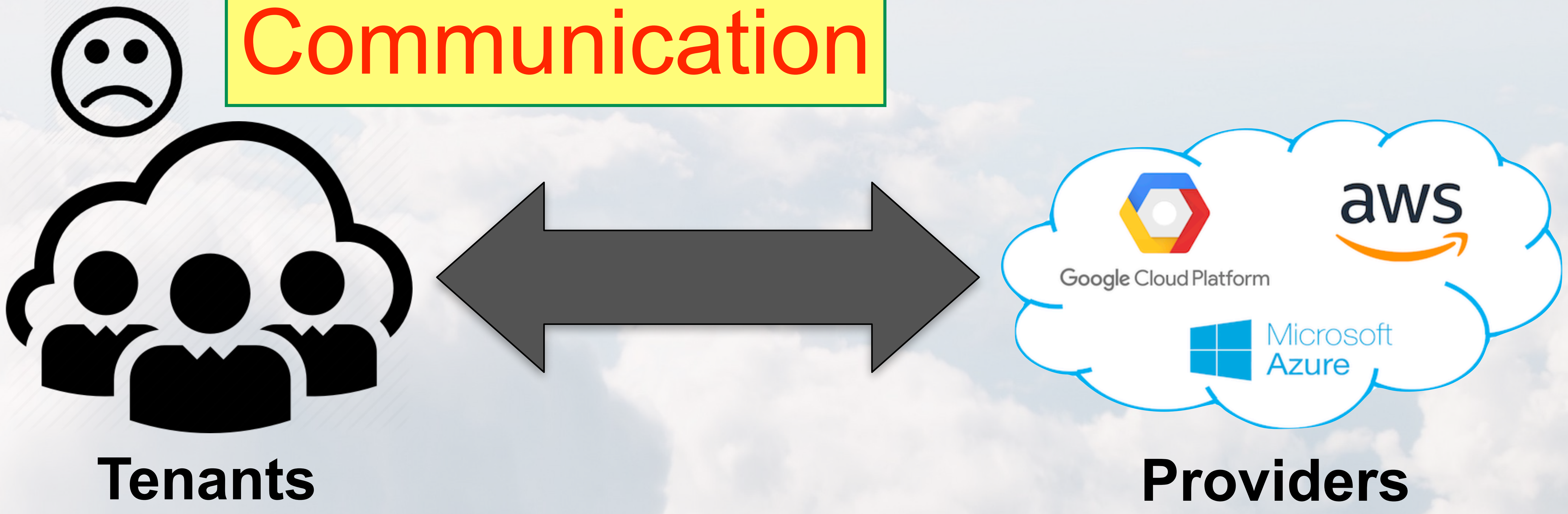
~13-40%



~42-65%

Overstated Requirements

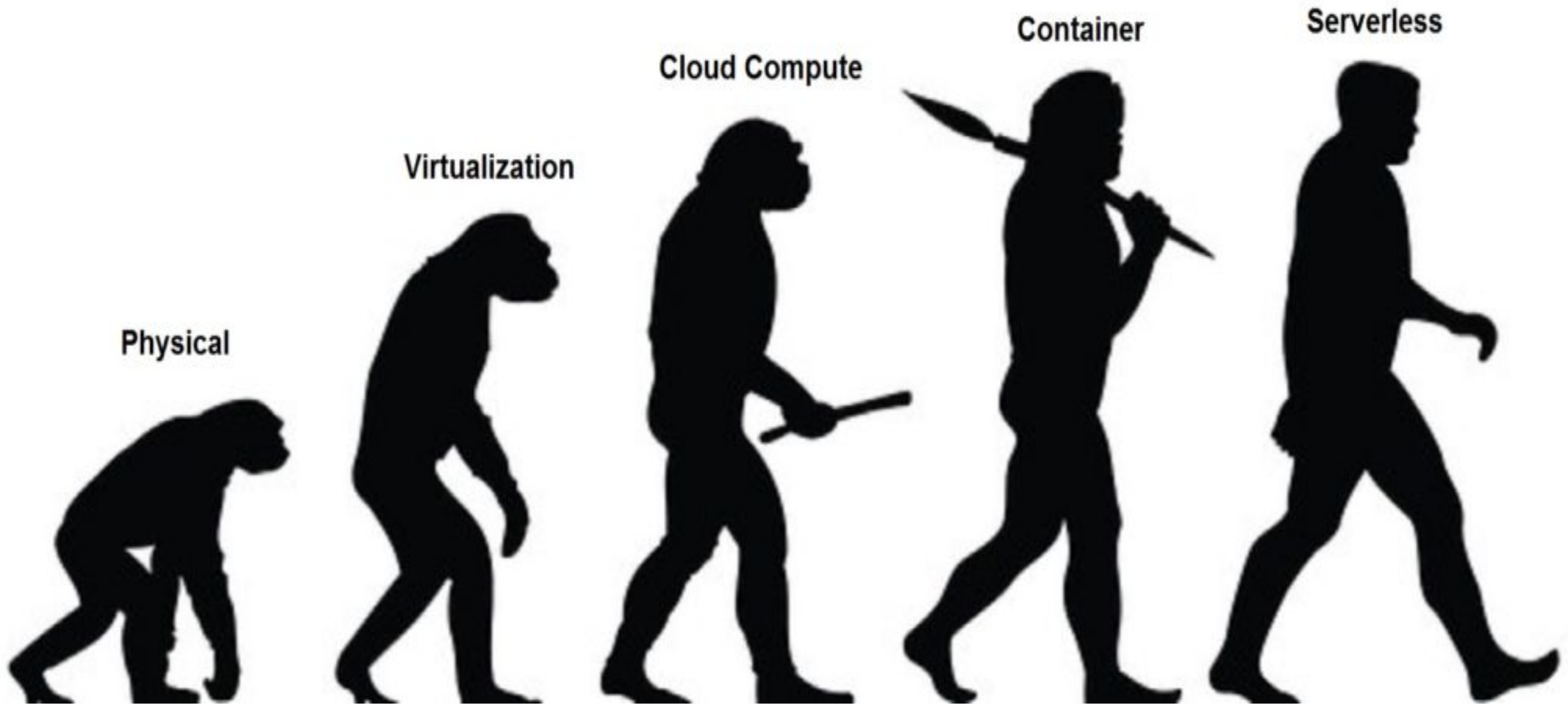
Communication



Blackbox Applications

Overprovisioning

SERVERLESS COMPUTING

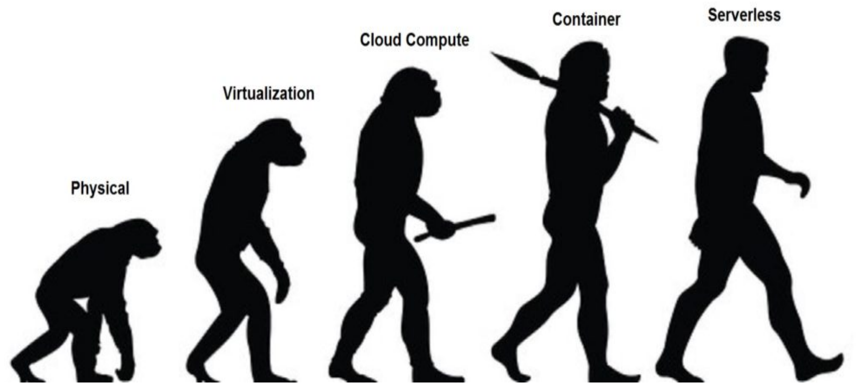


"...Distributed Event-based programming Service..." - **OpenWhisk**

"Run code without thinking about servers.Pay for only the compute time you consume" - ***AWS Lambda***

"...logic can be spun up on-demand in response to events originating from anywhere...." - ***Google Cloud Functions***

Serverless Computing



"...Distributed Event-based programming Service..." - **OpenWhisk**

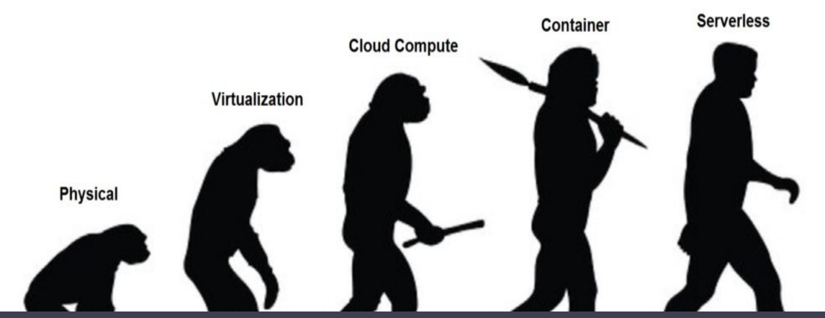
"Run code without thinking about servers.Pay for only the compute time you consume" - **AWS Lambda**

"...logic can be spun up on-demand in response to events originating from anywhere...." - **Google Cloud Functions**

Very Fast Startup

58%

SERVERLESS COMPUTING



Hard to estimate demand
Guaranteeing Performance

Very Fast Startup

58%

WHAT WE NEED ?

Low Cost

High Performance

Cloud Tenants

Cloud Providers

High Utilization

Energy Efficiency

What we need ?

Cloud Tenants

Seda, *SOSP'01*
Mapreduce, *OSDI'08*
Hadoop (Yahoo), *MSST'10*
Spark, *ACM Comm'16*
Clipper, *NSDI'17*

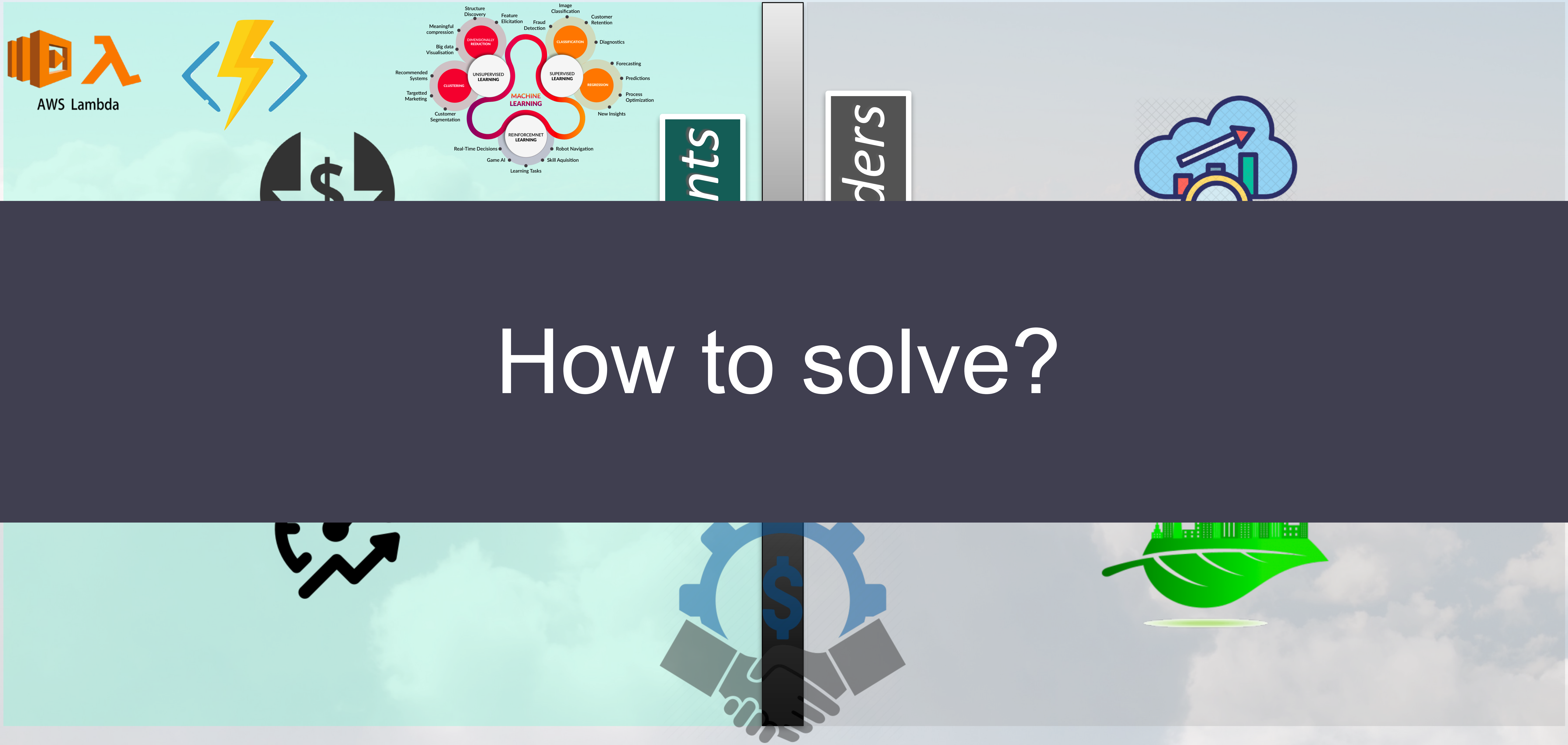
H ce

Cloud Providers

FGCS'02
Google, *IEEE Computer'07*
AWS, *CIDR'09*
Google Borg, *Eurosys'15*
Microsoft, *SOSP'17*

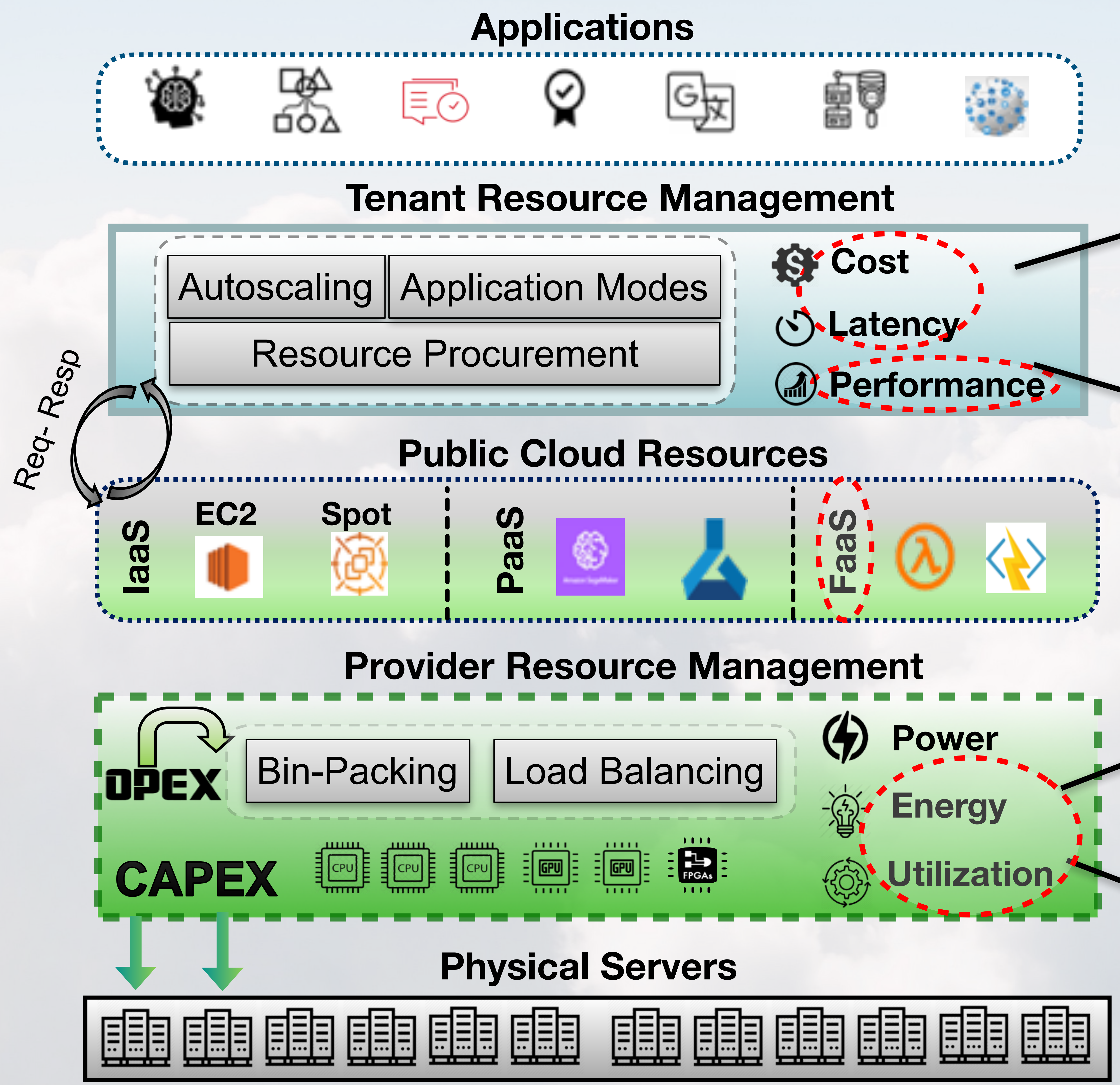
E cy

WHAT WE NEED ?



How to solve?

Dissertation Contributions



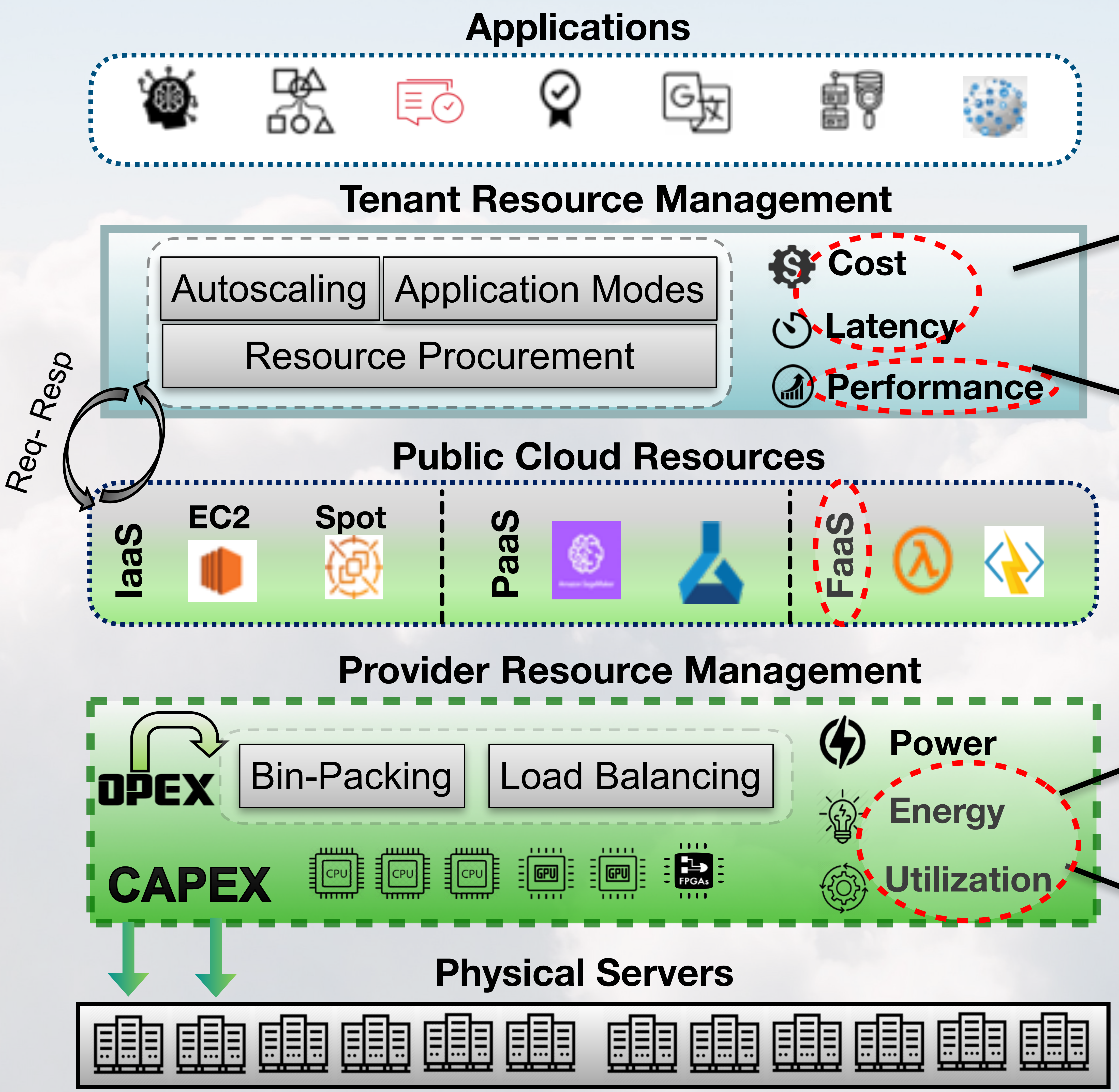
Spock- Cost Efficient and Latency Aware Autoscaling, ***IEEE CLOUD' 2019***

Cocktail- Improving Machine Learning Performance at Low Cost, ***NSDI' 2021*** *(Under-Revision)*, ***WoSC'2021***

Fifer- Improving Energy Efficiency for Serverless Platforms, ***Middleware 2020, ICDCS 2020***

Multiverse- Improving Server Utilization for Private HPC Clusters, ***CCGrid' 2020***

Dissertation Contributions



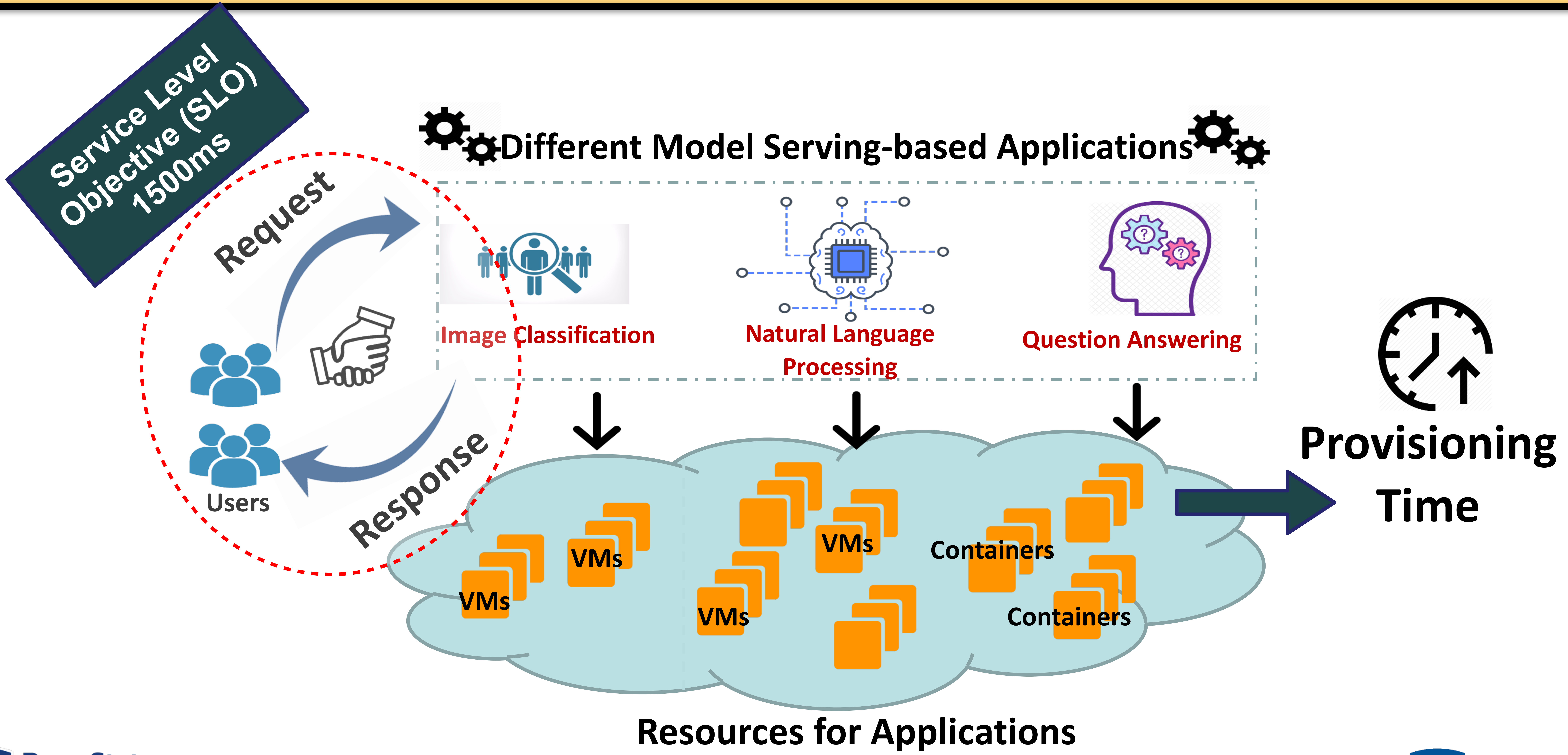
Spock- Cost Efficient and Latency Aware Autoscaling, ***IEEE CLOUD' 2019***

Cocktail- Improving Machine Learning Performance at Low Cost, ***NSDI' 2021*** *(Under-Revision)*, ***WoSC'2021***

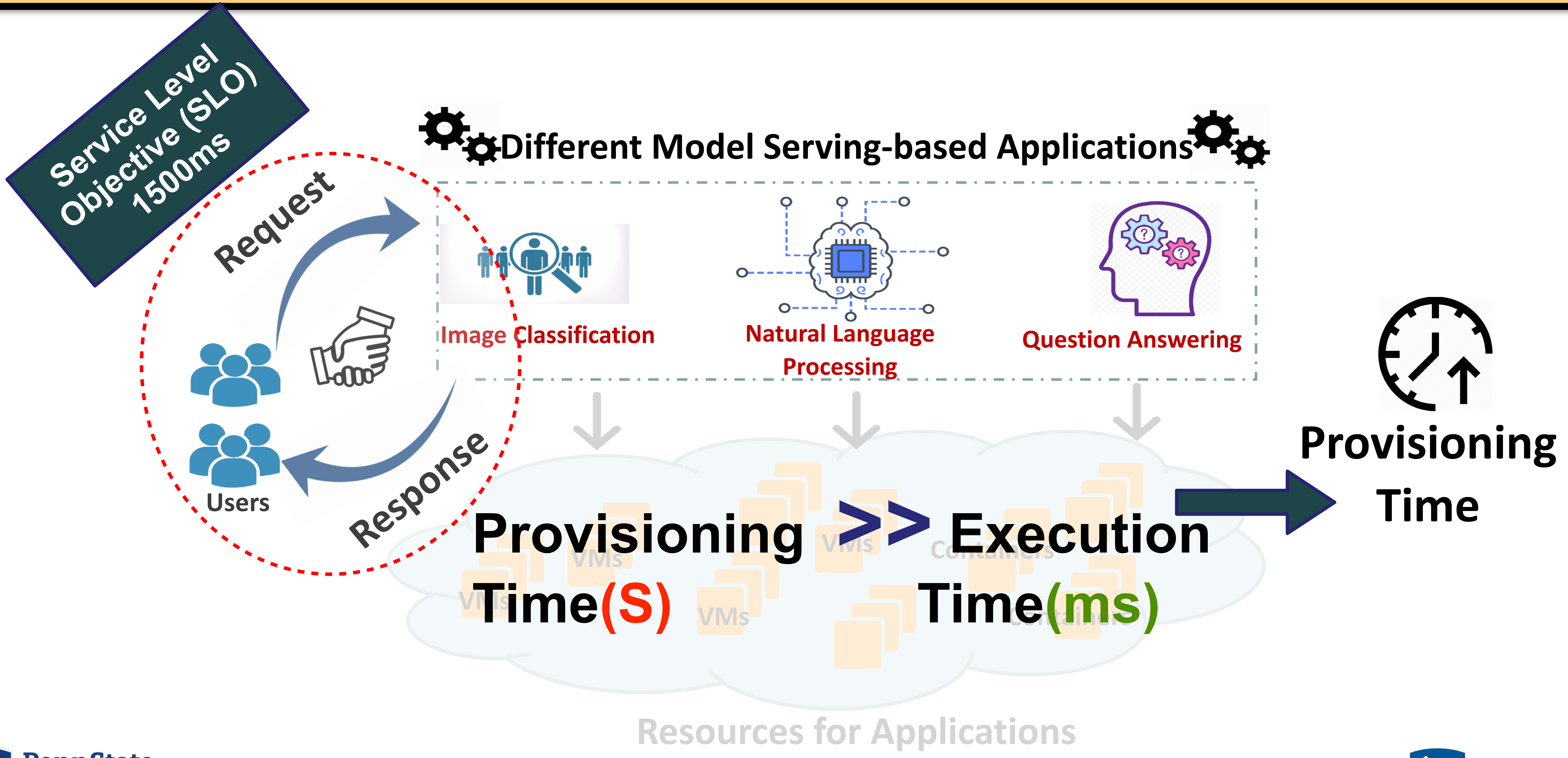
Fifer- Improving Energy Efficiency for Serverless Platforms, ***Middleware 2020, ICDCS 2020***

Multiverse- Improving Server Utilization for Private HPC Clusters, ***CCGrid' 2020***

Model Serving Hosted on Cloud



Model Serving Hosted on Cloud



PRIOR WORKS

- Utilization based autoscaling- *Urgaonkar et al PODC'03*
  - ➡ Not suitable for millisecond scale applications
- Relaxed VM scale down - *Gandhi et al SC'12, TOCS'12*
  - ➡ Intermittent over-provisioning
- Exploiting different VM instance types *Wang et al. Eurosys'17,*
  - ➡ They are complementary to our proposal.

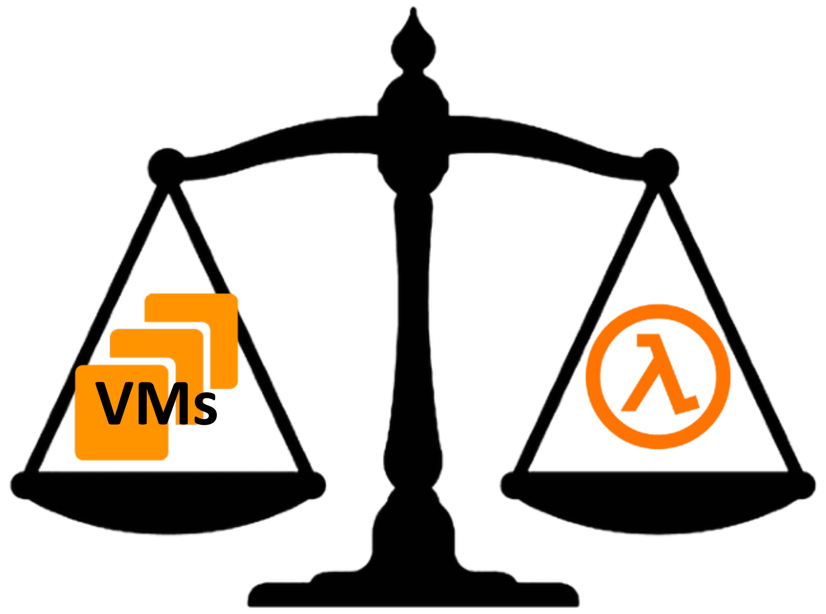
PRIOR WORKS

- Utilization based autoscaling- *Urgaonkar et al PODC'03*
  - ➡ Not suitable for millisecond scale applications

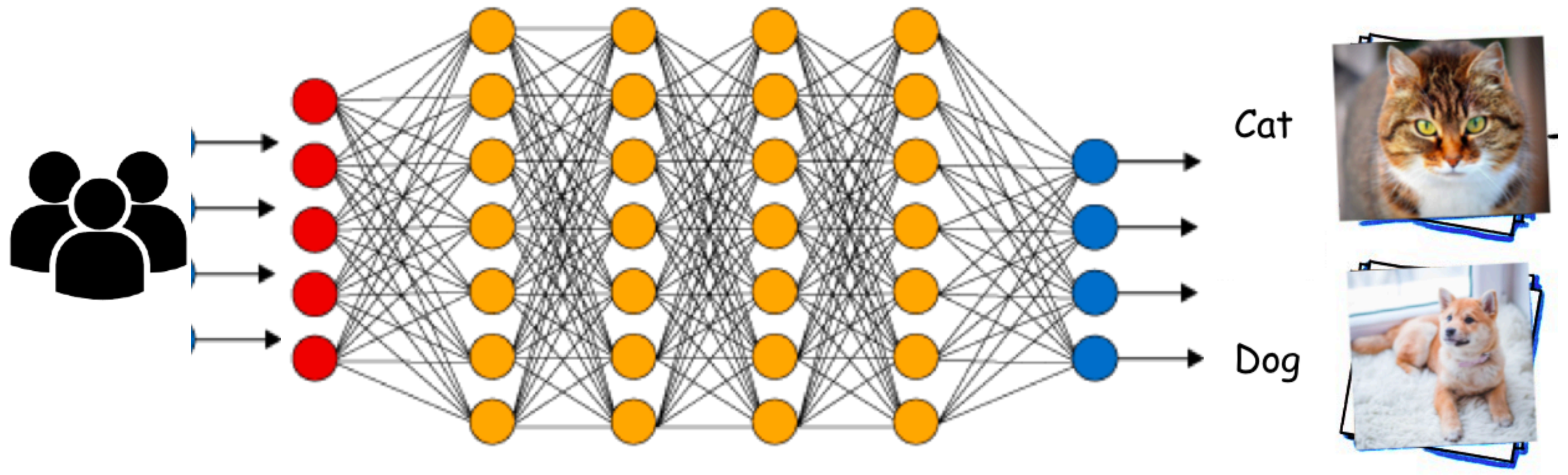
Only VM based solutions are largely expensive

- Exploiting different VM instance types *Wang et al. Eurosys'17,*
  - ➡ They are complementary to our proposal.

Key Findings

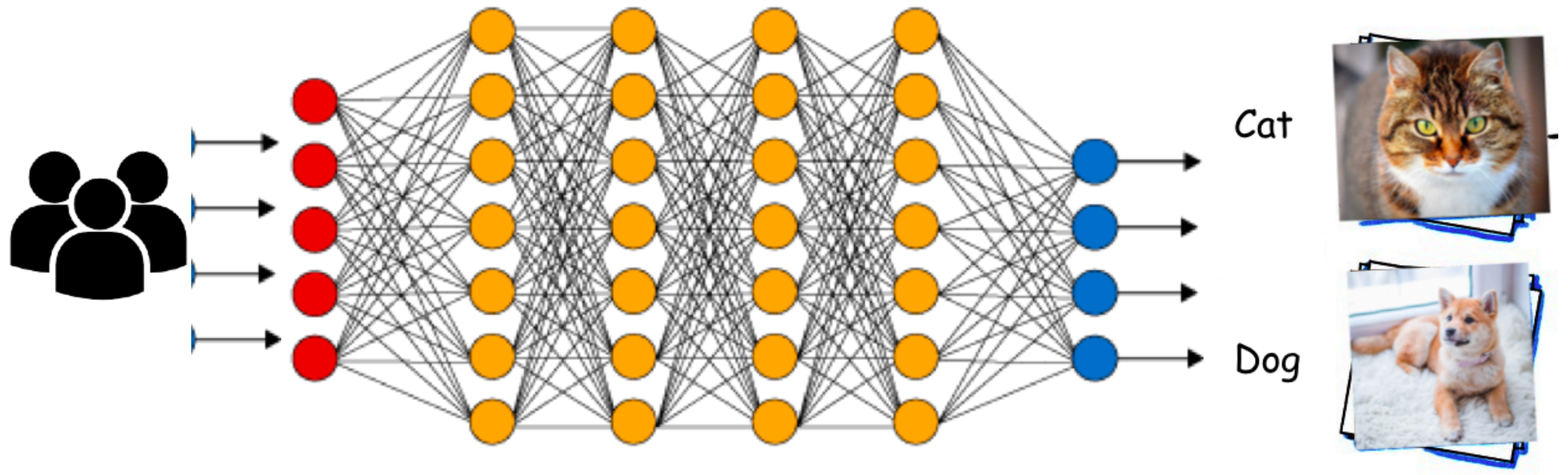


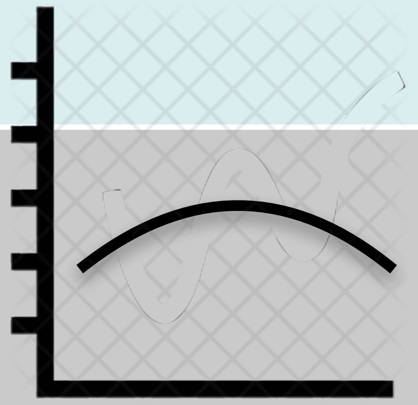
Deep Learning Inferences



Arrival	Resource	Cost	SLO
Bursty	λ	Pay per use 🙂	Pre warmed 🙂
	VMs	Over provisioned 🙁	Too much Scaling 🙁
Predictable	λ	Per-unit Cost 🙁	Pre warmed 🙂
	VMs	Known Demand 🙂	Reduced Scaling 🙂

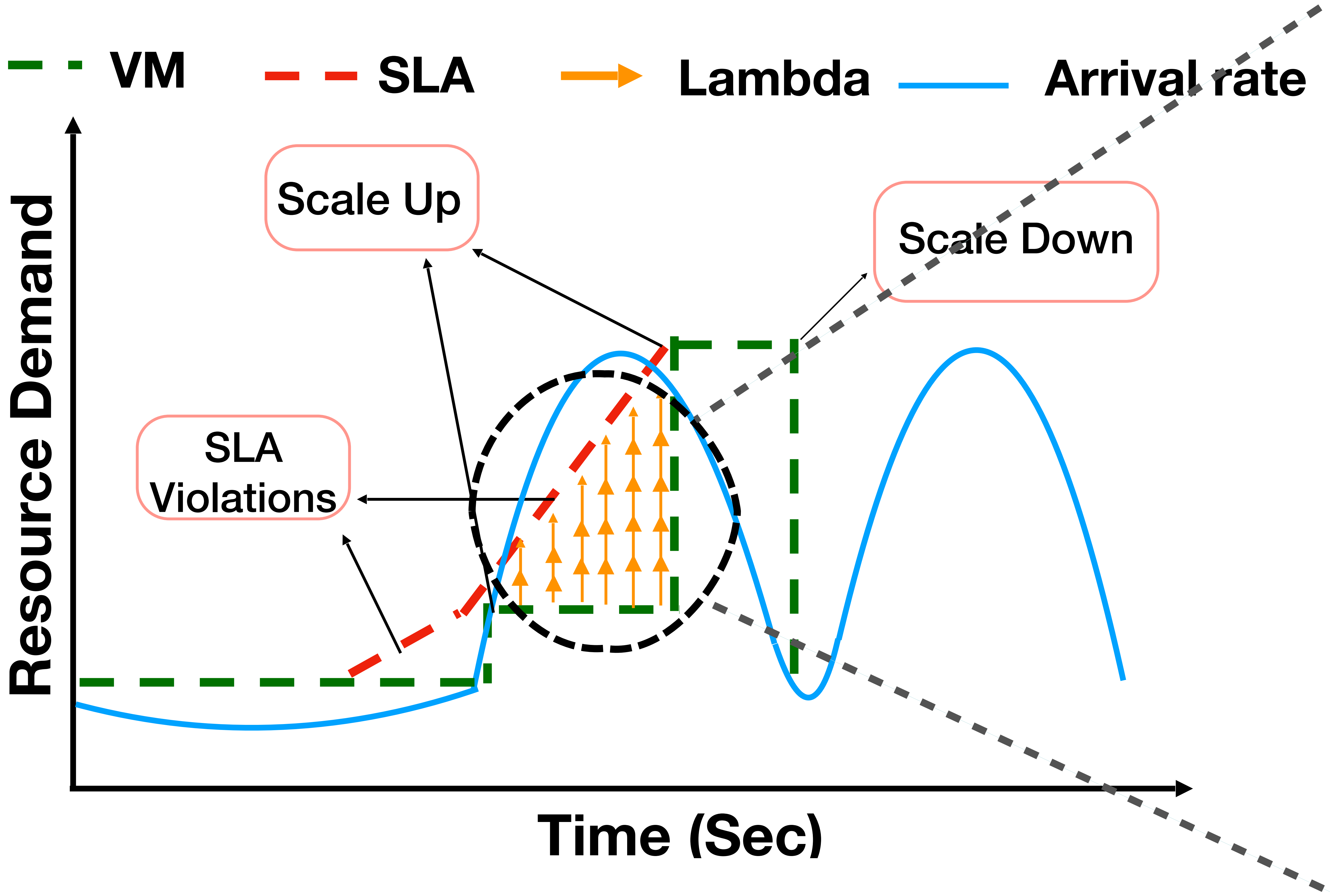
Key Findings



Arrival	Resource	Cost	SLO
		Cost	warmed
	 VMs	Known Demand	Reduced Scaling

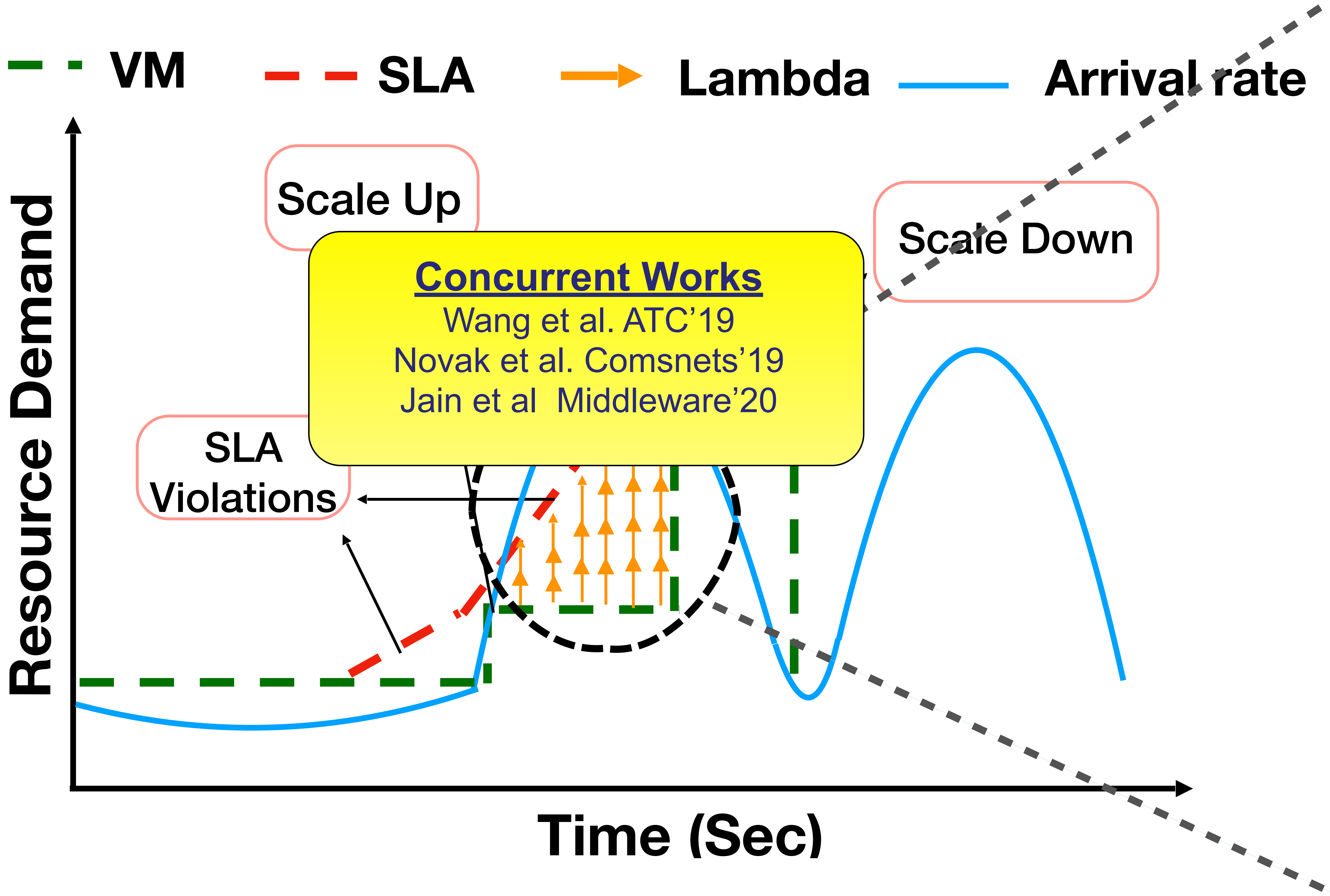
Can we multiplex both?

SPOCK: EXPLOITING SERVERLESS FUNCTIONS FOR SLO AND COST AWARE AUTOSCALING



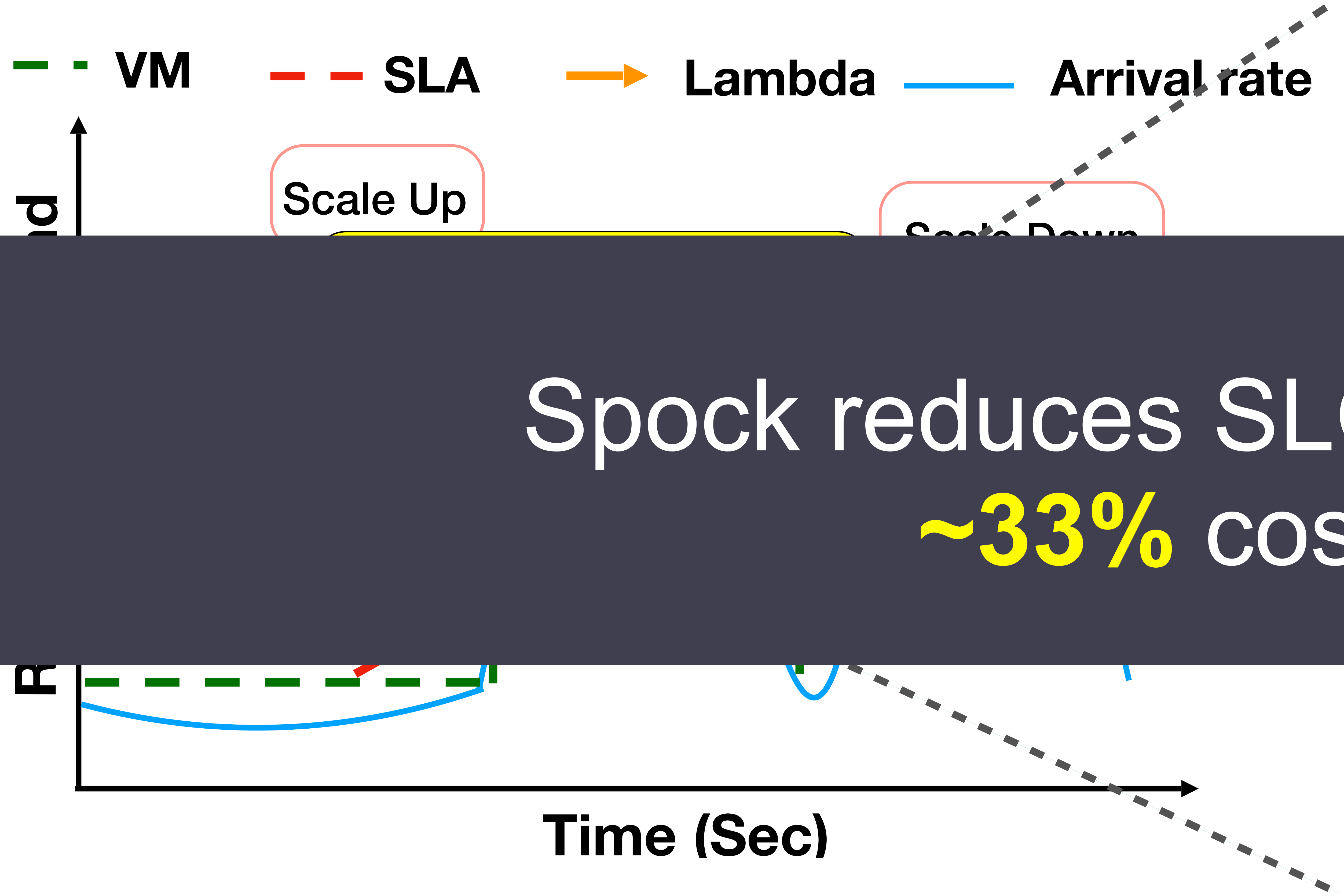
- ➢ Offload queries to lambdas while starting new VMs.
- ➢ Reduces SLO violations during request surge.
- ➢ Reduce intermittent over-provisioning VMs

SPOCK: EXPLOITING SERVERLESS FUNCTIONS FOR SLO AND COST AWARE AUTOSCALING



- ➢ Offload queries to lambdas while starting new VMs.
- ➢ Reduces SLO violations during request surge.
- ➢ Reduce intermittent over-provisioning VMs

Spock: Exploiting Serverless Functions For SLO and Cost Aware Autoscaling



- ➢ Offload queries to lambdas while starting new VMs.

Spock reduces SLO violations by **~74%** with **~33%** cost savings

- ➢ Reduce intermittent over-provisioning VMs

MODEL SERVING CHALLENGES

Model Serving

- **Resource Type** → **Provisioning Latency** ✅ ***(Spock)***
- **Model Type** → **Model Latency**, **Accuracy**

Cost

MODEL SERVING CHALLENGES

Model Serving

Resource Type

Provisioning Latency

✓ *(Spock)*

Model Type

Model Latency

Accuracy

Cost

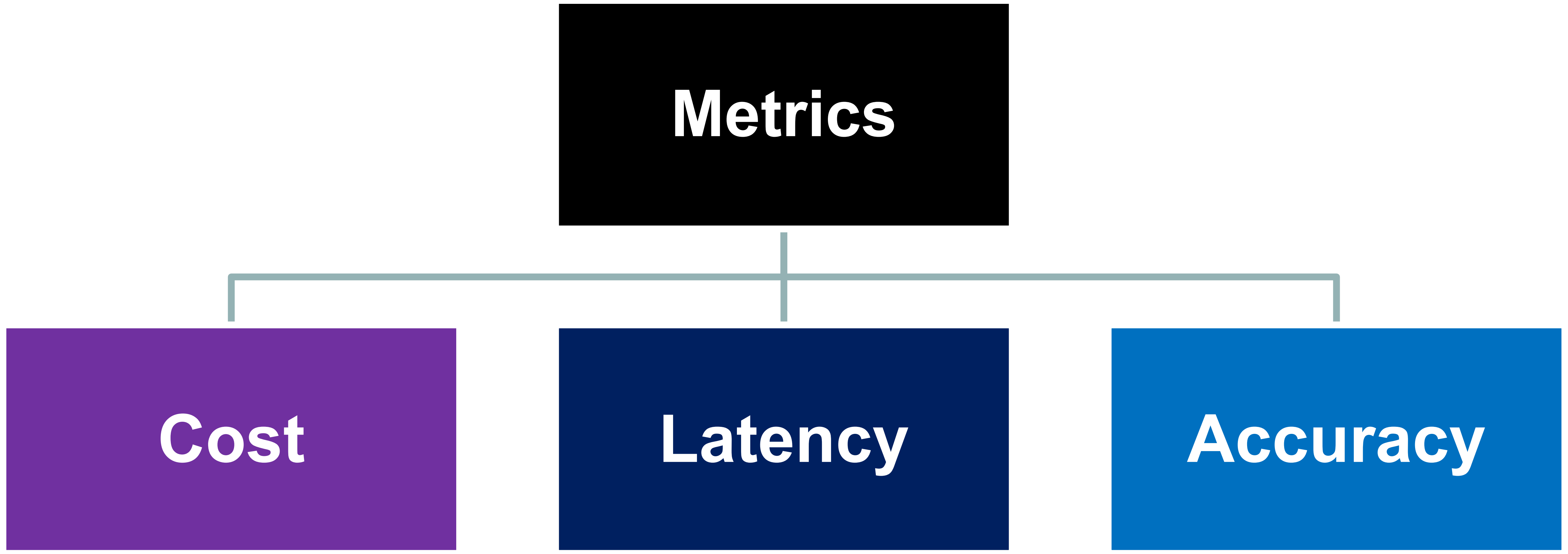
In Netflix, **75**% of viewer activity is based on these accurate suggestions.

Model Serving Challenges

Provisioning Latency

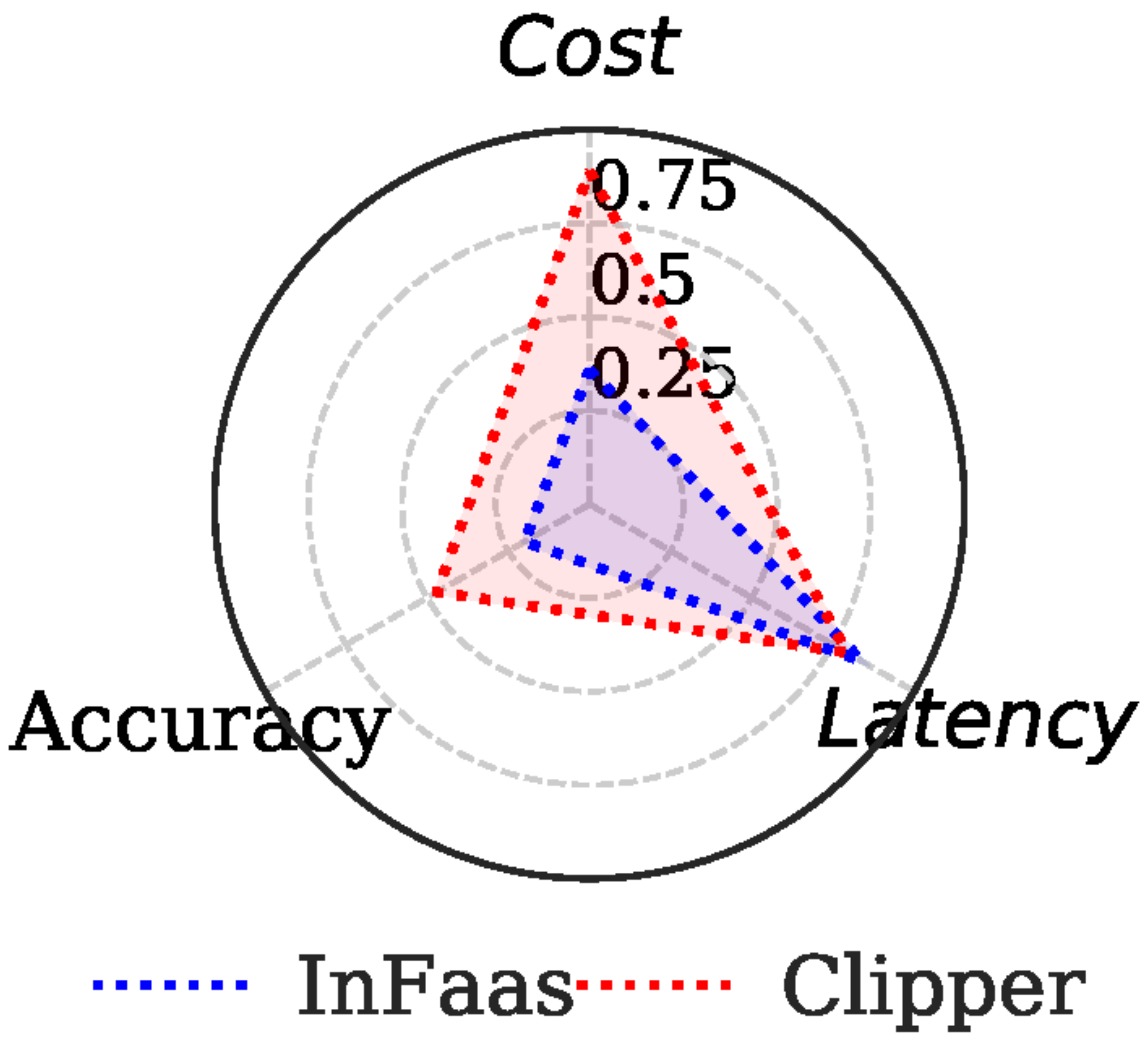
How to improve accuracy with low latency and low cost?

PRIOR WORK IN MODEL SERVING



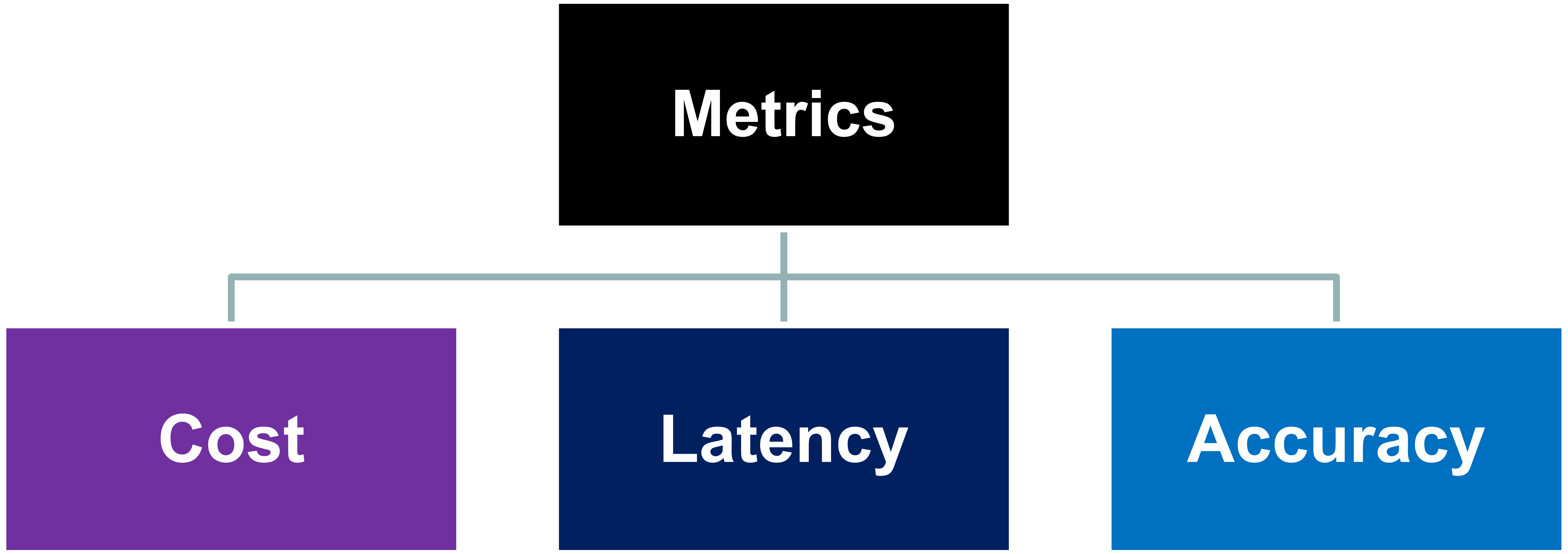
– **InFaas** uses different resource types to ensure low latency at low cost.

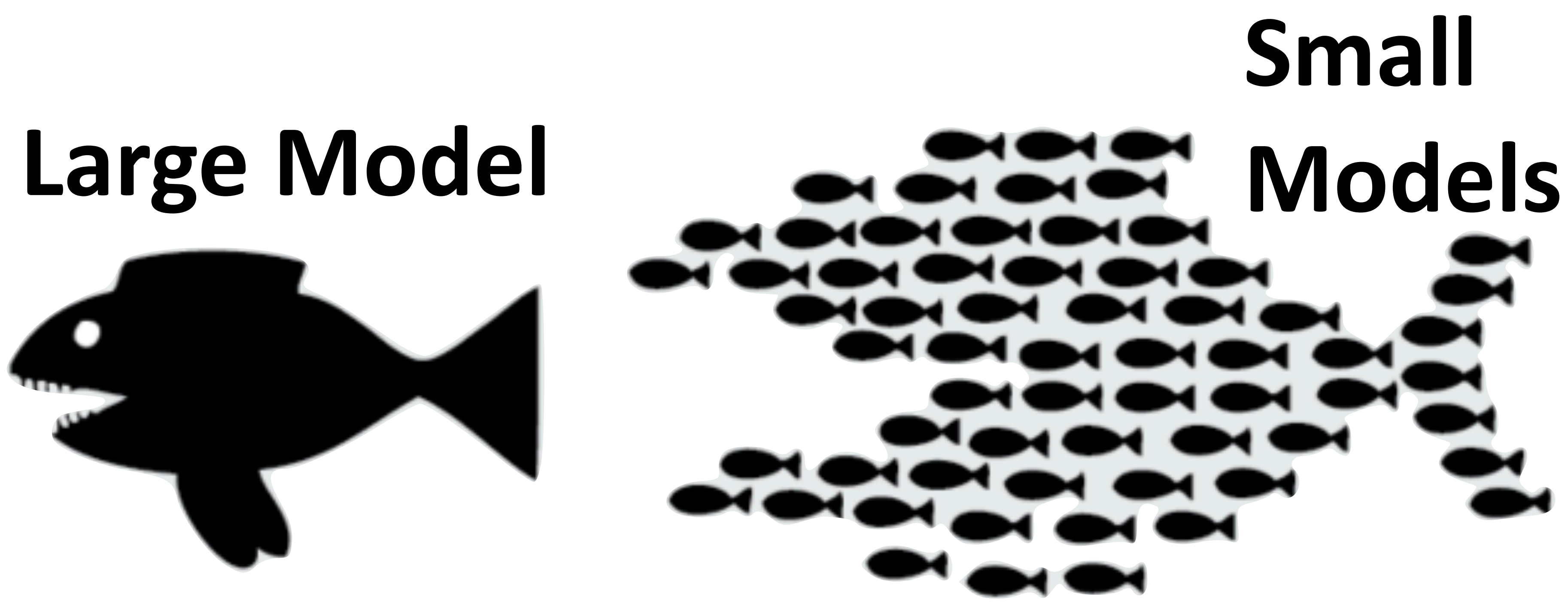
– **Clipper** uses model ensembling to achieve higher accuracy.



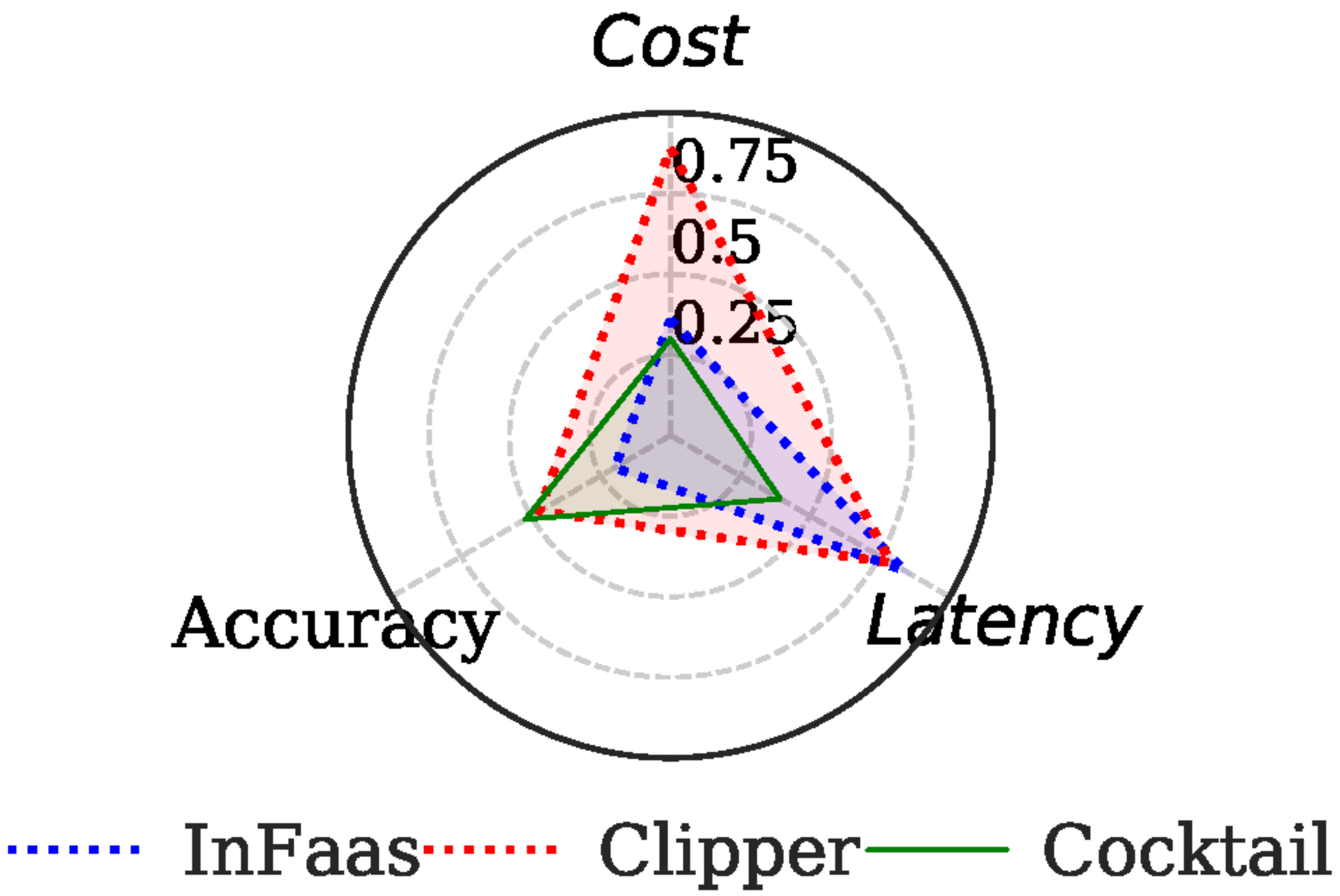
Crankshaw et al CIDR'15, NSDI'17, SoCC'20
Yadawkar et al Arxiv'19

Prior Work in Model Serving



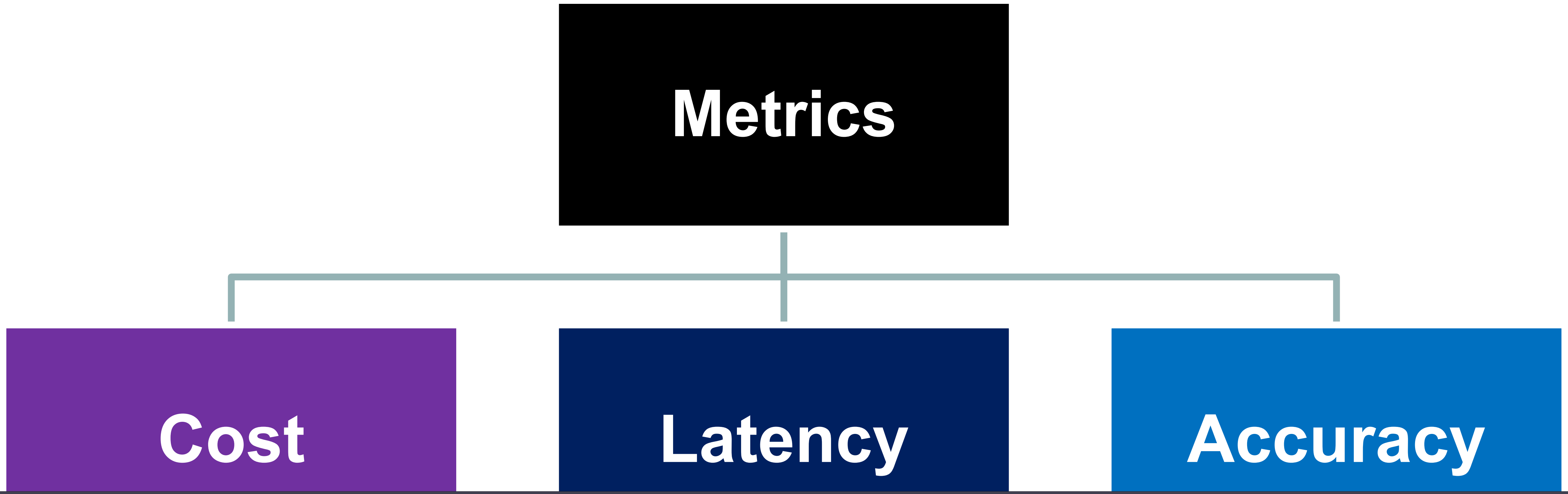


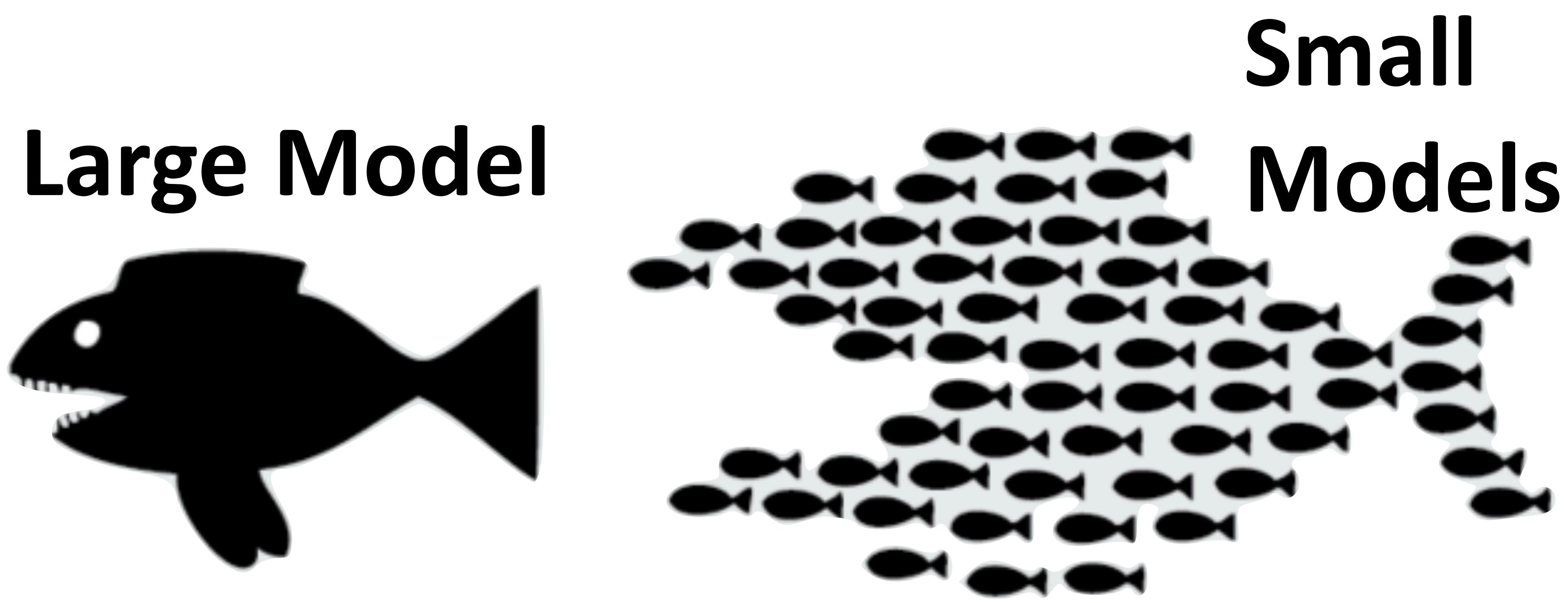
- **InFaas** uses different resource types to ensure low latency at low cost.
- **Clipper** uses model ensembling to achieve higher accuracy.



Crankshaw et al CIDR'15, NSDI'17, SoCC'20
Yadawkar et al Arxiv'19

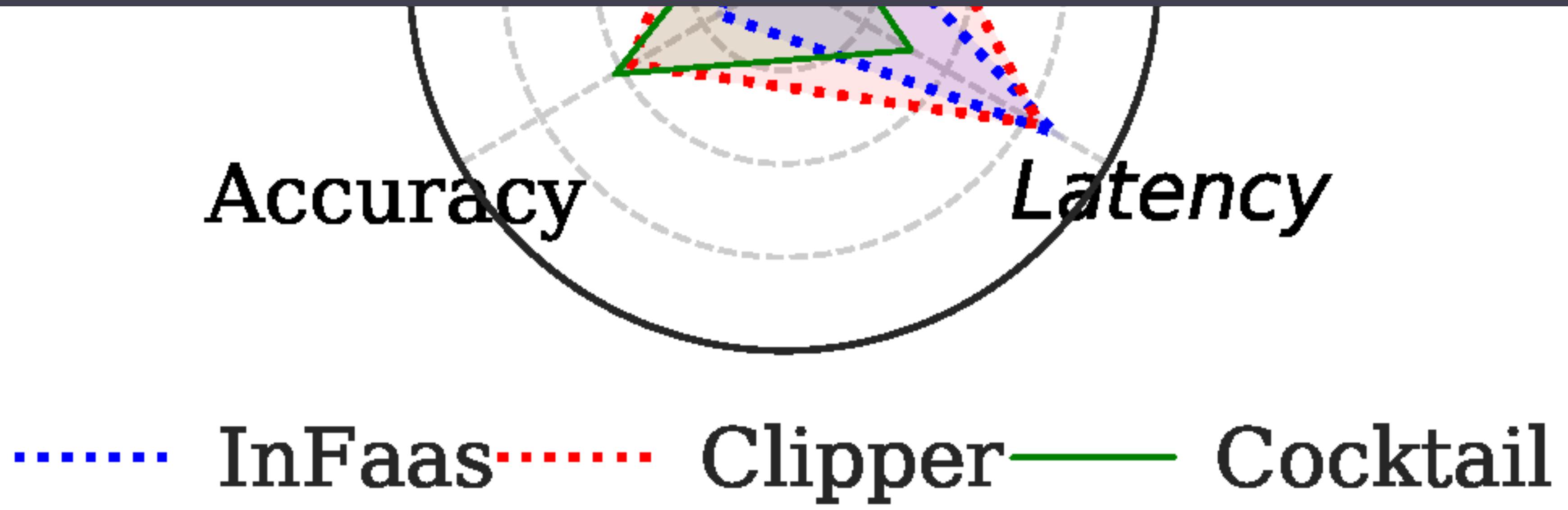
Prior Work in Model Serving





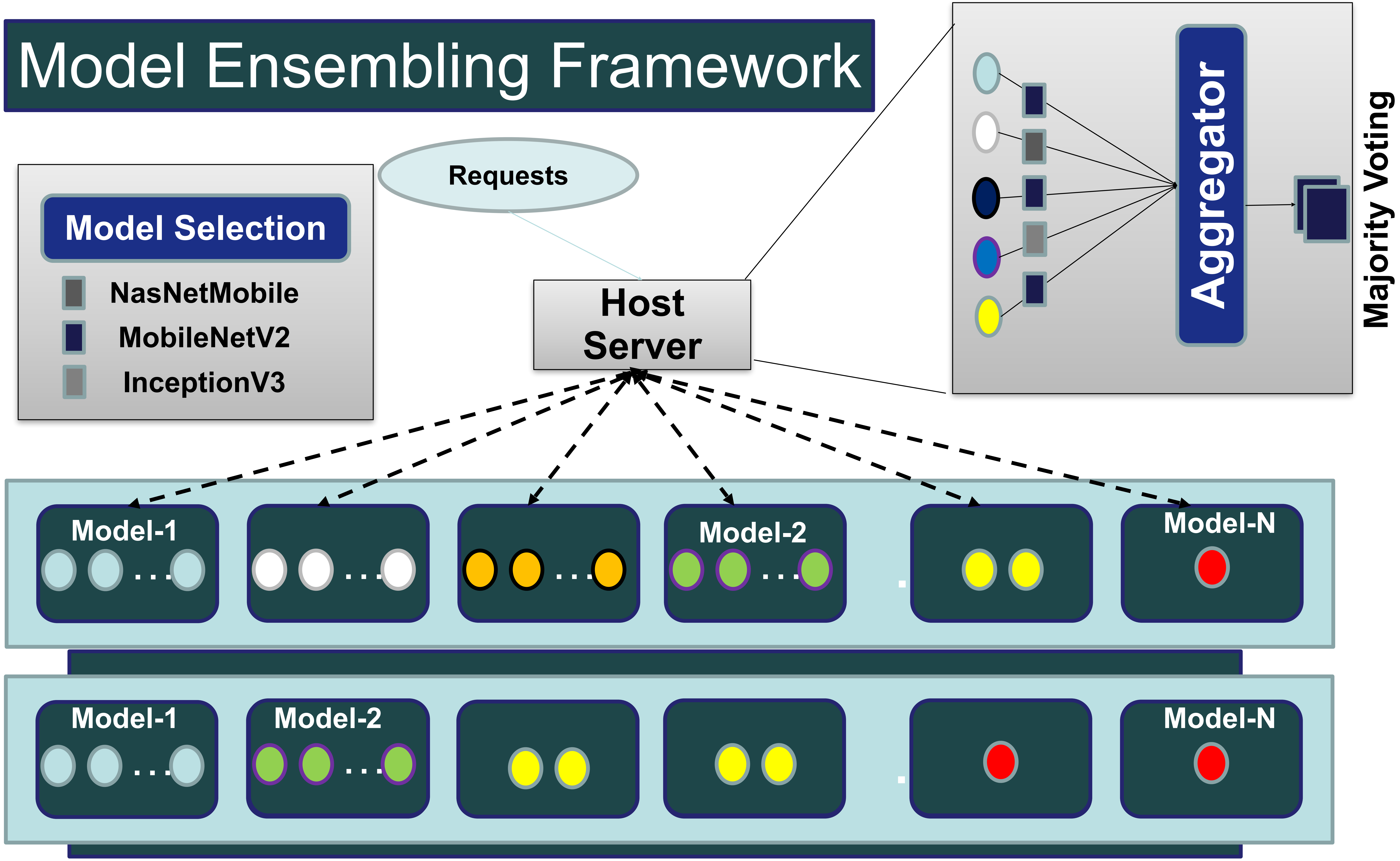
How to do ensembling?

– **Clipper** uses model ensembling to achieve higher accuracy.



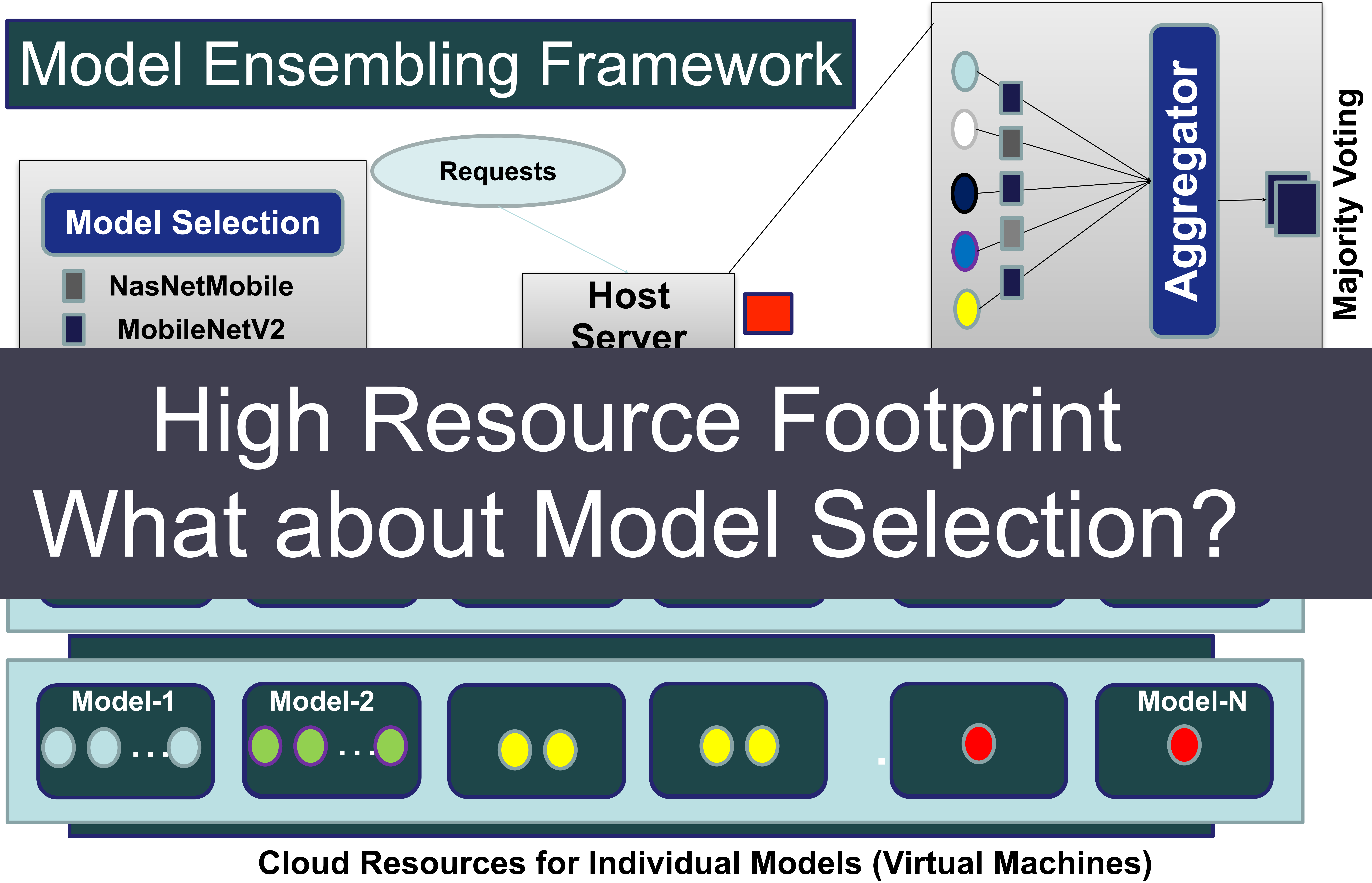
Crankshaw et al CIDR'15, NSDI'17, SoCC'20
Yadawkar et al Arxiv'19

Model Ensembling Framework



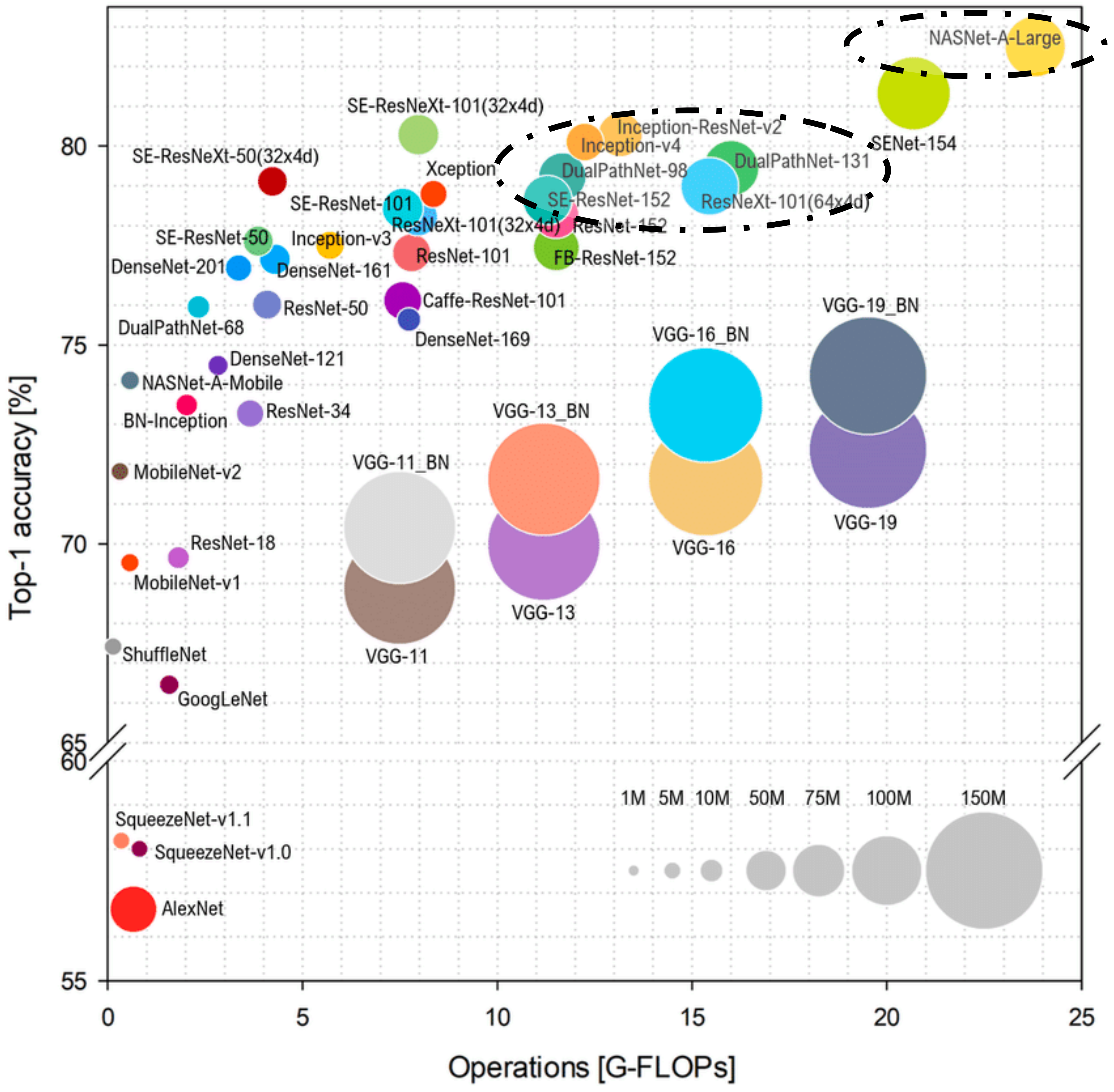
Cloud Resources for Individual Models (Virtual Machines)

Model Ensembling Framework



High Resource Footprint
What about Model Selection?

MODEL SPACE EXPLORATION



IEEE Access'18 Benchmark Analysis of Representative Deep Neural Network Architectures

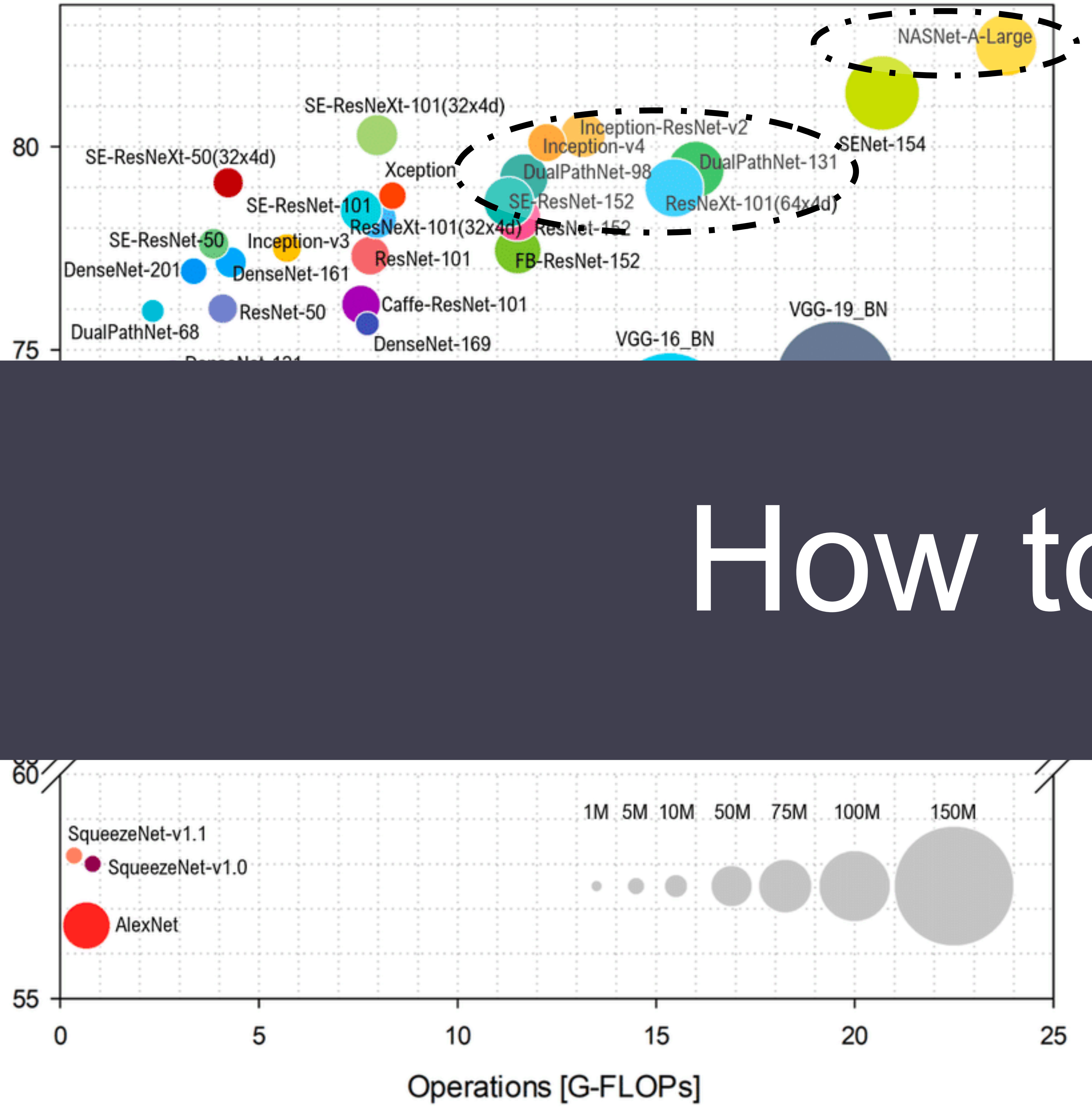
Most accurate model

✴**~2x** parameters, latency

✴**~2%** more accuracy

- How to bridge the 2% accuracy gap?
- What about cost?

Model Space Exploration



IEEE Access'18 Benchmark Analysis of Representative Deep Neural Network Architectures

Most accurate model

- ✴ **~2x** parameters, latency
- ✴ **~2%** more accuracy

How to ensemble?

- What about cost?

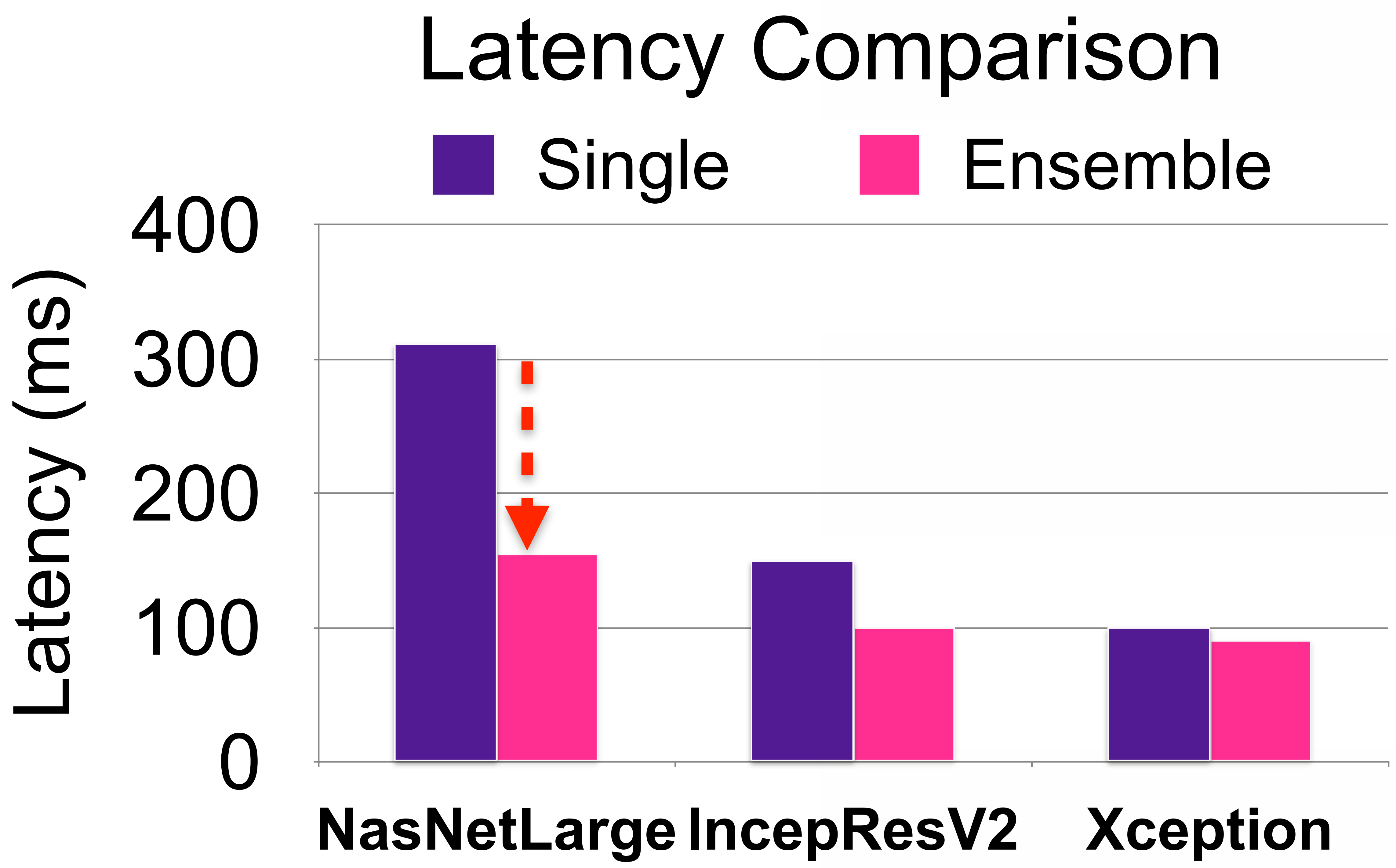
Full Ensemble

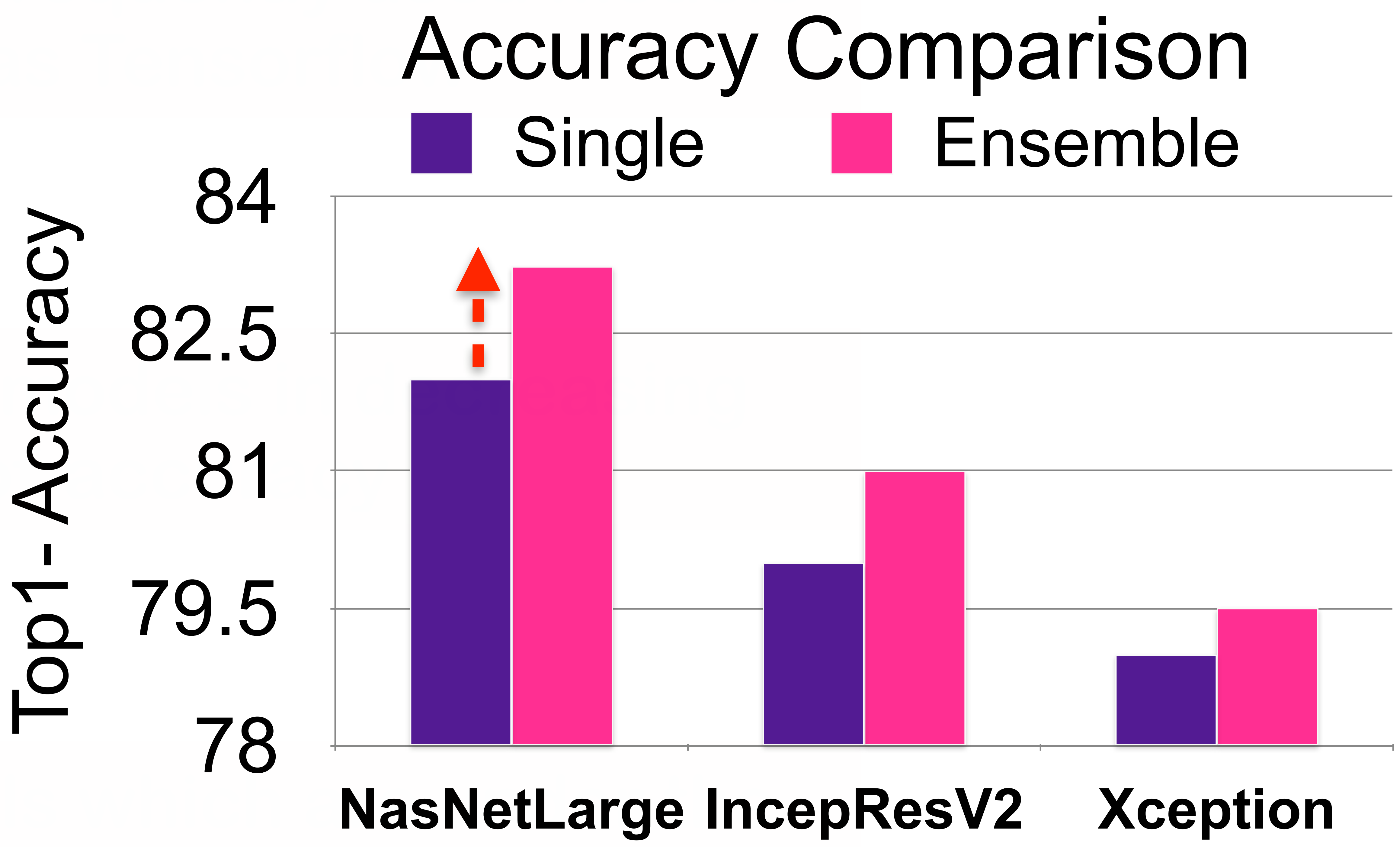
Model Set: Top 12 frequently used models from Keras Tensorflow

Choose baseline models in decreasing order of accuracy

Combine all models which are under the latency of baseline model.

FULL ENSEMBLE

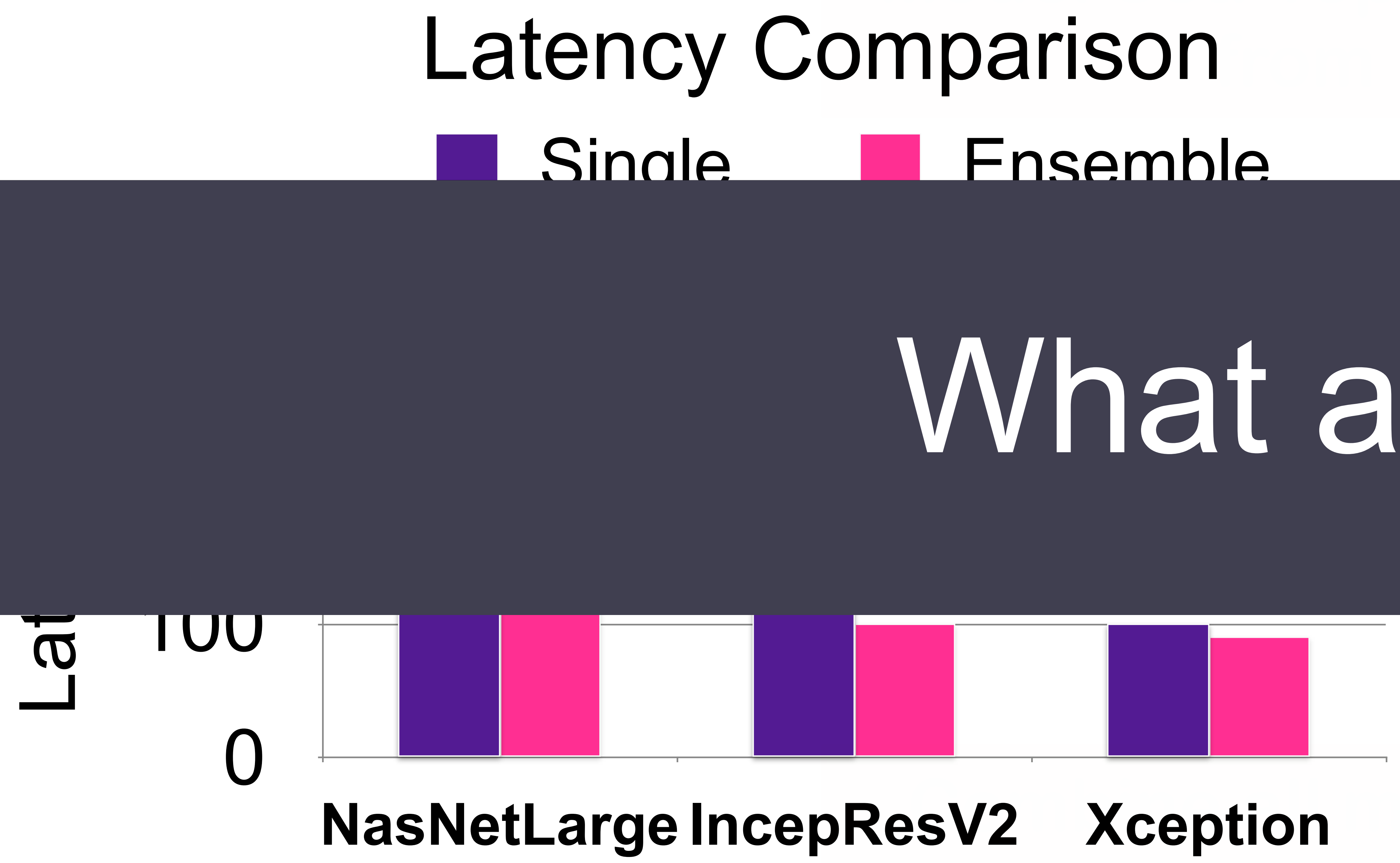


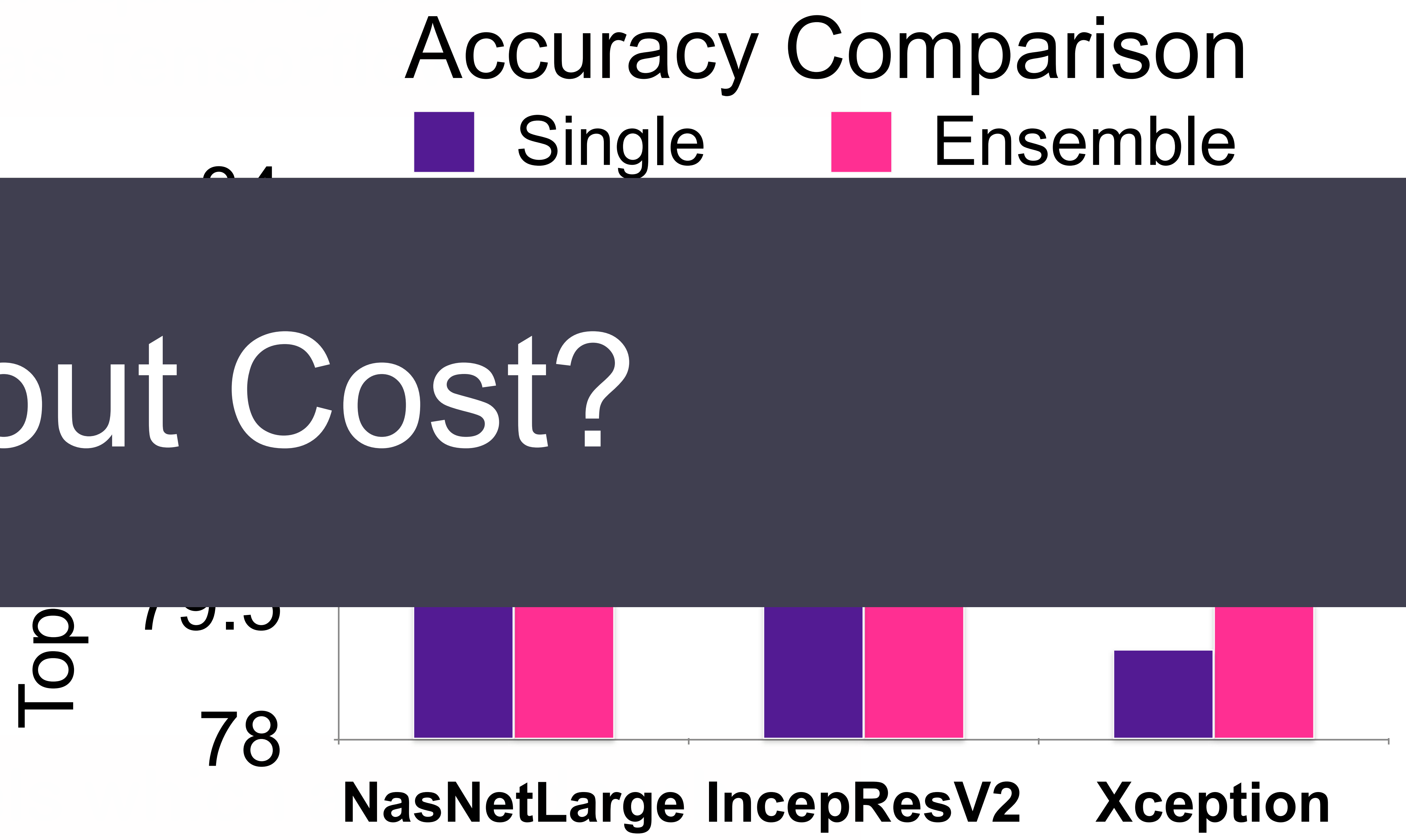


PennState College of Engineering

PennState High Performance Computing Lab

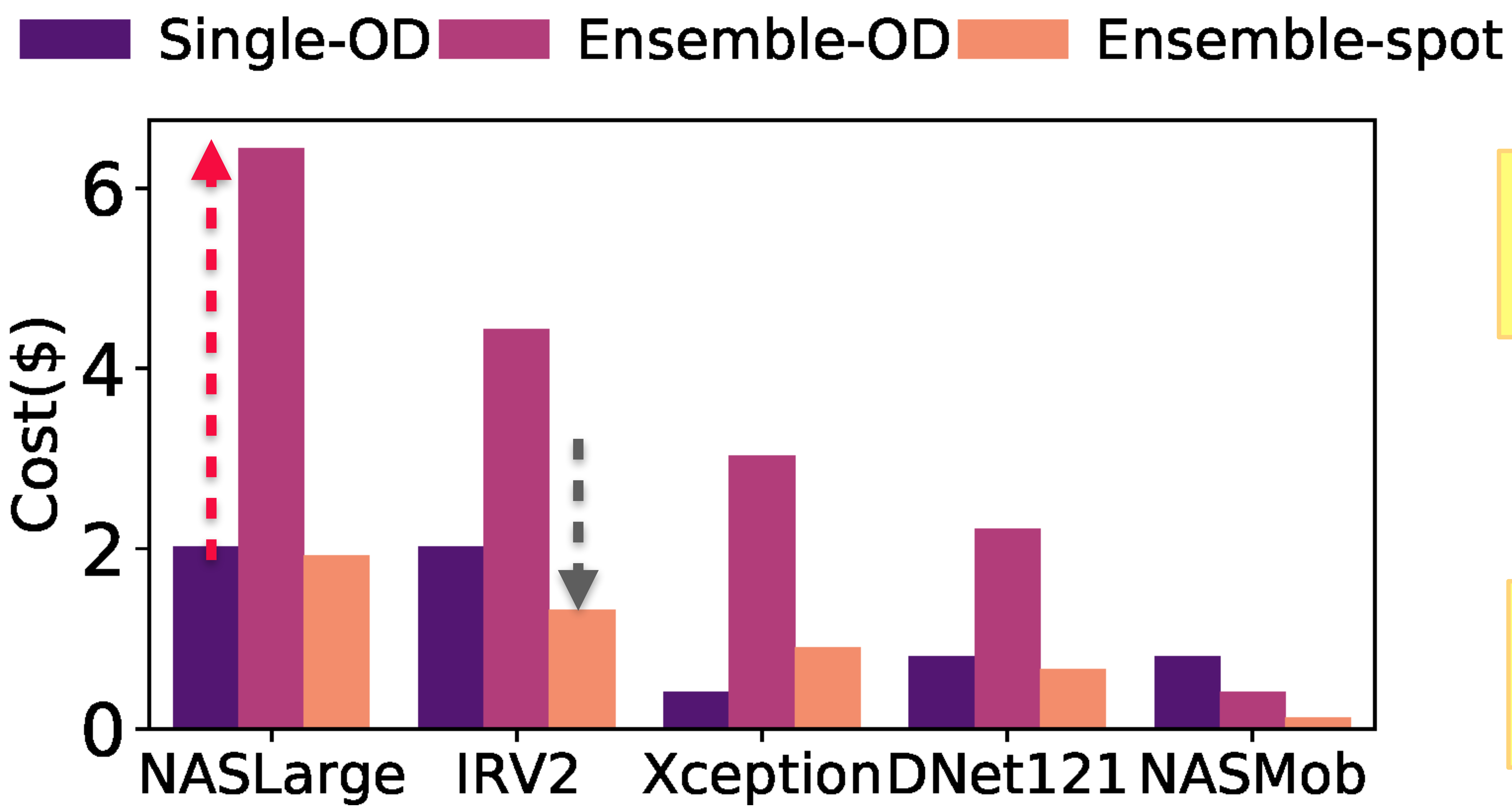
FULL ENSEMBLE





What about Cost?

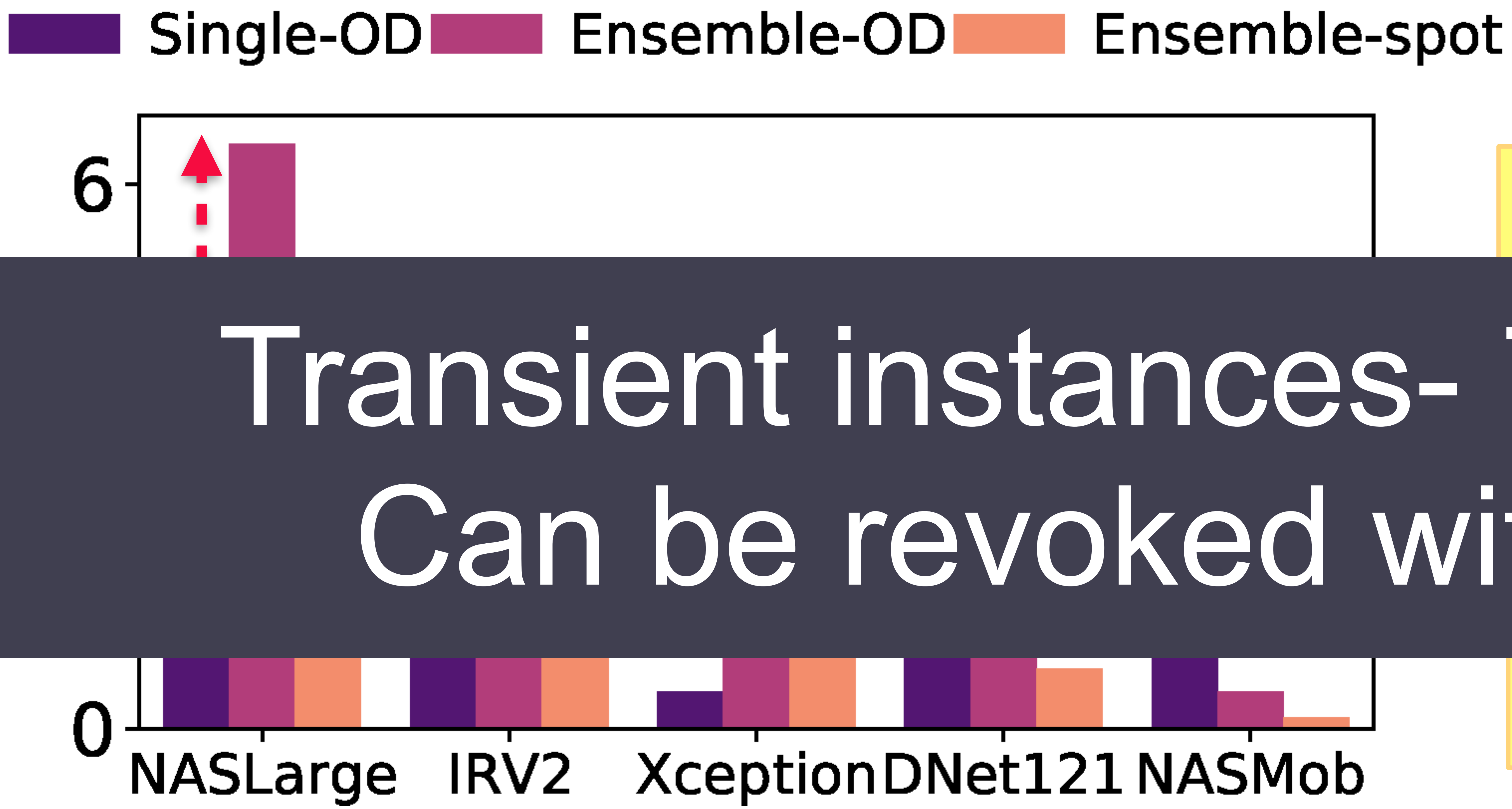
Full Ensembling Cost



Ensembling is up-to 2x expensive.

Spot instances can potentially reduce cost.

FULL ENSEMBLING COST



Ensembling is up-to 2x

potentially reduce cost.

Transient instances- 70-80% cheaper.
Can be revoked with short notice.

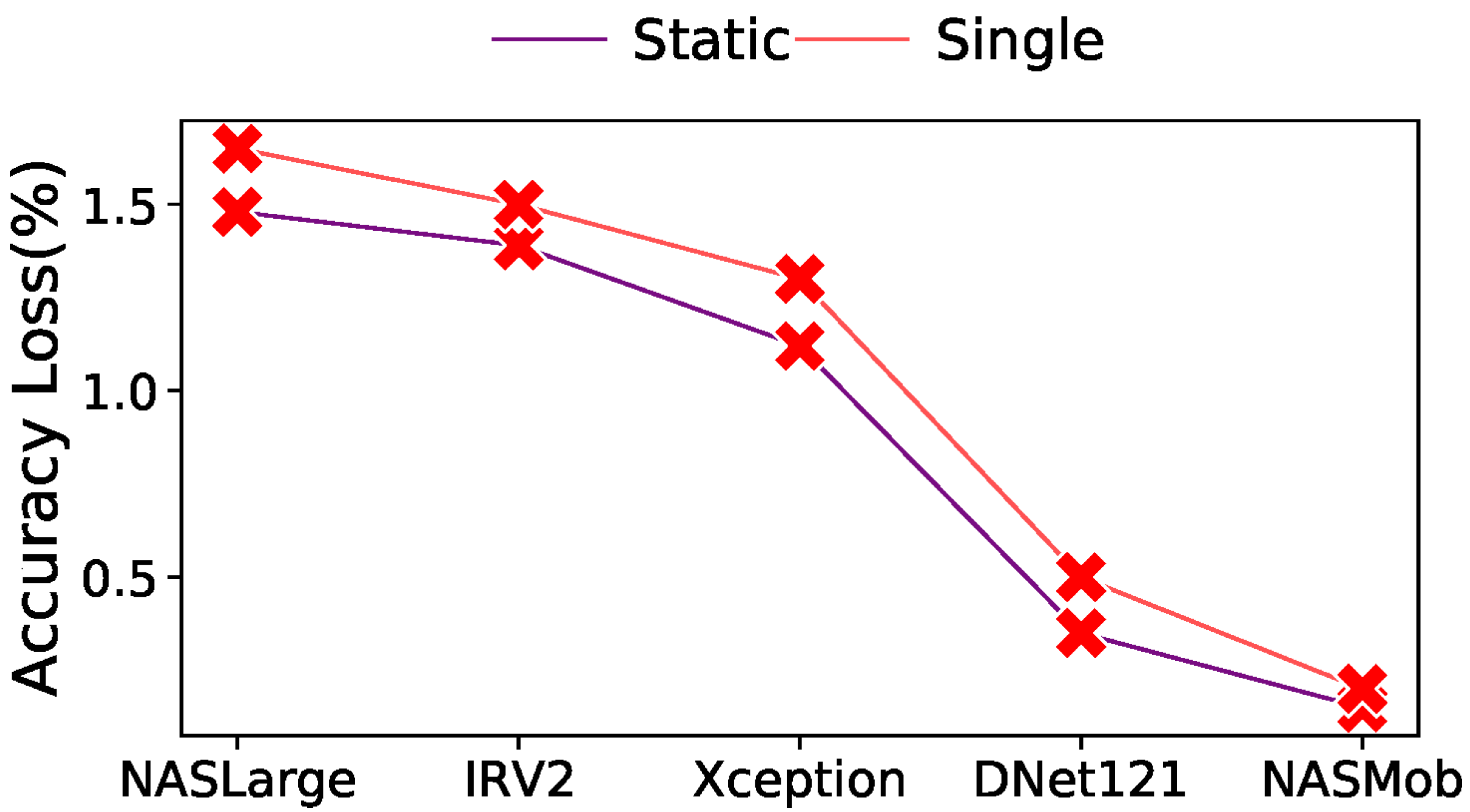
WHAT CAN WE DO?

Baseline(BL)	NASLarge	IRV2	Xception	DNet121	NASMob
#Models	10	8	7	5	2

- Do we need so many models?
- How to autoscale resources for each model?
- How to handle instance failures?

STATIC ENSEMBLING

Compared to Full-Ensemble (N models)

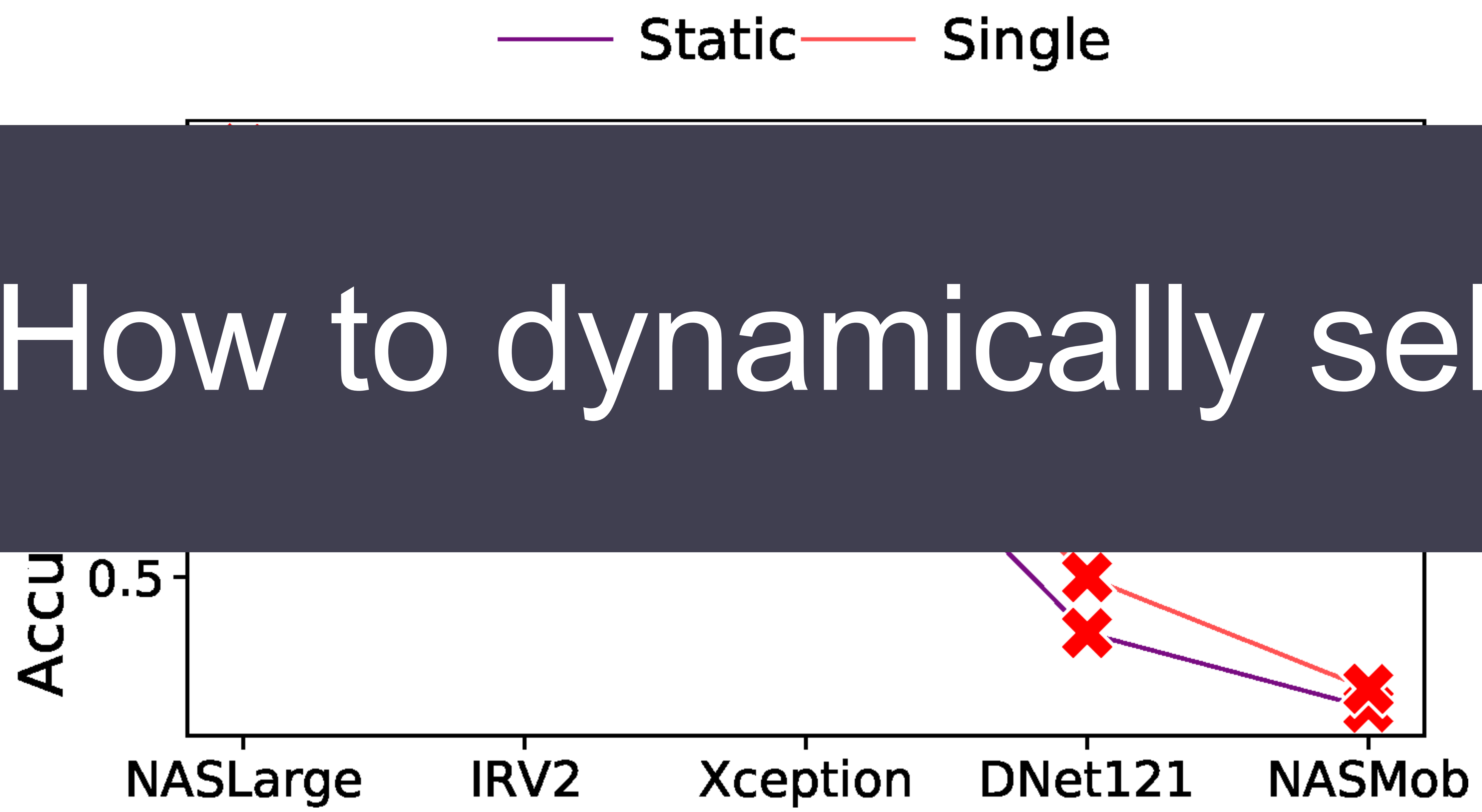


Most accurate N/2 models

Accuracy

STATIC ENSEMBLING

Compared to Full-Ensemble (N models)



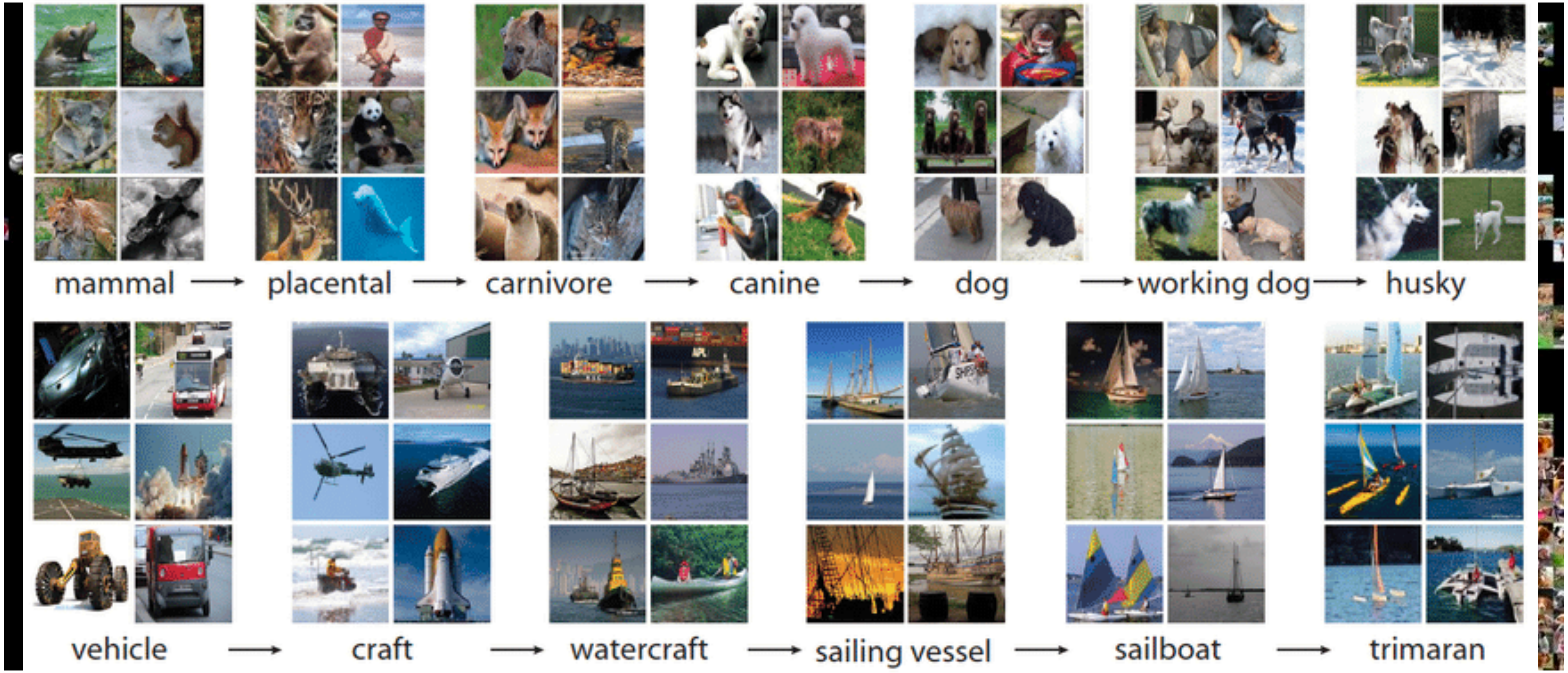
Most accurate N/2

Accuracy

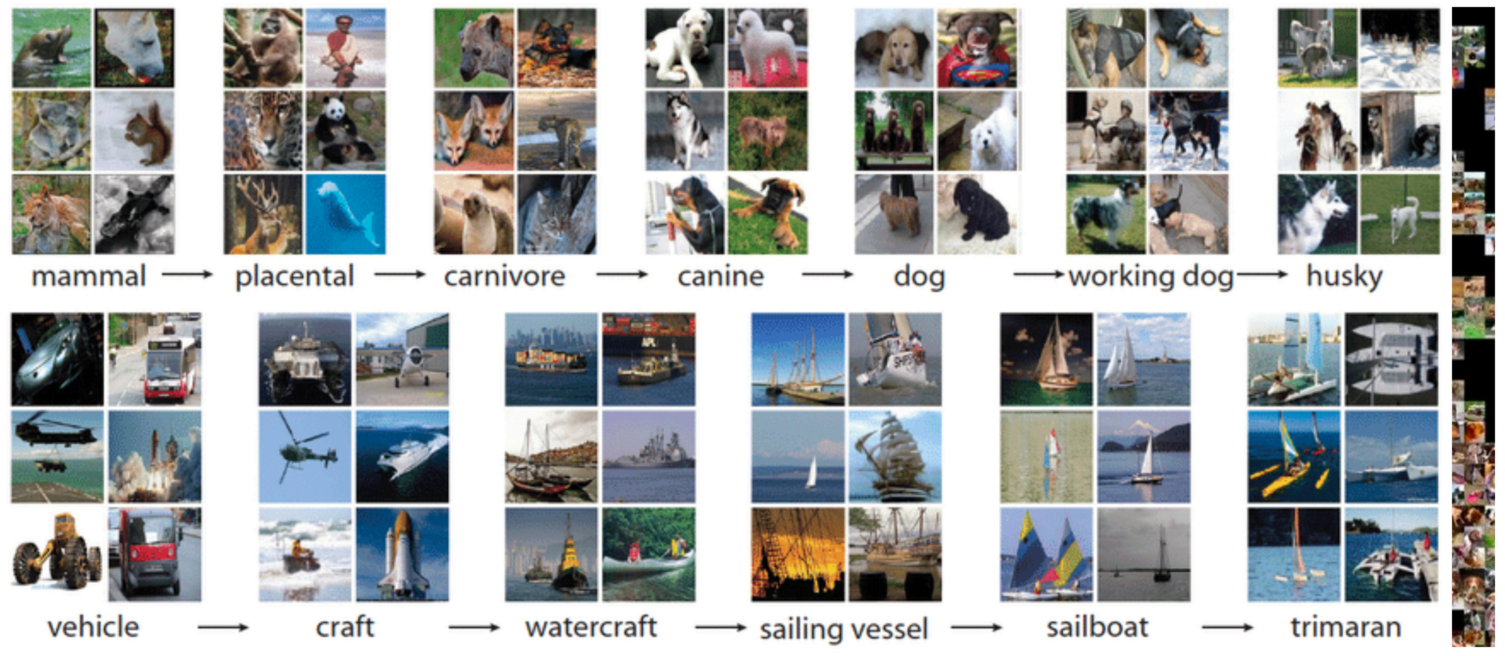
How to dynamically select the models?

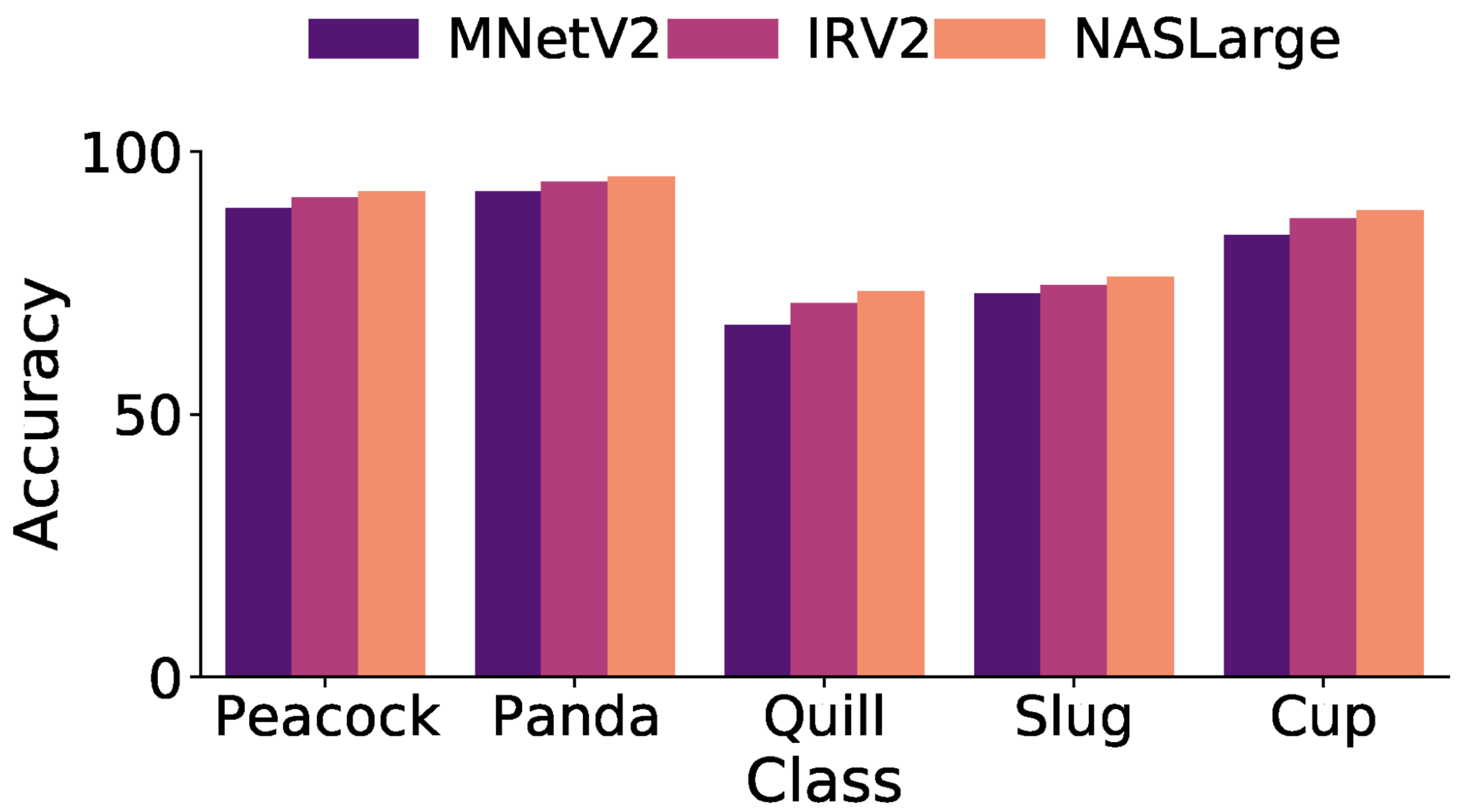
Dynamic Model Selection

DYNAMIC MODEL SELECTION



Dynamic Model Selection

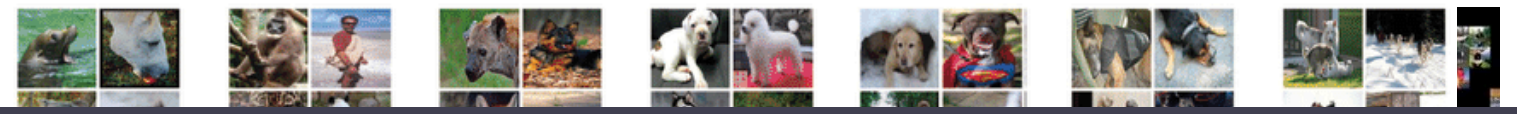


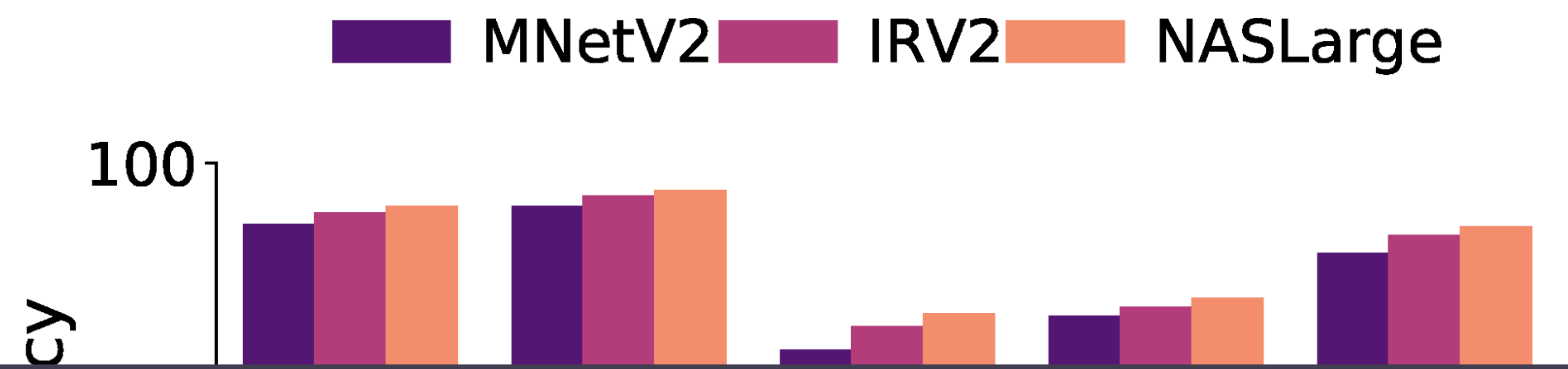


Mobilenet (MNet) ⇨ Slug ☺

Mobilenet (MNet) ⇨ Quill ☹

Dynamic Model Selection





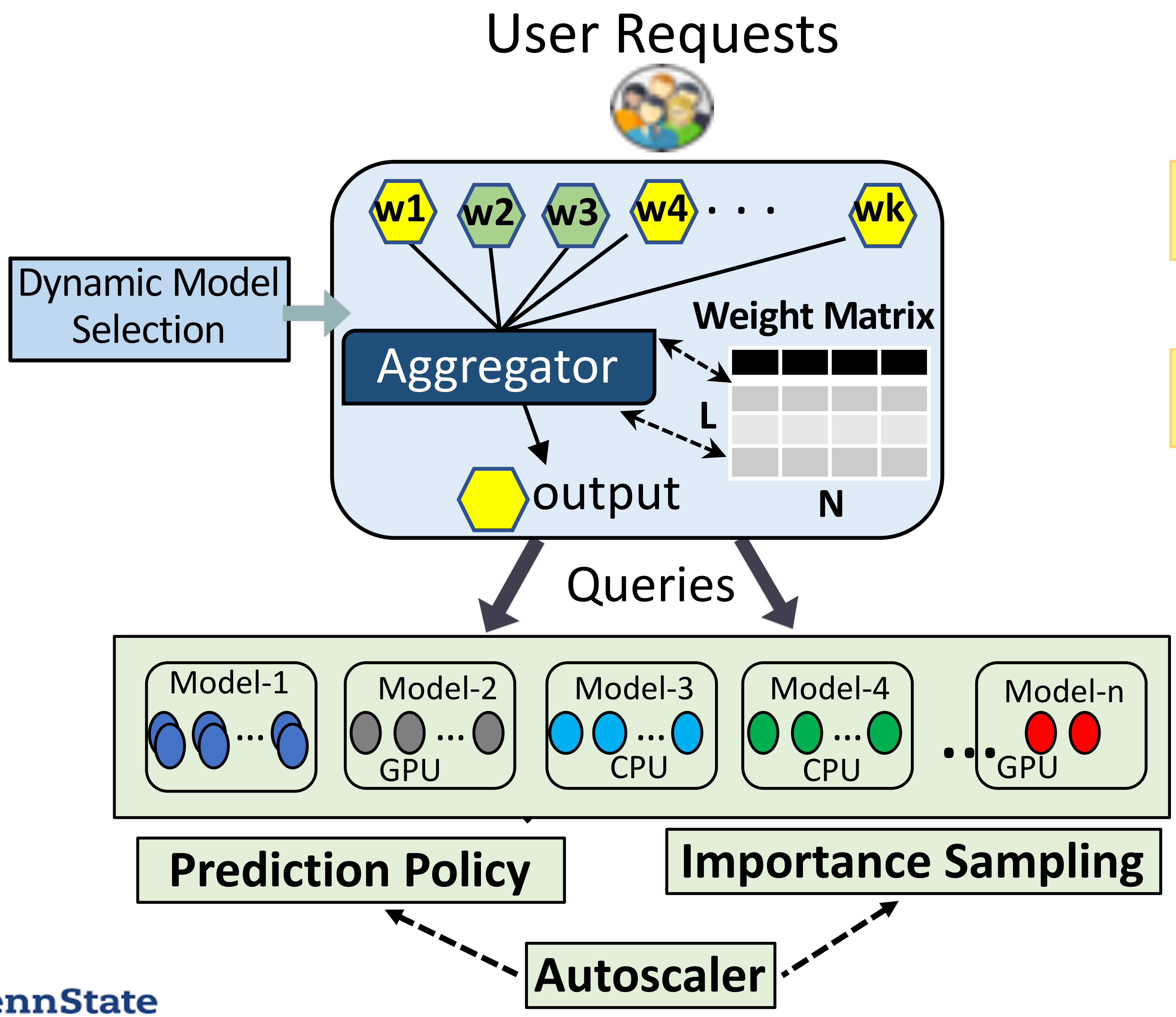
Leverage Class-wise Accuracy

Class

Mobilenet (MNet) ⇨ Slug

Mobilenet (MNet) ⇨ Quill

COCKTAIL- MULTIDIMENSIONAL OPTIMIZATION FOR ENSEMBLE LEARNING IN CLOUD



Class-wise dictionary

Weighted Selection

Dedicated Pools

Per model Scaling

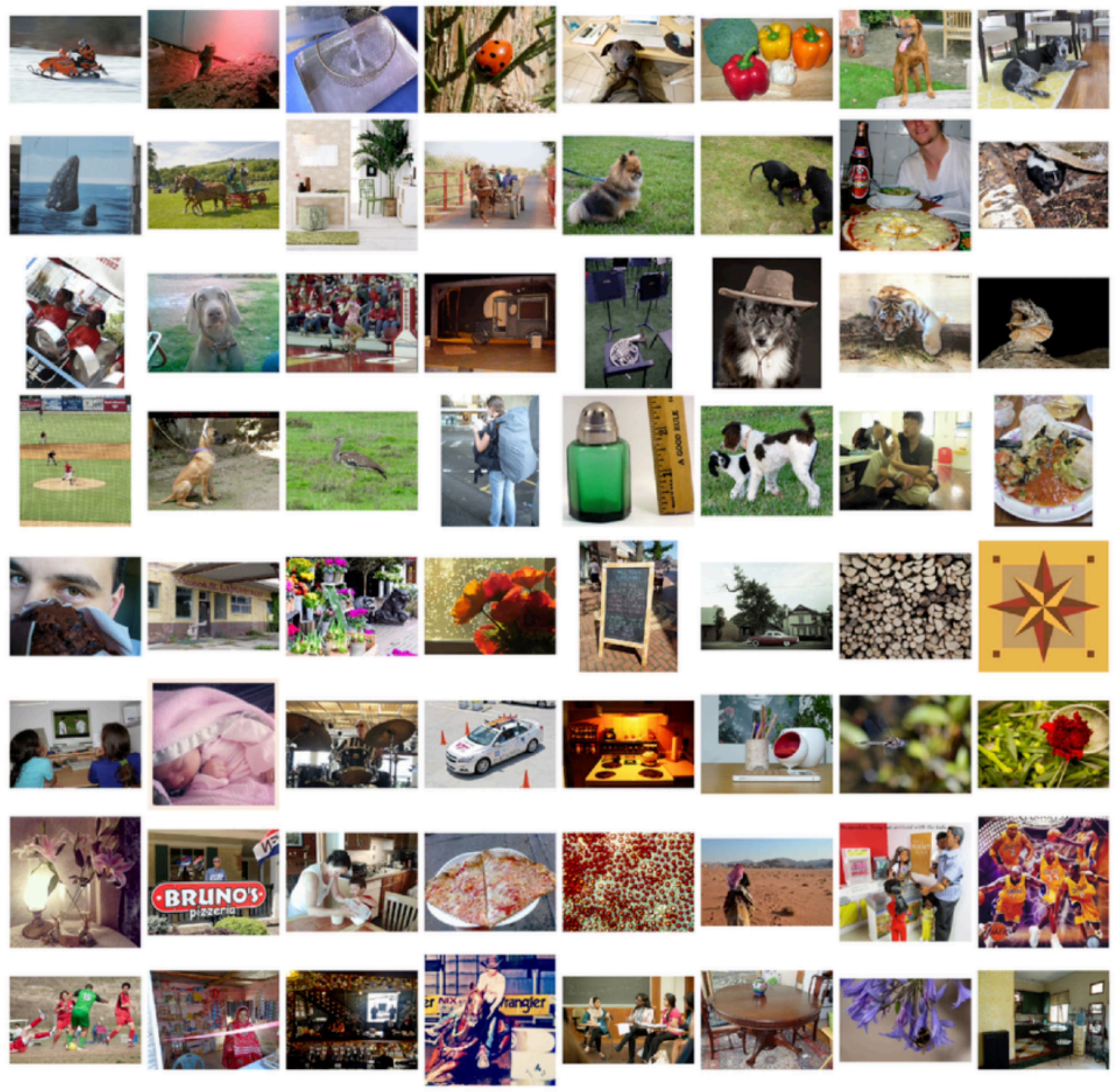
Fault tolerant

Evaluation and Setup

IMAGENET

Amazon EC2

TensorFlow 2.0

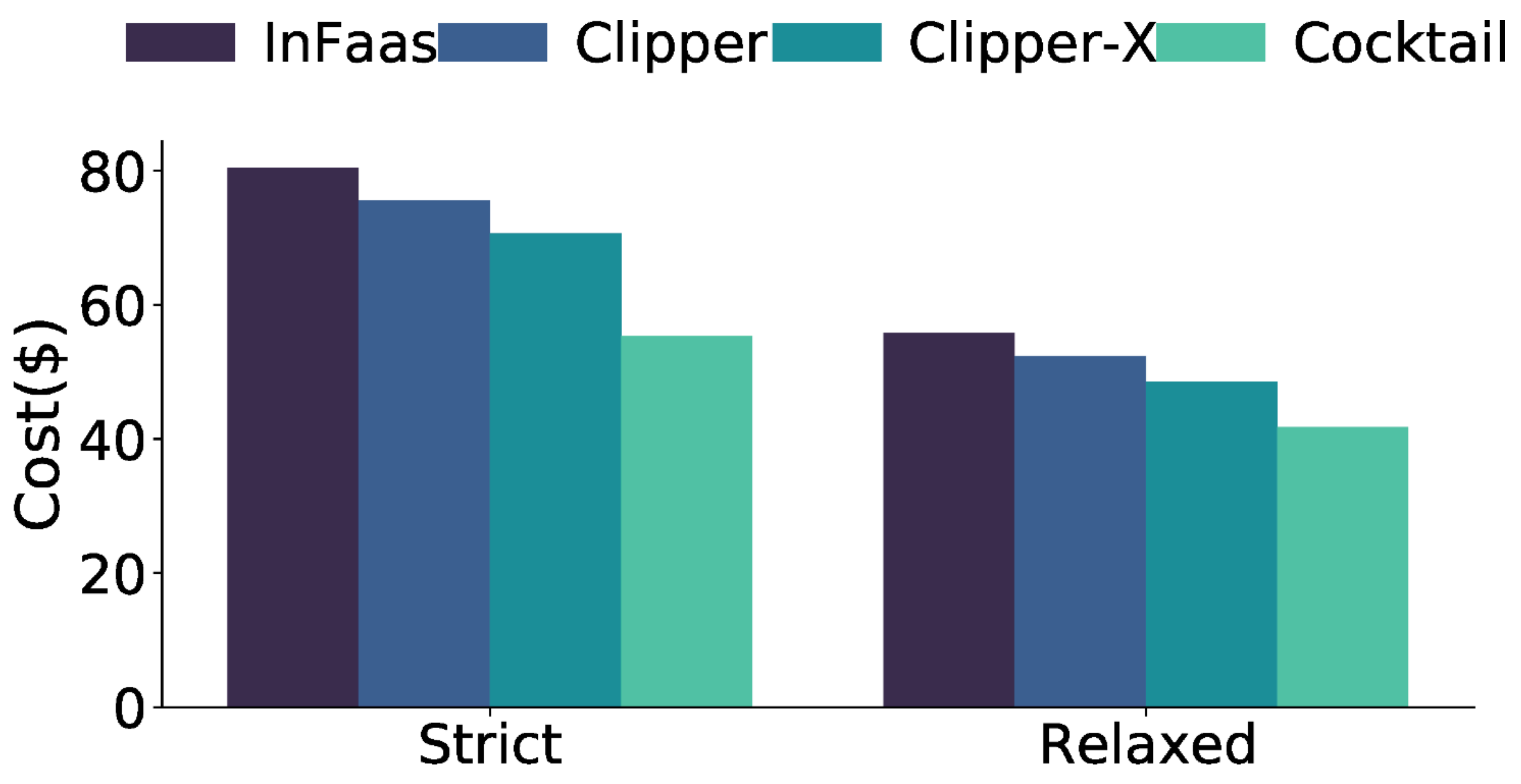


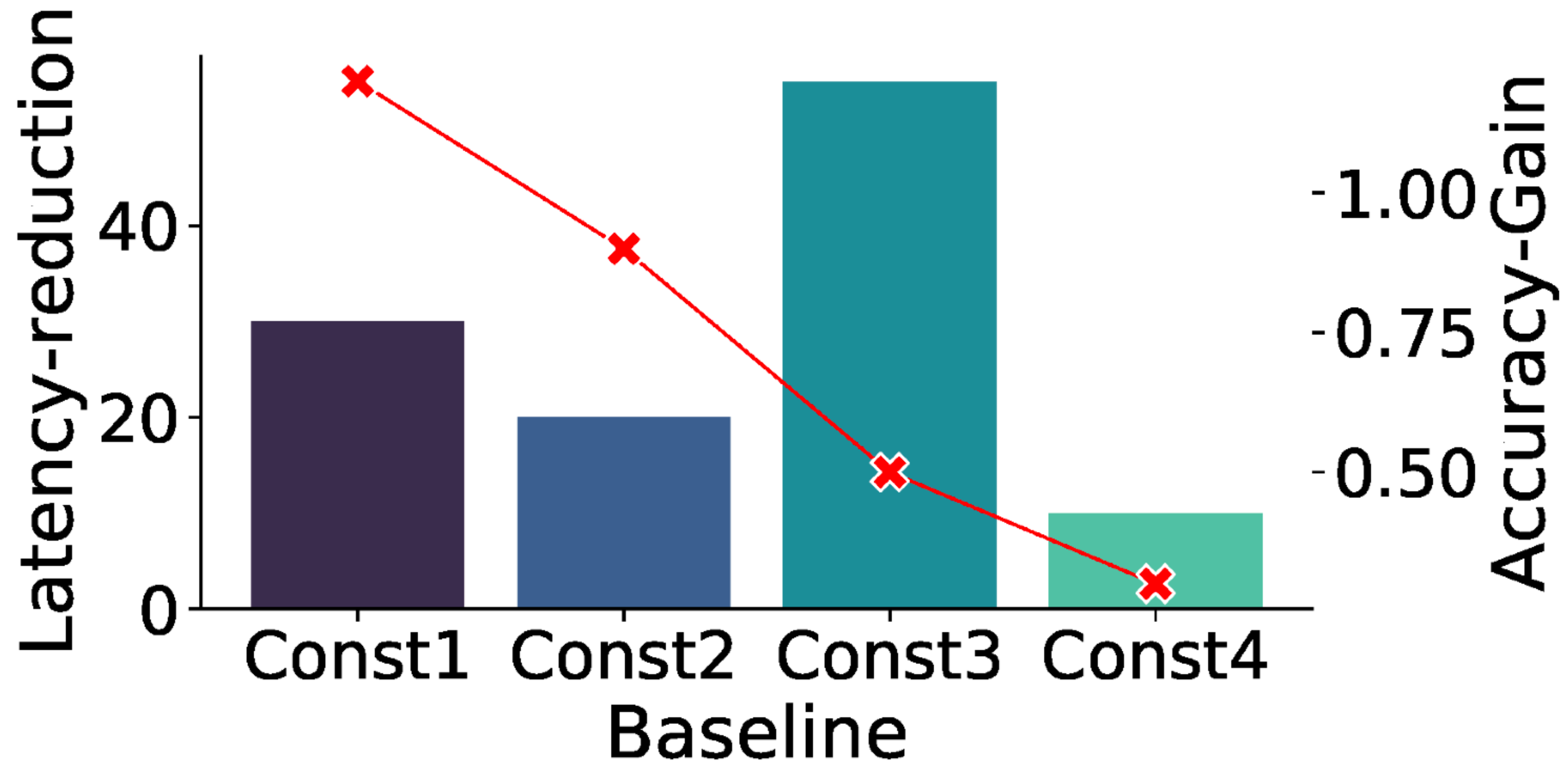
Dataset	Application	Classes	Train-set	Test-set
ImageNet [56]	Image	1000	1.2M	50K
CIFAR-100 [116]	Image	100	50K	10K
SST-2 [117]	Text	2	9.6K	1.8K
SemEval [118]	Text	3	50.3K	12.2K

Experiment Setup

- *40 EC2 CPU/GPU VMs*
- *Wiki Twitter Traces*

Major Results





Cocktail incurs ~**32%** lower cost

Cocktail reduces #models by ~**50%** on average

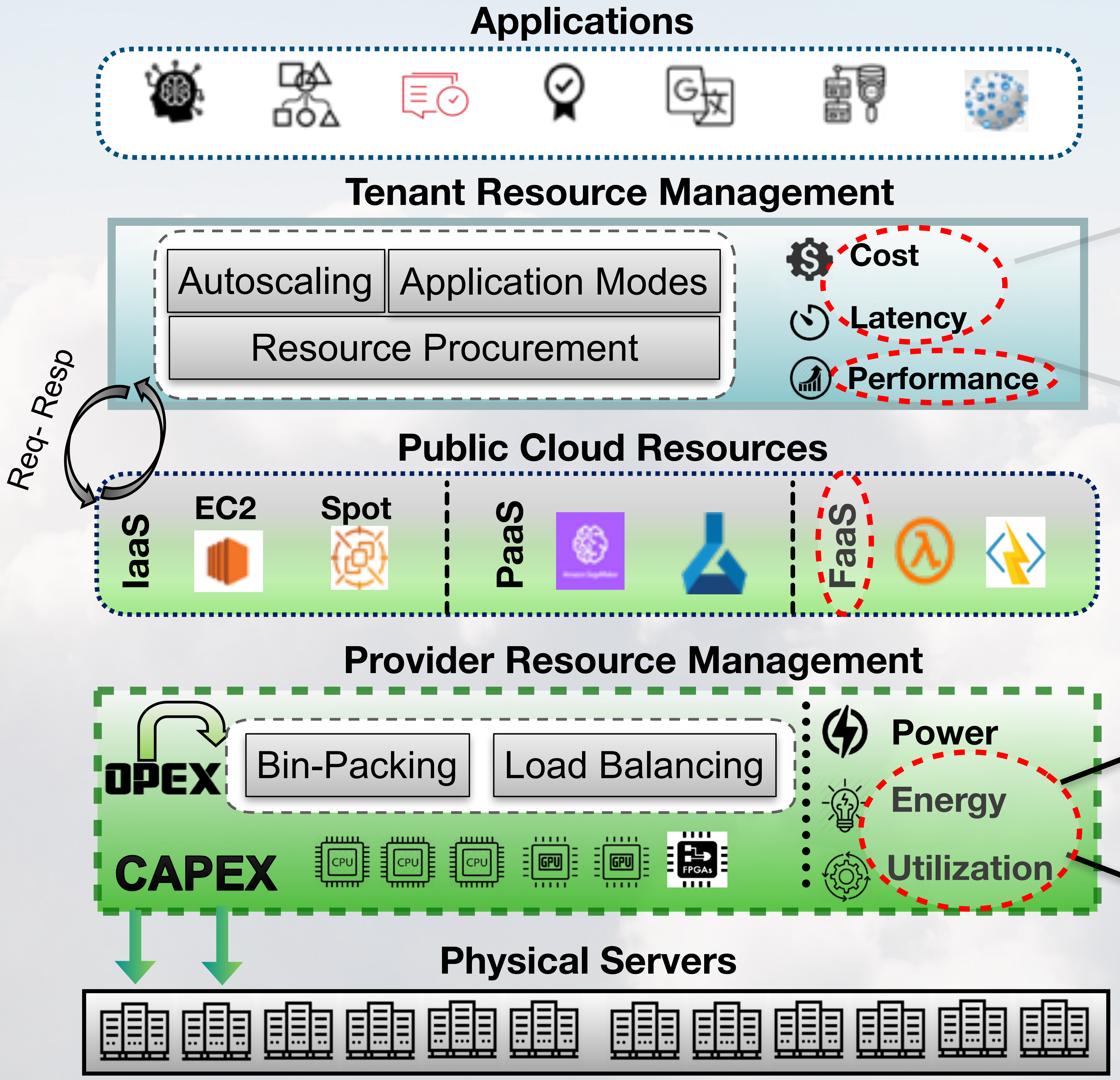
Cocktail yields ~**2x** lower latency

Cocktail gains upto ~**1.25%** more accuracy

PennState College of Engineering

PennState High Performance Computing Lab

Dissertation Contributions



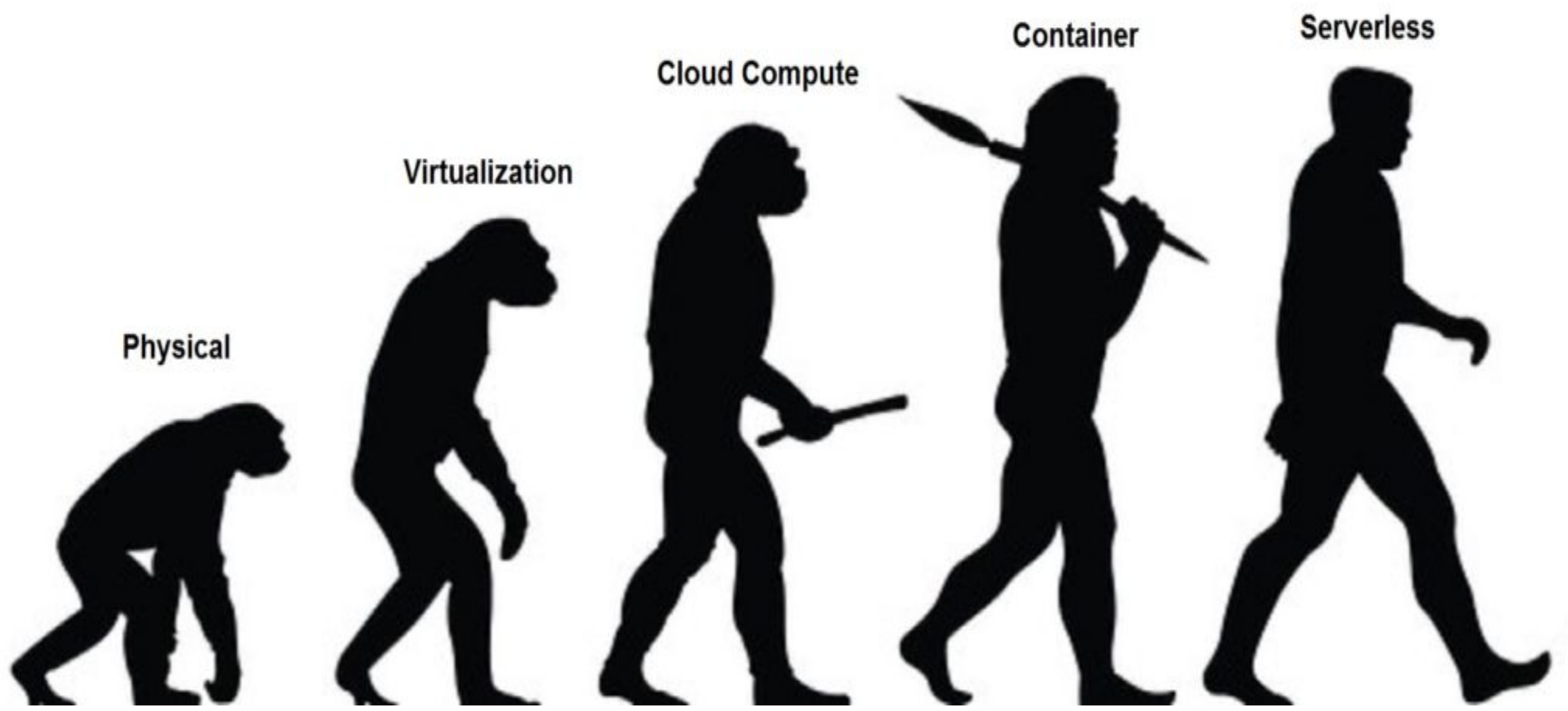
Spock- Cost Efficient and Latency Aware Autoscaling, ***IEEE CLOUD' 2019***

Cocktail- Improving Machine Learning Performance at Low Cost, ***NSDI' 2021*** *(Under-Revision)*

Fifer- Improving Energy Efficiency for Serverless Platforms, ***Middleware, ICDCS 2020***

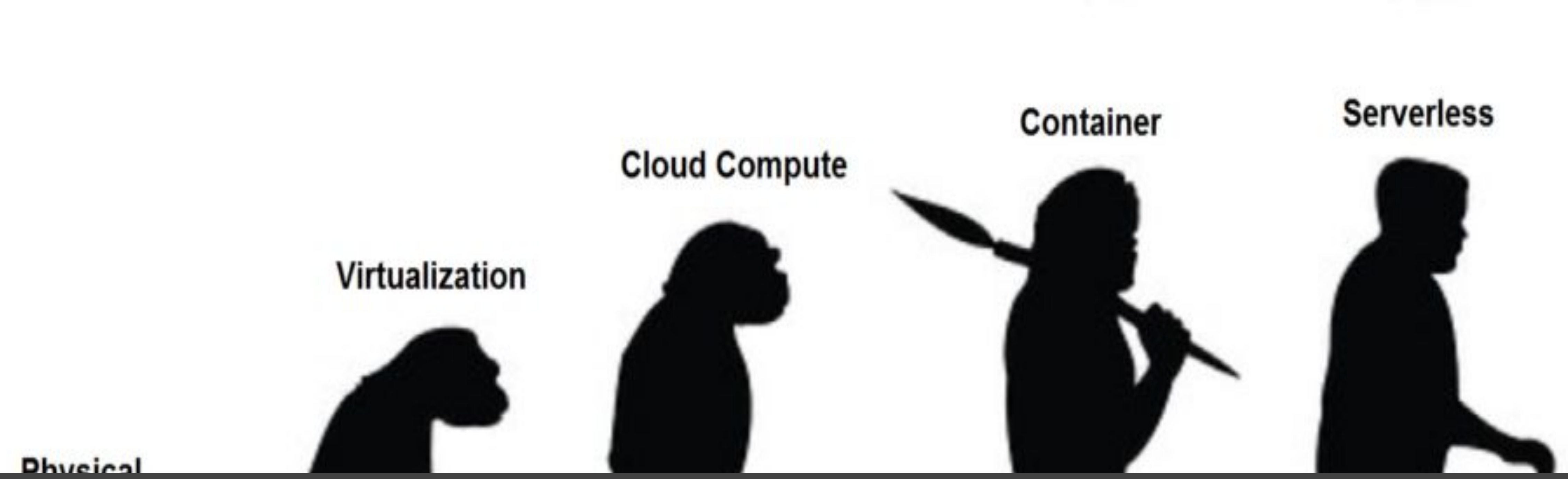
Multiverse- Improving Server Utilization for Private HPC Clusters, ***CCGrid' 2020***

RECAP



58% use Serverless to reduce cost and accelerate development.

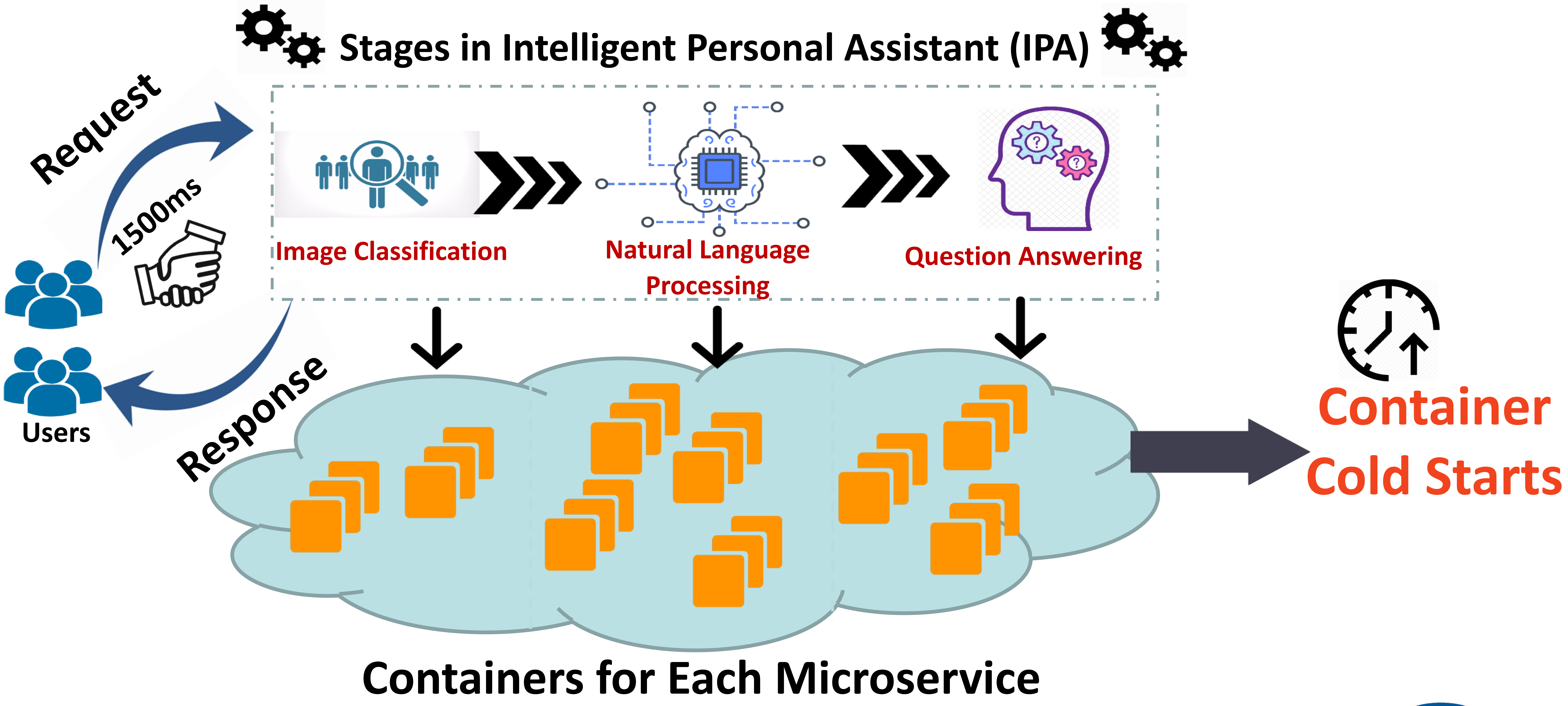
Recap



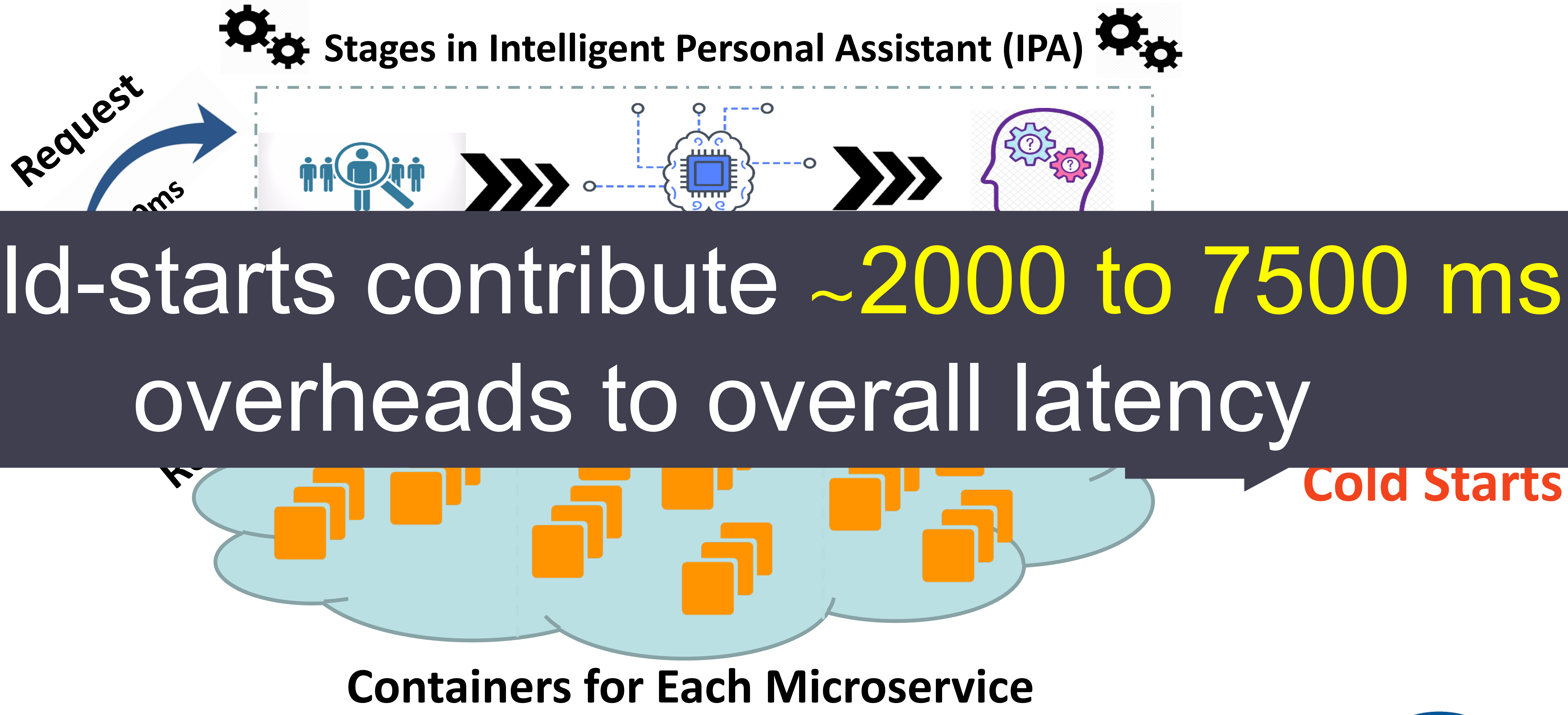
Provider Challenges?

58% use Serverless to reduce cost and accelerate development.

SERVERLESS FUNCTION CHAINS



Serverless Function Chains



Cold-starts contribute ~2000 to 7500 ms overheads to overall latency

CURRENT SERVERLESS PLATFORMS

- Spawn new containers if existing containers are busy.
  - ➡ Leads to SLO violations due to cold-starts.
  - ➡ Many idle containers. Wasted power and energy.

AWS Lambda

- Employing static queuing of requests on fixed pool of containers
  - ➡ Leads to SLO violations due to queuing.

Azure Functions

- Not aware of application execution times and response latency requirements.
  - ➡ Colossal container overprovisioning.

Wang et al, Peeking behind the curtains of Serverless Platforms in ATC'18

Shahrad et al, Serverless in the Wild, in ATC'21

CURRENT SERVERLESS PLATFORMS

- Spawn new containers if existing containers are busy.
  - ➡ Leads to SLO violations due to cold-starts.
  - ➡ Many idle containers. Wasted power and energy.

AWS Lambda

How can we do better?

- Not aware of application execution times and response latency requirements.
  - ➡ Colossal container overprovisioning.

Wang et al, Peeking behind the curtains of Serverless Platforms in ATC'18

Shahrad et al, Serverless in the Wild, in ATC'21

KEY FINDINGS

Slack = Response Latency ⊖ Execution Time (ET)

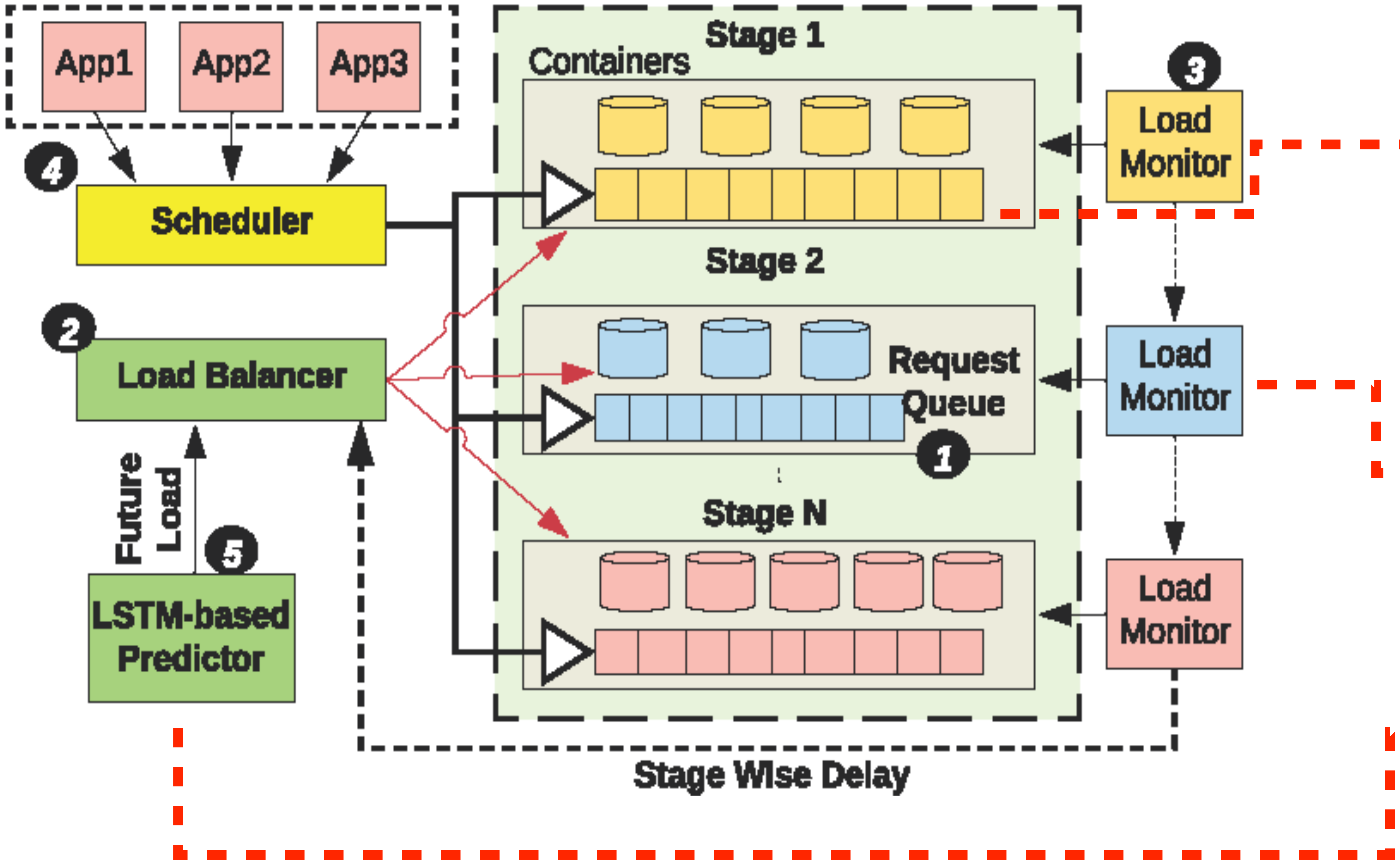
Multi-staged applications have ample slack (200-700ms)

Execution times of each function is predictable- (20-100ms)

Slack > ~7x ET !

Slack Aware Provisioning

FIFER: STAGE-AWARE PROACTIVE CONTAINER PROVISIONING AND MANAGEMENT

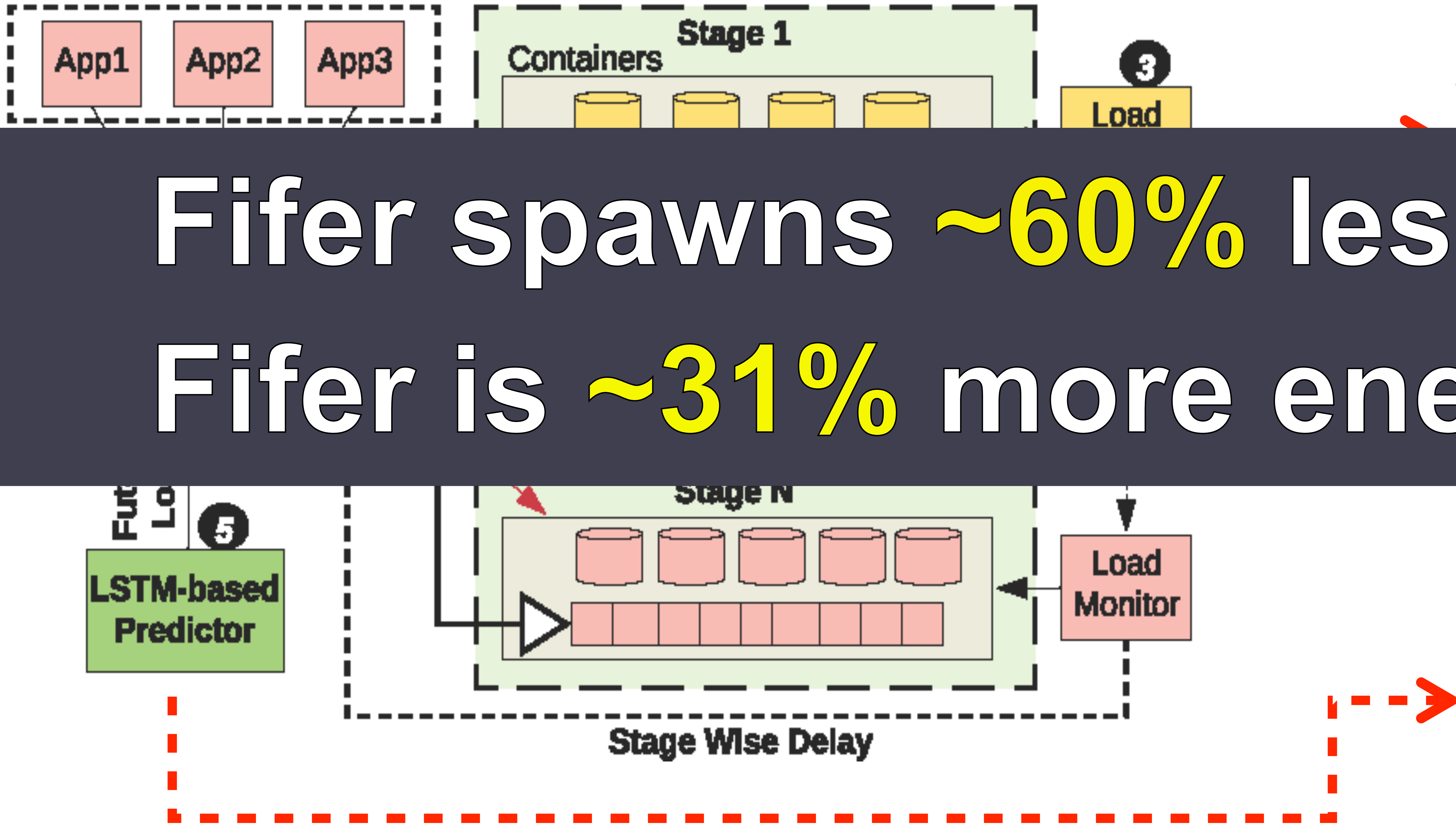


Slack-aware batching and queuing

Queuing delay-based Reactive container scaling

Proactive container scaling

FIFER: STAGE-AWARE PROACTIVE CONTAINER PROVISIONING AND MANAGEMENT



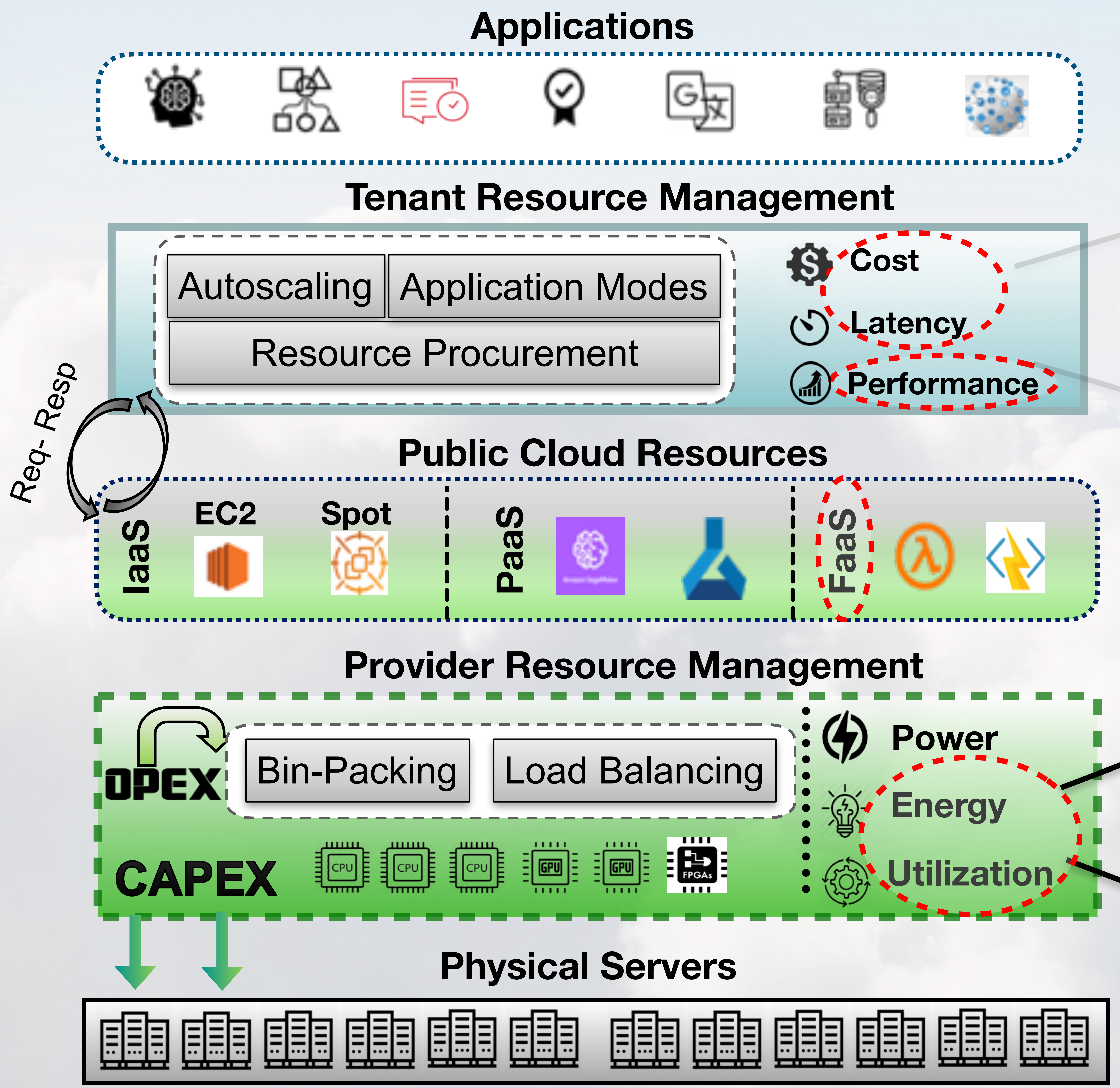
Slack-aware batching

Fifer spawns ~60% less containers.

Fifer is ~31% more energy efficient.

Proactive container scaling

DISSERTATION CONTRIBUTIONS



Spock- Cost Efficient and Latency Aware Autoscaling, ***IEEE CLOUD' 2019***

Cocktail- Improving Machine Learning Performance at Low Cost, ***NSDI' 2021*** *(Under-Revision)*

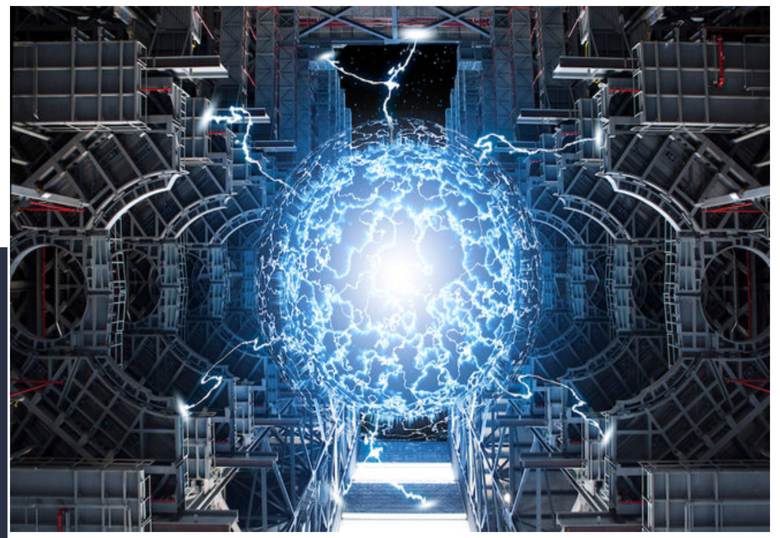
Fifer- Improving Energy Efficiency for Serverless Platforms, ***Middleware, ICDCS 2020***

Multiverse- Improving Server Utilization for Private HPC Clusters, ***CCGrid' 2020***

High Performance Computing

U.S. DEPARTMENT OF ENERGY

Galaxy PROJECT



THE VERGE
US government awards millions to HPE, Intel, and others in hopes they'll build next-gen

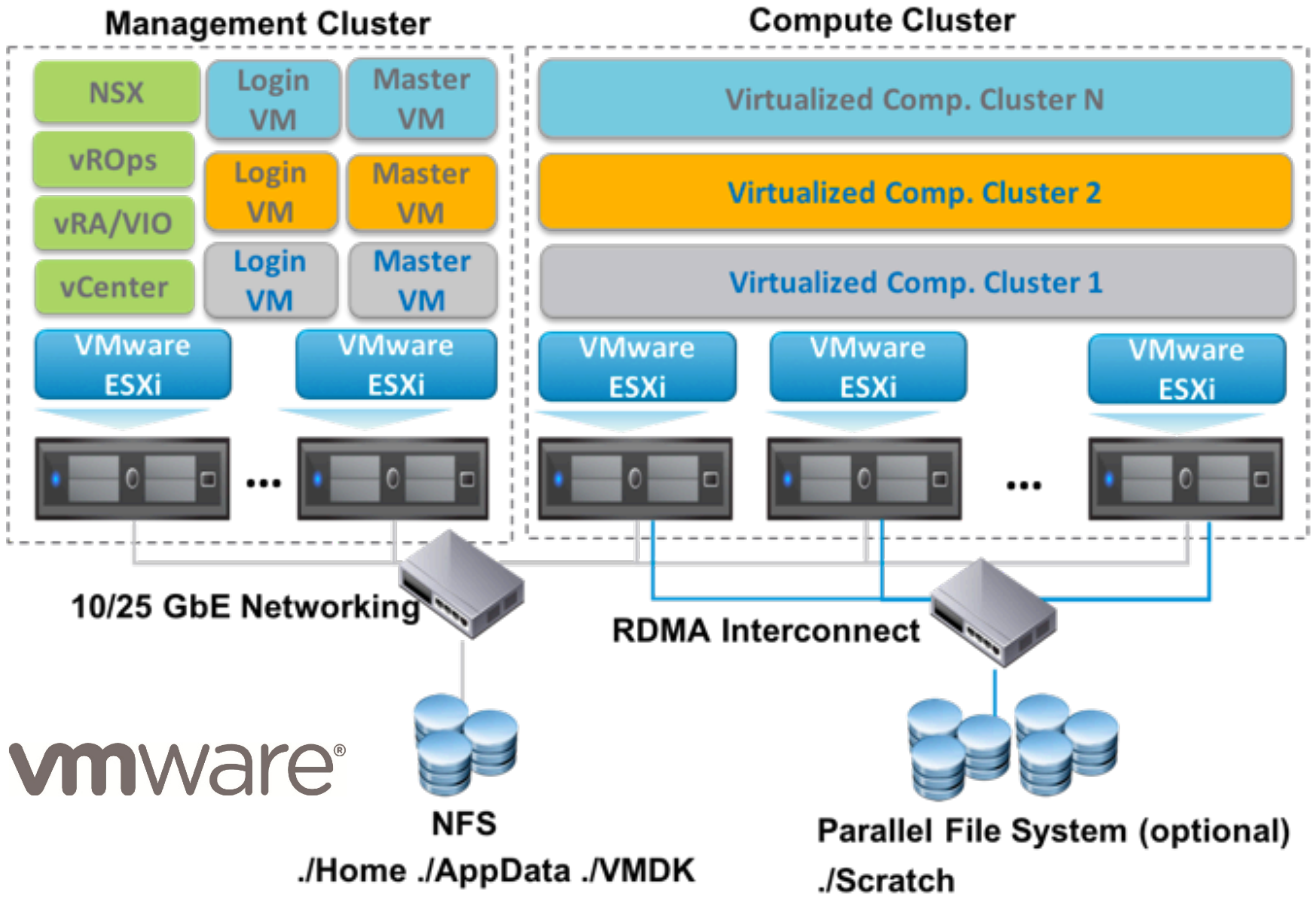
Department of Energy
Secretary of Energy Rick Perry Announces $1.8 Billion Initiative for New Supercomputers

The Worldwide HPC Server Market: $6.7 Billion in First Half 2019

Hyperion: AI-driven HPC Industry Continues to Push Growth Projections
By Doug Black

High-Performance Computing as a Service Market is Expected to Reach $17.00 Billion by 2026, Says Allied Market Research

VIRTUALIZED HPC



Heterogeneous Compute

Flexibility

Isolation and Security

https://blogs.vmware.com/apps/2018/09/vhpc-ra-part1.html

Challenges with HPC

HPC Schedulers

PBS Works™

slurm
workload manager

grid engine

HTCondor
High Throughput Computing

- Focus on throughput and utilization.
- Batch Jobs are usually long running.
- Fair sharing and fixed node reservations.

CHALLENGES WITH HPC

HPC Schedulers

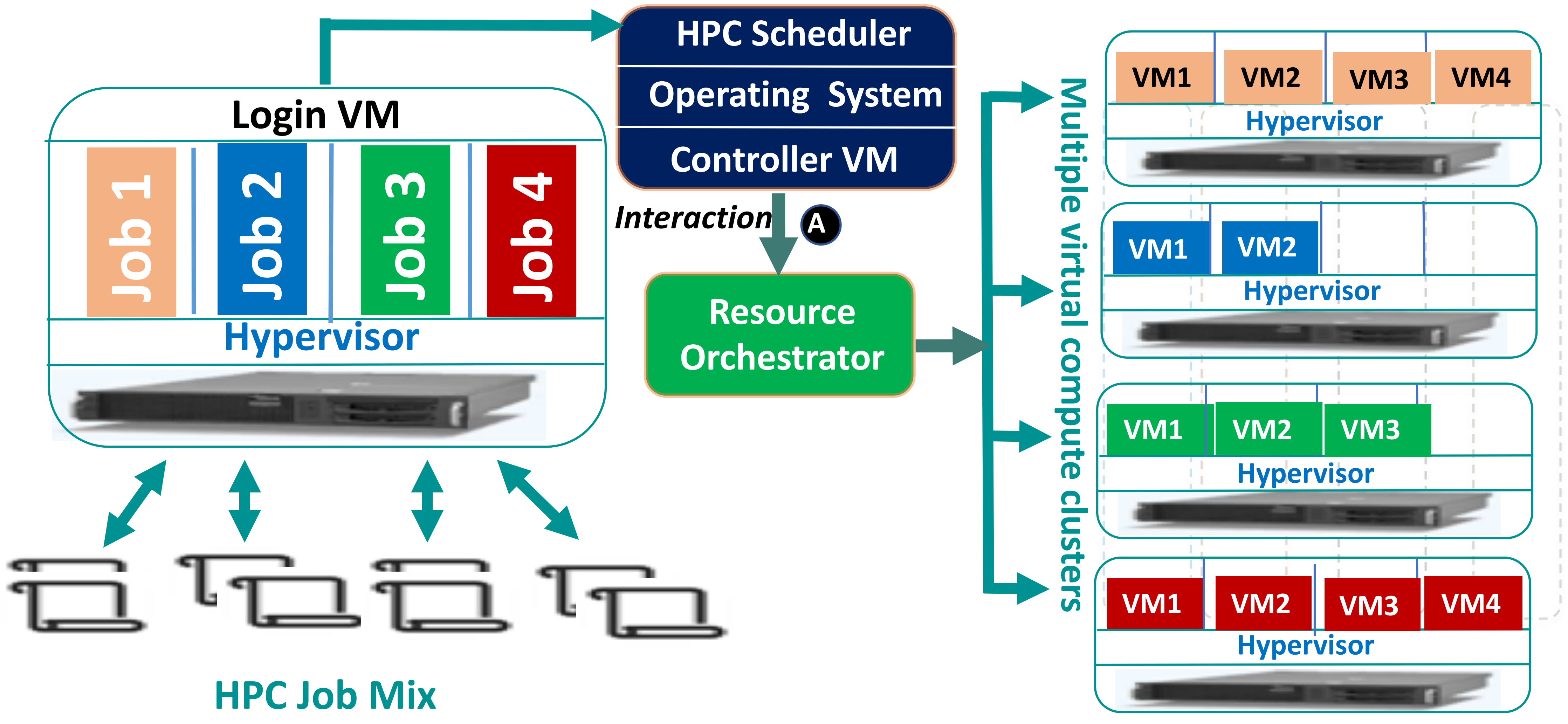
- Focus on throughput and utilization.

No interaction with VM orchestrators
Results in Underutilization

reservations.

High Throughput Computing

Why Underutilization?



- Static Provisioning
- High provisioning times
- Manual Scaling
- No information about physical cluster resources

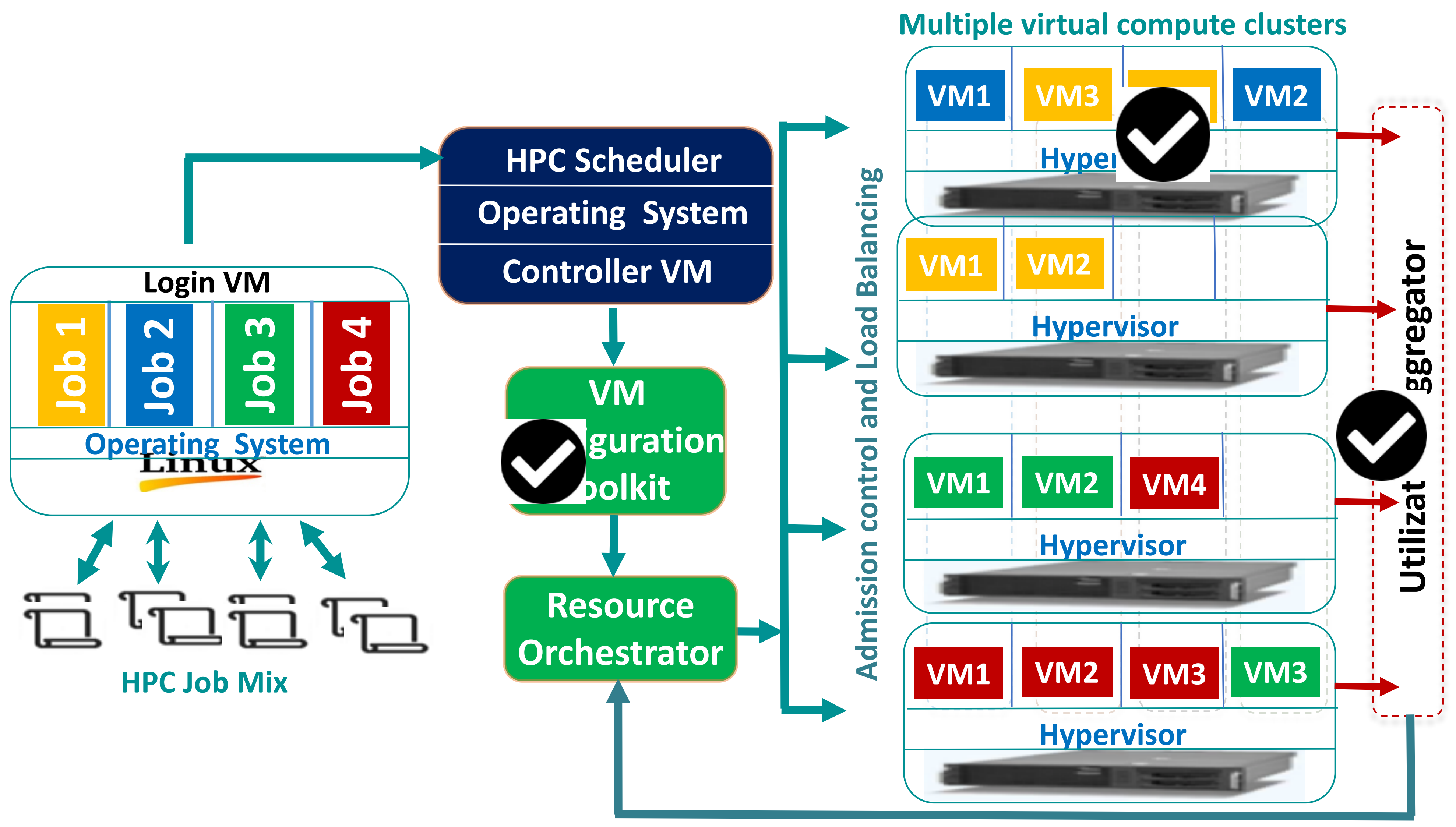
WHY UNDERUTILIZATION?

- Static Provisioning

How to solve this problem?

cluster resources

MULTIVERSE- DYNAMIC VM PROVISIONING FOR HIGH PERFORMANCE COMPUTING CLUSTERS



Seamless interaction with integration

Dynamic VM Provisioning

Leverage Instant Clone

Expose Real-time Cluster Statistics

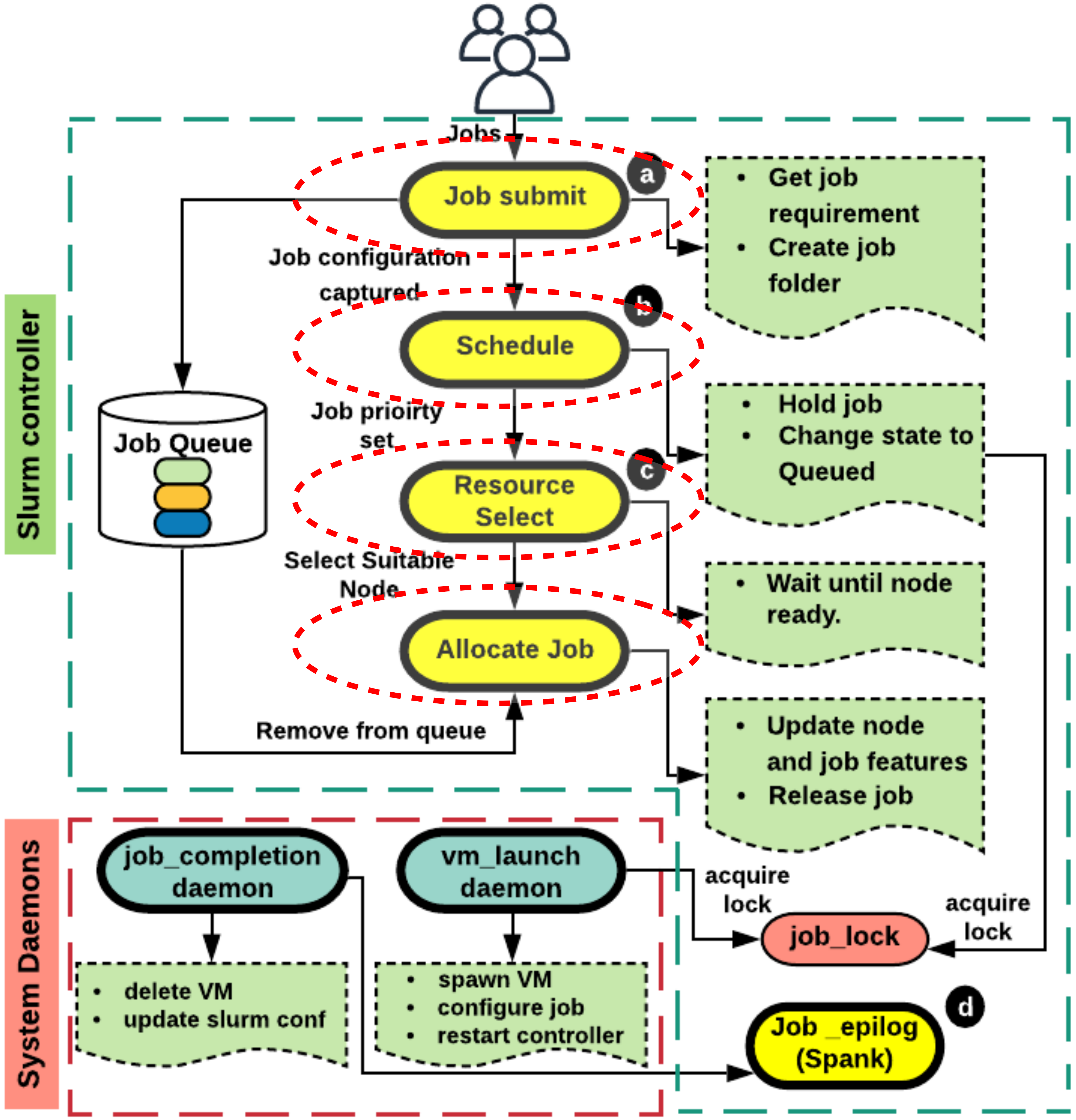
MULTIVERSE DESIGN

- Parse Job Requirements
- Customized VM launch
- Map Jobs to VMs (concurrency)

- Need to be thread-safe
- Schedulers are multi-threaded and are thread-safe.

We built a thread safe finite-state machine using linux flock utility.

Implementation on Slurm



Each phase corresponds to a plugin

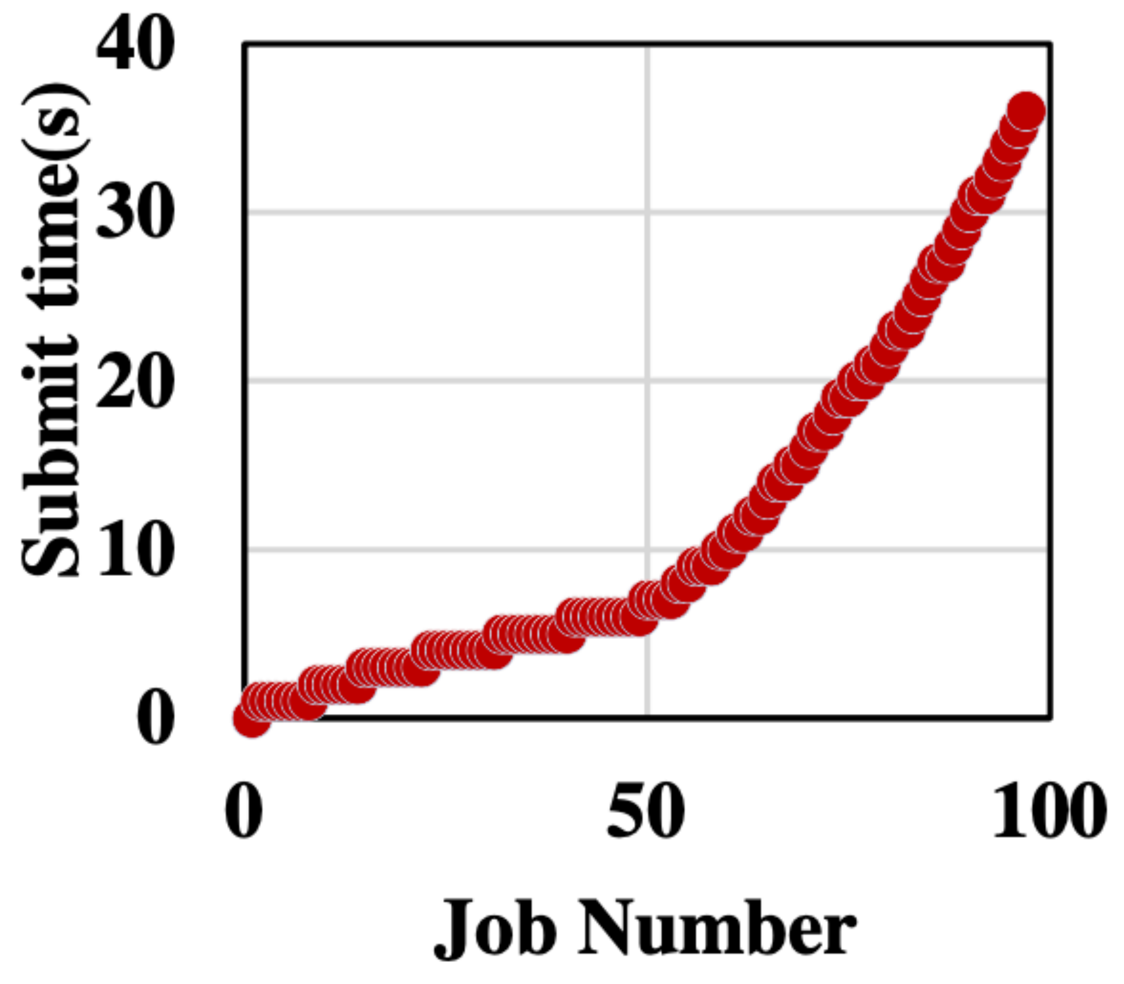
System Daemons ensure concurrency

Spank Plugins for VM Cleanup

EVALUATION SETUP

Experiment Setup

- *220 core* HPC cluster.
- *1TB* Memory
- *72TB* shared datastore



Workload

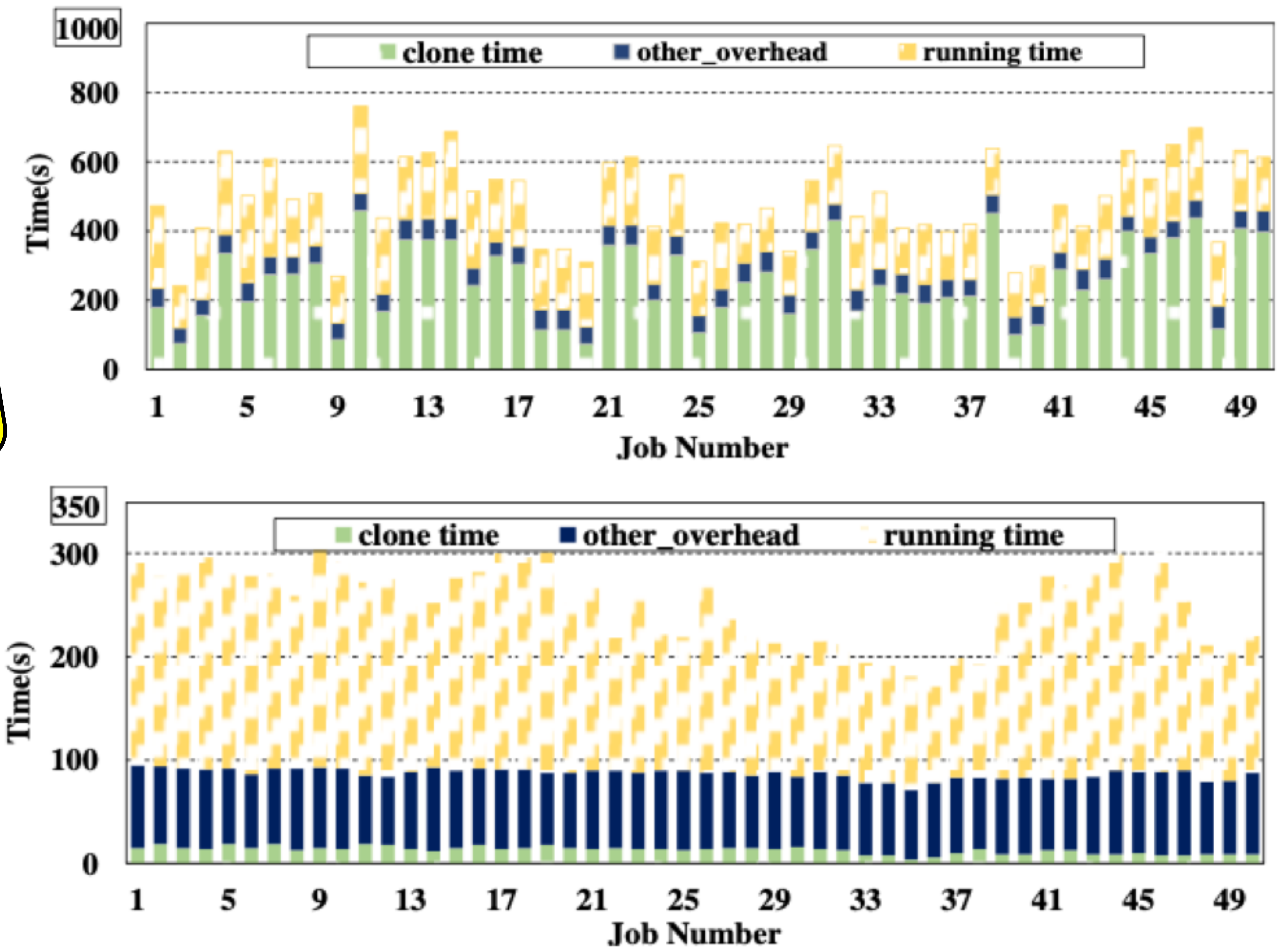
- HPCC, HPL, RandomAccess.
- Small (2vCPU, 4GB), Large (8vCPU, 16GB)
- 50 job/s, 100jobs/s

Major Results

Full Clone

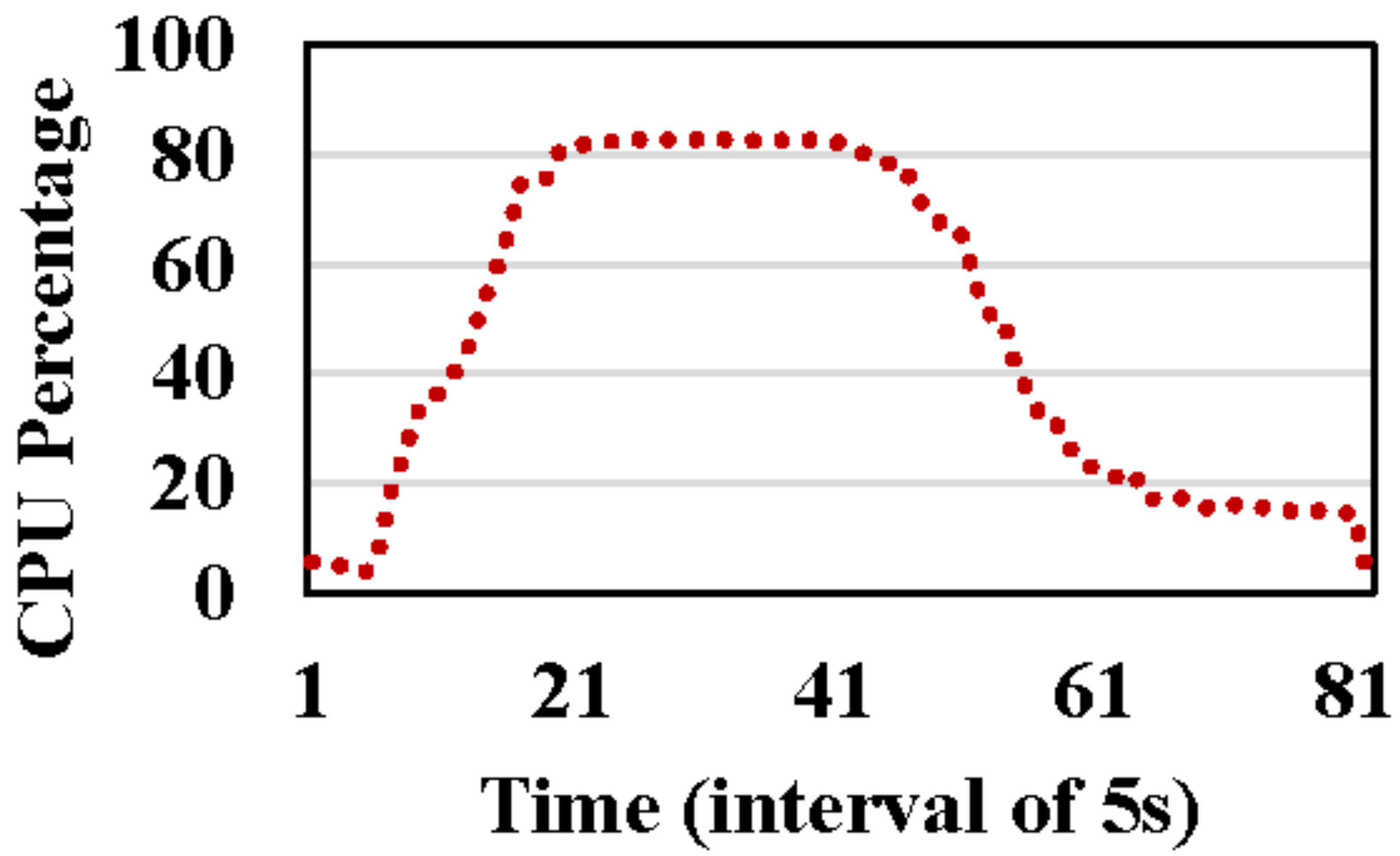
~3x Fast!

Instant Clone

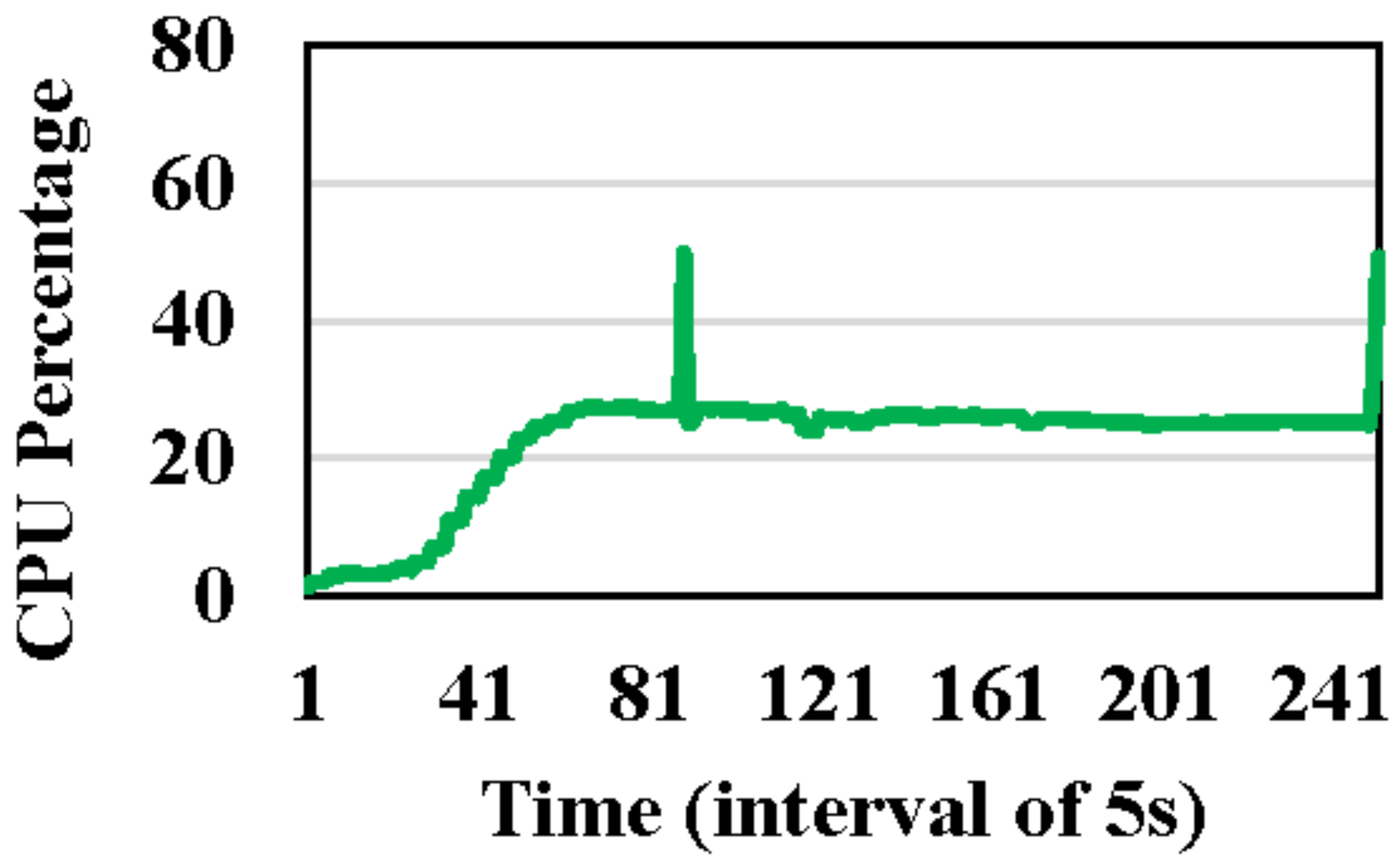


Major Results

Instant Clone



Full Clone



~**1.5x** more throughput.

~**40%** higher CPU utilization.

Future Research Directions

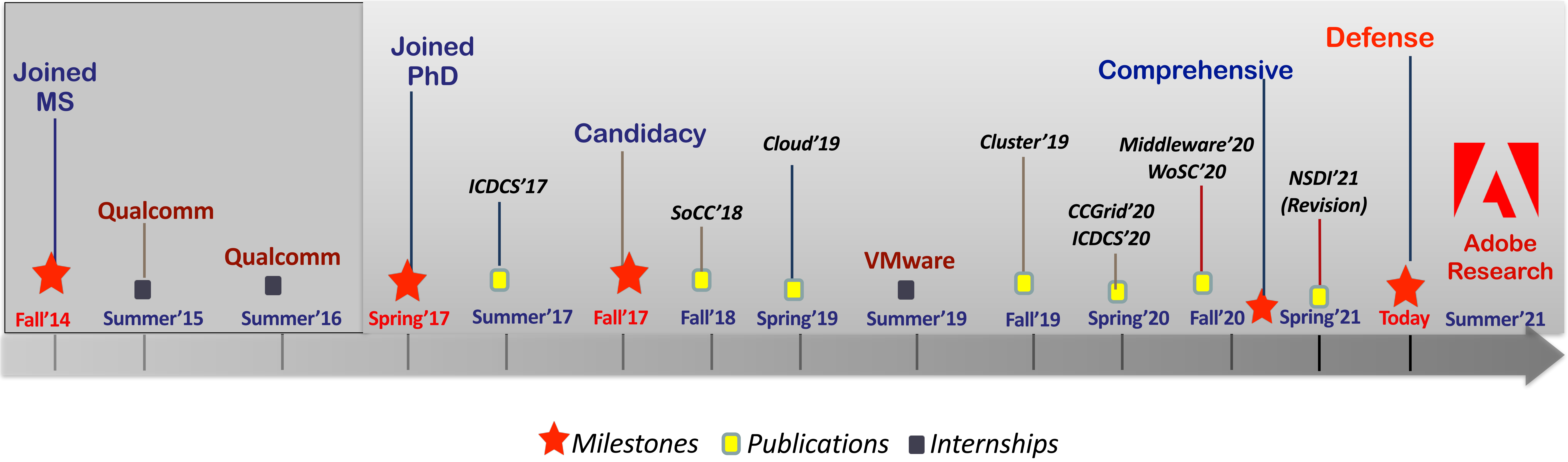
Short Term

- Dynamic DAGs in Serverless
- Stateful Serverless Storage Costs
- Machine Learning Training Costs

Long Term

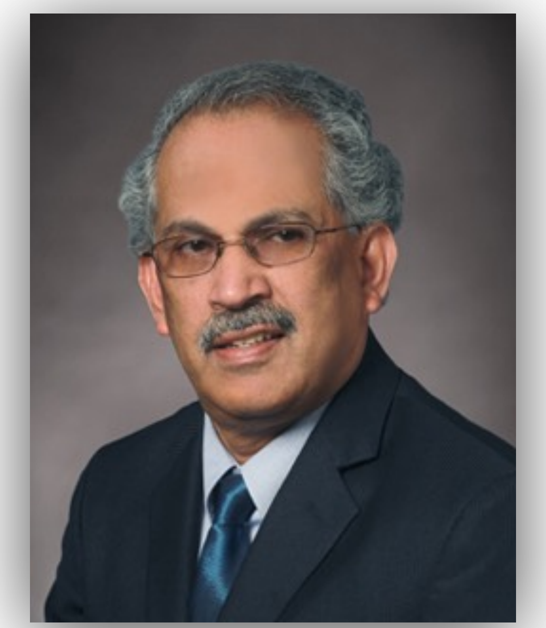
- Federated learning in Public Cloud
- Online Real-time training using serverless
- HPC in public cloud

My Timeline


Joined MS
Qualcomm
Qualcomm
Joined PhD
ICDCS'17
Candidacy
SoCC'18
Cloud'19
VMware
Cluster'19
CCGrid'20
ICDCS'20
Middleware'20
WoSC'20
Comprehensive
NSDI'21
(Revision)
Defense
Adobe Research
Fall'14
Summer'15
Summer'16
Spring'17
Summer'17
Fall'17
Fall'18
Spring'19
Summer'19
Fall'19
Spring'20
Fall'20
Spring'21
Today
Summer'21
Milestones
Publications
Internships

Doctoral Committee

Dr. Mahmut Kandemir
Dissertation Co-Advisor
Professor
Department of CSE



Dr. Chita Das
Dissertation Co-Advisor
Distinguished Professor
Head of CSE Department

Dr. George Kesidis
Committee Member
Professor
Department of CSE

Dr. Bhuvan Urgaonkar
Committee Member
Associate Professor
Department of CSE

Dr. Anton Nekrutenko
Committee Member
Professor
Department of BME

Dr. Bikash Sharma
Special Member
Infrastructure Engineer
Facebook

ACKNOWLEDGEMENTS

Nachiappan

Prashanth

Prasanna

Adhi (My Wife)

Ria (My Kid)

Acknowledgements

All other fellow lab mates

Thank You