Serverless Isn’t Server-Less

Measuring and Exploiting Resource Variability on Cloud FaaS Platforms

Sam Ginzburg, Michael J. Freedman

What is Serverless Computing?

External Events →  

→ Response, Output

What is Serverless Computing?

- **Consumption-based pricing** vs **Allocation-based pricing**
  - Customers pay for usage and not resource allocation
- Serverless & Consumption-based pricing enables new applications
  - Starling (SIGMOD 2020)
  - Pocket (OSDI 2018)
  - Serverless Linear Algebra (SoCC 2020)
  - and many more!
- What are the infrastructural implications?

Misplaced Incentives in Serverless

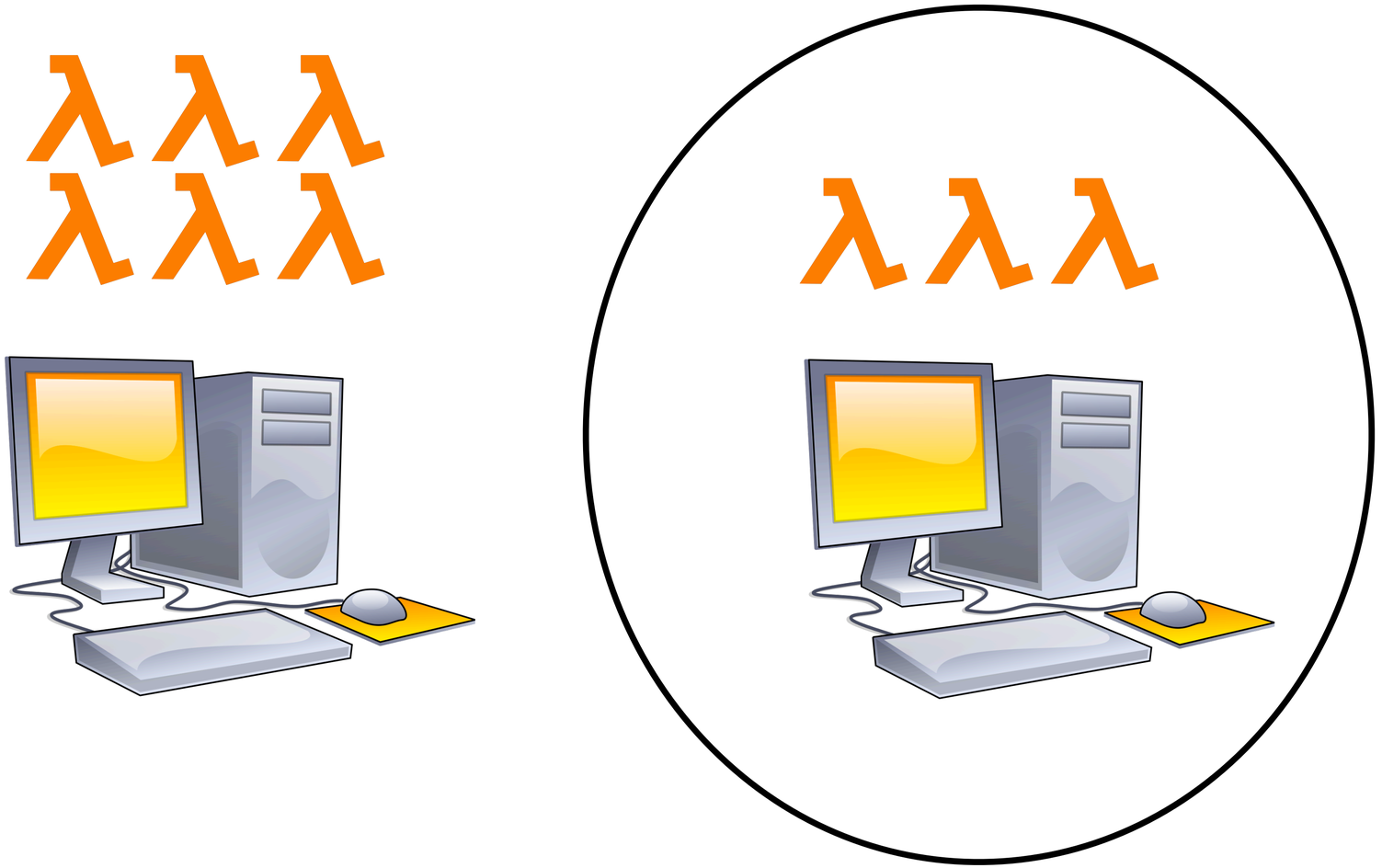
- There is a strong financial incentive to oversubscribe machines
  - Resources can’t be pre-allocated
  - The goal for serverless providers is to hit 100% resource utilization
- Not all time slices are equal to each other!
  - Performance variation means that you don’t always get what you pay for!

Serverless Tradeoffs

- Serverless Platforms make important tradeoffs that affect performance
  - Serverless infrastructure optimizes for resource utilization (by design)
  - The consumption-based pricing model means customers pay a fixed price
- Can customers optimize function placements to perform ***placement gaming?***

Serverless Tradeoffs

- Can customers optimize function placements to perform ***placement gaming?***



Motivation

1. Does performance variation exist in AWS Lambda?
    1. Is it possible to perform placement gaming?

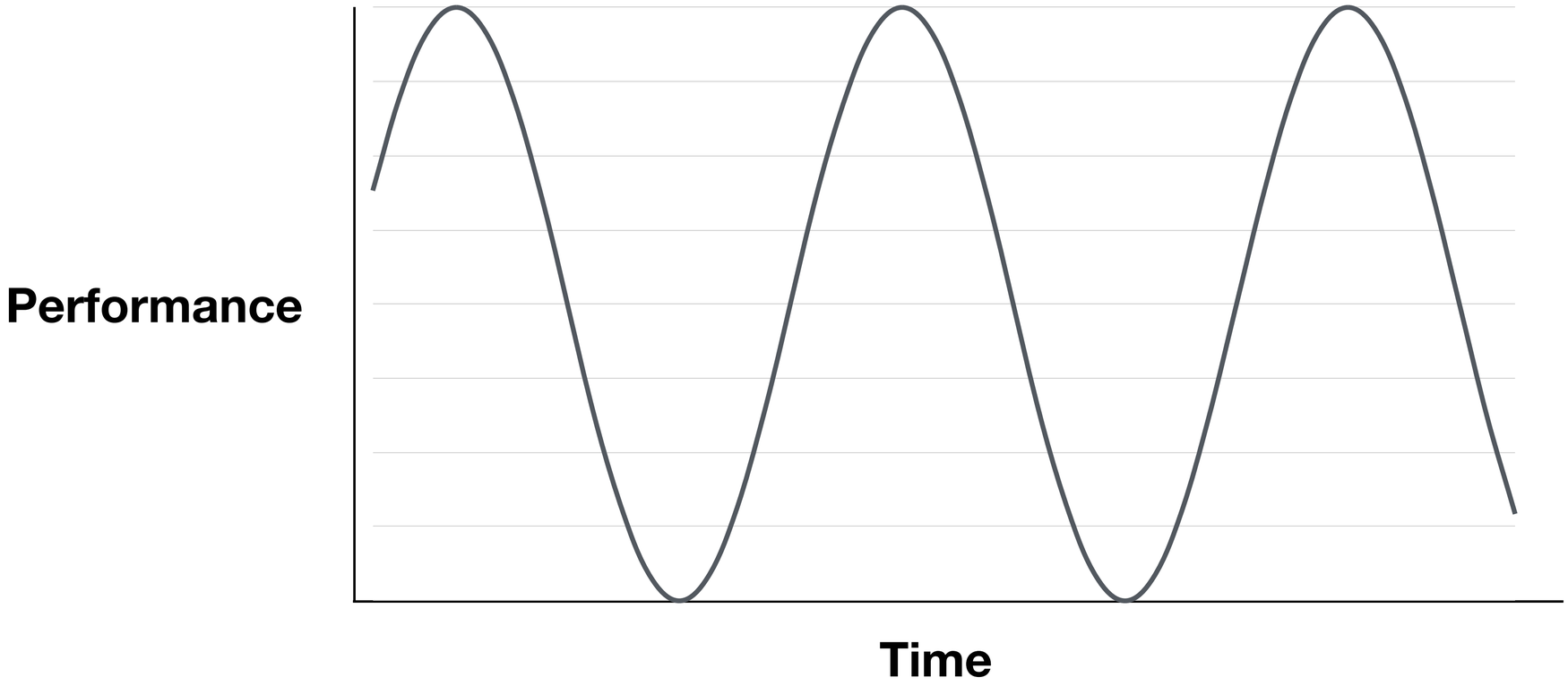
2. If so - is placement gaming on AWS Lambda worth it?

Measurement Study

The goal of our measurement study is to identify three dimensions across which we can explore performing placement gaming

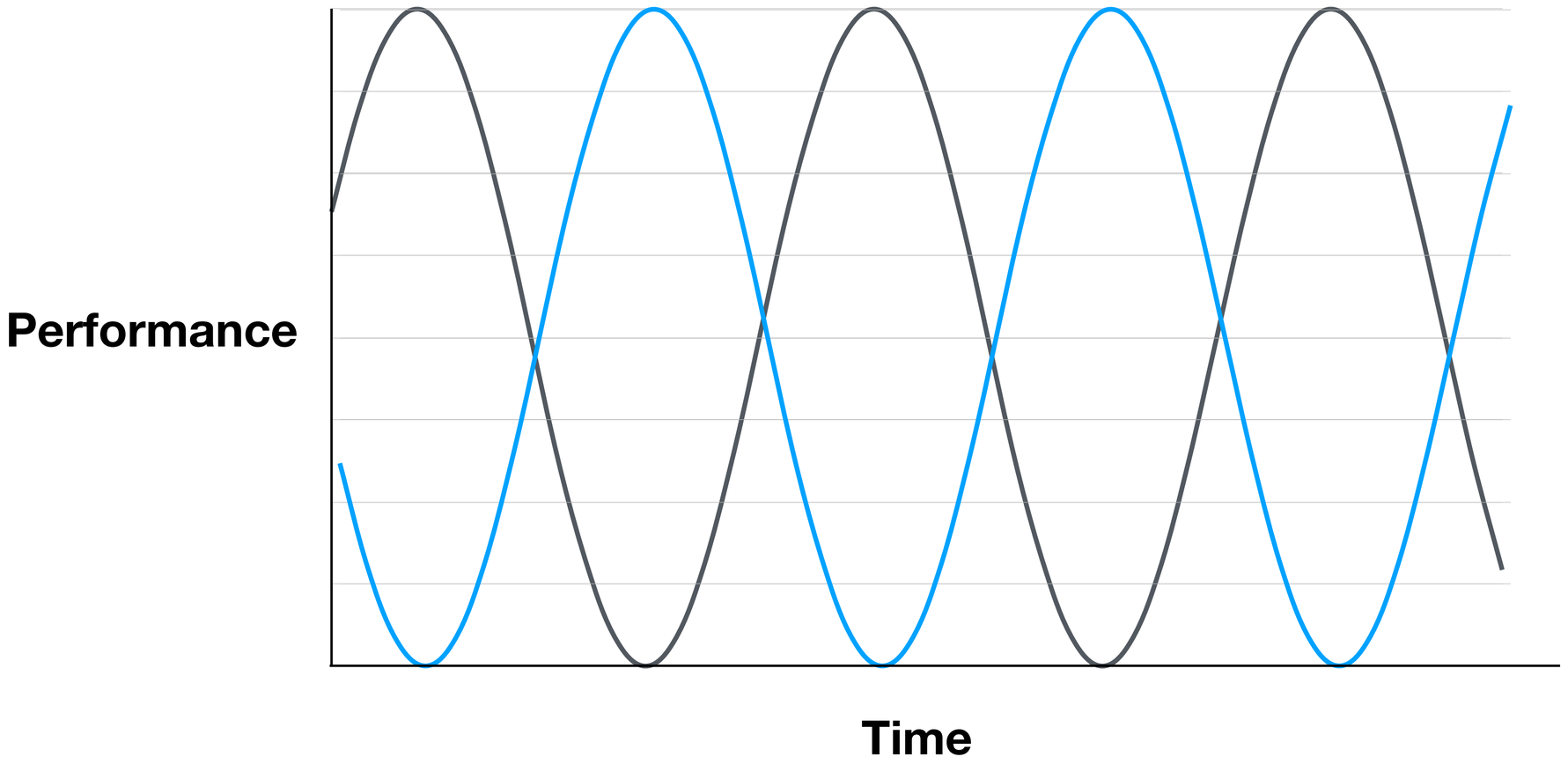
- ***Temporal***
- ***Spatial***
- ***Instantaneous***

Temporal (Diurnal) Placement Gaming

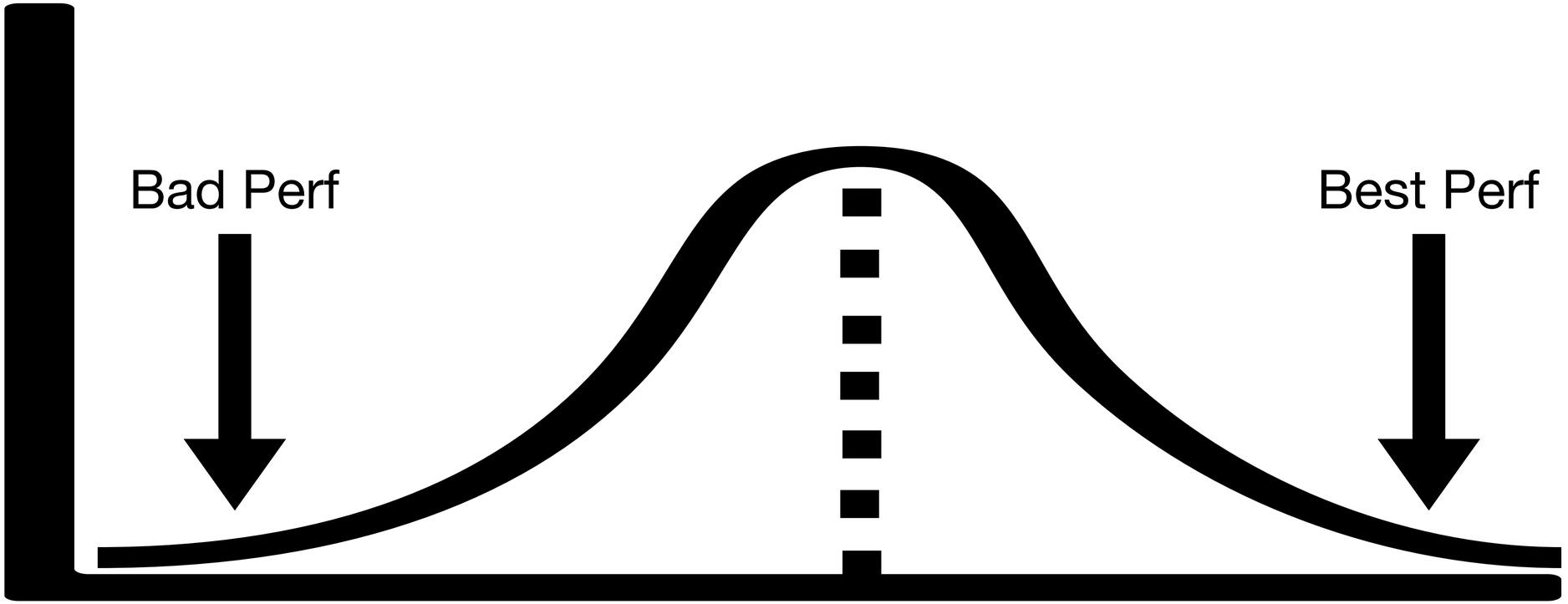


(Example - not real data)

Spatial Placement Gaming



Instantaneous Placement Gaming



Measurement Study

Benchmark Name	Measured Resources
Cache Benchmark (cache)	CPU, CPU Cache
FFmpeg Video Encoding (video)	CPU, CPU Cache, Disk IO
S3 File Download (net)	Network IO*
N-Queens (nqueens)	CPU

*For the net benchmark we control for S3 cache misses

Measurement Study

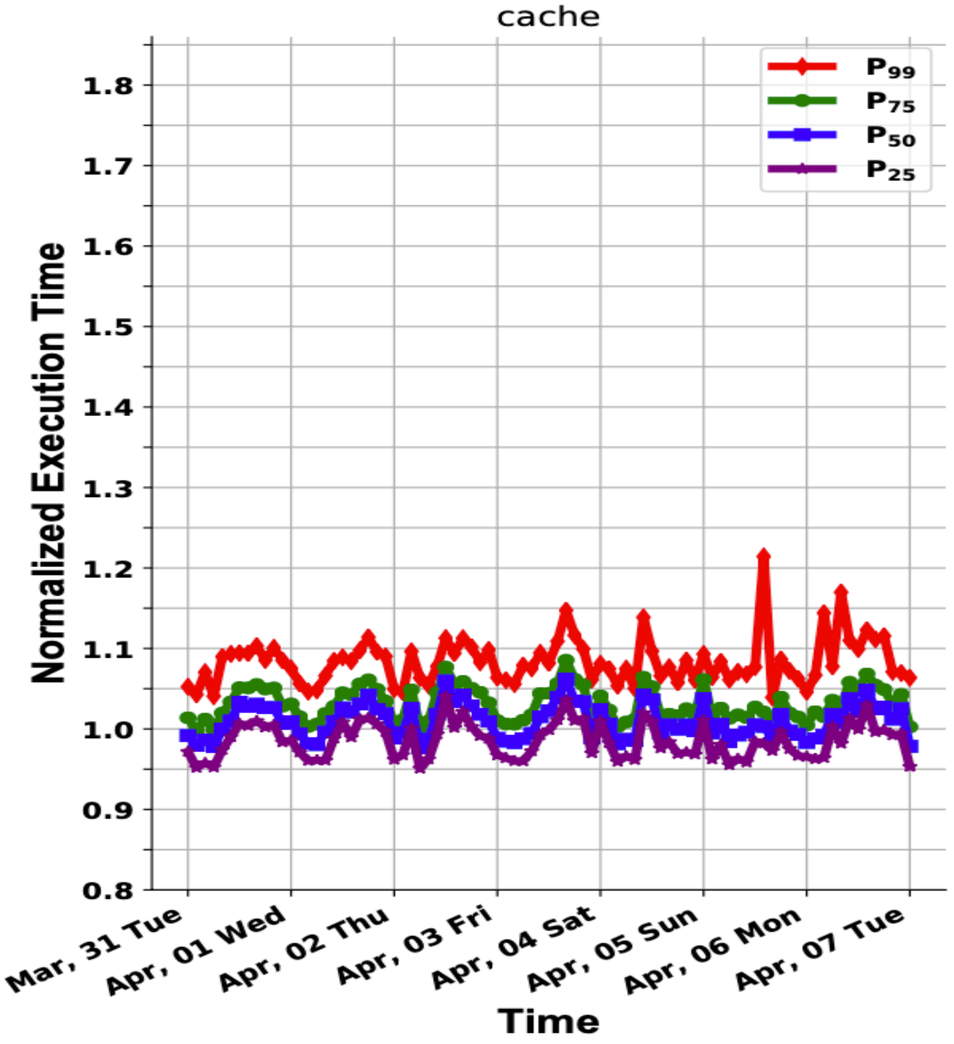
- ***Intra-Region Performance Variance***

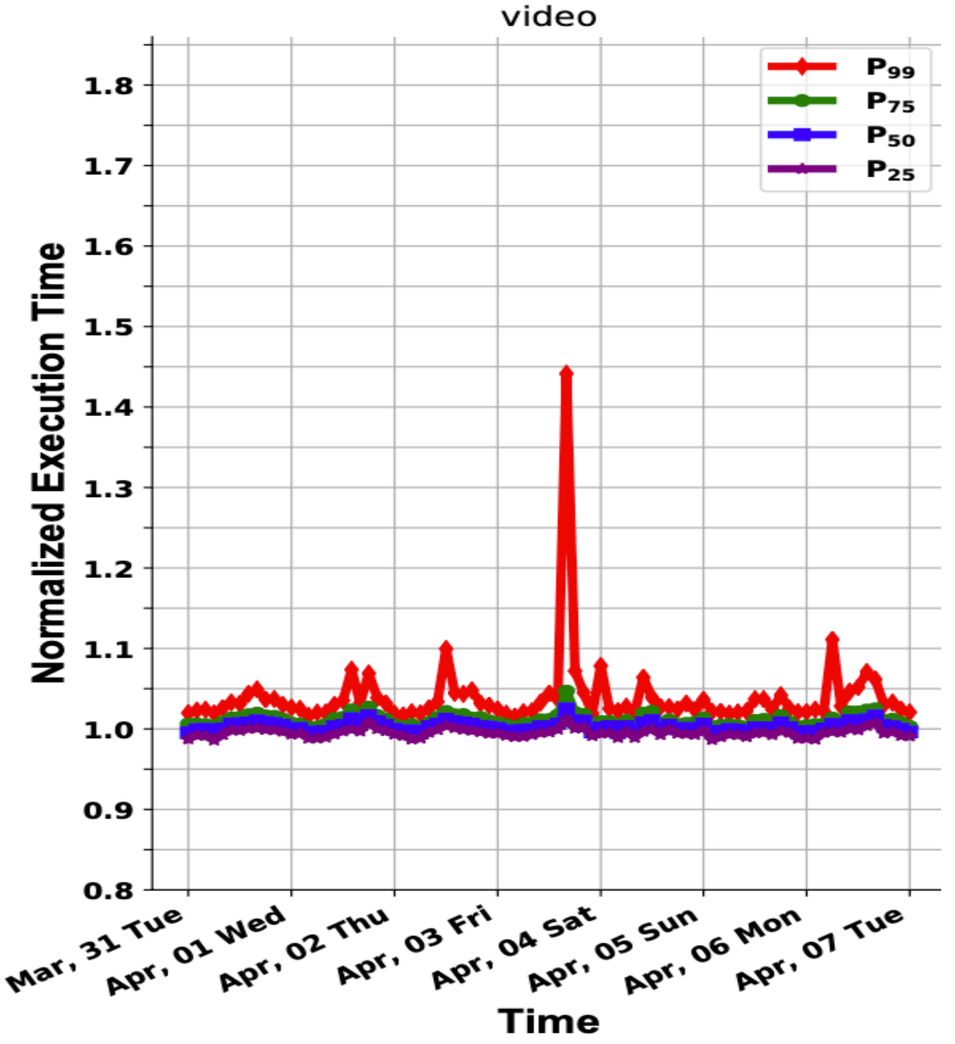
- Measuring within the *same* region
  - One week of data, sampling every 2 hours

- ***Inter-Region Performance Variance***

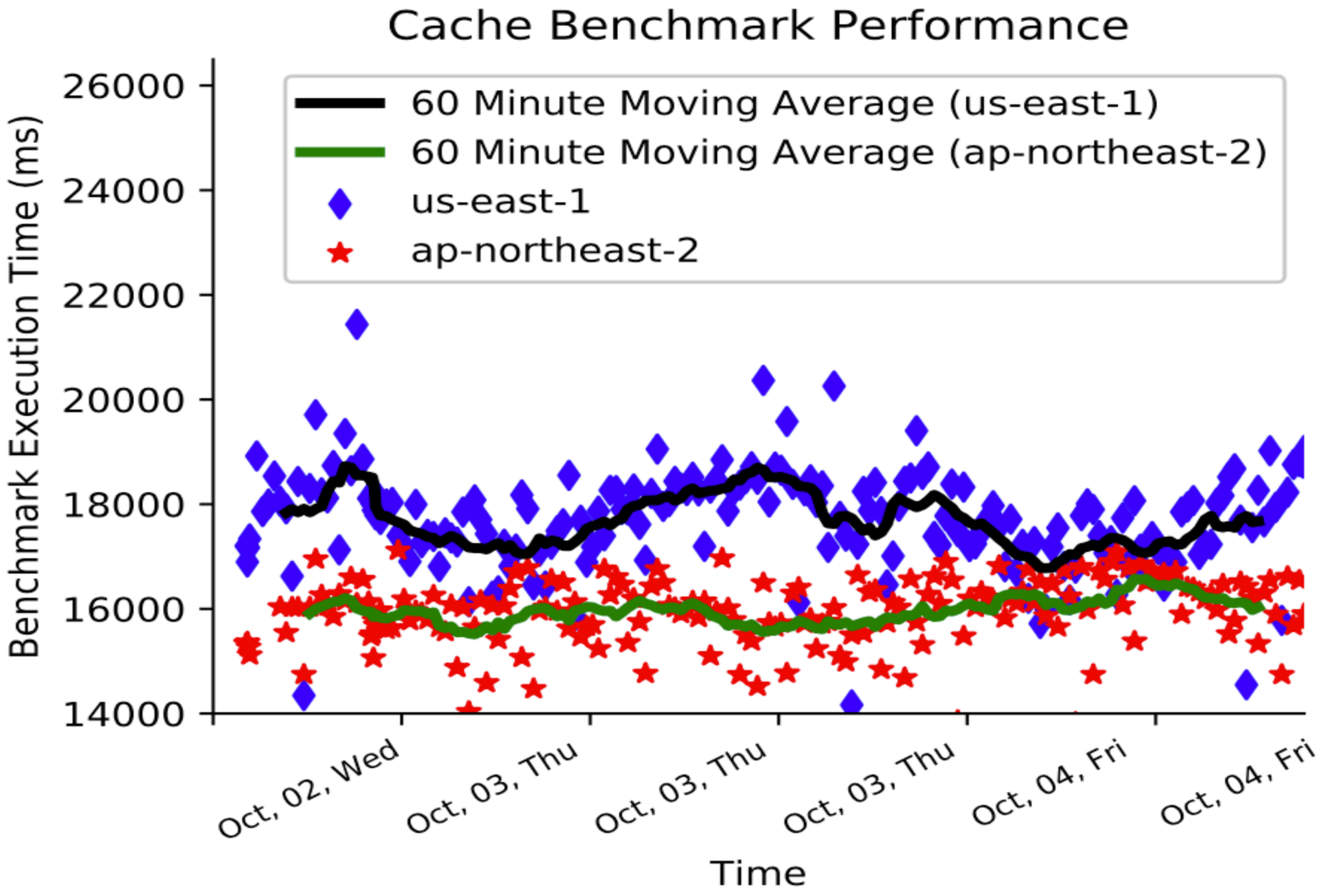
- Measuring across regions
  - 2 days of data, frequent sampling

Diurnal Patterns





Diurnal Patterns



Motivation

1. **Does performance variation exist in AWS Lambda?**
   1. **Is it possible to perform placement gaming?**

   Yes!

2. If so - is placement gaming on AWS Lambda worth it?

System Design & Implementation

Can we target ***any*** applications?

Applications that we know won't work:

- Function chaining
- Latency sensitive applications
- Network IO bound applications

Our ideal target:

- Batch workloads
  - Image/Video processing

Placement Gaming Strategies

Temporal + Spatial

- Limited by time & data sensitive workloads

Instantaneous Placement Gaming

- Our ideal target!

Two Strategies for Placement Gaming

Up Front Replacement

- Black-box
- Grey-box

Opportunistic Replacement

- Black-box only

Two Strategies for Placement Gaming

Up Front Replacement

- **Black-box**
- Grey-box

Opportunistic Replacement

- **Black-box only**

Two Strategies for Placement Gaming

Up Front Replacement

- Black-box
- **Grey-box**

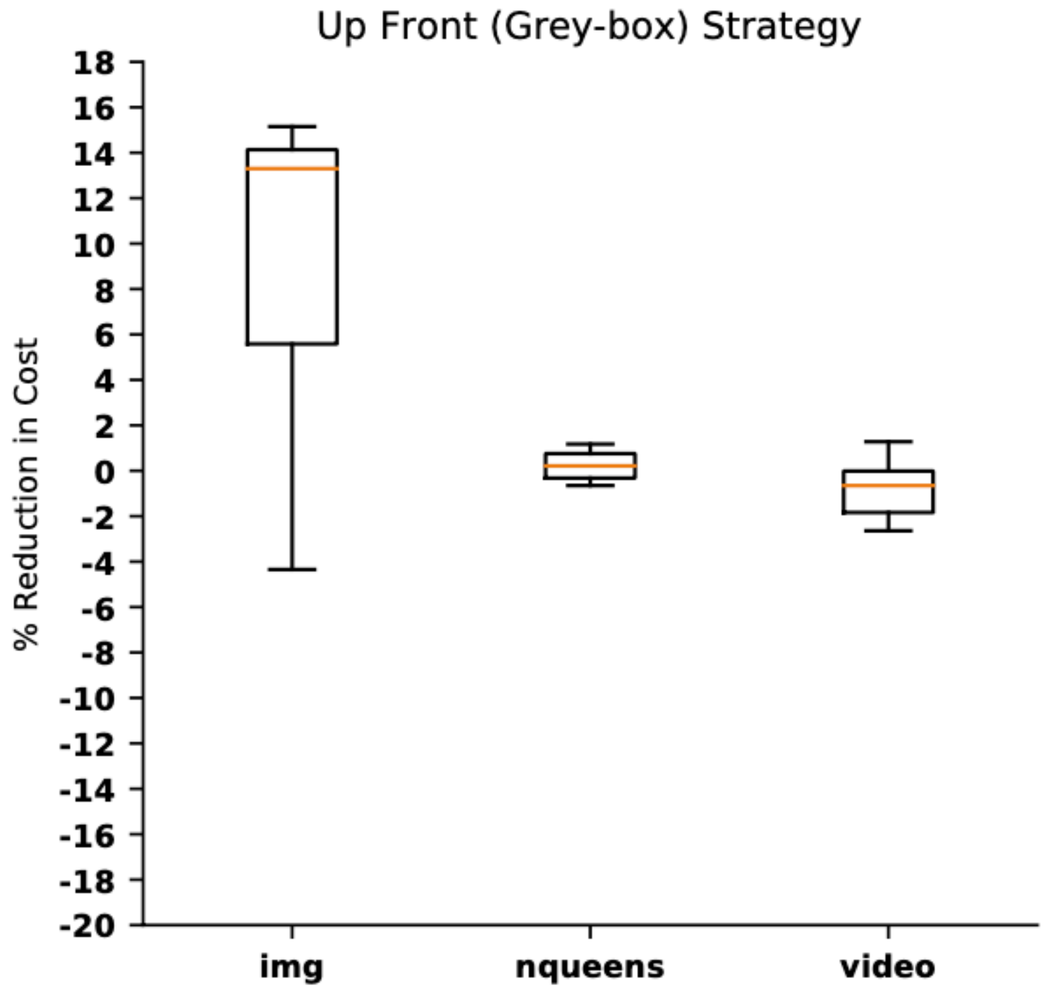
Opportunistic Replacement

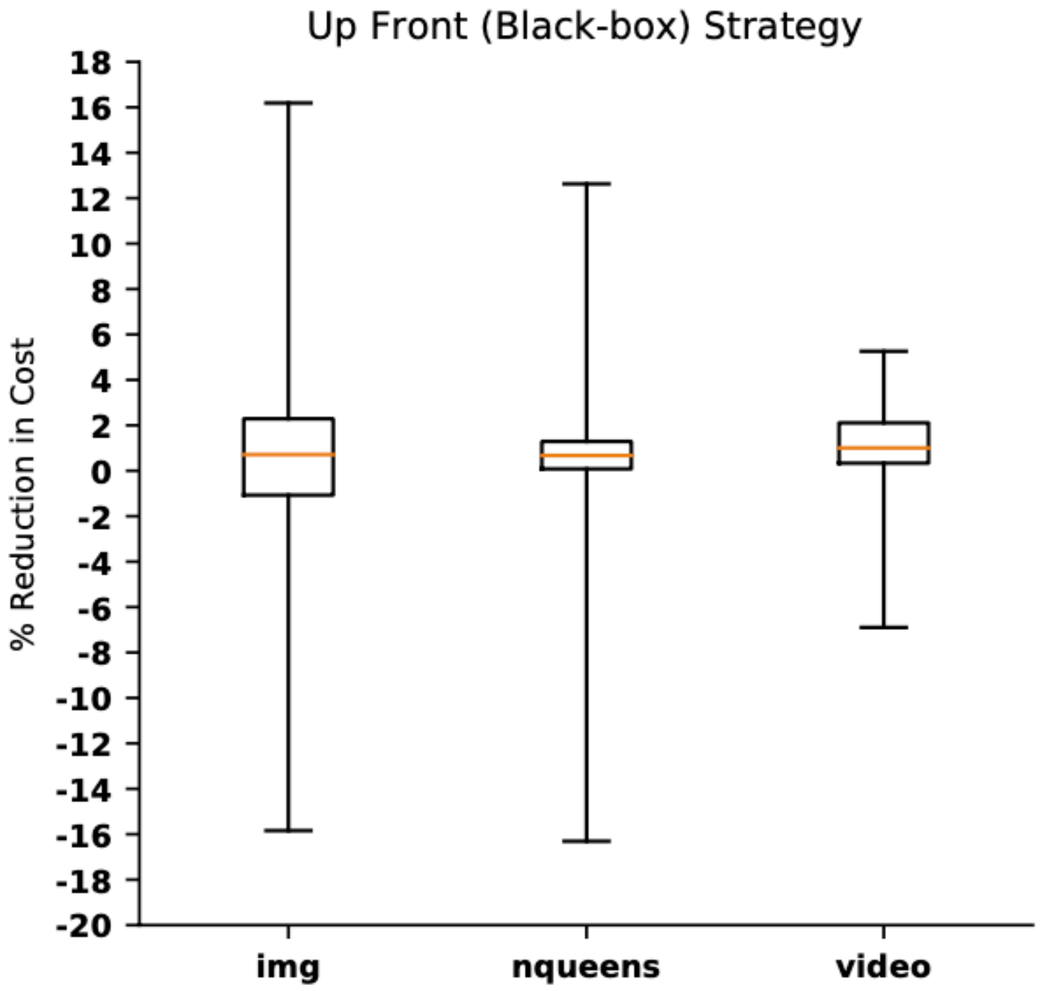
- Black-box only

Evaluation

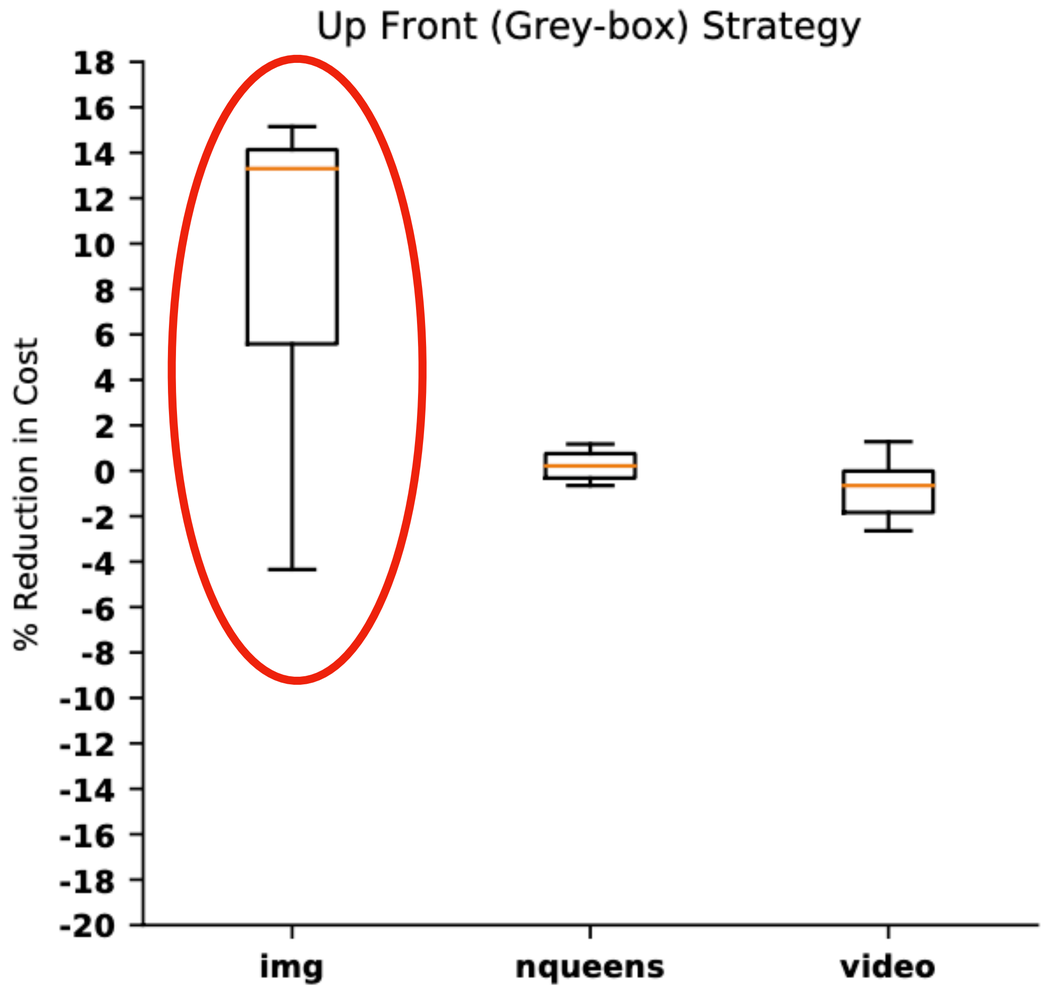
- Three Benchmarks
  - img benchmark (new)
  - nqueens (same from before)
  - video (same from before)

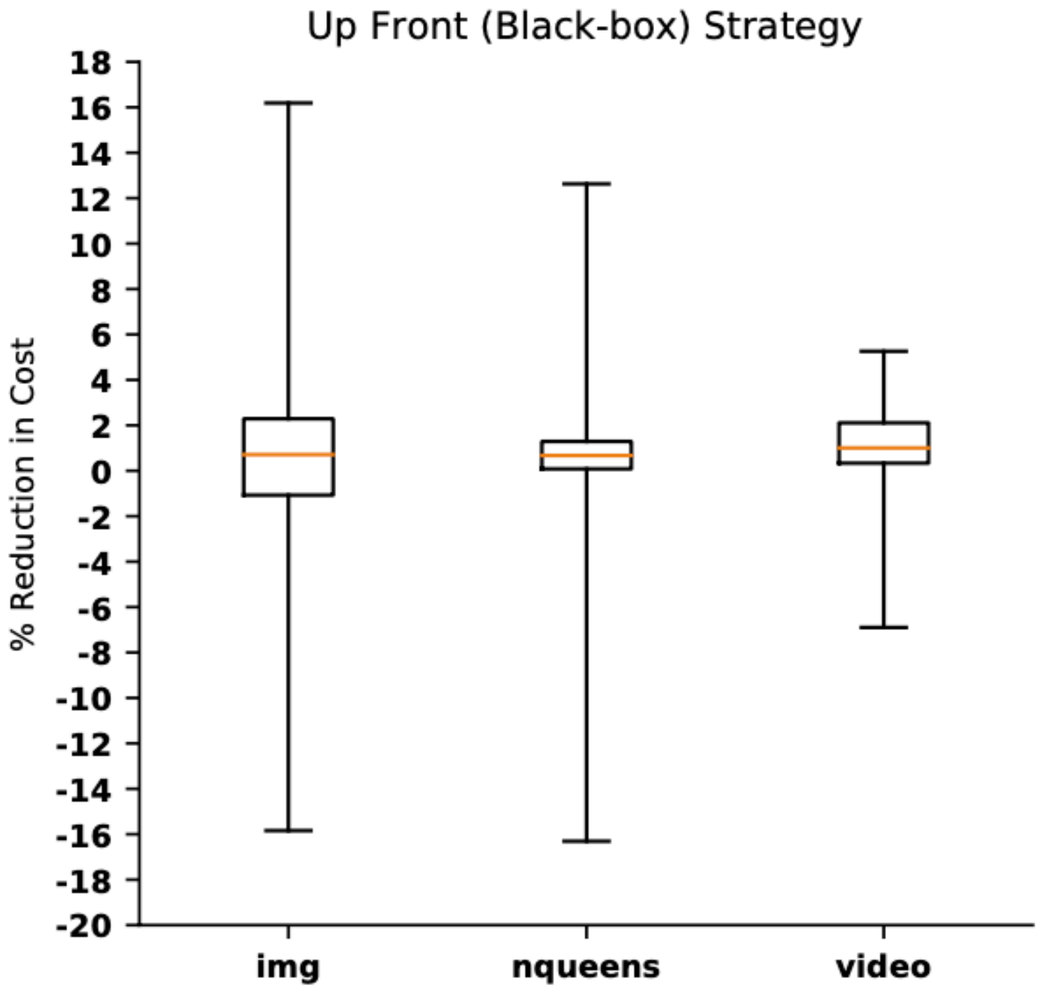
Evaluation (Up-Front)



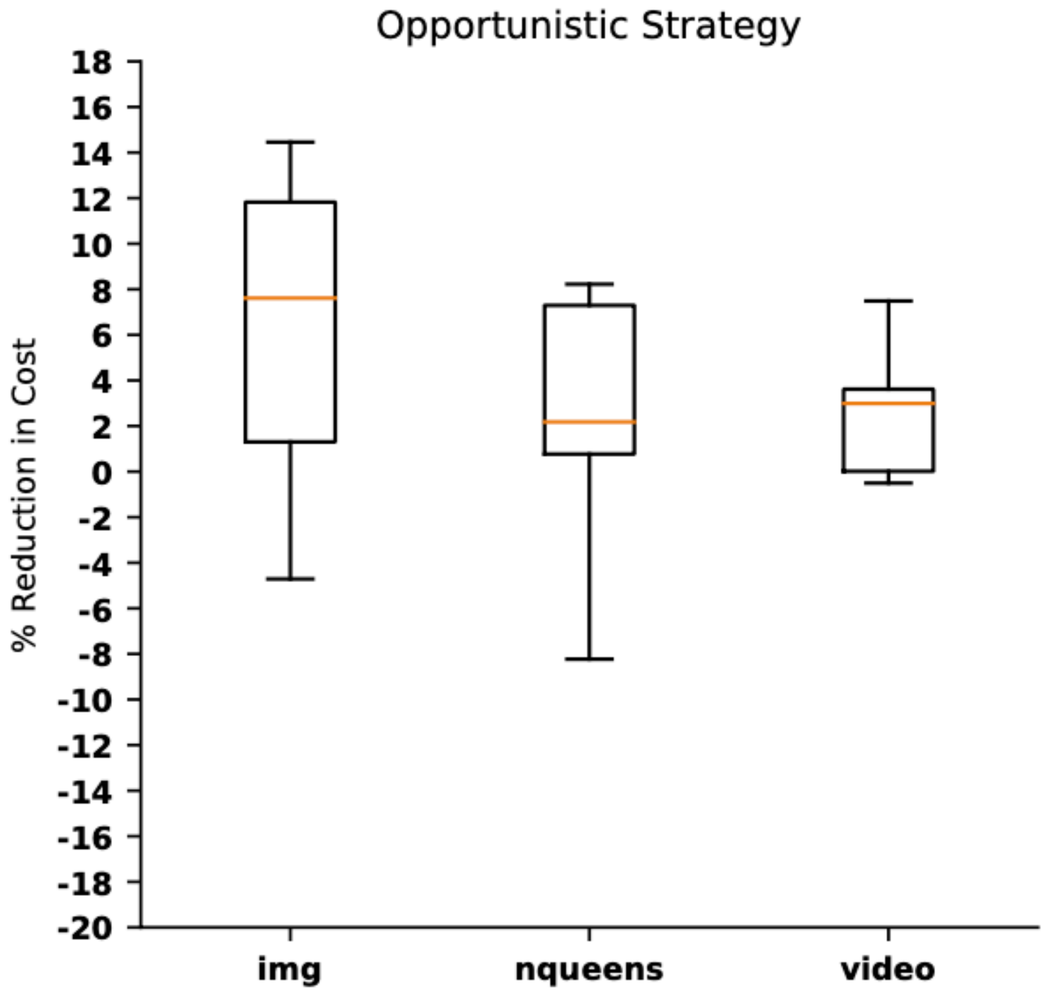


Evaluation (Up-Front)

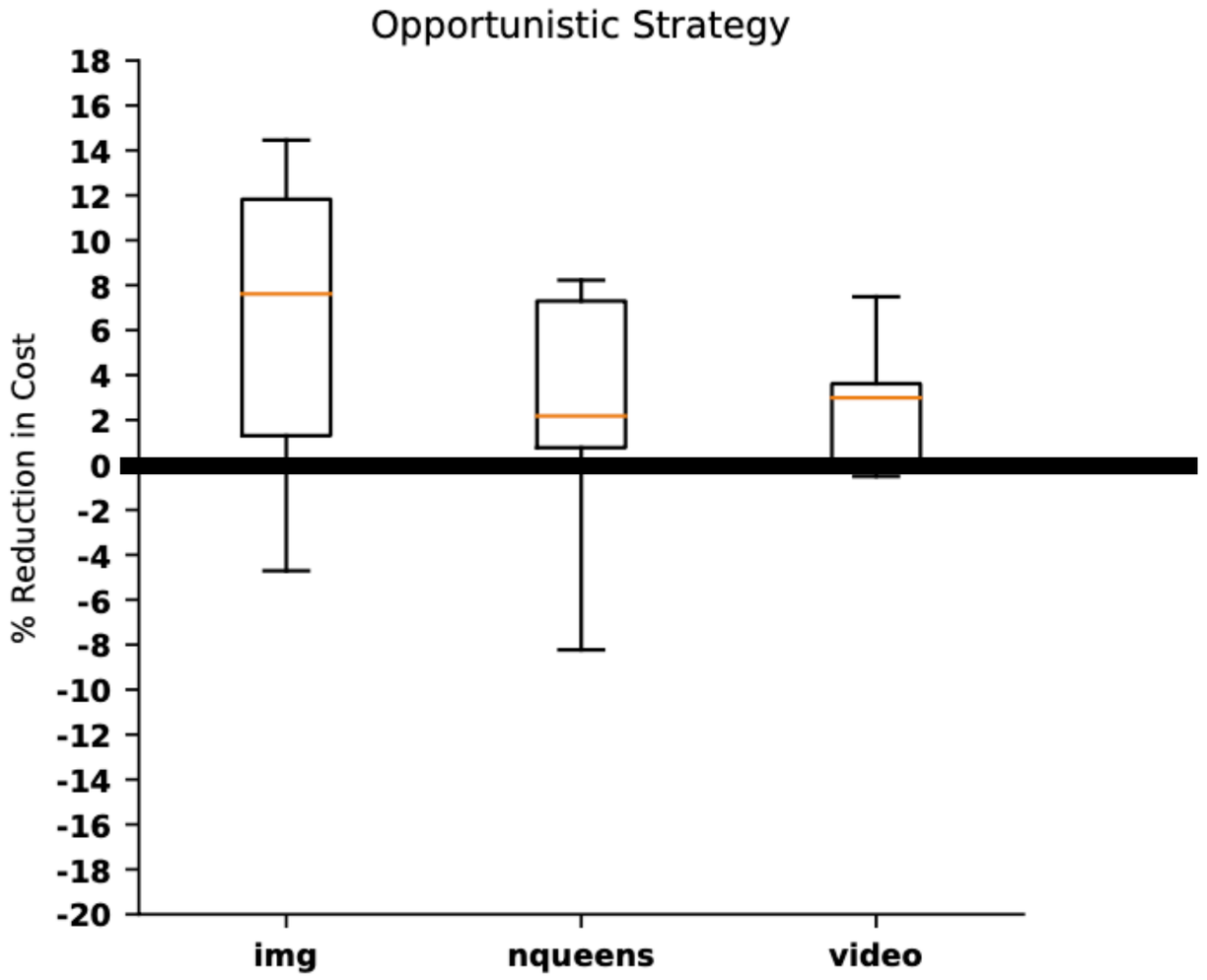




Evaluation (Opportunistic)



Evaluation (Opportunistic)



Conclusions

Placement gaming & exploitation of serverless is possible

What are the possible implications of this for serverless providers?