CS 744: PYWREN

Shivaram Venkataraman

Fall 2022

ADMINISTRIVIA

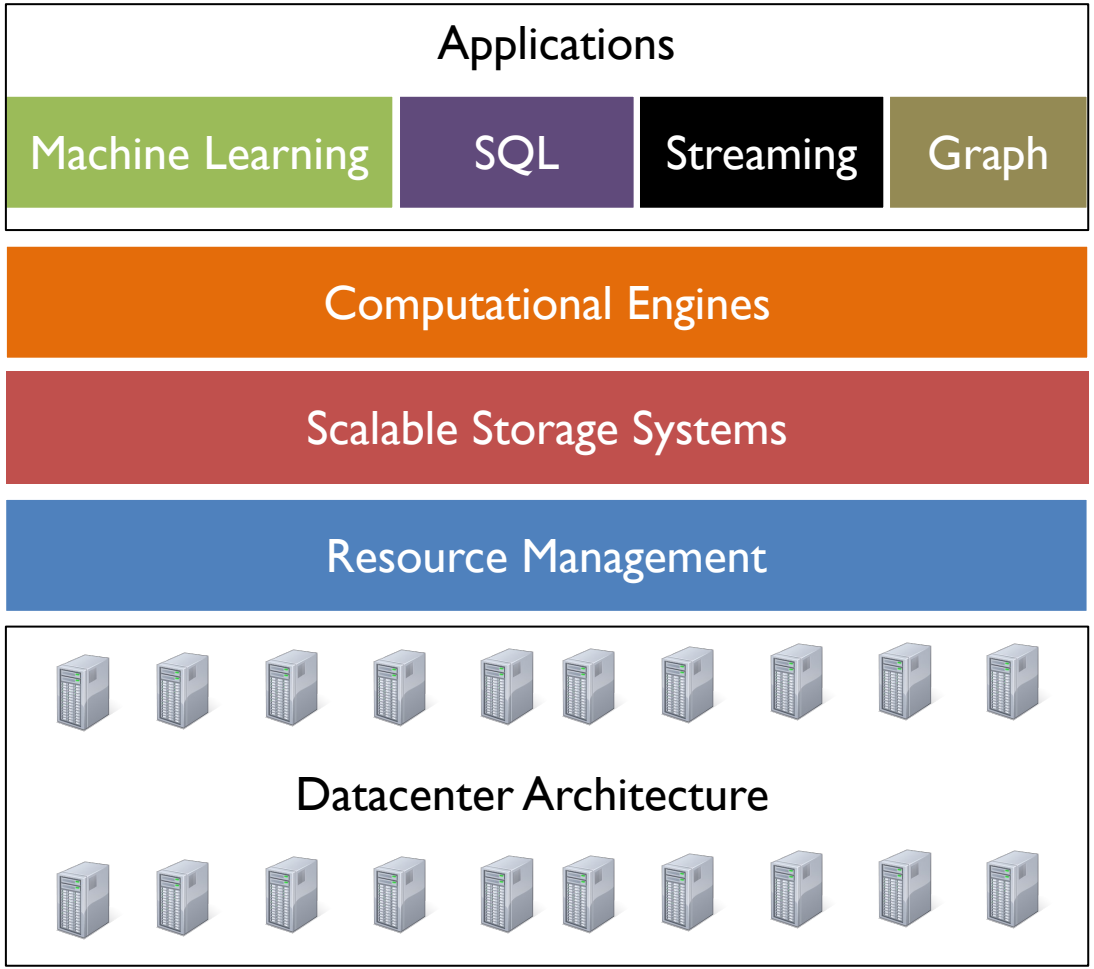
Project checkins due Nov 23rd

Poster presentation: Dec 13th

Final report: Dec 20th

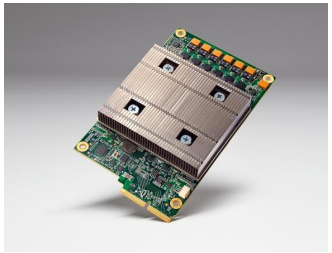
Project grade breakdown

- Intro: 5%
- Mid-semester checkin: 5%
- Poster: 10%
- Final Report: 10%


Applications
Machine Learning
SQL
Streaming
Graph
Computational Engines
Scalable Storage Systems
Resource Management
Datacenter Architecture

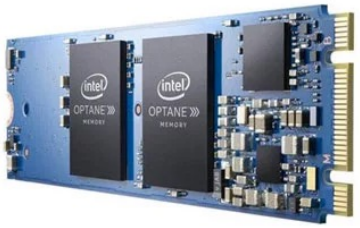
NEW DATA, HARDWARE MODELS

Serverless Computing



Compute Accelerators

Infiniband Networks



Non-Volatile Memory

SERVERLESS COMPUTING

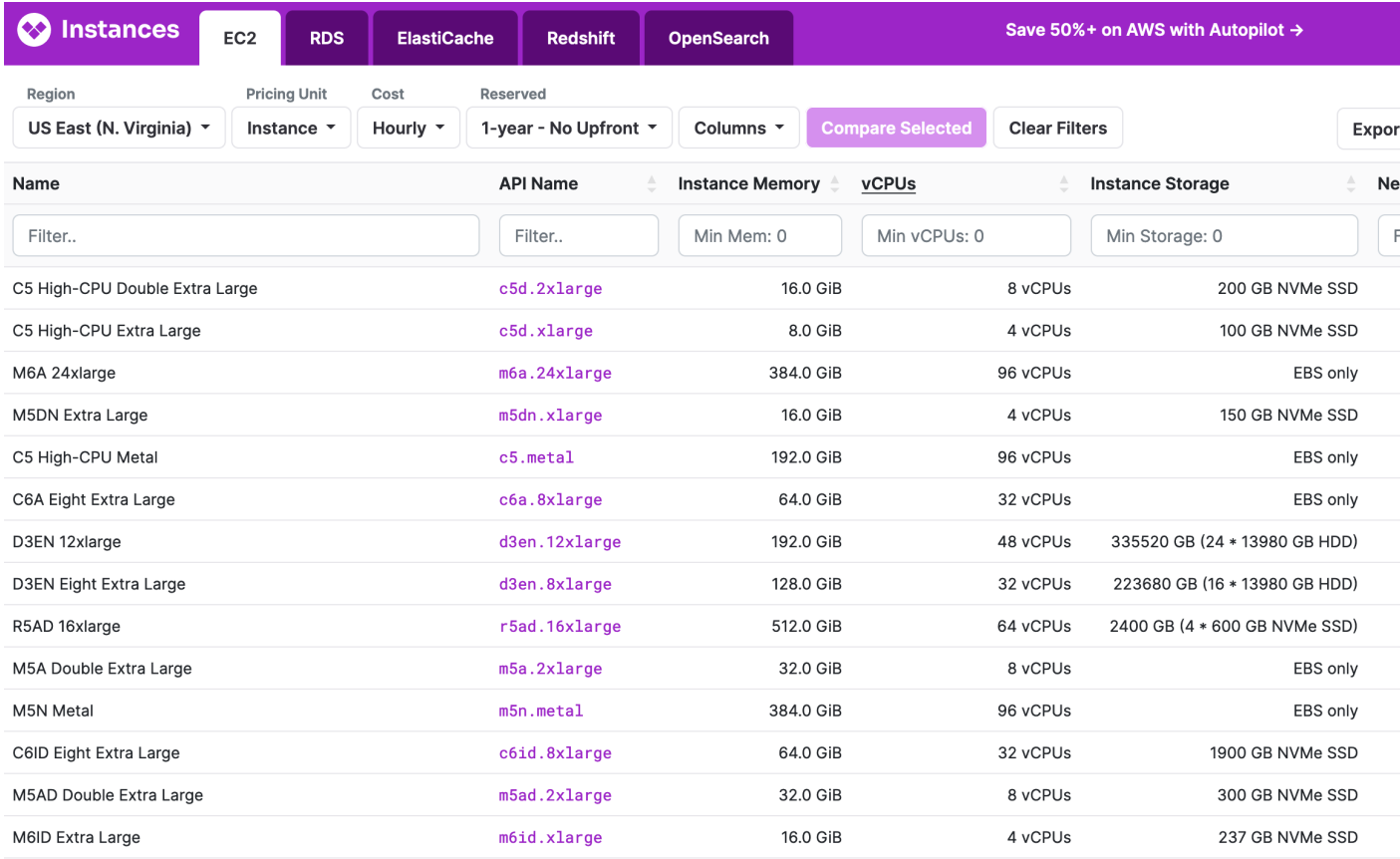
MOTIVATION: USABILITY

What instance type?

What base image?

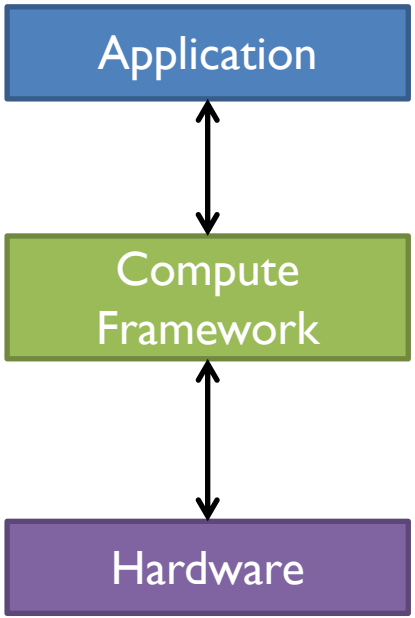
How many to spin up?

What price? Spot?




When to use the Cloud ?
Data
- Large amounts of data. Can't store locally
- Shared data across users
- Long term storage
Compute
- Need lots of CPUs for sho
- Varying comp
- No admin
Why is there no "cloud button"?

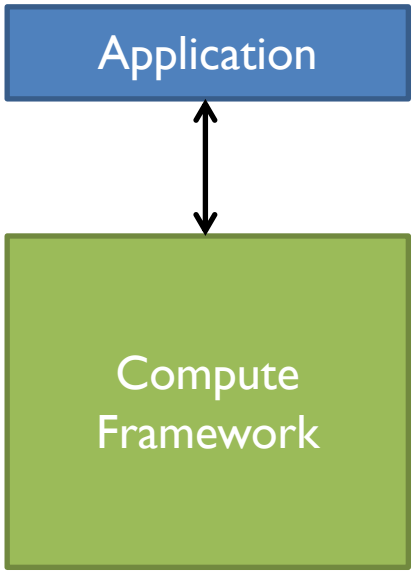
ABSTRACTION LEVEL ?



Logistic Regression

Spark

Amazon EC2
CloudLab
Private Cluster
…



“SERVERLESS” COMPUTING

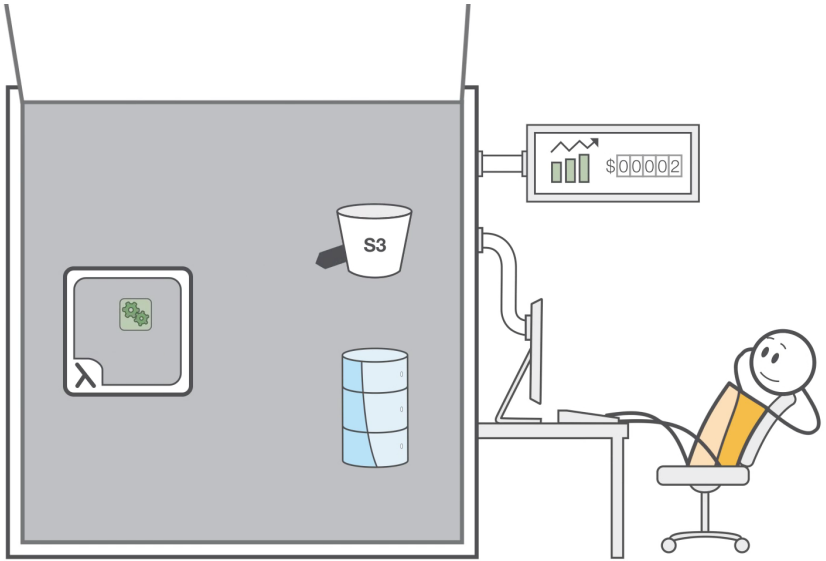
~~300~~ 900 seconds single-core

~~512~~ 512-10240 MB in /tmp

~~3~~ 10GB RAM

Python, Java, node.js, Ruby, Go etc.

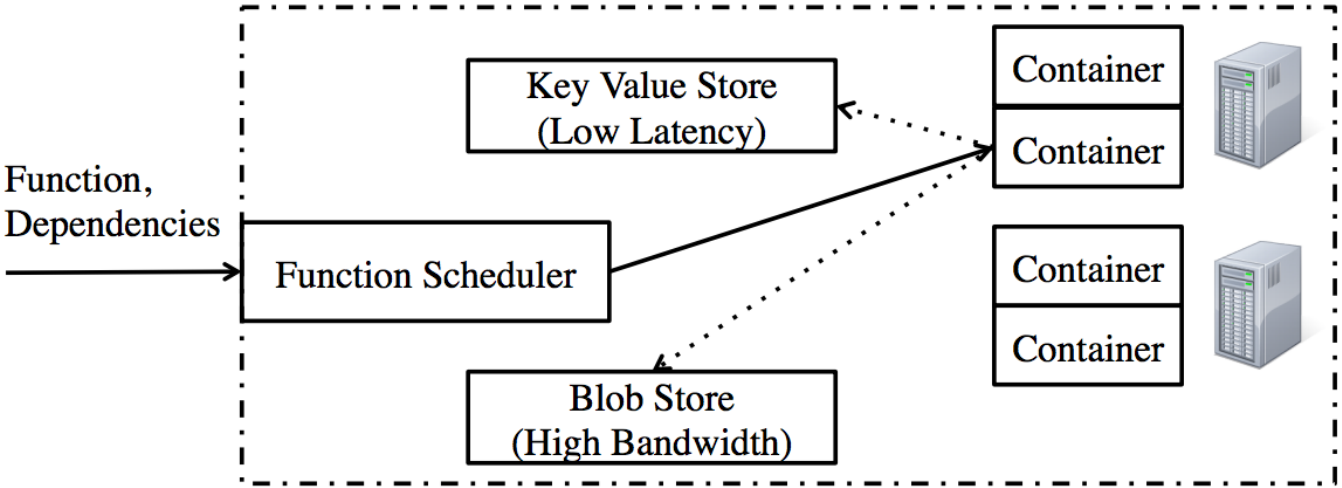
Support for containers



Google Cloud Platform
CLOUD FUNCTIONS ALPHA
A serverless platform for building event-based microservices

Microsoft Azure
Azure Functions
Process events with a serverless code architecture

STATELESS DATA PROCESSING



PYWREN API

```
import pywren
import numpy as np

def addone(x):
    return x + 1

wrenexec = pywren.default_executor()
xlist = np.arange(10)
futures = wrenexec.map(addone, xlist)

print [f.result() for f in futures]
```

The output is as expected:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

PYWREN: HOW IT WORKS

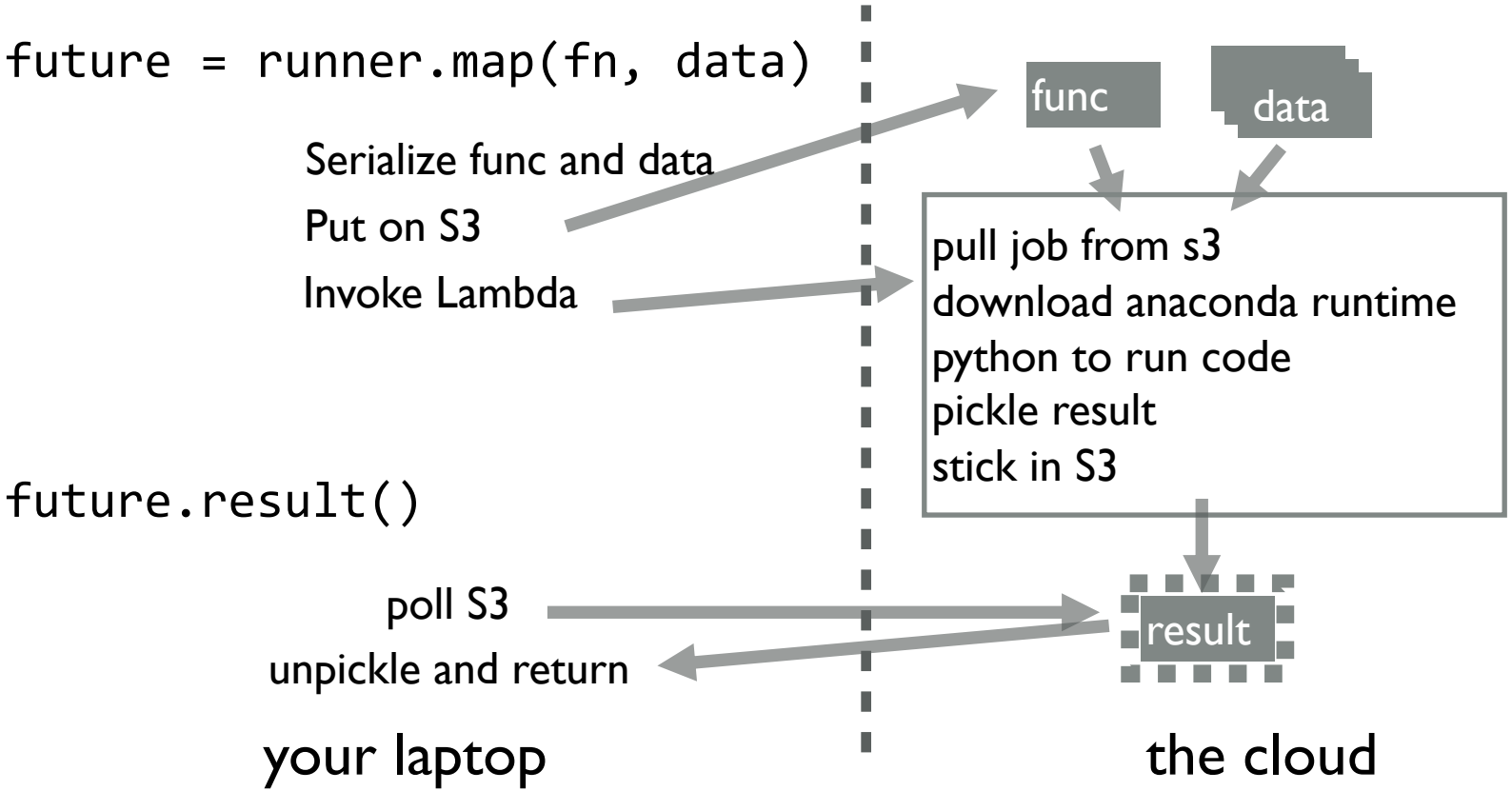
```
future = runner.map(fn, data)
```

```
future.result()
```

your laptop

the cloud

HOW IT WORKS

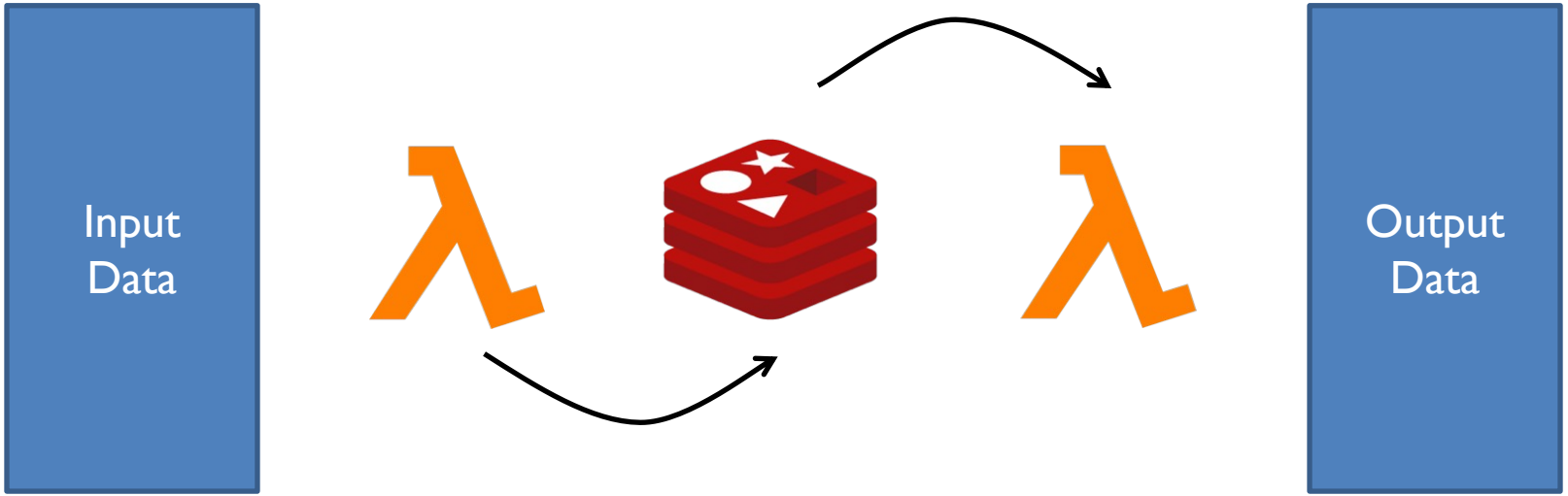

future = runner.map(fn, data)
Serialize func and data
Put on S3
Invoke Lambda
func
data
pull job from s3
download anaconda runtime
python to run code
pickle result
stick in S3
future.result()
poll S3
unpickle and return
result
your laptop
the cloud

STATELESS FUNCTIONS: WHY NOW ?

What are the trade-offs ?

Storage Medium	Write Speed (MB/s)
SSD on `c3.8xlarge`	208.73
SSD on `i2.8xlarge`	460.36
4 SSDs on `i2.8xlarge`	1768.04
S3	501.13

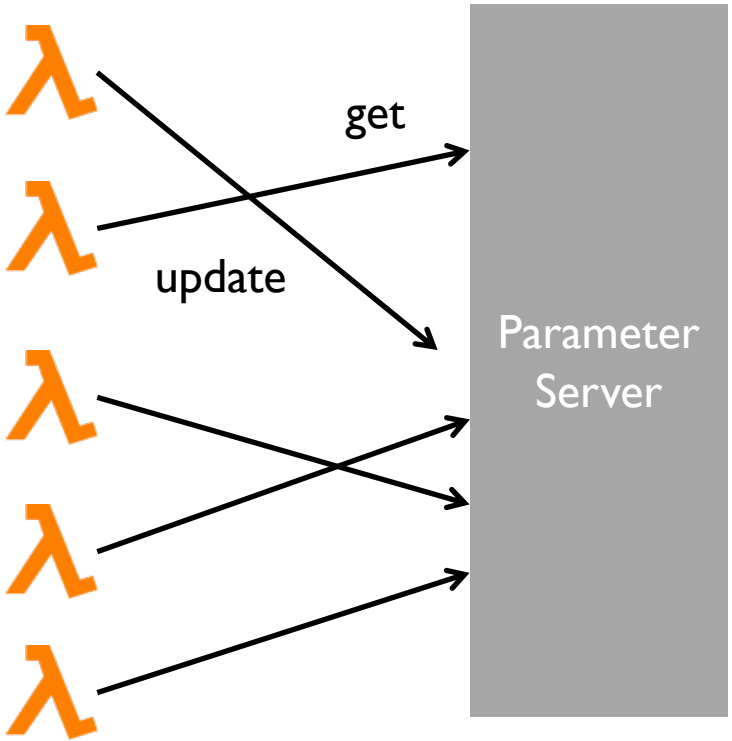
MAP AND REDUCE ?



PARAMETER SERVERS

Use lambdas to run “workers”

Parameter server as a service ?



WHEN SHOULD WE USE SERVERLESS ?

Yes!

Maybe not ?

SUMMARY

Motivation: Usability of big data analytics

Approach: Language-integrated cloud computing

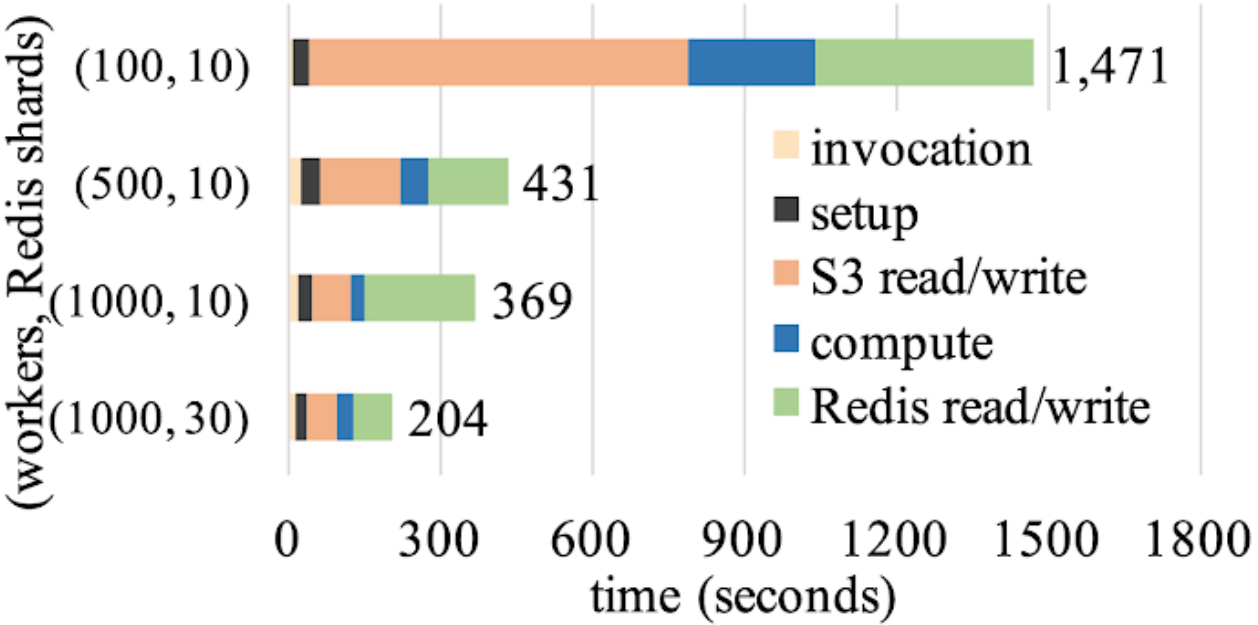
Features

- Breakdown computation into stateless functions
- Schedule on serverless containers
- Use external storage for state management

Open question on scheduling, overheads

DISCUSSION

https://forms.gle/GF1kkME52tvDRdDC8


(workers, Redis shards)
(100, 10)
(500, 10)
(1000, 10)
(1000, 30)
1,471
431
369
204
invocation
setup
S3 read/write
compute
Redis read/write
0
300
600
900
1200
1500
1800
time (seconds)

Consider you are a cloud provider (e.g., AWS) implementing support for serverless. What could be some of the new challenges in scheduling these workloads compared to schedulers we have studied in this class ? How would you go about addressing them?

NEXT UP

Happy Thanksgiving!

Next steps:

- Mid-semester project check-in, Nov 23rd
- After break: Owl, TPU papers
- Midterm 2, Dec 6th