



Enabling Serverless Sky Computing

Robert Cordingly, Wes Lloyd

School of Engineering and Technology
University of Washington Tacoma
11th IEEE International Conference on Cloud Engineering
IC2E 2023 - PhD Symposium

1

Outline

- Biography
 - Background and Motivation
 - Proposed Research
 - Preliminary Research and Results
 - Conclusions


2



Biography

- My name is Robert Cordingly
- Ph.D. student studying Distributed Systems and Cloud Computing in the School of Engineering and Technology at the University of Washington Tacoma
- This Ph.D. research is still in the early planning phases

Outline

- Biography
-  Background and Motivation
- Proposed Research
- Preliminary Research and Results
- Conclusions



What is Serverless Computing?



IBM Cloud Functions Apache OpenWhisk

Serverless function-as-a-service (FaaS) platforms offer many desirable features:

- Rapid elastic scaling
- Scale to zero
- No infrastructure management
- Fine grained billing
- Fault tolerance
- No up front cost to deploy an application

5

What is Sky Computing?



- The Sky sits above the clouds.
- Made up of compatibility layers providing interoperability between multiple cloud providers.

Goals of Sky Computing:

- Reduce vendor lock-in
- Enable applications to leverage resources of multiple cloud providers.

6

Sky Resource Aggregation

Resource Aggregation Benefits:

- Reduce Costs
- Improved Fault Tolerance
- Improved Availability
- Improved Runtime
- Reduce Network Latency
- Reduced Carbon Footprint
- Workload Consolidation
- Automatic Deployment and Management

7

History of Sky Computing

- First discussed in early 2010's as a means to reduce vendor lock-in on IaaS platforms
- More recently, sky-layers have been developed and investigated for specific use cases:
 - SkyPilot – Intercloud Broker for Large Language Model Training
 - SkyBridge – Data management system allowing multi-cloud data storage
- Goal: Investigate and enable Sky computing to deliver key enhancements for serverless computing

8



Challenges

- Vendor lock-in
- Cross cloud deployment and management
- Network latency
- Monitoring and observability
- Data availability
 - Move data to compute
 - Move compute to data

Outline

- Biography
- Background and Motivation
- ▶ Proposed Research
- Preliminary Research and Results
- Conclusions

Research Thrusts

- Thrust-1: FaaS Resource Aggregation Investigation
- Thrust-2: Sky-layer Prototyping and Trade-off Analysis
- Thrust-3: Autonomous Application Aggregation

11



Thrust 1 - FaaS Resource Aggregation Investigation



Google Cloud

- Investigate how serverless resources can be combined and aggregated across multiple cloud providers to achieve performance enhancements
 - Our prototype on AWS has shown this to be possible
- Expand scope beyond a single cloud provider (AWS) to other major platforms such as Google Cloud and Azure
- Aggregating resources between clouds introduces new challenges as platforms have varying pricing models, performance, APIs, available services, locations, etc

12

Thrust 1

FaaS Resource Aggregation Investigation

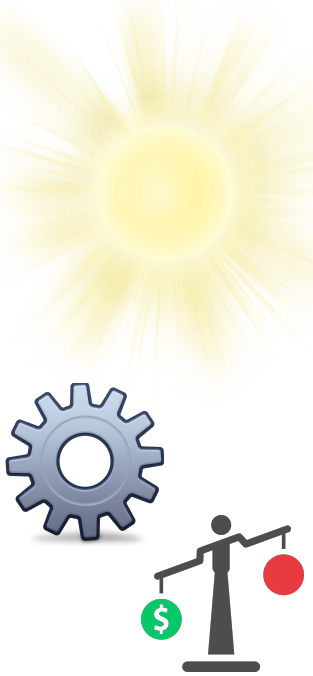
RQ-1: How can serverless resource aggregation optimize for performance objectives such as runtime, latency, throughput, carbon intensity, and cost while ensuring portability and observability?

Research Thrusts

- Thrust-1: FaaS Resource Aggregation Investigation
- Thrust-2: Sky-layer Prototyping and Trade-off Analysis
- Thrust-3: Autonomous Application Aggregation

Thrust 2 - Sky-layer Prototyping and Trade-off Analysis

- Prototype the sky-layer architecture to streamline application deployment, execution, and analysis
 - Our existing tools, such as FaaSSET, can be leveraged to build upon
- Investigate alternate architectures for key sky-layer components such as:
 - Deployment system
 - Load distribution system
 - Data management system
- Create platform neutral API abstractions for cloud services to enable platform cross-compatibility



15

Thrust 2

Sky-layer Development and Trade-off Analysis

RQ-2: How can platform-neutral abstractions be innovated to improve compatibility between platforms while proving feature parity on serverless cloud platforms?

RQ-3: What are the trade-offs of different resource aggregation and deployment strategies (e.g. multi-region, multi-cloud, multi-configuration deployment) utilized in the sky-layer?

16

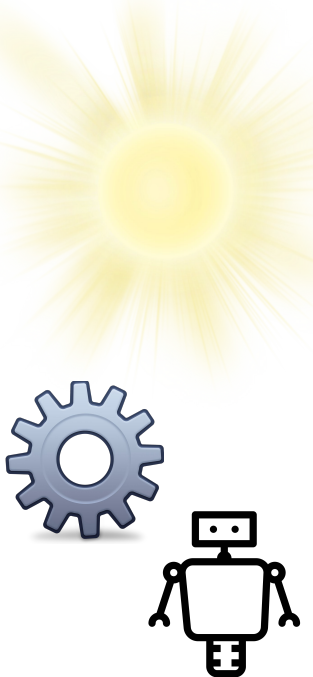
Research Thrusts

- Thrust-1: FaaS Resource Aggregation Investigation
- Thrust-2: Sky-layer Prototyping and Trade-off Analysis
- Thrust-3: Autonomous Application Aggregation

17

Thrust 3 - Autonomous Application Aggregation

- Investigate autonomous composition and aggregation of serverless resources to achieve performance enhancements
 - Our previous research touches on autonomous function configuration and modeling
- Investigate alternate modeling techniques and strategies:
 - Autonomous function deployment and relocation
 - Autonomous function configuration and dynamic reconfiguration



18

Thrust 3

Autonomous Application Aggregation

RQ-4: How can serverless aggregation strategies be dynamically learned and applied for specific goals (reduced cost, reduced latency, reduced carbon footprint) for different aggregations of serverless resources?

19

Outline

- Biography
- Background and Motivation
- Proposed Research
- Preliminary Research and Results
- Conclusions

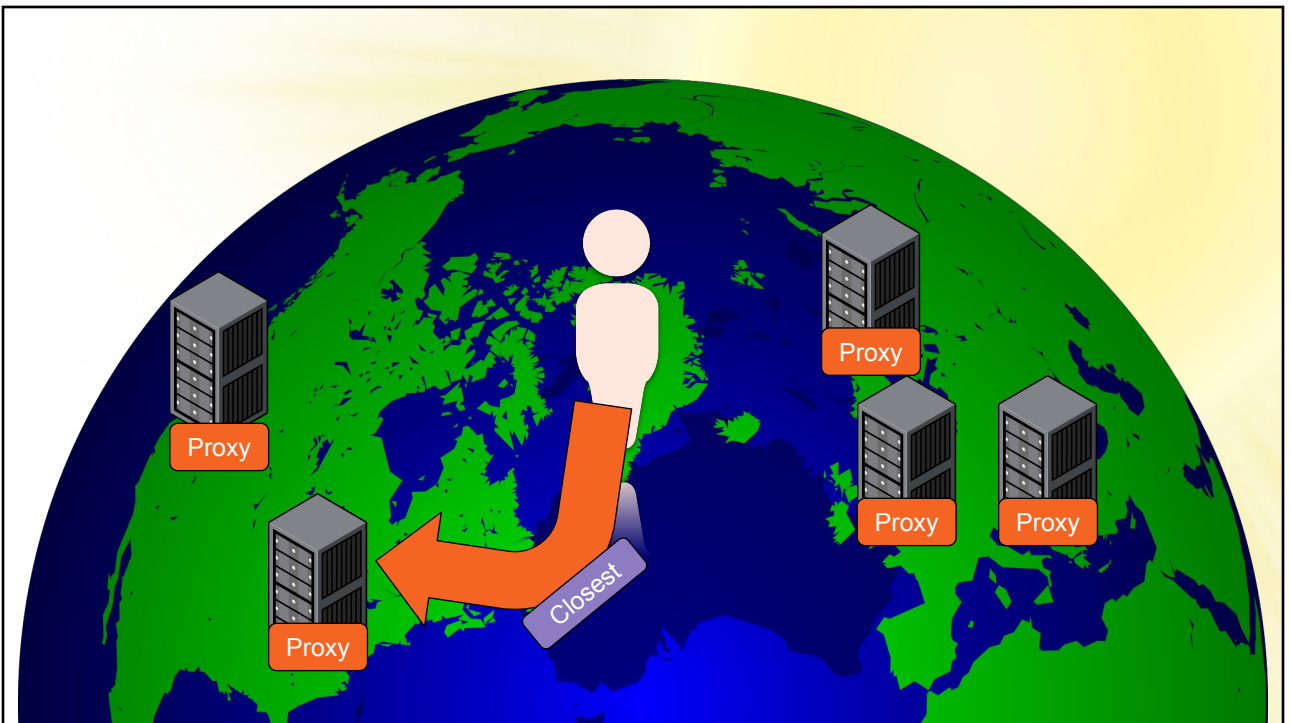
20

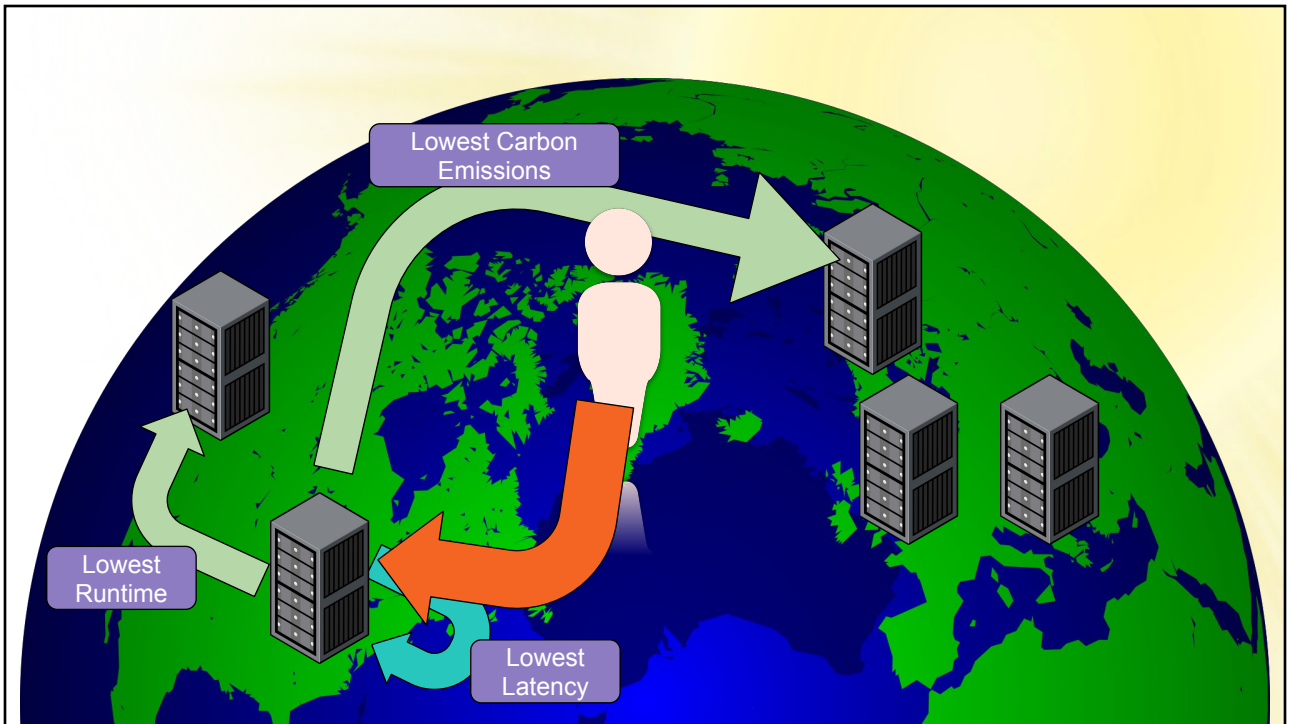
Towards Serverless Sky Computing

An Investigation on Global Workload Distribution to Mitigate Carbon Intensity, Network Latency, and Cost

Cordingly, R., Kaur, J., Dwivedi, D., Lloyd, W., Towards Serverless Sky Computing: An Investigation on Global Workload Distribution to Mitigate Carbon Intensity, Network Latency, and Cost, 2023 11th IEEE International Conference on Cloud Engineering (IC2E 2023), Sept 25-28, 2023

- Relates to Thrust-1, shows some of the benefits a Serverless Sky Computing platform could have.
- Shows that aggregation of resources can be used to reduce latency, costs, runtime, and carbon footprint.





Function Memory Optimization for Heterogeneous Serverless Platforms with CPU Time Accounting

Cordingly, R., Xu, S., & Lloyd, W. (2022, September). Function Memory Optimization for Heterogeneous Serverless Platforms with CPU Time Accounting. In 2022 IEEE International Conference on Cloud Engineering (IC2E) (pp. 104-115). IEEE.

- Developed the CPU-TAMS model to select optimal memory settings based off the number of vCPUs a function utilized.
- Relates to Thrust-3 where memory configuration will be needed for autonomous workload configuration

Function	Cheapest* Price Δ%	MIN Price Δ%	AWS-CO Price Δ%	CPU-TAMS Price Δ%	MID Price Δ%	MAX Price Δ%	Fastest* Price Δ%
Writer	-10	-7	-9	-2	160	410	160
Zip	-8	-5	-7	-6	150	406	232
Resize	-9	-7	-9	-7	142	406	142
DNA	-21	-15	-21	-9	165	406	0
PR	-6	-3	-6	11	144	368	325
MST	-3	4	-3	0	185	467	341
BFS	-7	-6	-5	-5	167	419	253
Sysbench	-8	-7	-8	-2	-5	0	0
Average	-9	-5.75	-8.5	-2.5	138.5	360.25	181.625

Function	Cheapest* Runtime Δ%	MIN Runtime Δ%	AWS-CO Runtime Δ%	CPU-TAMS Runtime Δ%	MID Runtime Δ%	MAX Runtime Δ%	Fastest* Runtime Δ%
Writer	23	367	50	13	-6	-4	-6
Zip	26	375	56	8	-4	-4	-6
Resize	172	1287	51	8	-7	-4	-7
DNA	50	384	50	19	7	7	0
PR	57	1355	57	-2	-11	-12	-13
MST	41	1247	41	0	-5	-7	-12
BFS	26	371	58	10	2	-2	-4
Sysbench	383	7296	711	6	92	0	0
Average	97.25	1585.25	134.25	7.75	8.5	-3.25	-6

Selection method average percent error compared to brute force discovered optimal memory setting.

AWS Lambda Autonomous Function Configuration using CPU-TAMS

27

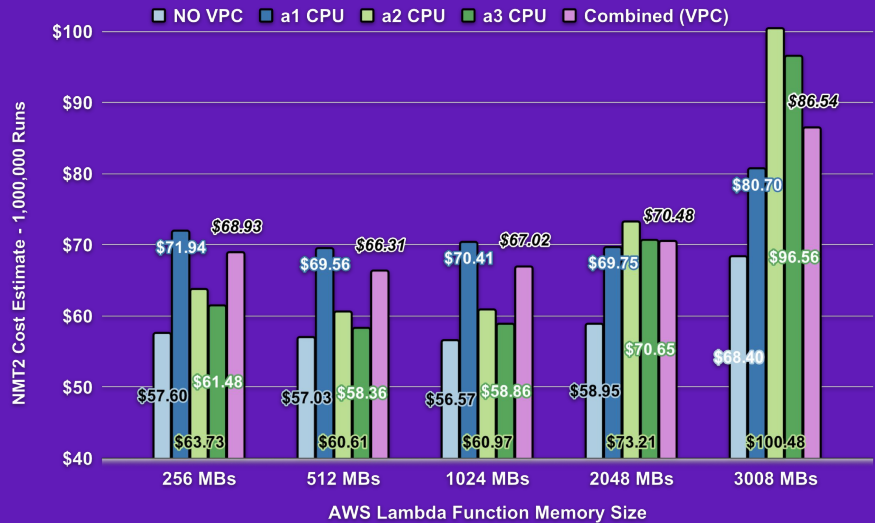
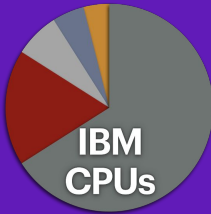
Predicting Performance and Cost of Serverless Computing Functions with SAAF

Cordingly, R., Shu, W. and Lloyd, W.J., 2020, August. Predicting performance and cost of serverless computing functions with SAAF. In 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCCom/CyberSciTech) (pp. 640-649). IEEE.

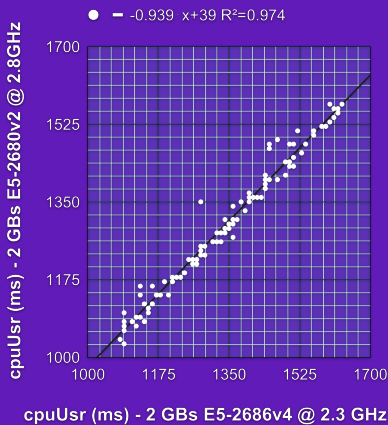
- Developed models for predicting the performance of one FaaS configuration based off another
- The goal was to improve pricing clarity and account for random hardware heterogeneity of FaaS platforms
- These modeling techniques can be expanded and applied to more platforms in Thrust-3

28

Research with SAAF: Predicting Performance



Predicting Performance Scenarios



CPU:

- 256 MBs a1 → a2
- 256 MBs a1 → a3
- 256 MBs a2 → a3
- 512 MBs a1 → a2
- 512 MBs a1 → a3
- 512 MBs a2 → a3
- 1024 MBs a1 → a2
- 1024 MBs a1 → a3
- 1024 MBs a2 → a3
- 2048 MBs a1 → a2
- 2048 MBs a1 → a3
- 2048 MBs a2 → a3

Memory:

- a1 256MBs → 512MBs
- a1 256MBs → 1024MBs
- a1 256MBs → 2048MBs
- a2 256MBs → 512MBs
- a2 256MBs → 1024MBs
- a2 256MBs → 2048MBs
- a3 256MBs → 512MBs
- a3 256MBs → 1024MBs
- a3 256MBs → 2048MBs

Platform:

- 256MBs a1 → i1
- 256MBs a1 → i2
- 256MBs a1 → i3
- 256MBs a1 → i4
- 1024MBs a1 → i1
- 1024MBs a1 → i2
- 1024MBs a1 → i3
- 1024MBs a1 → i4
- 2048MBs a1 → i1
- 2048MBs a1 → i2
- 2048MBs a1 → i3
- 2048MBs a1 → i4

Prediction Scenarios

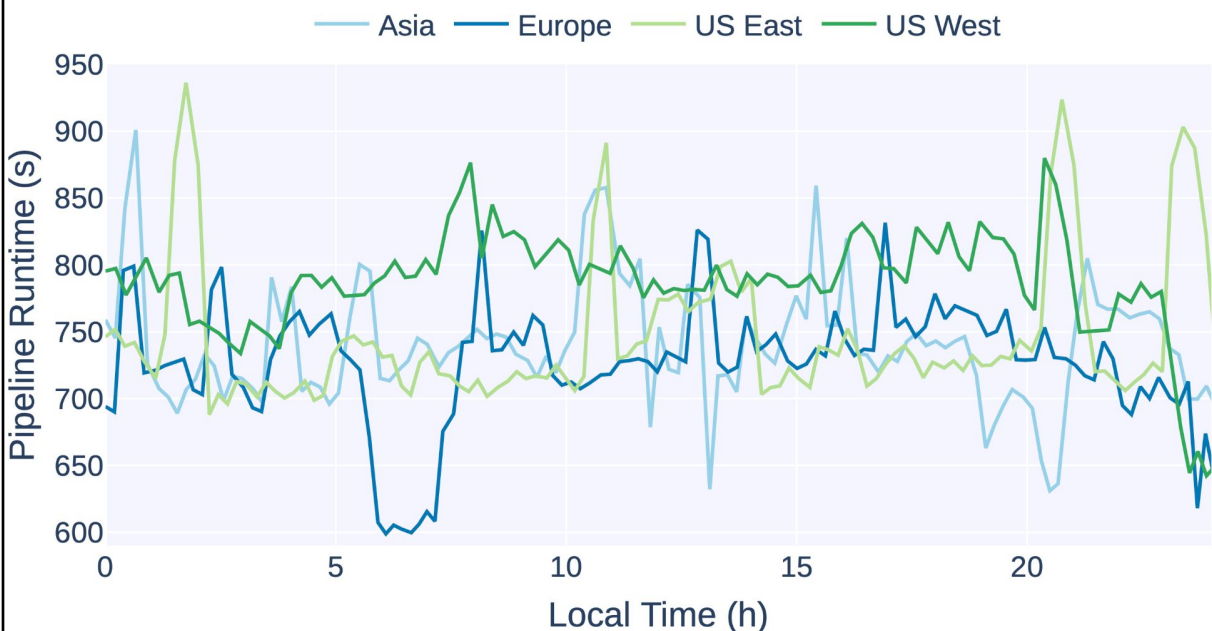
$$\text{Runtime} = \frac{(\text{cpuUsr} + \text{cpuKrn} + \text{cpuIdle} + \text{cpuOWait} + \text{cpuIntSrcv} + \text{cpuSftIntSrcv})}{(\# \text{ of cores})}$$

Characterizing X86 and ARM Serverless Performance Variation: A Natural Language Processing Case Study

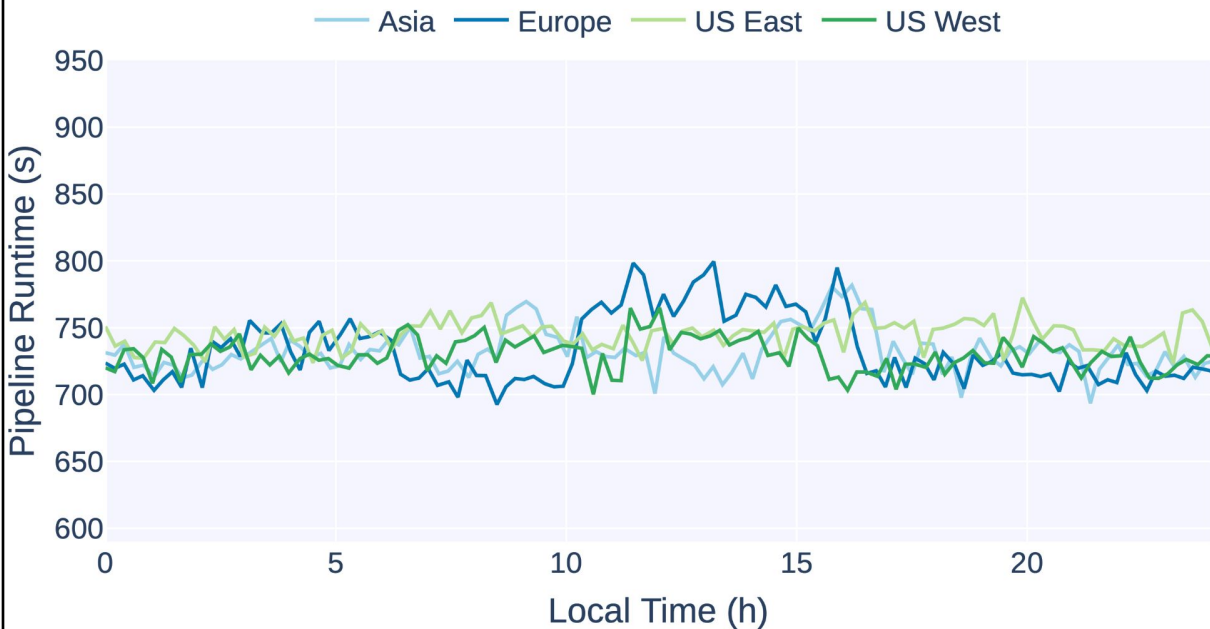
Lambion, D., Schmitz, R., Cordingly, R., Heydari, N. and Lloyd, W., 2022, July. Characterizing X86 and ARM Serverless Performance Variation: A Natural Language Processing Case Study. In Companion of the 2022 ACM/SPEC International Conference on Performance Engineering (pp. 69-75).

- Evaluated the performance of X86 and ARM processors on AWS Lambda through a NLP pipeline
- In the future, an autonomous sky-layer can leverage resources across multiple architectures to achieve cost and performance improvements

X86 AWS Lambda Runtime Variation



ARM AWS Lambda Runtime Variation



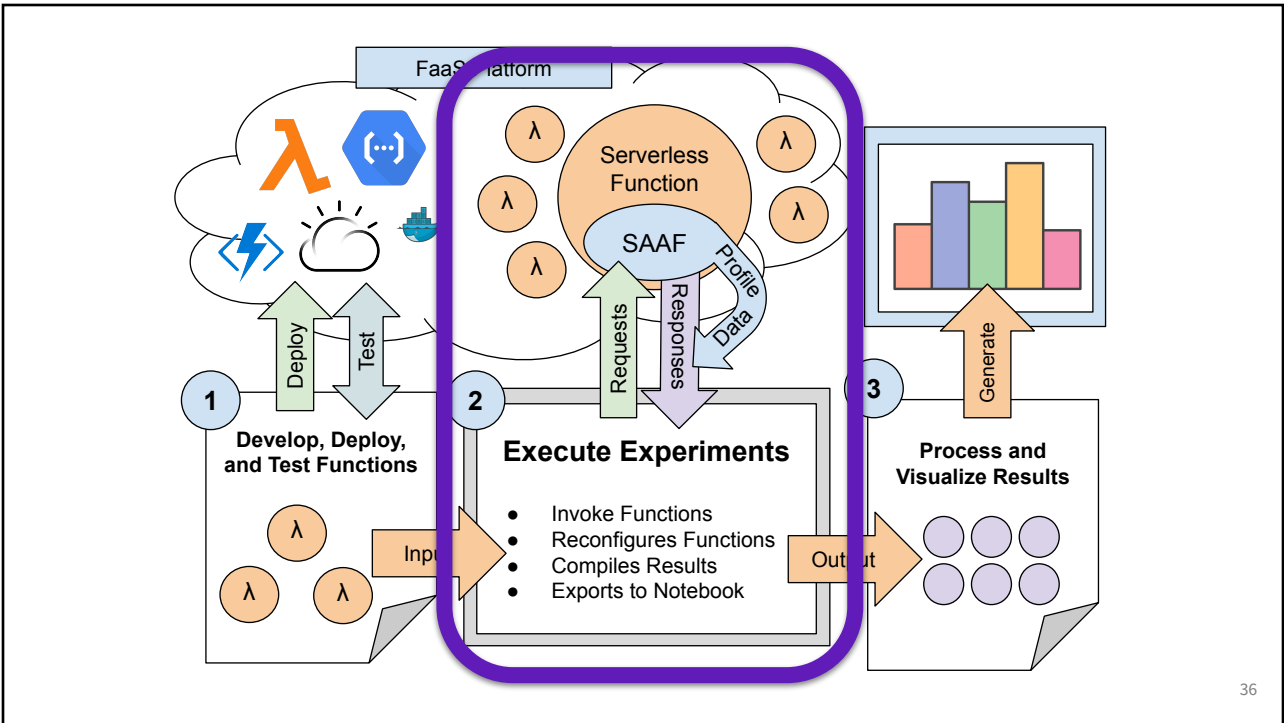
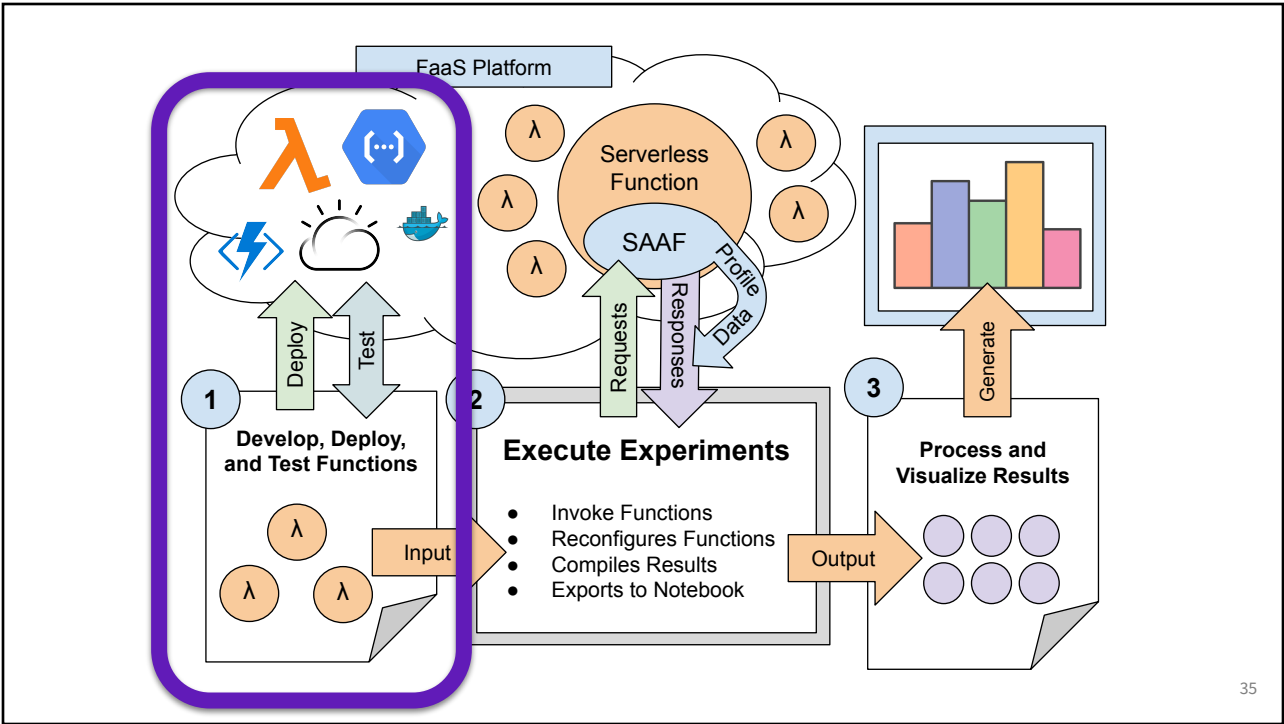
33

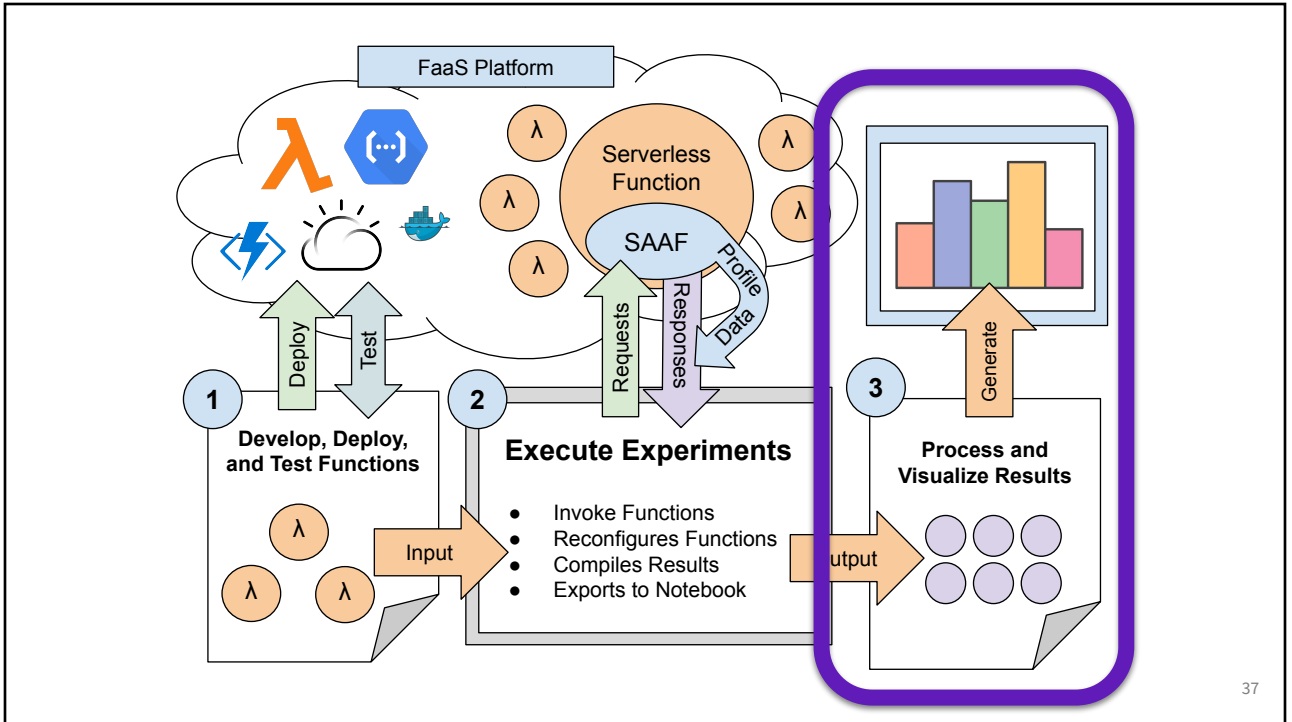
FaaSSET: A Jupyter Notebook to Streamline Every Facet of Serverless Development

Cordingly, R., & Lloyd, W. (2022, July). FaaSSET: A Jupyter notebook to streamline every facet of serverless development. In Companion of the 2022 ACM/SPEC International Conference on Performance Engineering (pp. 49-52).

- Developed an environment for developing and deploying platform neutral serverless functions
- FaaSSET supports creating and deploying functions to AWS, Google Cloud, IBM, and Azure
- The sky-layer in Thrust-2 could be built upon FaaSSET's tools

34





Function Development and Deployment with FaaSSET



```
def hello_world(request, context):
    return {"message": "Hello " + str(request["name"]) + "!"}
```

Function Development and Deployment with FaaS



```
def hello_world(request, context):  
    return {"message": "Hello " + str(request["name"]) + "!"}
```



```
import FaaS  
@FaaS.cloud_function(platform="AWS", config={"memory":256})  
def hello_world(request, context):  
    return {"message": "Hello " + str(request["name"]) + "!"}
```

39

Function Development and Deployment with FaaS



```
def hello_world(request, context):  
    return {"message": "Hello " + str(request["name"]) + "!"}
```



```
import FaaS  
@FaaS.cloud_function(platform="AWS", config={"memory":256})  
def hello_world(request, context):  
    return {"message": "Hello " + str(request["name"]) + "!"}
```

40

Function Development and Deployment with FaaSSET



```
def hello_world(request, context):  
    return {"message": "Hello " + str(request["name"]) + "!"}
```

```
import FaaSSET  
@FaaSSET.cloud_function(platform="AWS", config={"memory":256})  
def hello_world(request, context):  
    return {"message": "Hello " + str(request["name"]) + "!"}
```



```
hello_world({'name': 'Bob'}, None)
```

```
>> Deploying to AWS Lambda...
```

```
>> {"message": "Hello Bob!"}
```

41

Outline

- Biography
- Background and Motivation
- Proposed Research
- Preliminary Research and Results



Conclusions

42



Expected Contributions

- Create and evaluate a serverless Sky-layer
 - Autonomous application deployment system
 - Serverless load distribution system
 - Serverless data management system
 - Investigate resource aggregation methodologies to achieve improved performance and costs of applications on serverless platforms
-

43

Sky Resource Aggregation



Resource Aggregation Benefits:

- Reduced Carbon Footprint
- Improved Fault Tolerance
- Improved Availability
- Improved Runtime
- Reduce Network Latency
- Reduce Costs
- Workload Consolidation
- Automatic Deployment and Management

44

Thank You!

This research has been supported by AWS Cloud Credits for Research.
