Going Serverless:

Evaluating the Potential of Serverless Computing for Environmental Modelling Application Hosting

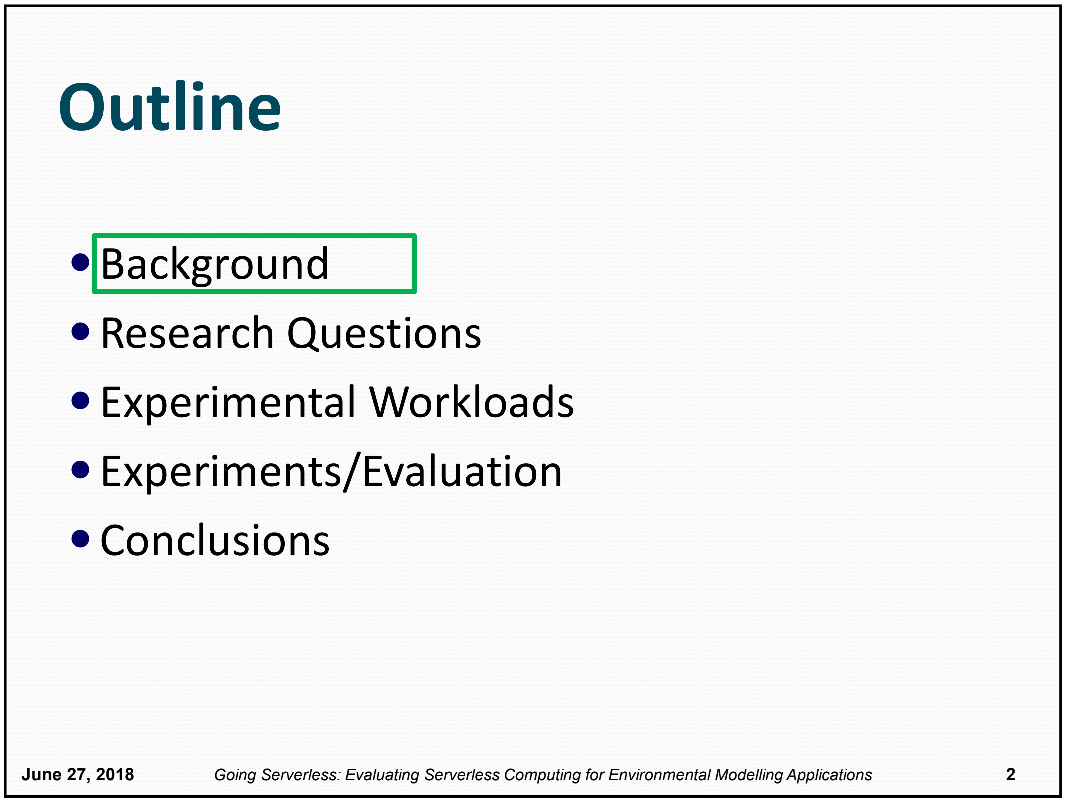
Baojia Zhang, Wes Lloyd[1], Olaf David, George Leavesley

[1] http://faculty.washington.edu/wlloyd/

June 27, 2018

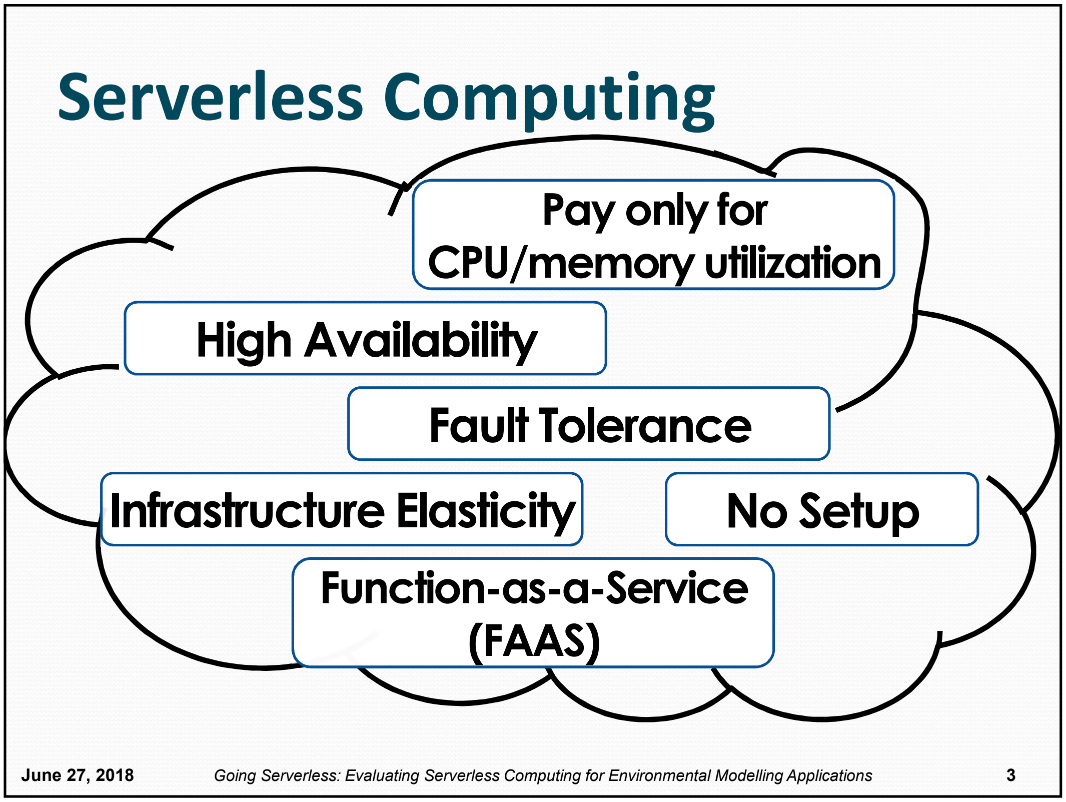
Institute of Technology,
University of Washington, Tacoma, Washington USA

iEMSs 2018: 9th International Congress on Environmental Modelling and Software



Outline

- Background
- Research Questions
- Experimental Workloads
- Experiments/Evaluation
- Conclusions



Serverless Computing

Pay only for CPU/memory utilization

High Availability

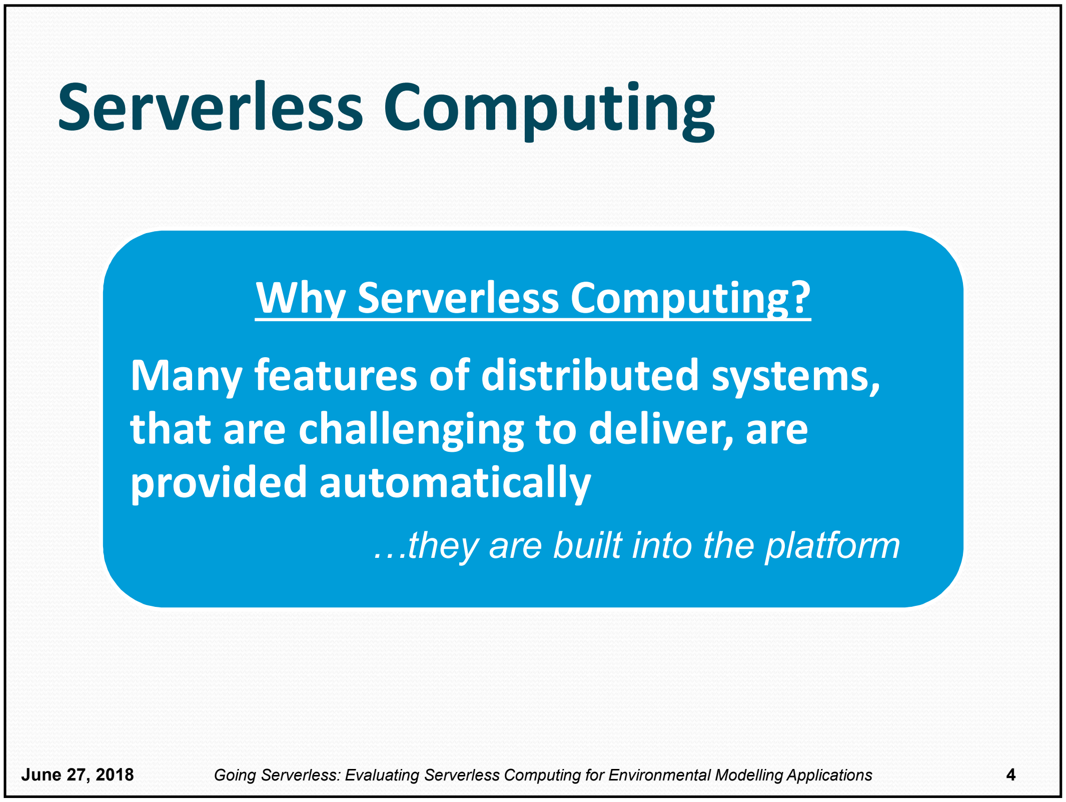
Fault Tolerance

Infrastructure Elasticity

No Setup

Function-as-a-Service (FAAS)

June 27, 2018 *Going Serverless: Evaluating Serverless Computing for Environmental Modelling Applications* 3



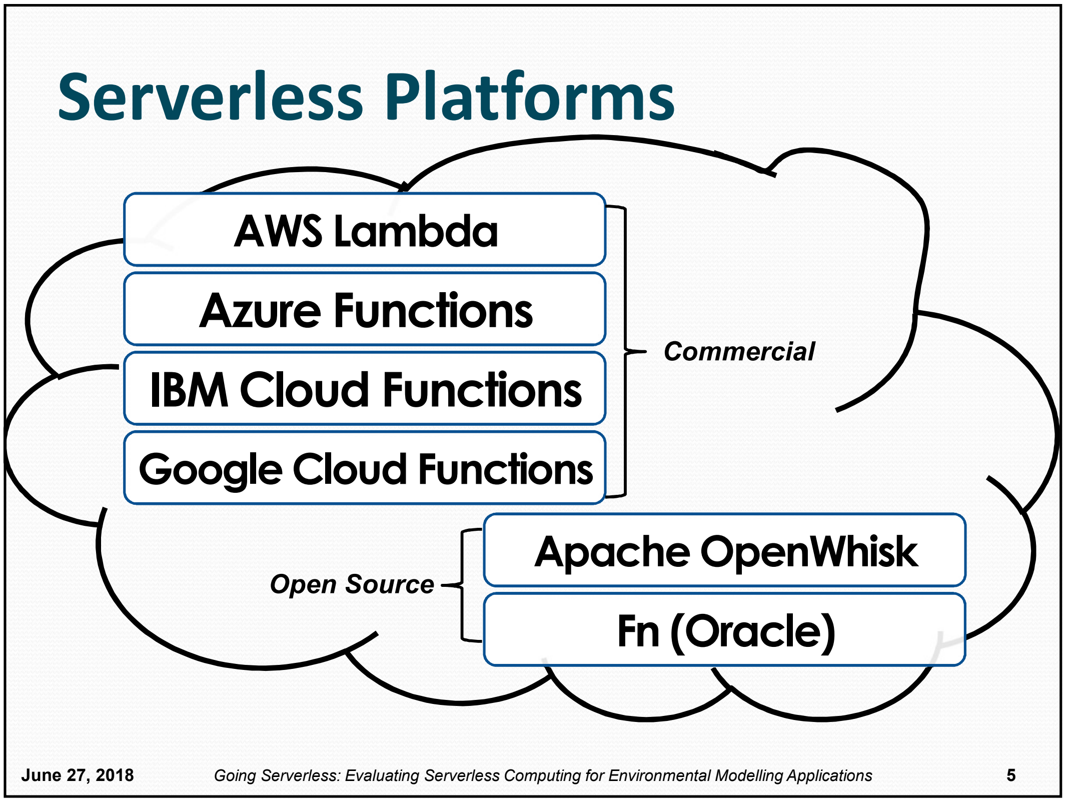
Serverless Computing

Why Serverless Computing?

Many features of distributed systems, that are challenging to deliver, are provided automatically

…they are built into the platform

June 27, 2018 *Going Serverless: Evaluating Serverless Computing for Environmental Modelling Applications* 4



Serverless Platforms

Commercial:
- AWS Lambda
- Azure Functions
- IBM Cloud Functions
- Google Cloud Functions

Open Source:
- Apache OpenWhisk
- Fn (Oracle)

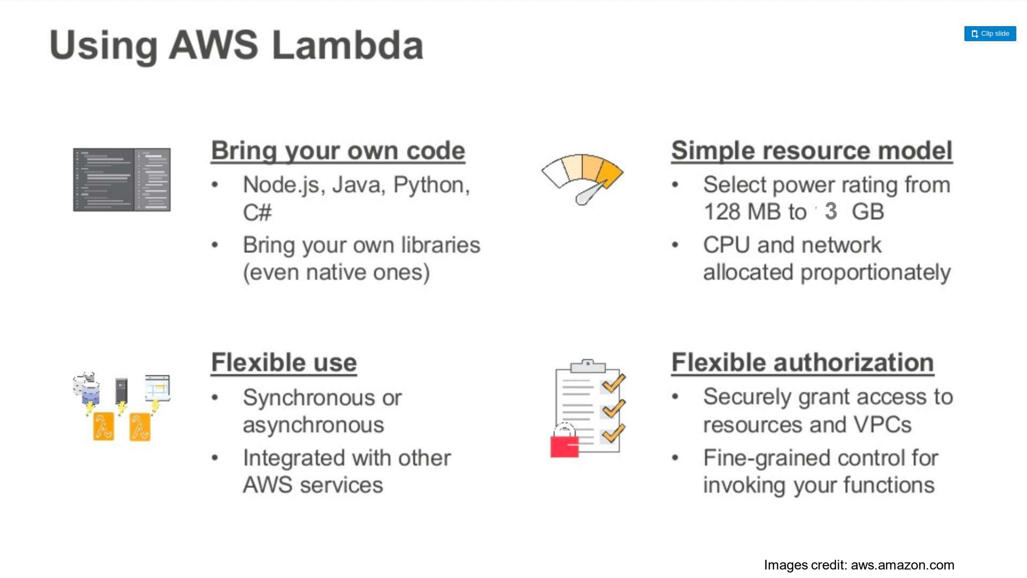
AWS Lambda

Serverless Computing Platform

Serverless Computing
Deploy Applications Without
Fiddling With Servers

Image from: https://mobisoftinfotech.com/resources/blog/serverless-computing-deploy-applications-without-fiddling-with-servers/

Aws Lambda



Using AWS Lambda

Bring your own code

- Node.js, Java, Python, C#
- Bring your own libraries (even native ones)

Simple resource model

- Select power rating from 128 MB to 3 GB
- CPU and network allocated proportionately

Flexible use

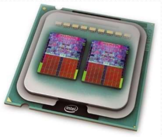
- Synchronous or asynchronous
- Integrated with other AWS services

Flexible authorization

- Securely grant access to resources and VPCs
- Fine-grained control for invoking your functions

Images credit: aws.amazon.com

Smith Waterman Example

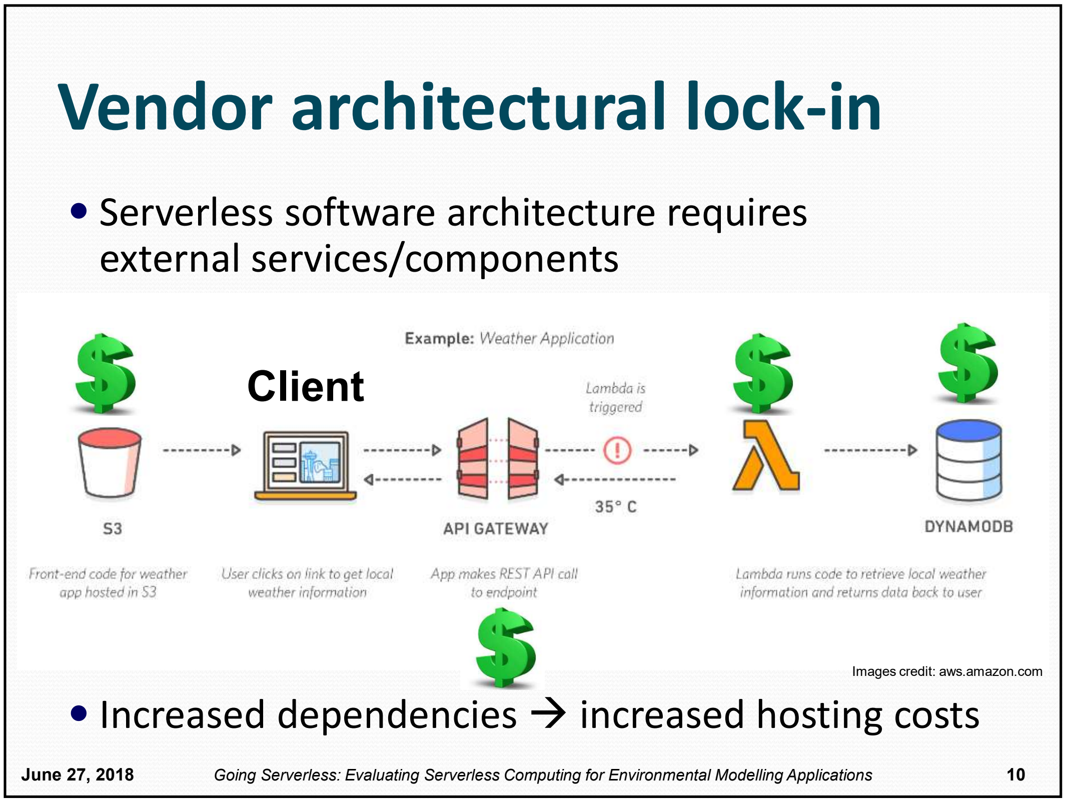


- Applies dynamic programming to find best local sequencing alignment of two DNA/RNA samples
  - Embarrassingly parallel, each task can run in isolation
  - Use case for GPU acceleration
- **Example:** Compare 20,336 protein sequences
  - Python client, C execution engine
- Intel i5-7200U 2.5 GHz laptop client (2-core, 4-HT): 8.7 hrs
- AWS Lambda, same laptop as client: 2.2 minutes
  - Partitions 20,336 sequences into 41 sets
  - Execution cost: **~ 82¢ (237x speed-up)**

Serverless Computing

Challenges for Environmental Modelling

Image from: https://mobisoftinfotech.com/resources/blog/serverless-computing-deploy-applications-without-fiddling-with-servers/



Vendor architectural lock-in

- Serverless software architecture requires external services/components

Example: Weather Application
Client
$
S3
Front-end code for weather app hosted in S3
User clicks on link to get local weather information
API GATEWAY
App makes REST API call to endpoint
Lambda is triggered
35° C
DYNAMODB
Lambda runs code to retrieve local weather information and returns data back to user

Images credit: aws.amazon.com

- Increased dependencies → increased hosting costs

Pricing Obfuscation

- **VM pricing:** hourly rental pricing, billed to nearest second intuitive…
- **Serverless Computing:**

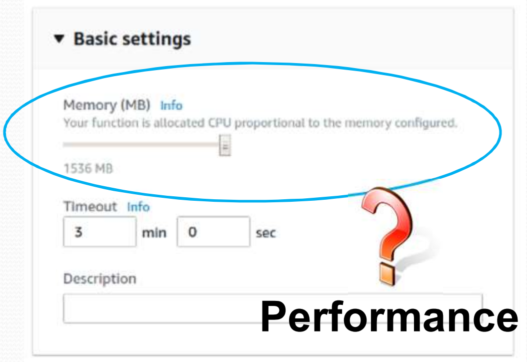
AWS Lambda Pricing

FREE TIER: first 1,000,000 function calls/month → FREE
first 400 GB-sec/month → FREE

- Afterwards: $0.0000002 per request
$0.000000208 to rent 128MB / 100-ms

Memory Reservation Question…

- Lambda memory reserved for functions
- UI provides "slider bar" to set function's memory allocation
- CPU power coupled to slider bar: *"every **doubling** of memory, **doubles** CPU…"*



▼ **Basic settings**

Memory (MB) Info
Your function is allocated CPU proportional to the memory configured.
1536 MB

Timeout Info
3 min 0 sec

Description

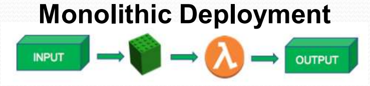
Performance

- **But how much memory do model services require?**

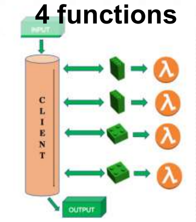
Service Composition

- **How should model code be composed for deployment to serverless computing platforms?**

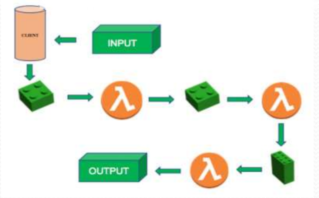
Monolithic Deployment



Client flow control, 4 functions



Server flow control, 3 functions



- Recommended practice: Decompose into many microservices
- Platform limits: code + libraries ~256MB
- **How does composition impact the number of function invocations, and memory utilization?**

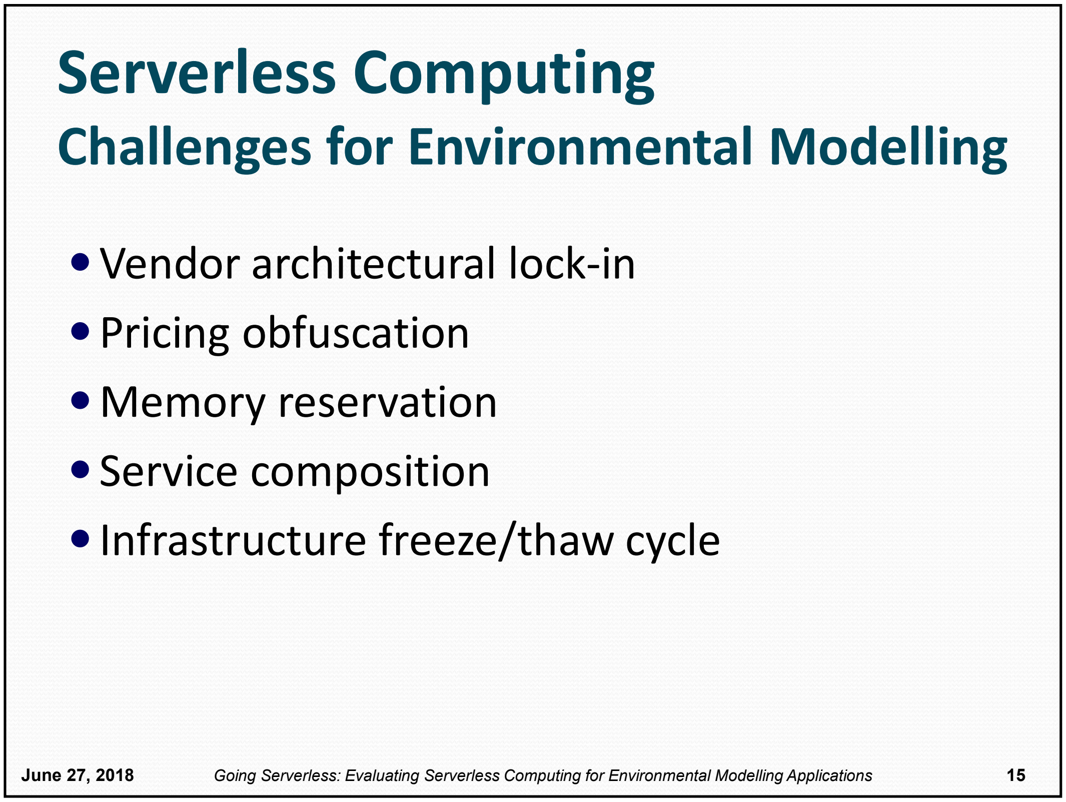
Performance

Infrastructure Freeze/Thaw Cycle

- Unused infrastructure is deprecated
  - ***But after how long?***
- Infrastructure: VMs, “containers”
- **Provider-COLD / VM-COLD**
  - “Container” images - built/transferred to VMs
- **Container-COLD**
  - Image cached on VM
- **Container-WARM**
  - “Container” running on VM

Performance

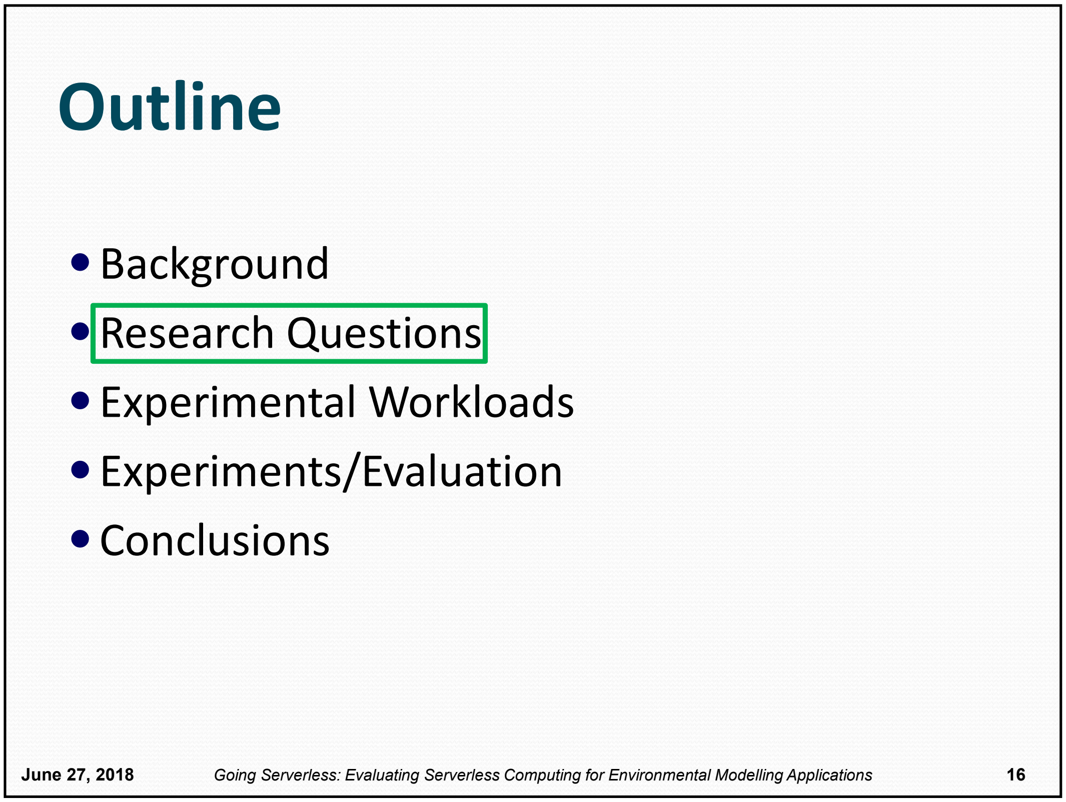
Image from: Denver7 – The Denver Channel News



Serverless Computing

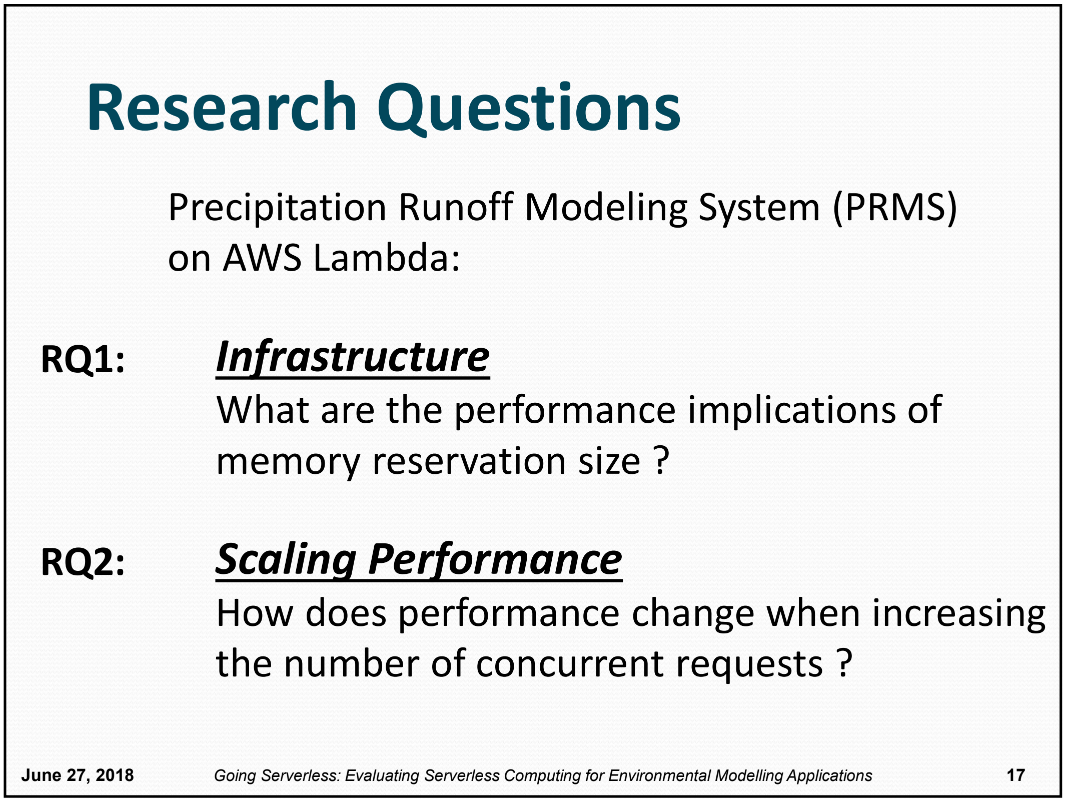
Challenges for Environmental Modelling

- Vendor architectural lock-in
- Pricing obfuscation
- Memory reservation
- Service composition
- Infrastructure freeze/thaw cycle



Outline

- Background
- Research Questions
- Experimental Workloads
- Experiments/Evaluation
- Conclusions

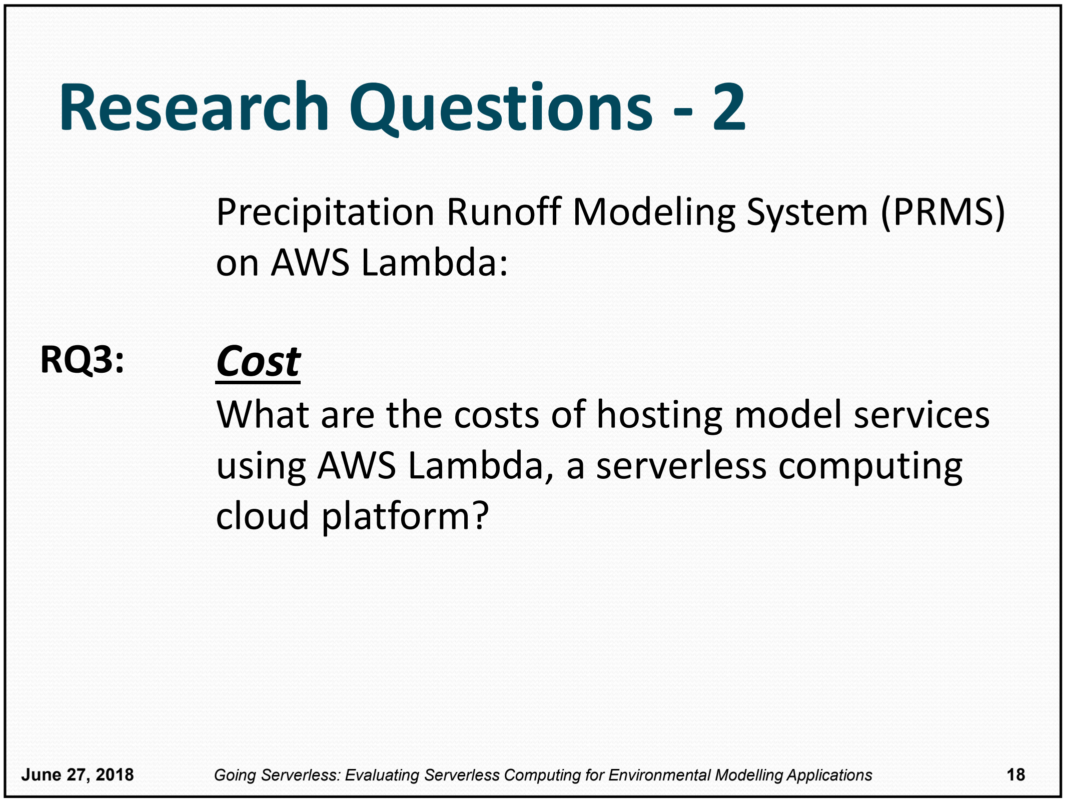


Research Questions

Precipitation Runoff Modeling System (PRMS) on AWS Lambda:

RQ1: ***Infrastructure***
What are the performance implications of memory reservation size ?

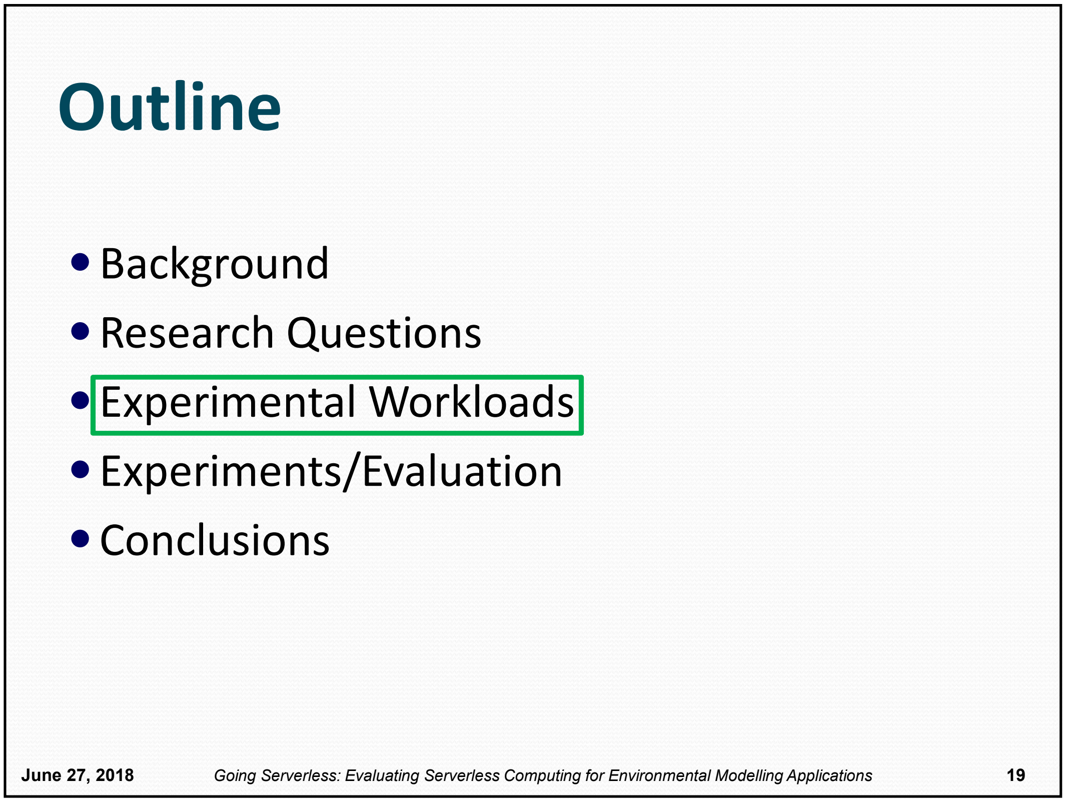
RQ2: ***Scaling Performance***
How does performance change when increasing the number of concurrent requests ?



Research Questions - 2

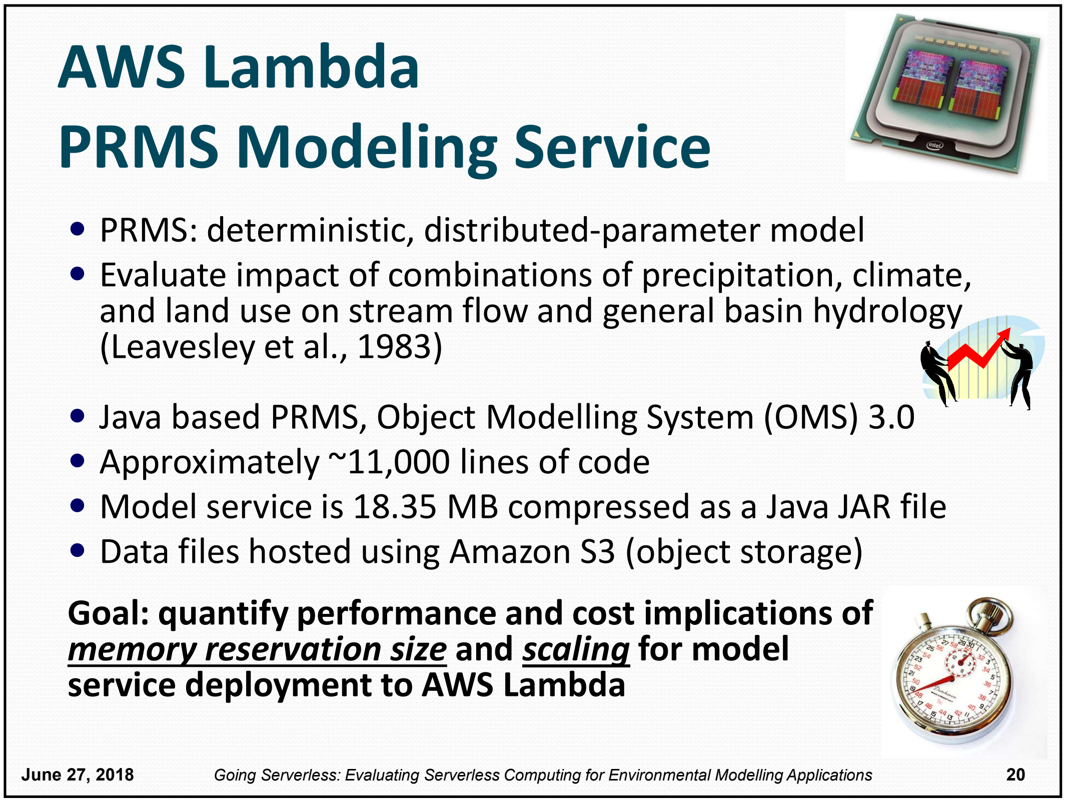
Precipitation Runoff Modeling System (PRMS) on AWS Lambda:

RQ3: ***Cost***
What are the costs of hosting model services using AWS Lambda, a serverless computing cloud platform?



Outline

- Background
- Research Questions
- Experimental Workloads
- Experiments/Evaluation
- Conclusions



AWS Lambda PRMS Modeling Service

- PRMS: deterministic, distributed-parameter model
- Evaluate impact of combinations of precipitation, climate, and land use on stream flow and general basin hydrology (Leavesley et al., 1983)
- Java based PRMS, Object Modelling System (OMS) 3.0
- Approximately ~11,000 lines of code
- Model service is 18.35 MB compressed as a Java JAR file
- Data files hosted using Amazon S3 (object storage)

Goal: quantify performance and cost implications of *__memory reservation size__* and *__scaling__* for model service deployment to AWS Lambda

Serverless Computing: An Investigation of Factors Influencing Microservice Performance

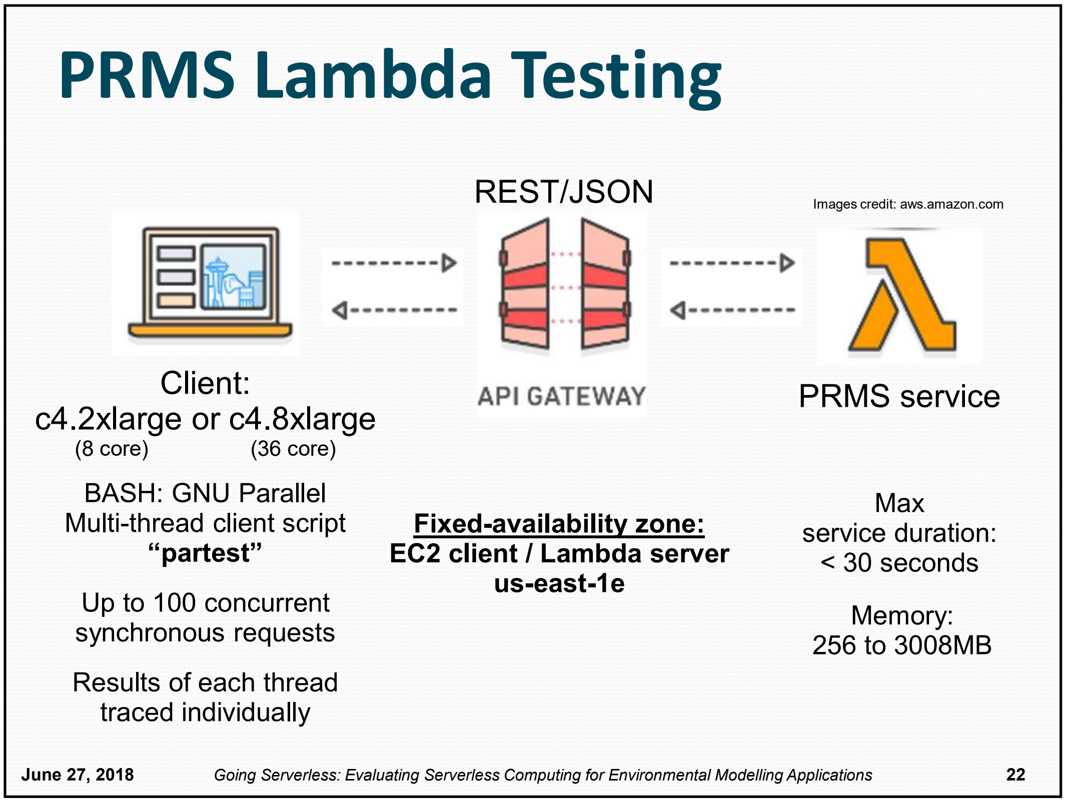
Wes Lloyd, Shruti Ramesh,
Swetha Chinthalapati,
Lan Ly, Shrideep Pallickara

April 20, 2018

Institute of Technology,
University of Washington, Tacoma, Washington USA

IC2E 2018: IEEE International Conference on Cloud Engineering

Available at: https://goo.gl/tZvfCH



PRMS Lambda Testing

Client:
c4.2xlarge or c4.8xlarge
(8 core) (36 core)

BASH: GNU Parallel
Multi-thread client script
“partest”

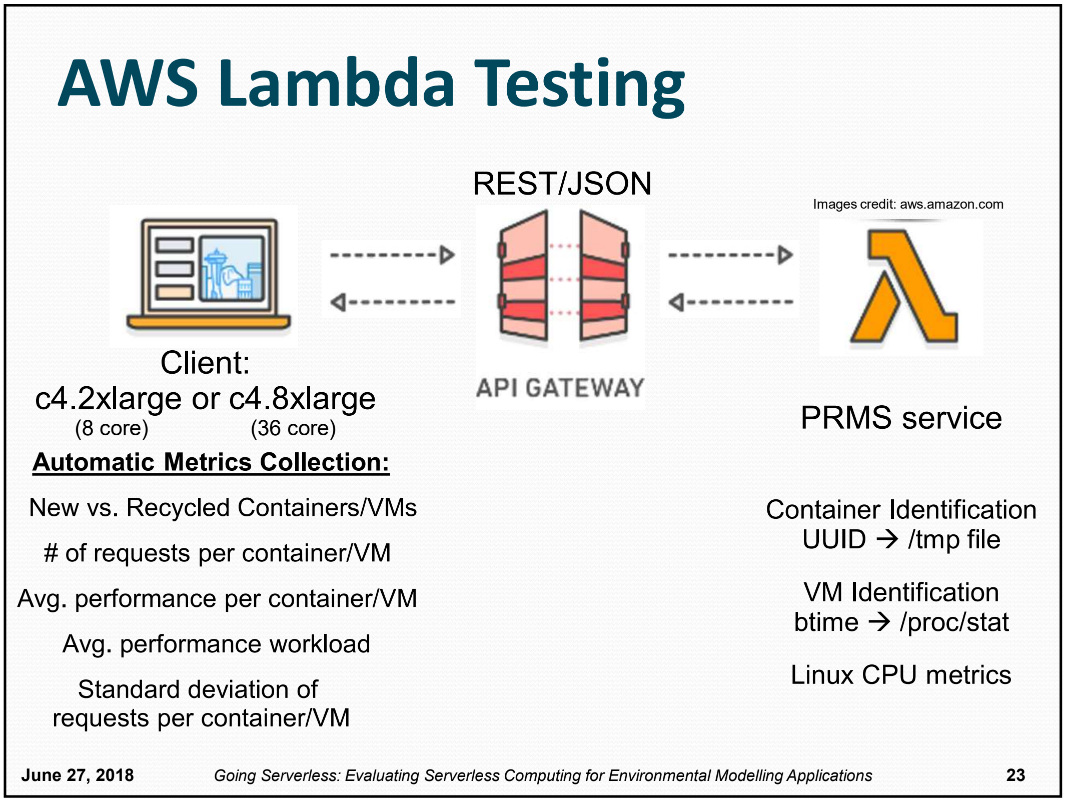
Up to 100 concurrent synchronous requests

Results of each thread traced individually

Fixed-availability zone:
EC2 client / Lambda server
us-east-1e

Max
service duration:
< 30 seconds

Memory:
256 to 3008MB



AWS Lambda Testing

REST/JSON

Images credit: aws.amazon.com

Client:
c4.2xlarge or c4.8xlarge
(8 core) (36 core)

Automatic Metrics Collection:

New vs. Recycled Containers/VMs

# of requests per container/VM

Avg. performance per container/VM

Avg. performance workload

Standard deviation of requests per container/VM

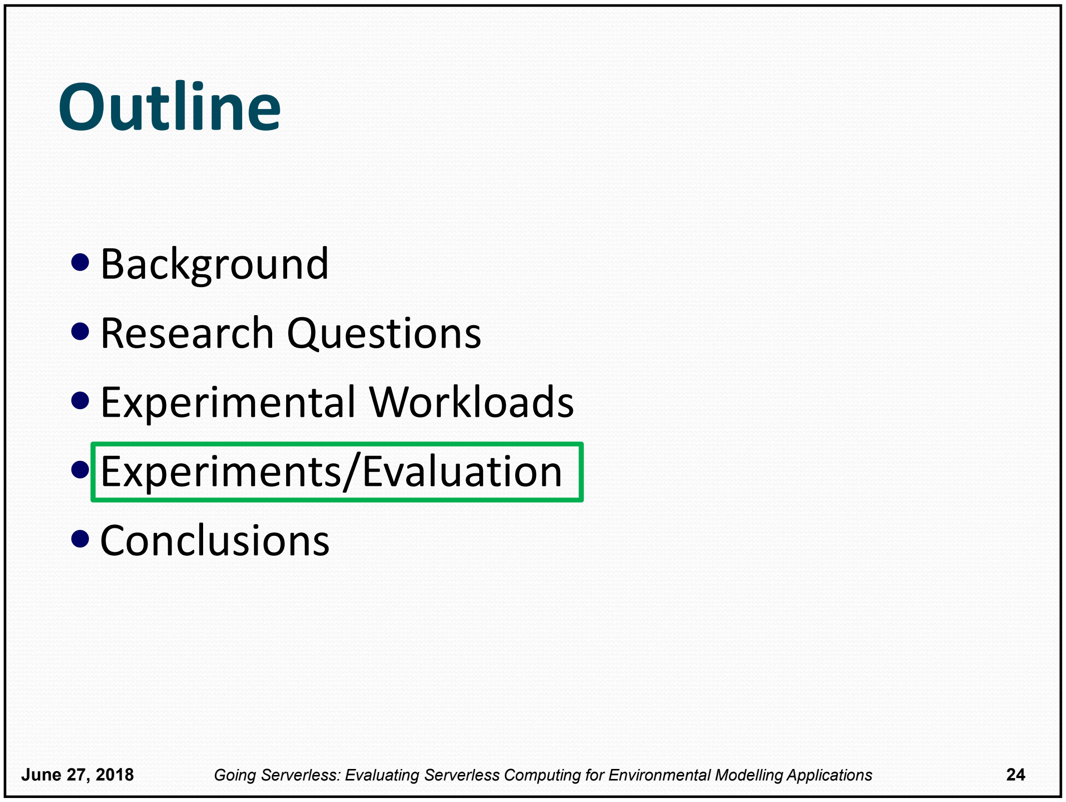
PRMS service

Container Identification
UUID → /tmp file

VM Identification
btime → /proc/stat

Linux CPU metrics

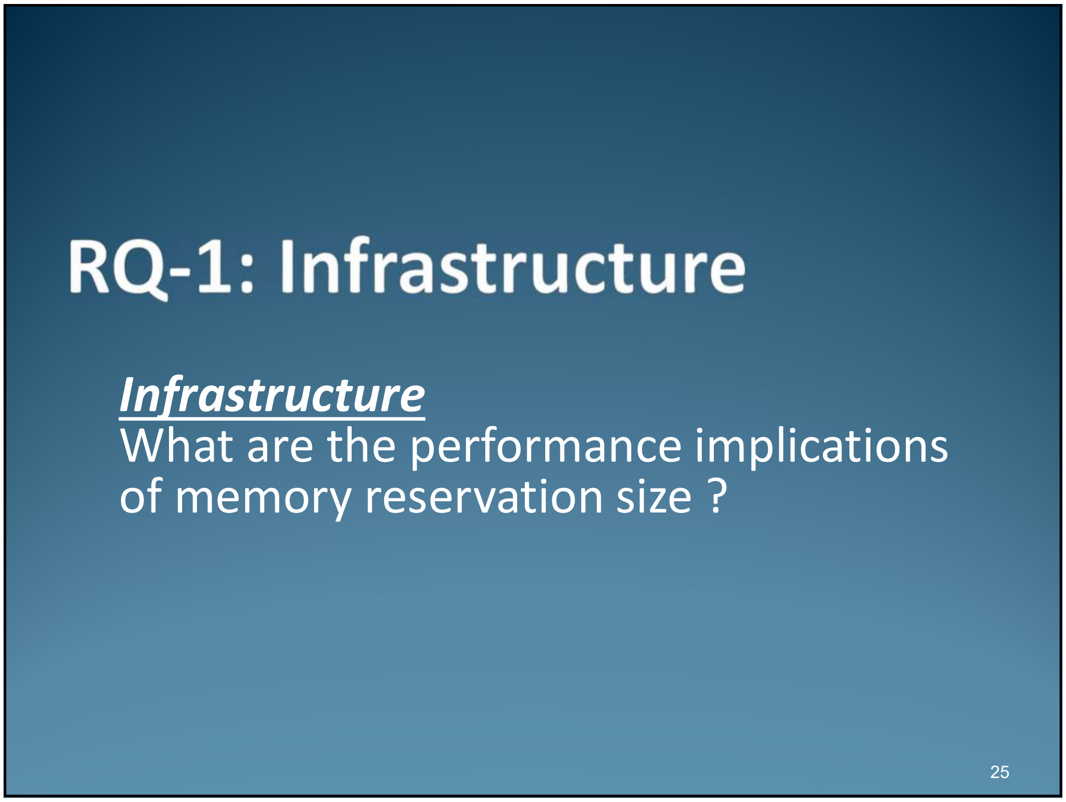
June 27, 2018 *Going Serverless: Evaluating Serverless Computing for Environmental Modelling Applications* 23



Outline

- Background
- Research Questions
- Experimental Workloads
- Experiments/Evaluation
- Conclusions

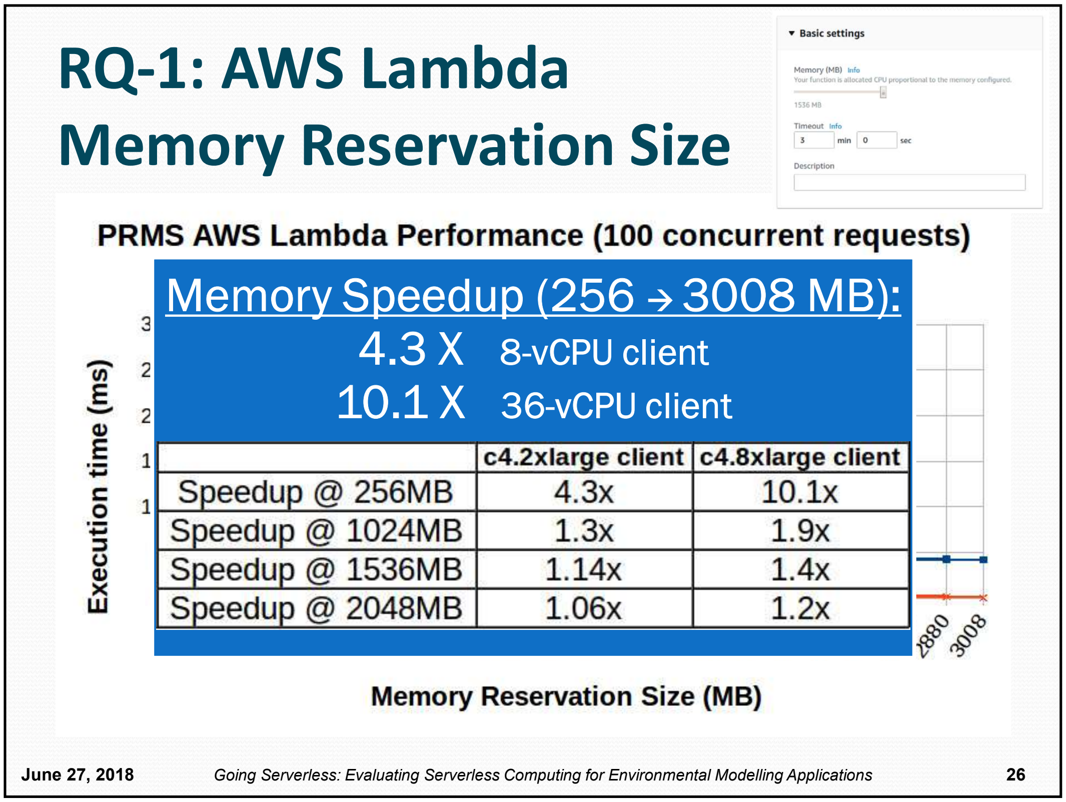
June 27, 2018 *Going Serverless: Evaluating Serverless Computing for Environmental Modelling Applications* 24



RQ-1: Infrastructure

Infrastructure
What are the performance implications of memory reservation size ?

25



RQ-1: AWS Lambda Memory Reservation Size

▼ Basic settings

Memory (MB) Info
Your function is allocated CPU proportional to the memory configured.

1536 MB

Timeout Info
3 min 0 sec

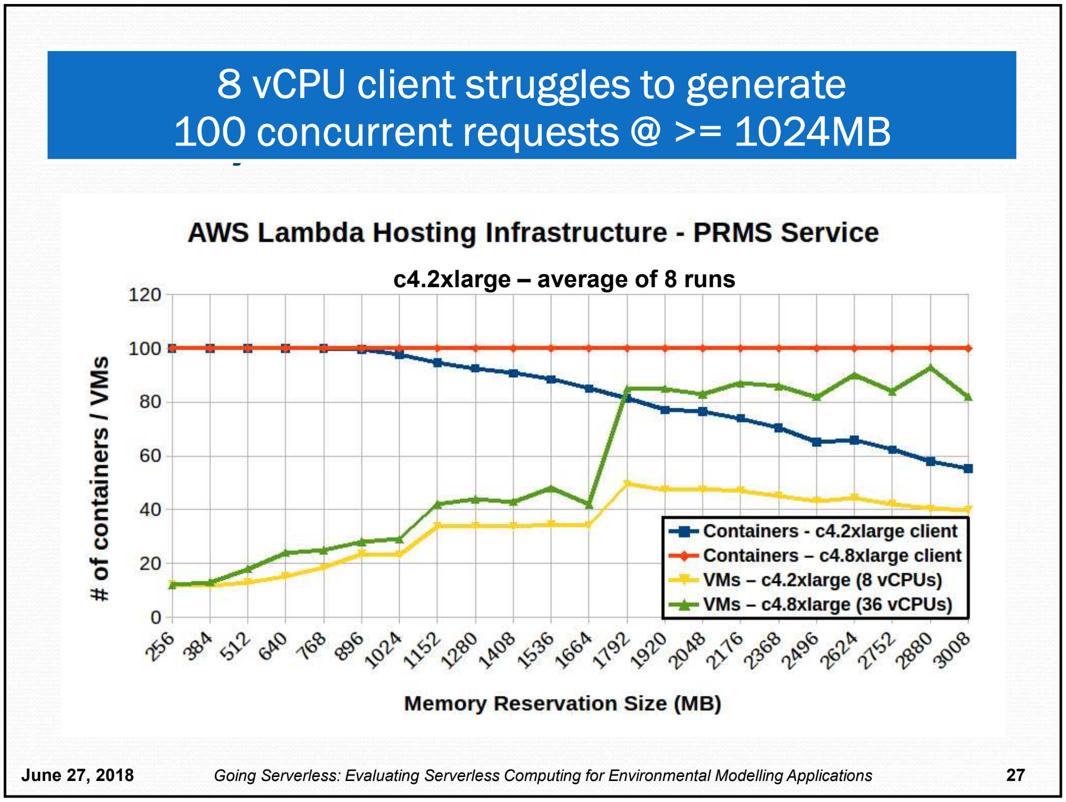
Description

PRMS AWS Lambda Performance (100 concurrent requests)

Memory Speedup (256 → 3008 MB):
4.3 X 8-vCPU client
10.1 X 36-vCPU client

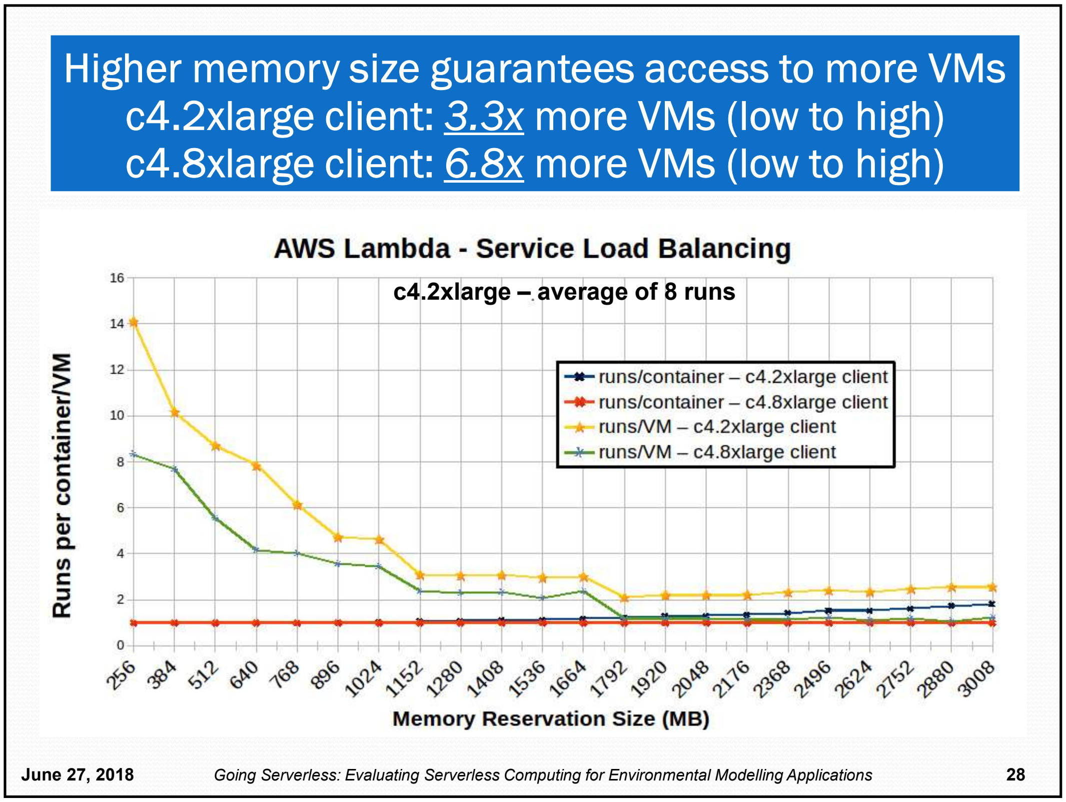
	c4.2xlarge client	c4.8xlarge client
Speedup @ 256MB	4.3x	10.1x
Speedup @ 1024MB	1.3x	1.9x
Speedup @ 1536MB	1.14x	1.4x
Speedup @ 2048MB	1.06x	1.2x

June 27, 2018 *Going Serverless: Evaluating Serverless Computing for Environmental Modelling Applications* 26



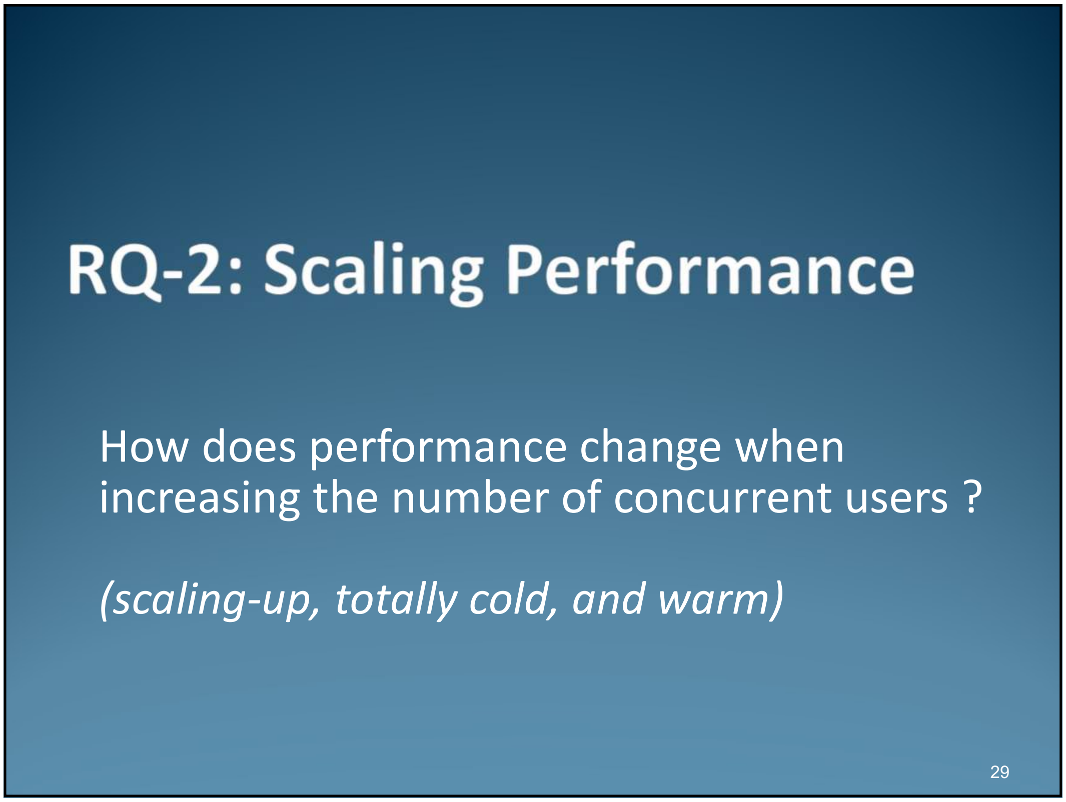
8 vCPU client struggles to generate 100 concurrent requests @ >= 1024MB

June 27, 2018 *Going Serverless: Evaluating Serverless Computing for Environmental Modelling Applications* 27



Higher memory size guarantees access to more VMs c4.2xlarge client: <u>3.3x</u> more VMs (low to high) c4.8xlarge client: <u>6.8x</u> more VMs (low to high)

June 27, 2018 *Going Serverless: Evaluating Serverless Computing for Environmental Modelling Applications* 28

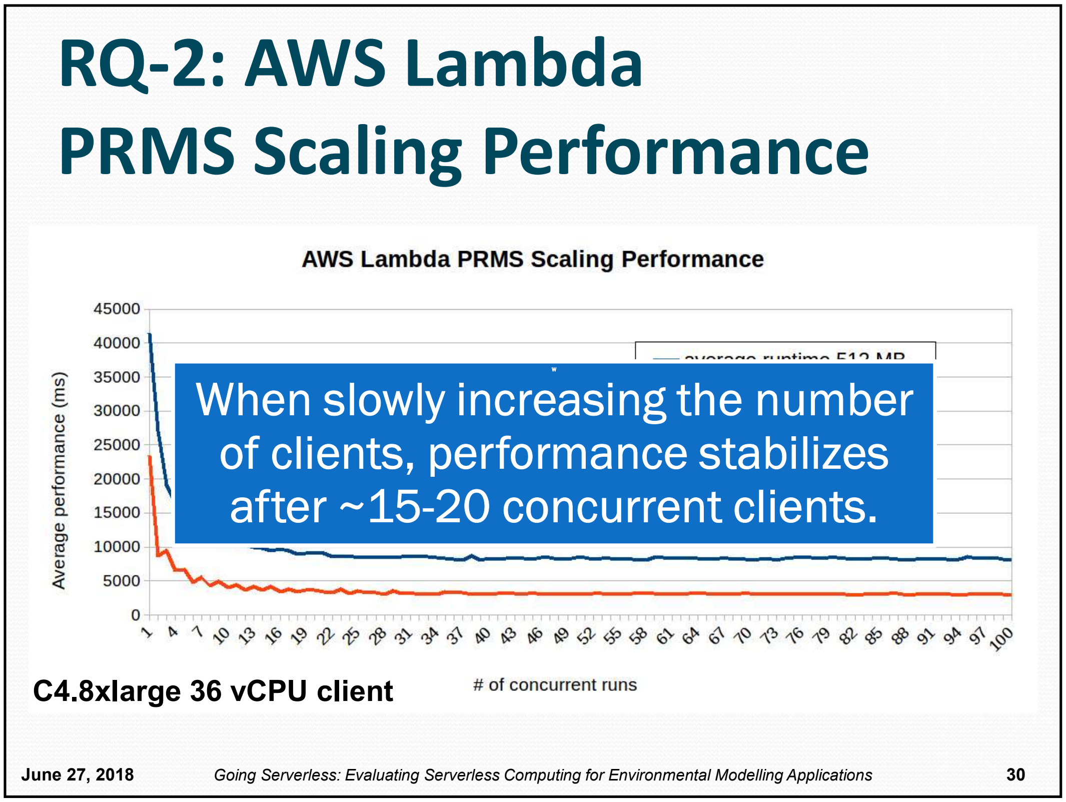


RQ-2: Scaling Performance

How does performance change when increasing the number of concurrent users ?

(scaling-up, totally cold, and warm)

29

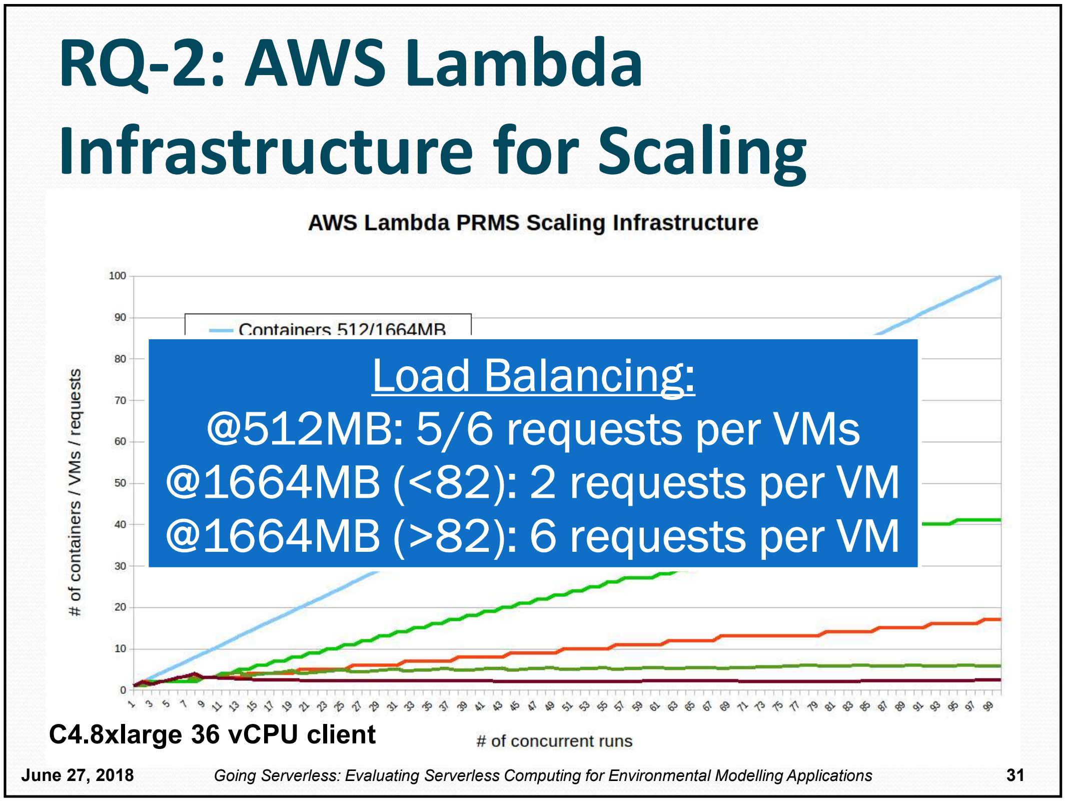


RQ-2: AWS Lambda PRMS Scaling Performance

When slowly increasing the number of clients, performance stabilizes after ~15-20 concurrent clients.

C4.8xlarge 36 vCPU client

June 27, 2018 *Going Serverless: Evaluating Serverless Computing for Environmental Modelling Applications* **30**

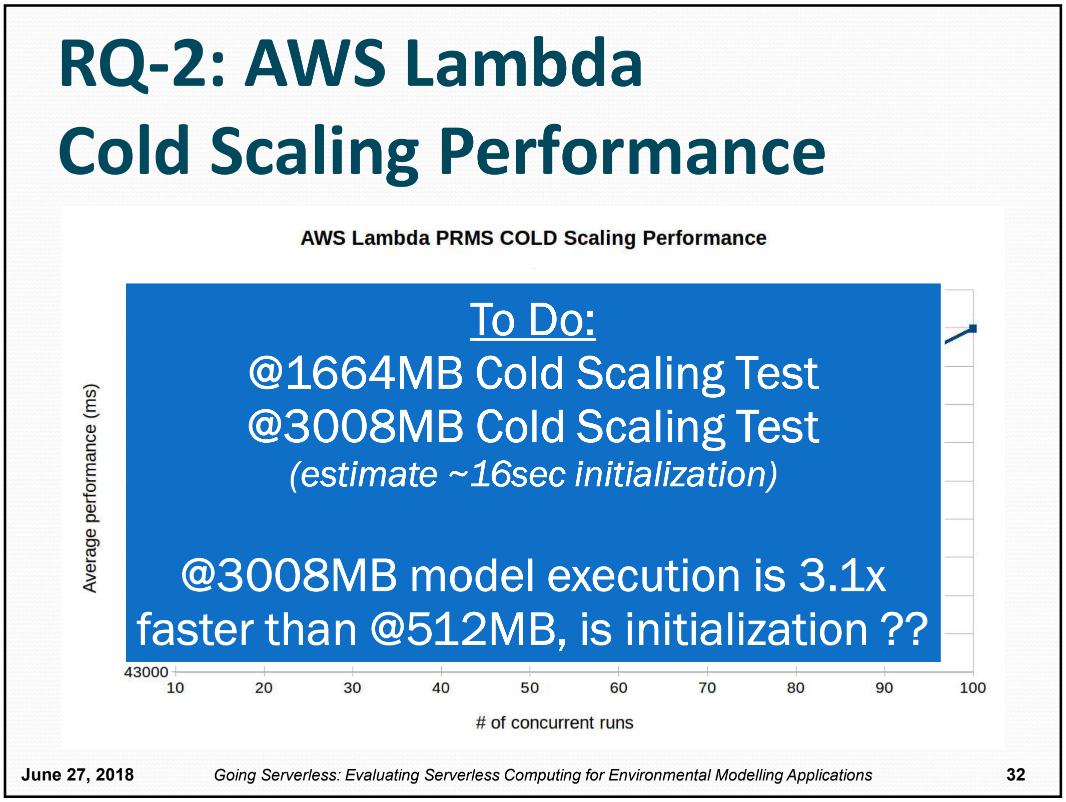


RQ-2: AWS Lambda Infrastructure for Scaling

Load Balancing:
@512MB: 5/6 requests per VMs
@1664MB (<82): 2 requests per VM
@1664MB (>82): 6 requests per VM

C4.8xlarge 36 vCPU client

June 27, 2018 *Going Serverless: Evaluating Serverless Computing for Environmental Modelling Applications* 31

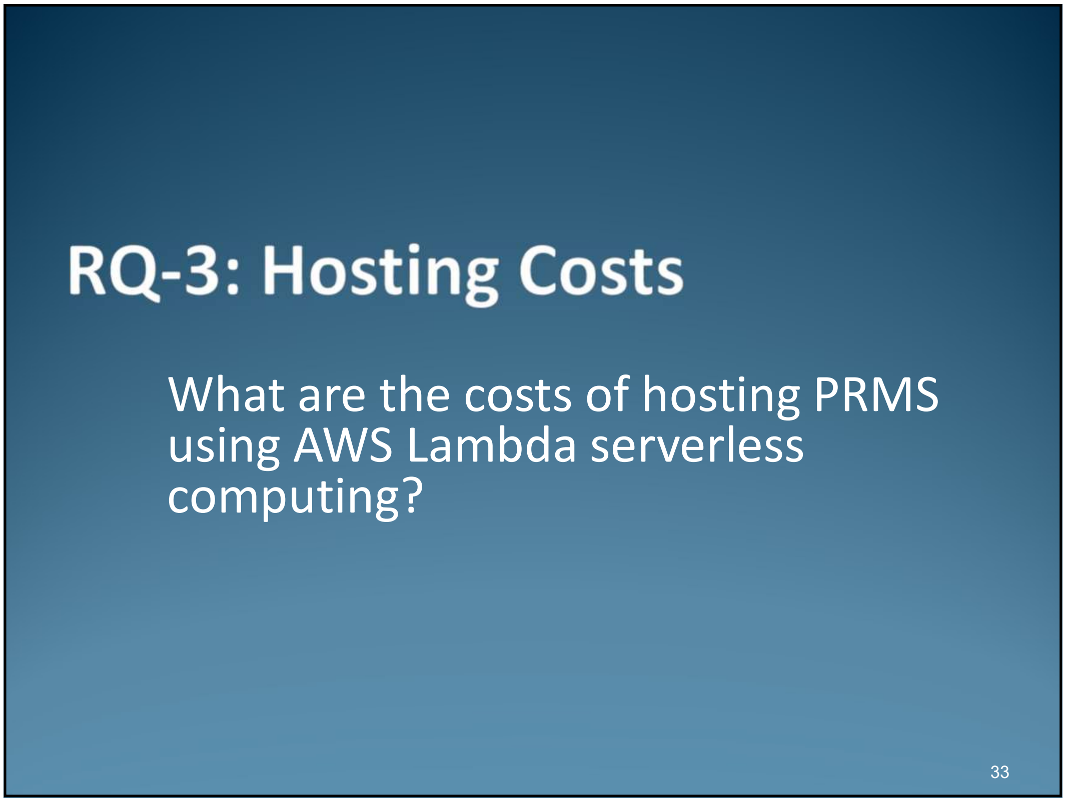


RQ-2: AWS Lambda Cold Scaling Performance

To Do:
@1664MB Cold Scaling Test
@3008MB Cold Scaling Test
(estimate ~16sec initialization)

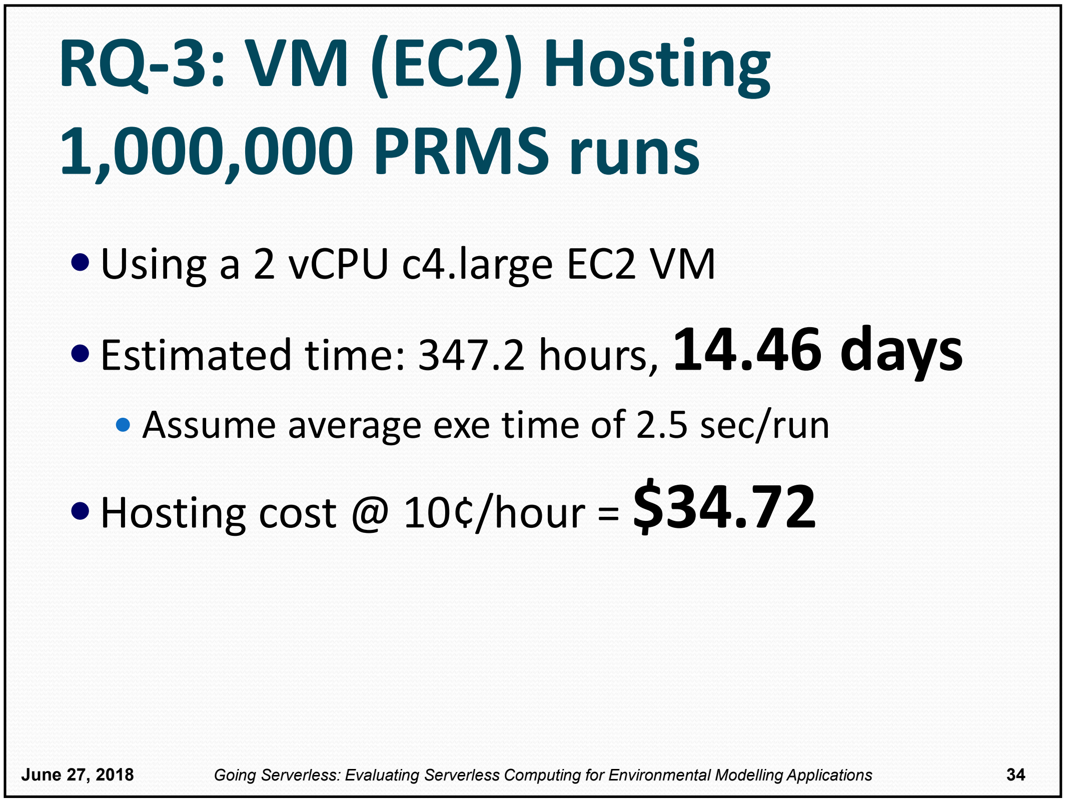
@3008MB model execution is 3.1x faster than @512MB, is initialization ??

June 27, 2018 *Going Serverless: Evaluating Serverless Computing for Environmental Modelling Applications* 32



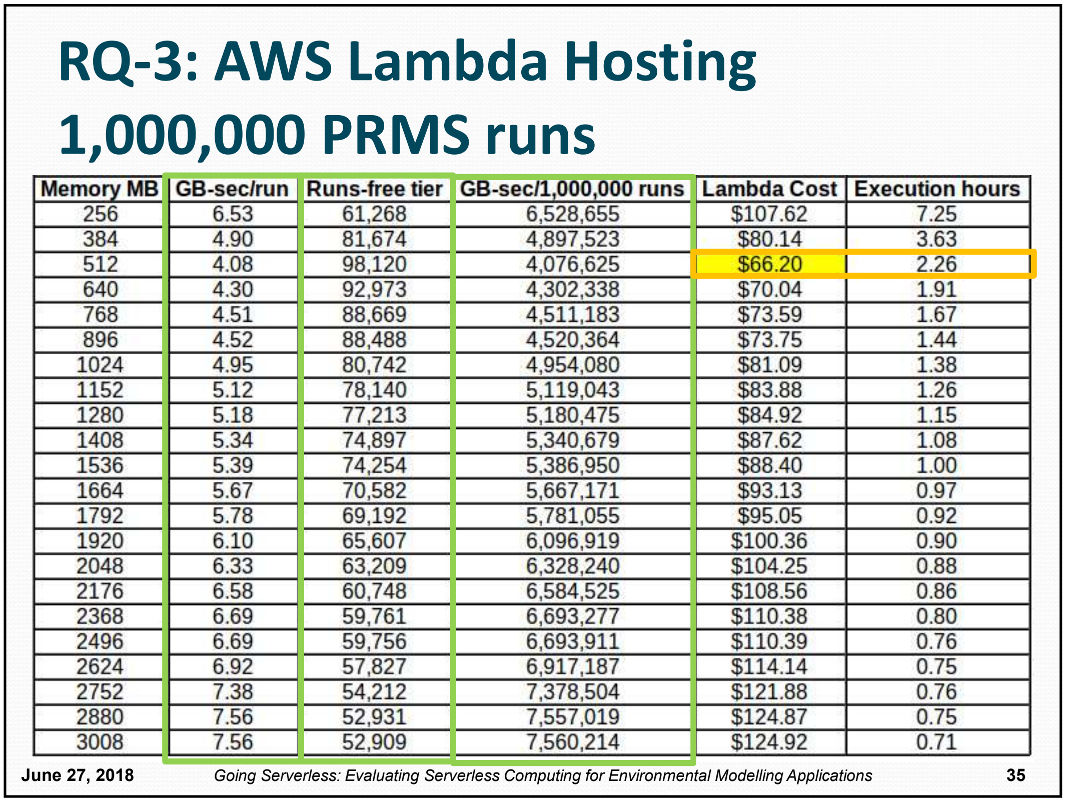
RQ-3: Hosting Costs

What are the costs of hosting PRMS using AWS Lambda serverless computing?



RQ-3: VM (EC2) Hosting 1,000,000 PRMS runs

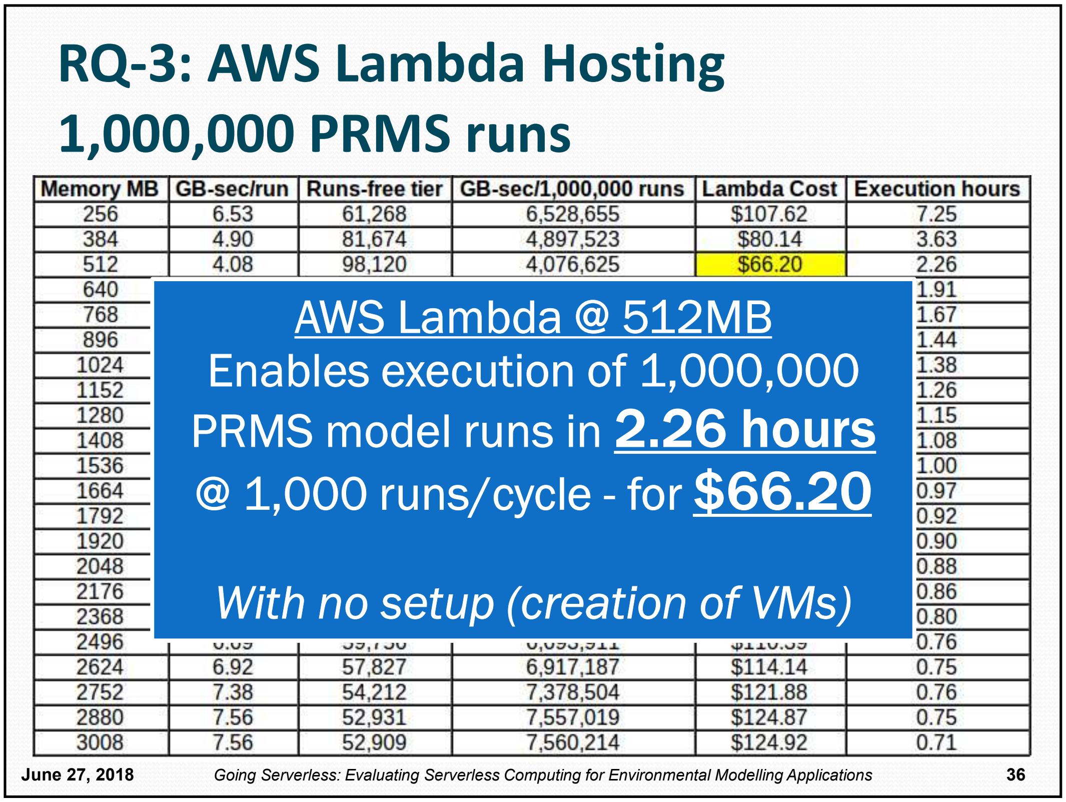
- Using a 2 vCPU c4.large EC2 VM
- Estimated time: 347.2 hours, **14.46 days**
  - Assume average exe time of 2.5 sec/run
- Hosting cost @ 10¢/hour = **$34.72**



RQ-3: AWS Lambda Hosting 1,000,000 PRMS runs

Memory MB	GB-sec/run	Runs-free tier	GB-sec/1,000,000 runs	Lambda Cost	Execution hours
256	6.53	61,268	6,528,655	$107.62	7.25
384	4.90	81,674	4,897,523	$80.14	3.63
512	4.08	98,120	4,076,625	$66.20	2.26
640	4.30	92,973	4,302,338	$70.04	1.91
768	4.51	88,669	4,511,183	$73.59	1.67
896	4.52	88,488	4,520,364	$73.75	1.44
1024	4.95	80,742	4,954,080	$81.09	1.38
1152	5.12	78,140	5,119,043	$83.88	1.26
1280	5.18	77,213	5,180,475	$84.92	1.15
1408	5.34	74,897	5,340,679	$87.62	1.08
1536	5.39	74,254	5,386,950	$88.40	1.00
1664	5.67	70,582	5,667,171	$93.13	0.97
1792	5.78	69,192	5,781,055	$95.05	0.92
1920	6.10	65,607	6,096,919	$100.36	0.90
2048	6.33	63,209	6,328,240	$104.25	0.88
2176	6.58	60,748	6,584,525	$108.56	0.86
2368	6.69	59,761	6,693,277	$110.38	0.80
2496	6.69	59,756	6,693,911	$110.39	0.76
2624	6.92	57,827	6,917,187	$114.14	0.75
2752	7.38	54,212	7,378,504	$121.88	0.76
2880	7.56	52,931	7,557,019	$124.87	0.75
3008	7.56	52,909	7,560,214	$124.92	0.71

June 27, 2018 *Going Serverless: Evaluating Serverless Computing for Environmental Modelling Applications* 35



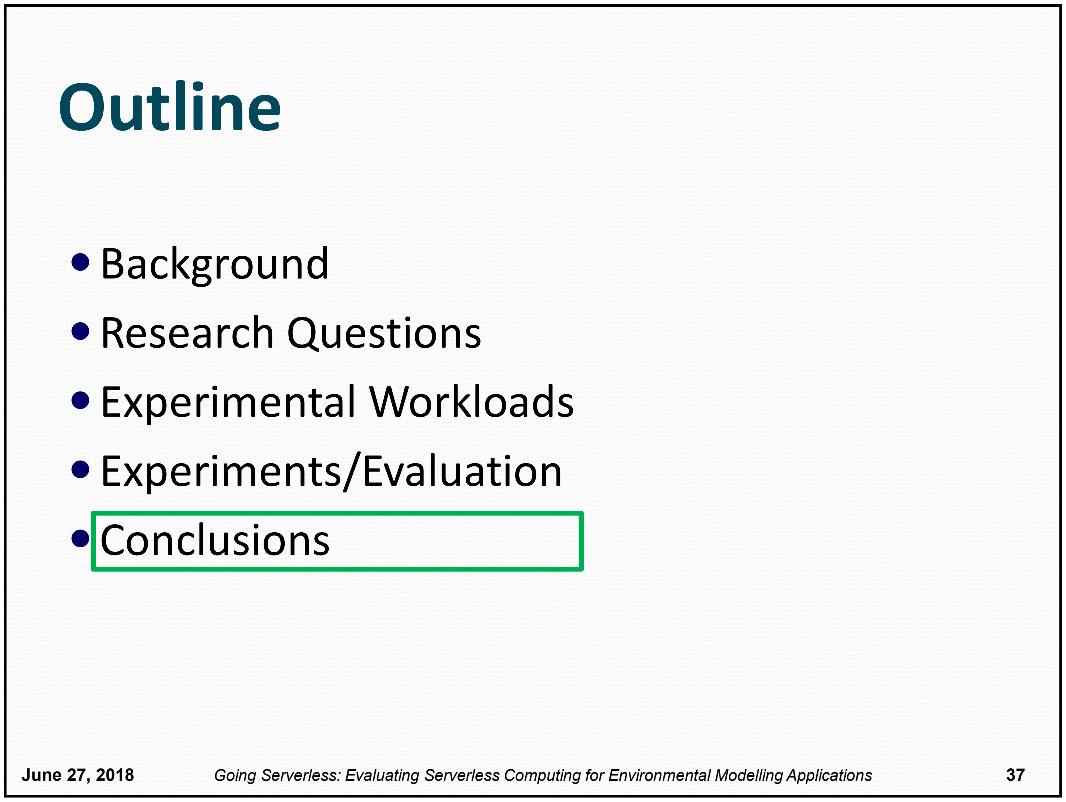
RQ-3: AWS Lambda Hosting 1,000,000 PRMS runs

Memory MB	GB-sec/run	Runs-free tier	GB-sec/1,000,000 runs	Lambda Cost	Execution hours
256	6.53	61,268	6,528,655	$107.62	7.25
384	4.90	81,674	4,897,523	$80.14	3.63
512	4.08	98,120	4,076,625	$66.20	2.26
640					1.91
768					1.67
896					1.44
1024					1.38
1152					1.26
1280					1.15
1408					1.08
1536					1.00
1664					0.97
1792					0.92
1920					0.90
2048					0.88
2176					0.86
2368					0.80
2496	6.69	59,756	6,693,911	$110.39	0.76
2624	6.92	57,827	6,917,187	$114.14	0.75
2752	7.38	54,212	7,378,504	$121.88	0.76
2880	7.56	52,931	7,557,019	$124.87	0.75
3008	7.56	52,909	7,560,214	$124.92	0.71

AWS Lambda @ 512MB
Enables execution of 1,000,000 PRMS model runs in **2.26 hours** @ 1,000 runs/cycle - for **$66.20**

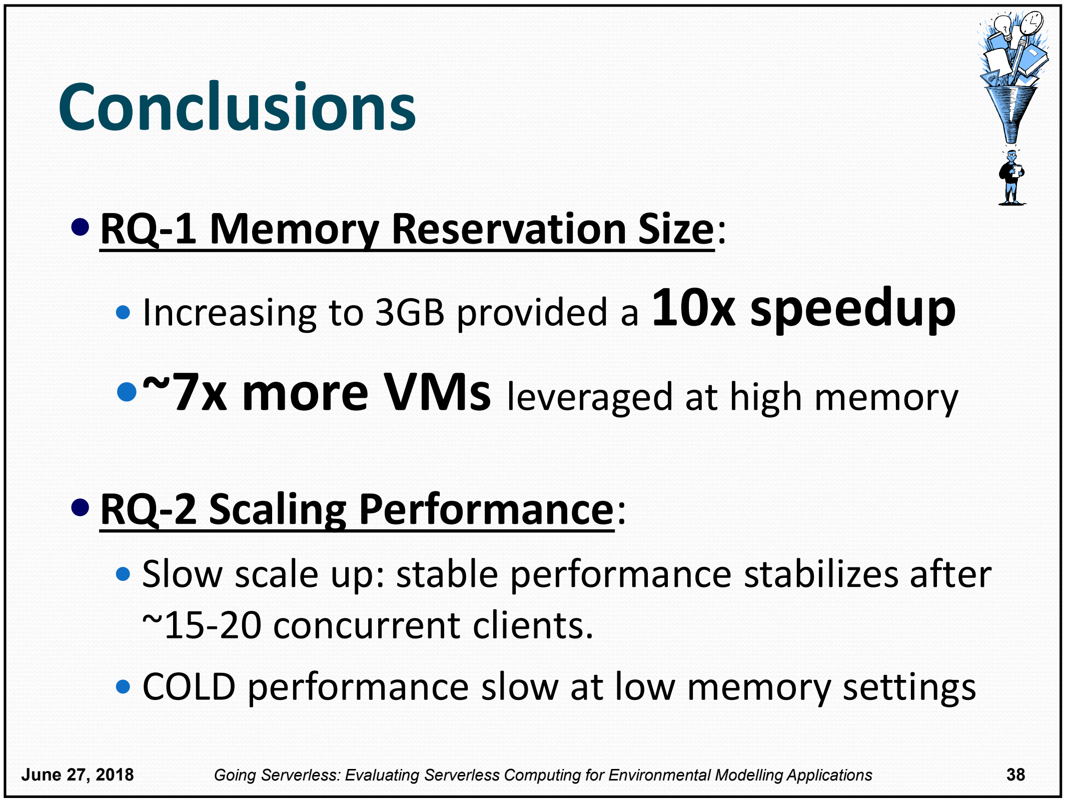
With no setup (creation of VMs)

June 27, 2018 *Going Serverless: Evaluating Serverless Computing for Environmental Modelling Applications* 36



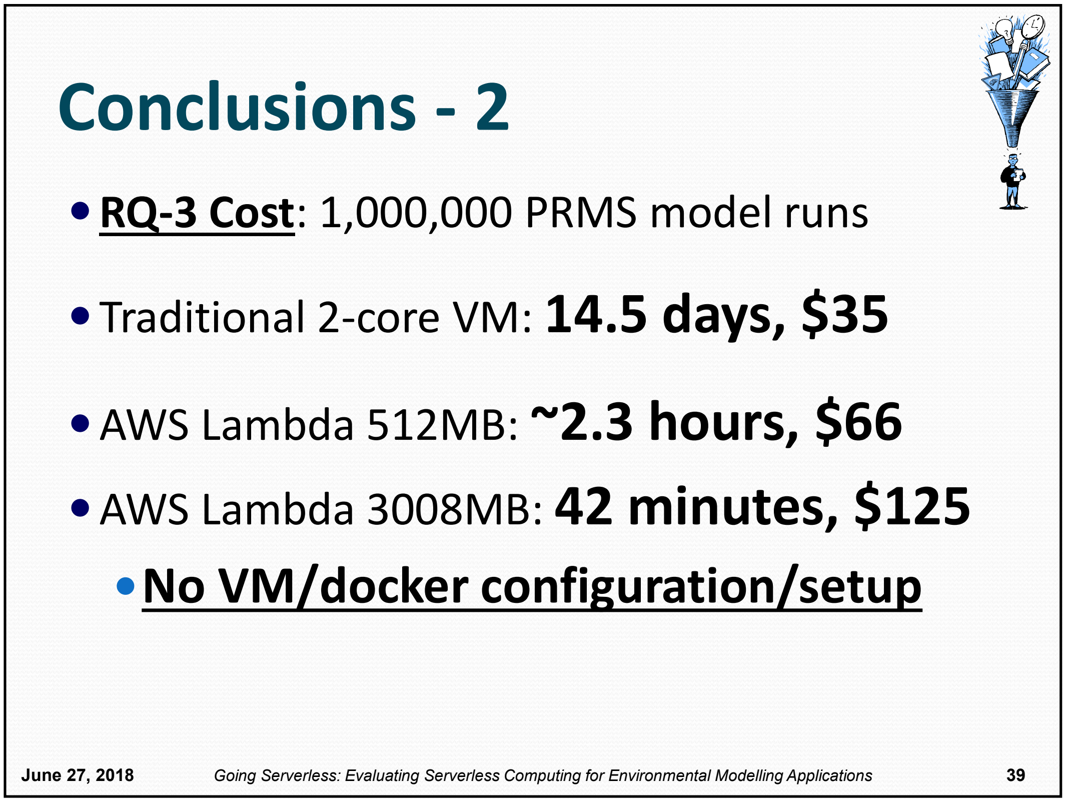
Outline

- Background
- Research Questions
- Experimental Workloads
- Experiments/Evaluation
- Conclusions



Conclusions

- **RQ-1 Memory Reservation Size**:
  - Increasing to 3GB provided a **10x speedup**
  - **~7x more VMs** leveraged at high memory
- **RQ-2 Scaling Performance**:
  - Slow scale up: stable performance stabilizes after ~15-20 concurrent clients.
  - COLD performance slow at low memory settings



Conclusions - 2

- **RQ-3 Cost**: 1,000,000 PRMS model runns
- Traditional 2-core VM: **14.5 days, $35**
- AWS Lambda 512MB: **~2.3 hours, $66**
- AWS Lambda 3008MB: **42 minutes, $125**
  - **No VM/docker configuration/setup**

June 27, 2018 *Going Serverless: Evaluating Serverless Computing for Environmental Modelling Applications* 39

Questions