A1
A2
A3
A4

TCSS562

PAPER PRESENTATION

PERFORMANCE EVALUATION OF HETEROGENEOUS CLOUD FUNCTIONS

ANKIT SINGH | AYUSH BANDIL | PRASHANTI PATHAK

1

INDEX

2

2

Slide 1

A1 There are too many slides.

Need to reduce to about 25 slides.

Can choose to cover only some of the hypothesis in the paper. There are 5, so many just talk about 3 of them. Otherwise the presentation will be too long. Group can choose the most interesting ones.
Author, 12/3/2019

A2 General comment:

Try to use a larger font size

Try to use phrases not complete sentences on all slides

Reduce the over number of words on each slide to prevent reading slides during the presentation
Author, 12/3/2019

A3 Overall this is a interesting comprehensive paper. The most novel contribution is in measuring and visualizing serverless infrastructure life span / life cycle
Author, 12/3/2019

A4 Note the relationship between the freeze-thaw cycle of serverless infrastructure and how often infrastructure is replaced.

Lambda replaces infrastructure frequently, and therefore has more freeze-thaw cycles
Author, 12/3/2019

A5

INTRODUCTION

- Cloud functions are heterogeneous due to variations in underlying hardware, runtime systems, resource management and billing models.
- Heterogeneity of cloud function may bring challenges such as network connection limitations/overheads, hardware performance throttling and billing issues, if appropriate cloud functions and providers aren't chosen as per program code. Which cloud provider to choose?
- Serverless architectures and cloud functions, in particular, have numerous use cases in both commercial and scientific applications. These lead to usage patterns, which are common to cloud functions and thus are interesting from performance evaluation perspective.
- Paper focuses on performance evaluation of cloud functions so that we can gain insight to resource allocation policies of different cloud providers which might help us save cost and increase efficiency of program.

3

3

HYPOTHESIS AND TESTS

- H1 – Computational performance of a cloud function is A6 proportional to function size.

  measure execution time of CPU-intensive workload.
- H2 – Network performance (throughput) of a cloud function is proportional to function size.

  measure download and upload time of benchmark file.
- H3 – Overheads do not depend on cloud function size and are consistent for each provider.

  compare request processing times that may be observed from the client with workload processing time measured in function runtime.

4

4

Slide 3

A5 This slide is very wordy.

Throughout the presentation, try to reduce bullet length to one line.
Use phrases, not complete sentences.

With sentences, there is a tendency to directly read the slide during the presentation.

Phases summarize main points with less words.

The help the speaker to talk about the topic, without just reading the presentation.
Author, 12/3/2019

Slide 4

A6 Is this function size in lines of code (LOC)?

What do the authors mean by function size? Is this lines of code (LOC) or total package size including dependencies & libraries ?
Author, 12/3/2019

HYPOTHESIS AND TESTS

- H4 – Application server instances are reused between calls and are recycled every couple of hours.

  assign a unique identifier for each execution environment and measure for how long it can be observed.

- H5 – Functions are executed on heterogeneous hardware.

  determine the process or type used for each function call if possible.

5

5

A7

RELATED WORK

- Paper References

1) Extensive performance studies focusing on selected IaaS cloud providers presented in the work of Lenk et al.

2) Performance of alternative cloud solutions such as Platform-as-a-Service (PaaS) has also been analyzed in the works of Prodan et al and Malawski et al.

3) Another study used DEWE workflow engine to run also Montage using a hybrid setup combining IaaS and FaaS.

A8

These studies provided up-to-date performance comparisons of current cloud provider offerings & also reported on effects of multitenancy, noisy neighbors and variability. Impacts of virtualization & performance variability were emphasized.

6

6

Slide 6

A7 What is DEWE ? This acronym will be unfamiliar. Define acronyms of first use.
Author, 12/3/2019

A8 many many words - use phrases on slides
Author, 12/3/2019

RELATED WORK

- Profiling Google Cloud functions by measuring execution times and execution counts through the emulator before actually deploying it. Done by Colt McAnlis (https://goo.gl/pKeiGV)
- None of the studies reported so far provides a comprehensive performance evaluation of all the major cloud function providers, giving the details on CPU performance, data access, overheads, lifetime and pricing, with emphasis on showing the heterogeneity of the environment.

A9

7

7

A10

BENCHMARKING

- For benchmarking cloud function providers, two frameworks are used

  Serverless Framework Suite **HyperFlow workflow engine**

- Serverless Framework Suite is used to execute and gather performance results of heterogeneous cloud function benchmarks over a long period of time. This is a new suite created by authors of this paper.
- HyperFlow workflow engine is used to run preliminary experiments on cloud functions and execute workflows that can have many parallel tasks.

8

8

Slide 7

A9 use short phrases on slides
Author, 12/3/2019

Slide 8

A10 to shorten the presentation could cut benchmarking down to 1 slide
Author, 12/3/2019

A11

BENCHMARKING (SERVERLESS)

- Source Code is pushed into the Git repository then Docker builds binaries compatible with target environments. It is picked up by Travis continuous integration (CI) so that the code is automatically deployed on each cloud whenever new code is pushed to the Git repository.
- After execution of cloud functions, the benchmark results are sent to the InfluxDB time series database. Grafana is used for convenient access to benchmark results.

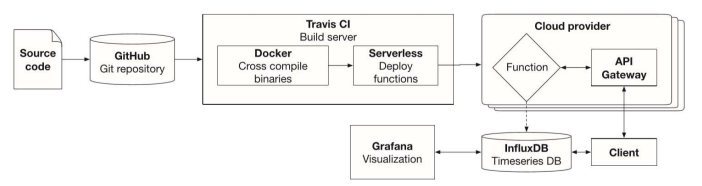


FIGURE 1 Architecture of the cloud functions benchmarking framework based on Serverless Framework

9

9

BENCHMARKING (HYPERFLOW)

- HyperFlow is a lightweight workflow engine based on Node.js, and it can orchestrate complex large-scale scientific workflows, including directed acyclic graphs (DAG).
- The cloud function running on the provider side is a JavaScript wrapper (HyperFlow executor), which runs the actual benchmark, measures the time and sends the results to the cloud storage such as AWS S3 or Google Cloud Storage depending on the cloud provider.

10

10

Slide 9

A11 many words
Author, 12/3/2019

EXPERIMENTAL SETUP

(Integer Based)

Configuration of the serverless benchmarking suite

- Three Parts
  1) Integer-based CPU intensive benchmark
  2) Instance lifetime
  3) Data transfer benchmark

(Floating Point Based)

Configuration of HyperFlow suite

11

11

EXPERIMENTAL SETUP

Configuration of the serverless benchmarking suite

- Three Parts
  1) Integer-based CPU intensive benchmark

  Used a random number generator A12

  Why?

  -used in many scientific applications like Monte-Carlo methods

  -Mersenne Twister (MT19937) random number generator algorithm

  -Benchmark runs approx. 16.7M iterations

Configuration of HyperFlow suite

12

12

Slide 12

A12 did they just generate random numbers?
Author, 12/3/2019

EXPERIMENTAL SETUP

Configuration of the serverless benchmarking suite

- Three Parts
  1) Integer-based CPU intensive benchmark
  2) Instance lifetime
  - Providers reuse the same execution environment to process subsequent requests
  - Assign a global variable with the timer value when the execution environment was started
  - Return the time elapsed with every request

Configuration of HyperFlow suite

13

13

EXPERIMENTAL SETUP

Configuration of the serverless benchmarking suite

- Three Parts
  1) Integer-based CPU intensive benchmark
  2) Instance lifetime
  3) Data transfer benchmark
  - Measure time required to download and upload 64 MB file from object storage
  - **Why 64 MB?**
  - To keep the transfer time between 1 to 30 sec where transfer rate dominates the latency

Configuration of HyperFlow suite

14

14

EXPERIMENTAL SETUP

Configuration of the serverless benchmarking suite

- Three Parts
  1) Integer-based CPU intensive benchmark
  2) Instance lifetime
  3) Data transfer benchmark

Configuration of HyperFlow suite

- Used the HPL Linpack, the most popular CPU-intensive benchmark focusing on the floating point performance
- Solves a dense linear system of equations in double precision
- Solves for s (number of equations): {1000,1500,…15000}
- Need minimum 128MB to run all the cases
- benchmark stops when it cannot allocate enough memory

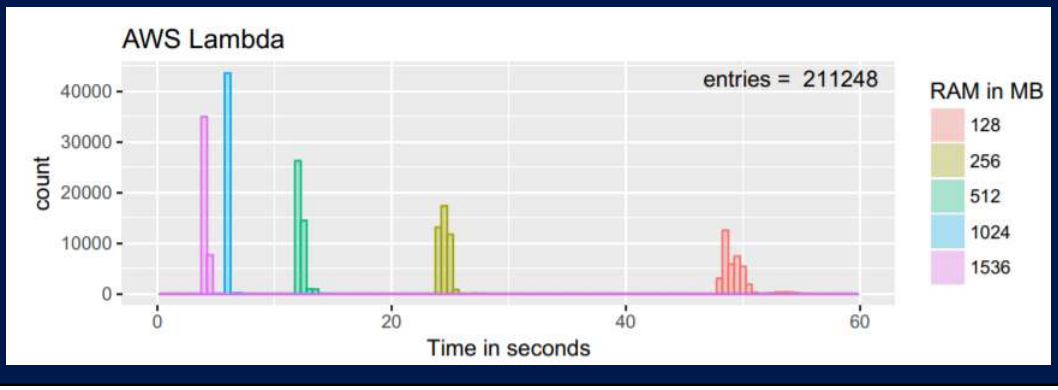
15

15

PERFORMANCE EVALUATION RESULTS

6.1 Integer performance evaluation

performance of **AWS Lambda** is fairly consistent and agrees with the documentation which states that the CPU allocation is proportional to the function size (memory)



16

16

PERFORMANCE EVALUATION RESULTS

6.1 Integer performance evaluation

- **Google Cloud Functions** execution time have bi-modal distributions with higher dispersion
- All the functions with memory smaller than 2048 MB have two peaks
- one around the expected higher values (depending on the memory allocated)
- second peak overlapping with the performance of the fastest 2048 MB function
- This suggests that Google Cloud Functions does not enforce strictly the performance limits and opportunistically invokes smaller functions using faster resources

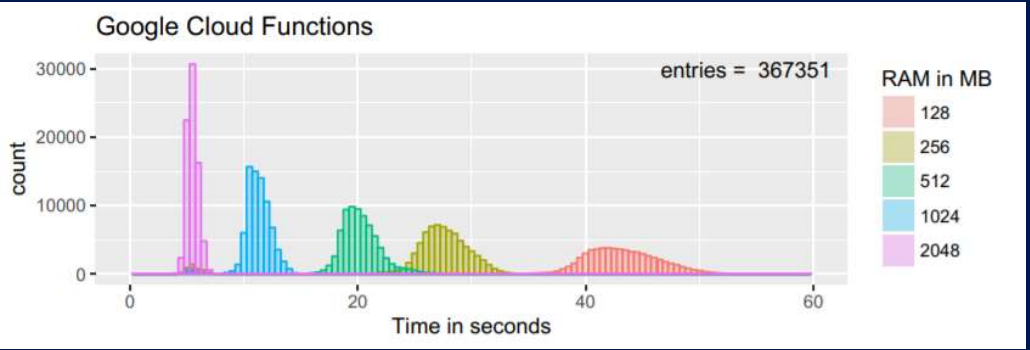
17

17

PERFORMANCE EVALUATION RESULTS

6.1 Integer performance evaluation

- **Google Cloud Functions** execution time have bi-modal distributions with higher dispersion



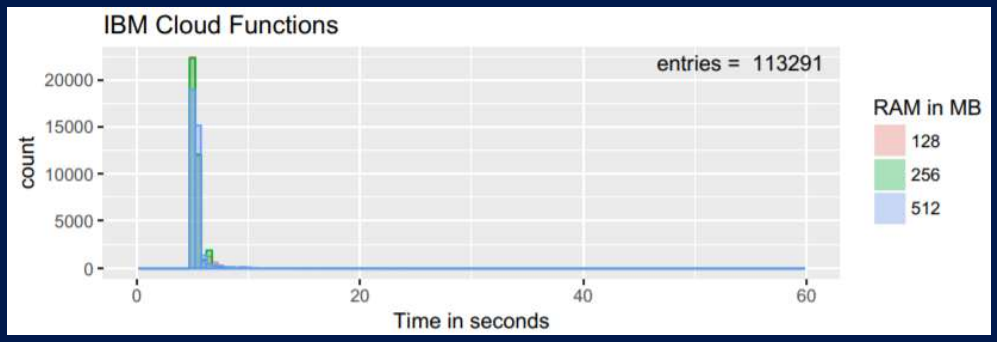
18

18

PERFORMANCE EVALUATION RESULTS

6.1 Integer performance evaluation

- **IBM Cloud Functions**, the performance does not depend on the function size, and the distribution is quite narrow



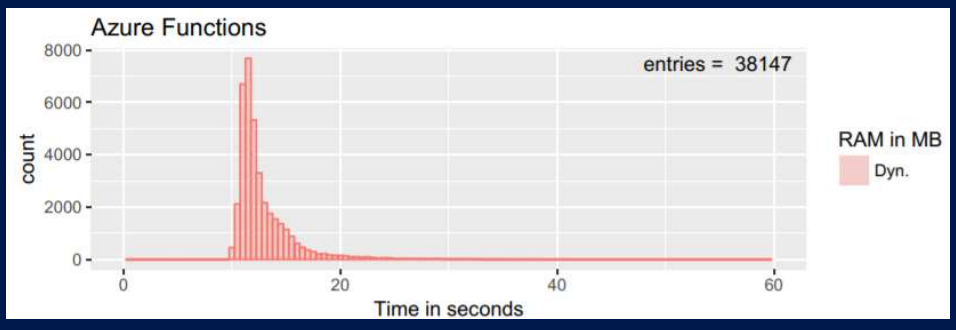
19

19

PERFORMANCE EVALUATION RESULTS

6.1 Integer performance evaluation

- **Azure** has much wider distribution, and the average execution times are relatively slower. This can be attributed to different hardware but also to the underlying operating system (Windows) and virtualization technology



20

20

Slide 19

A13 function size?

do you mean memory size?
Author, 12/3/2019

PERFORMANCE EVALUATION RESULTS

6.2 Floating-point performance evaluation

AWS

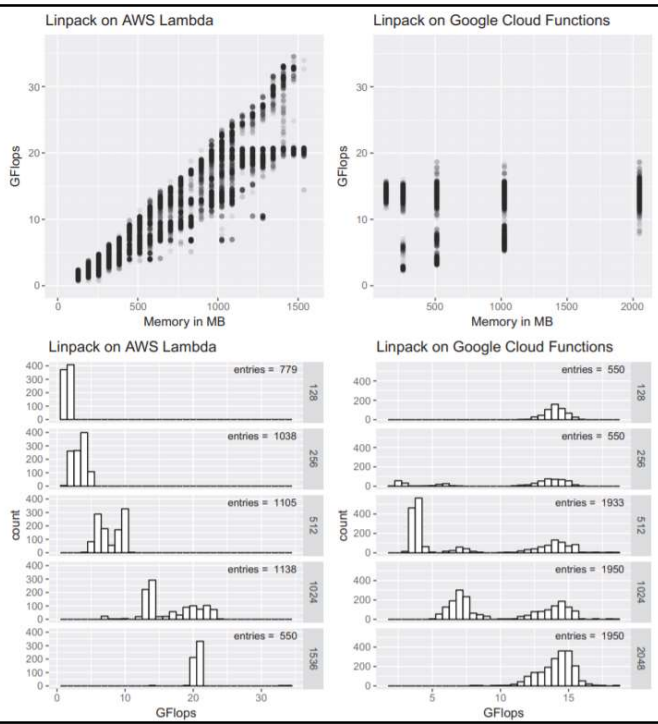
- maximum performance grows linearly with the function size
- With the growing memory, we can see that the execution times form two clusters, one growing linearly over 30 GFlops, and one saturating around 20 GFlops

Google

- The performance of one group of tasks grows linearly with memory
- there is a large group of tasks, which achieve the top performance of 15 GFlops regardless of the function size.

21

21



22

22

PERFORMANCE EVALUATION RESULTS

6.3 File transfer times to/from cloud storage

- File transfer times are proportional to memory allocation
- both the download and upload times depend on the size of the function
- AWS exhibits much shorter data transfer times than Google cloud and also a smaller variance
- In Google we observe similar bi-modal distributions

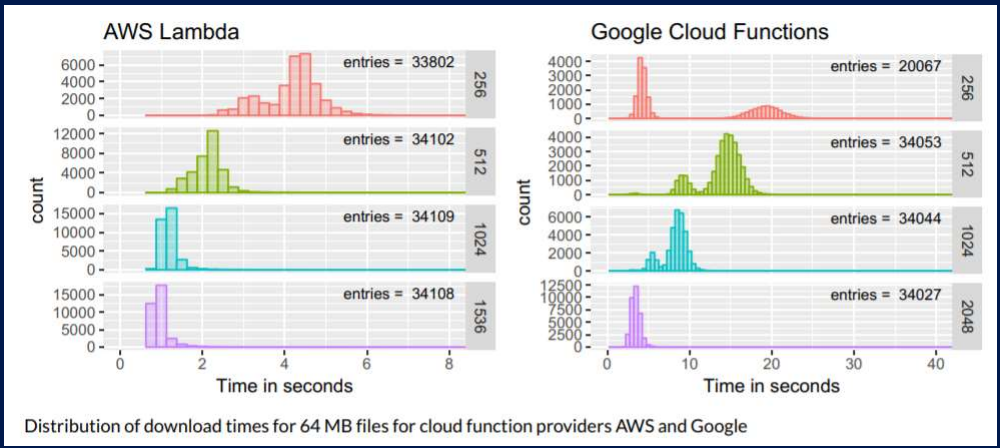
23

23

A14

PERFORMANCE EVALUATION RESULTS

6.3 File transfer times to/from cloud storage



Distribution of download times for 64 MB files for cloud function providers AWS and Google

24

24

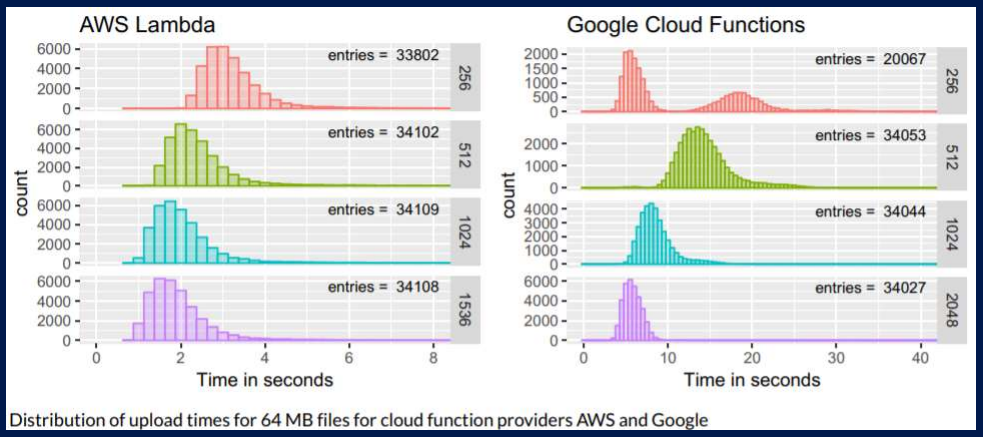
Slide 24

A14 add to slide bullet or title "DOWNLOAD PERFORMANCE"
Author, 12/3/2019

A15

PERFORMANCE EVALUATION RESULTS

6.3 File transfer times to/from cloud storage



Distribution of upload times for 64 MB files for cloud function providers AWS and Google

25

PERFORMANCE EVALUATION RESULTS

6.4 Overheads evaluation

- overhead includes: network latency, platform routing, and scheduling overheads
- total overhead = tr – tb
  - tb – Execution time
  - tr - as seen from the client
- latency is lowest for AWS Lambda
- overhead is stable with a few outliers

26

Slide 25

A15 make clear "UPLOAD PERFORMANCE"
Author, 12/3/2019

PERFORMANCE EVALUATION RESULTS

6.4 Overheads evaluation

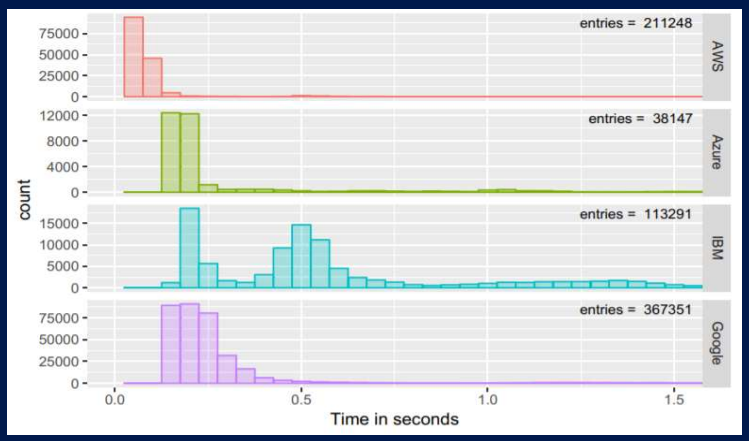
- latency is lowest for AWS Lambda
- overhead is stable with a few outliers
- No correlation between the function size and the overheads

27

27

PERFORMANCE EVALUATION RESULTS

6.4 Overheads evaluation



28

28

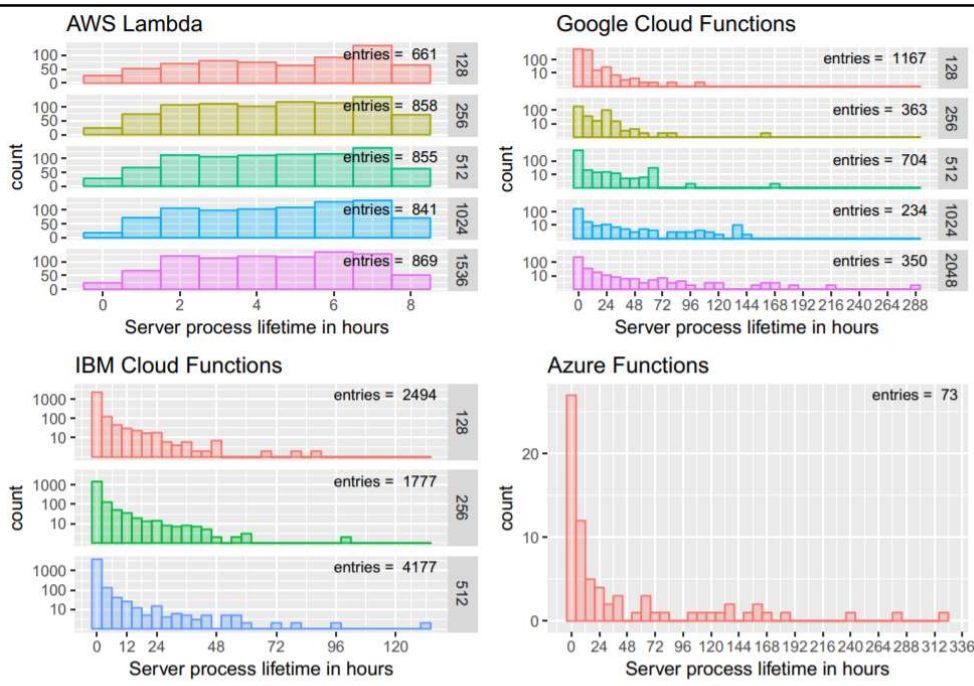
PERFORMANCE EVALUATION RESULTS

6.5 Instance lifetime

- distributions vary between providers
- Azure: the environment process is being preserved of a very long time up to two weeks
- AWS Lambda: the Node.js environment is recycled every few (up to 8) hours
- IBM Cloud Functions recycles execution environment within a few hours
- Google Cloud Functions, environments with low memory allocation are terminated more frequently, while the longer lifetime is being observed for larger allocations.

29

29



A16

30

Distribution of instance lifetime t_i for cloud function providers

30

Slide 30

A16 This is a really cool slide / graph.
I think it would help to have a TITLE or comment in the side bar that describes what the graphs represent.
Author, 12/3/2019

PERFORMANCE EVALUATION RESULTS

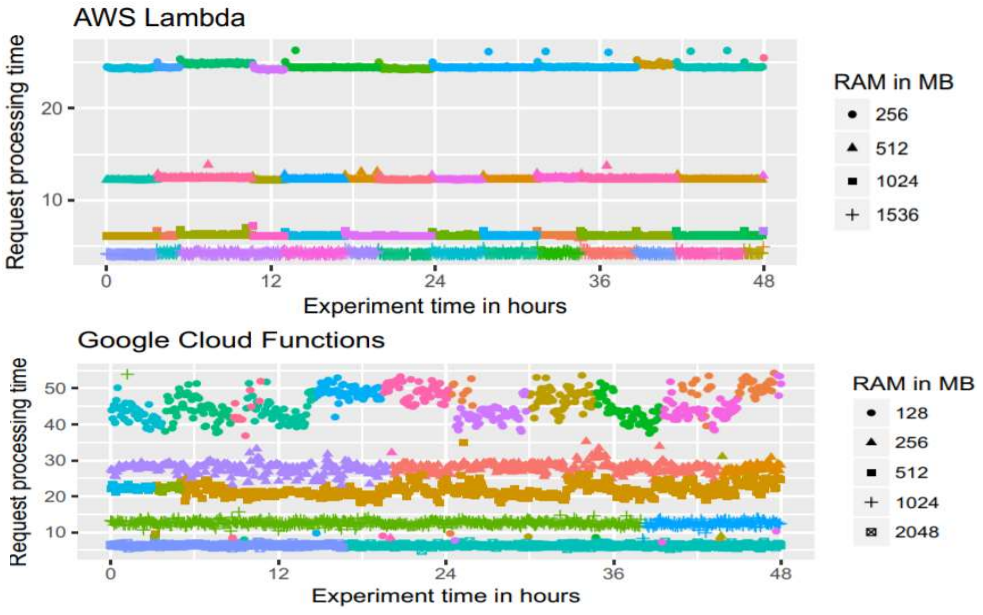
6.6 Cost Comparison

- Price at the time of execution-
  - AWS Lambda- charges $0.00001667 for every GB-second
  - Google Cloud Functions cost $0.0000025 for GB-second plus $0.0000100 for GHz-second
  - Azure Functions cost $0.000016 per GB-second
  - IBM Cloud Function cost $0.000017 per GB-second

31

31

A17
A18



Add a footer

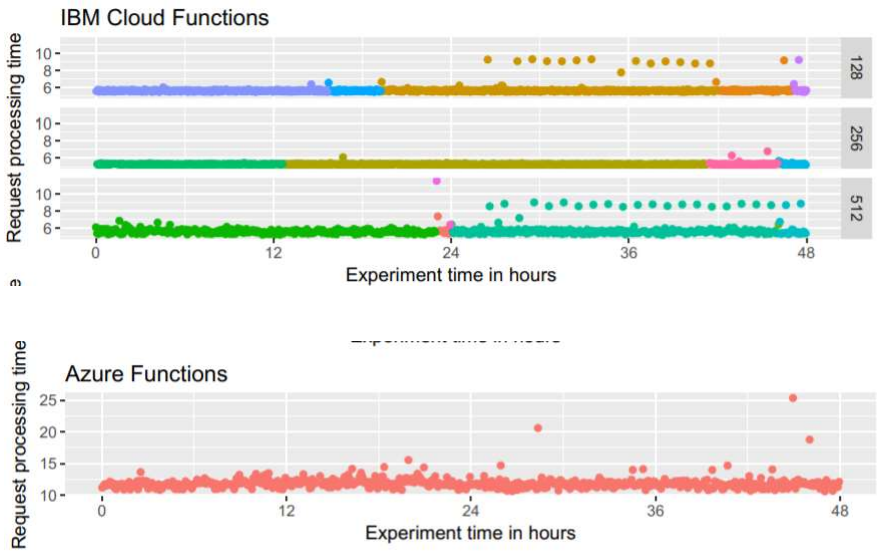
32

32

Slide 32

A17 what do the different colors show?
Author, 12/3/2019

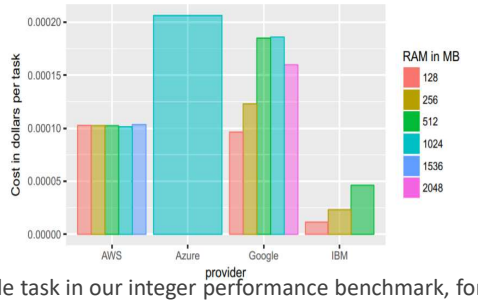
A18 what is the format of these graphs inconsistent?
Author, 12/3/2019



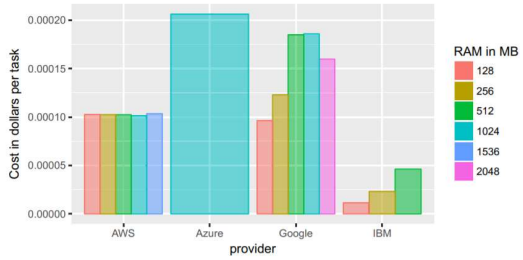
33

A19

- Price for cloud function per 100 millisecond depending on RAM. For Azure, we assumed the cost of 1024 MB



- Costs for execution of single task in our integer performance benchmark, for all cloud function providers depending on RAM. For Azure, we assumed the cost of 1024 MB-



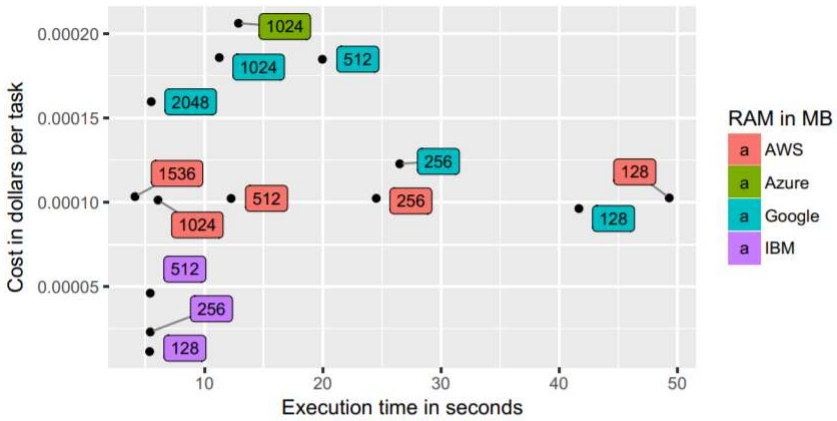
34

Slide 34

A19 I would cut this slide to shorten the presentation
Author, 12/3/2019

A20

- Comparison of cost vs execution time of single task in our integer performance benchmark, for all cloud function providers depending on RAM. For Azure, we assumed the cost of 1024 MB



35

A21

PERFORMANCE EVALUATION RESULTS

6.7 Infrastructure heterogeneity

- Performance Evaluation-
  - AWS
  - Google Cloud Functions
  - Azure Functions
  - IBM Cloud Function

36

36

Slide 35

A20 I would cut this slide to shorten the presentation
Author, 12/3/2019

Slide 36

A21 I would cut this slide to shorten the presentation

The authors results aren't very good on this
Author, 12/3/2019

A23

DISCUSSION ON RESULTS

Hypothesis 1: *Computational performance of a cloud function is proportional to function size*-A22

- AWS Lambda – True
- Google Cloud Functions – True (5% exception cases)
- IBM – False
- Azure – False

37

37

A24

DISCUSSION ON RESULTS

Hypothesis 2: *Network performance (throughput) of a cloud function is proportional to function size*

- AWS Lambda – True
- Google Cloud Functions – True (same restriction as hypothesis 1)
- IBM – Not Measured
- Azure – Not Measured

38

38

Slide 37

A22 do you mean the memory size?
Author, 12/3/2019

A23 Critique: This isn't much of a research question given the answer can be found out by reading the documentation for these platforms.
Author, 12/3/2019

Slide 38

A24 Critique: this is in the AWS documentation. Not sure we need a research paper to answer this question
Author, 12/3/2019

A25

DISCUSSION ON RESULTS

Hypothesis 3: *Overheads do not depend on cloud function size and are consistent for each provider*

- AWS Lambda – Generally True
- Google Cloud Functions – Generally True
- IBM – Generally True
- Azure – Generally True

39

39

A26

DISCUSSION ON RESULTS

Hypothesis 4: *Application server instances are reused between calls and are recycled at regular intervals*

- This was nicely demonstrated by our experiments, and we also observed that the instance lifetime differs between providers.

40

40

Slide 39

A25 this research question requires measurement through experiments
Author, 12/3/2019

Slide 40

A26 this is one of the strongest parts of the paper - they visualize infrastructure llfecycles for the 4 platforms
Author, 12/3/2019

A27

DISCUSSION ON RESULTS

Hypothesis 5: Functions are executed on heterogeneous hardware

- AWS Lambda – True
- Google Cloud Functions – Not Measured
- IBM – True
- Azure – Not Measured

41

41

A28

ADDITIONAL TAKEAWAYS

- The specific resource allocation policies as these of Google or different variances of the results must be taken into account when making decisions about choosing the provider and the function size.
- The price/performance analysis needs to be carefully performed to avoid unnecessary costs
- Heterogeneity exists at multiple level. Many Cloud service provider use Linux for the hosting Operating System, whereas Azure provides Windows. Many providers also have different versions of Node, though it is platform agnostic.
- AWS Lambda performance is directly proportional to the memory allocated by a user, except sometimes it is a bit slower, whereas in case of Google Cloud function, the performance is slightly better for the exceptional cases. This result is also valid for data transfer time.
- In case of AWS Lambda, for CPU intensive applications, using larger functions is more economical, since the price is same, but results are obtained much faster than using slower function.
- From the cost perspective, IBM is better than others, whereas AWS is the fastest instance.

42

42

Slide 41

A27 critique: somewhat of a weak evaluation in the paper
Author, 12/3/2019

Slide 42

A28 The font size is very small

There is a lot of text

Please use short phrases
Author, 12/3/2019

A29

CRITIQUE: STRENGTHS

- This is a very well drafted paper that follows a hypothesis driven dive deep
- The authors also used two frameworks for benchmarking. One, a new suite designed specifically for this research while the other one, an existing suite that has already been used to run preliminary experiments on cloud functions.
- The paper is well explained and clearly call out any gaps in performance evaluation
- The paper gives technical details on how we address the heterogeneity of the environment, and describes our automated data taking pipeline

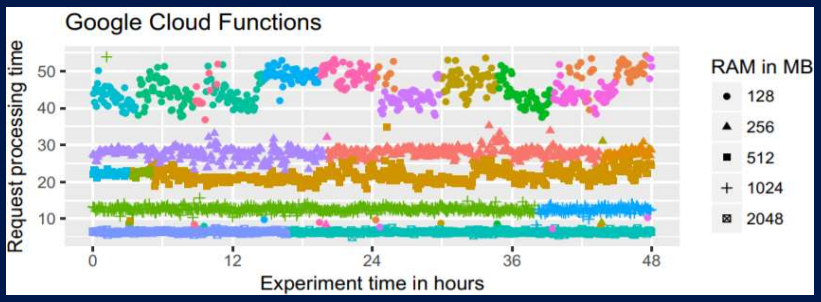
43

43

A30

CRITIQUE: WEAKNESSES

- More measurements can be added to the benchmarking suite
- The paper fails to address the cause of the exceptions observed in the behavior of function performances
- The pictorial representation of "Processing time vs Experiment time" is confusing.



44

44

Slide 43

A29 The font size is very small

There is a lot of text

Please use short phrases
Author, 12/3/2019

Slide 44

A30 the diagram show infrastructure life time. Each color represents one set of serverless infrastructure. When that infrastructure is deprecated it is replaced with the next set of serverless infrastructure.

Apparently for google, the memory size of the function impacts how quickly infrastructure is recycled. Functions with 128MB are recycled after ~4 hours, whereas functions with 2048MB are recycled after ~18 hours ??
Author, 12/3/2019

IDENTIFIED GAPS

- For the third and fifth hypothesis, the authors were able to get results for only two providers.
- Hypothesis are generally confirmed with some deviations from the expected behavior

45

45

THANK YOU

46

46