university of groningen

Using a Microbenchmark to Compare Function as a Service (FaaS) Solutions

Timon Back & Vasilios Andrikopoulos

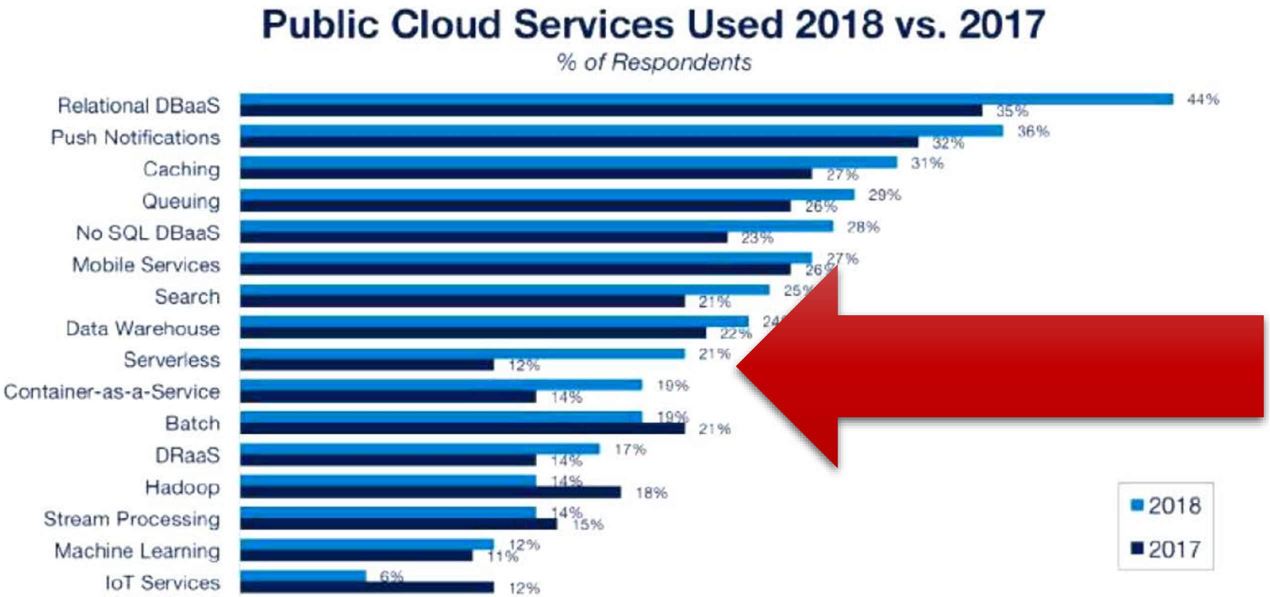
ESOCC, September 2018

Some terminology

- Serverless computing model
  - Code executed without any control on the resources on which the code runs

- Function as a Service (FaaS)
  - Similar to PaaS but finer granularity
  - Scaling on the level of functions
  - Event-driven

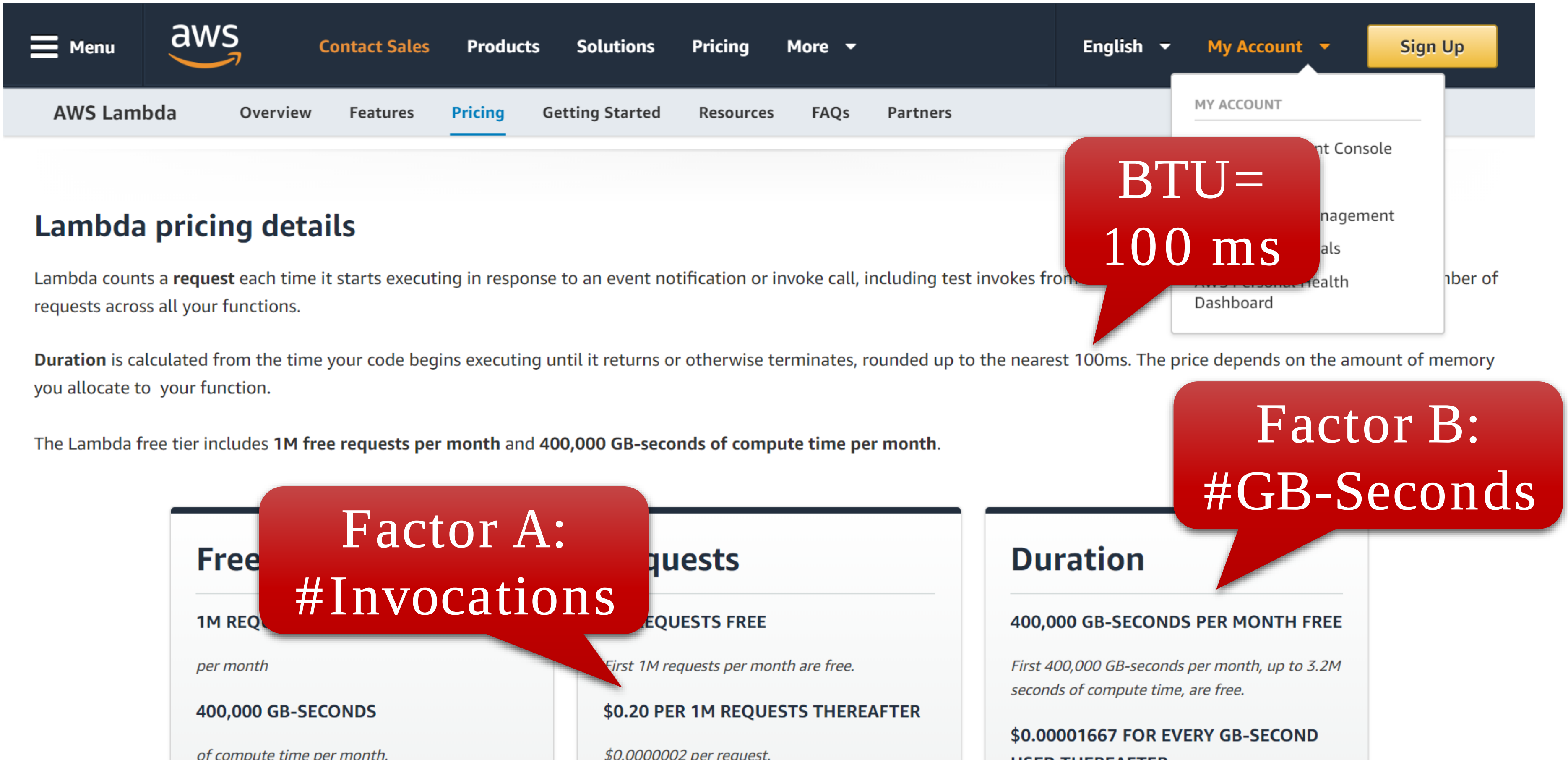
Industry adoption



Source: RightScale 2018 State of the Cloud Report

“Year over year, serverless was the top-growing extended cloud service with a 75 percent increase over 2017 (12 to 21 percent adoption)”

FaaS Pricing Model Peculiarities



Lambda pricing details

Lambda counts a **request** each time it starts executing in response to an event notification or invoke call, including test invokes fro[m] … [nu]mber of requests across all your functions.

Duration is calculated from the time your code begins executing until it returns or otherwise terminates, rounded up to the nearest 100ms. The price depends on the amount of memory you allocate to your function.

The Lambda free tier includes **1M free requests per month** and **400,000 GB-seconds of compute time per month**.

Free

1M REQ

per month

400,000 GB-SECONDS

of compute time per month.

quests

EQUESTS FREE

First 1M requests per month are free.

$0.20 PER 1M REQUESTS THEREAFTER

$0.0000002 per request.

Duration

400,000 GB-SECONDS PER MONTH FREE

First 400,000 GB-seconds per month, up to 3.2M seconds of compute time, are free.

$0.00001667 FOR EVERY GB-SECOND USED THEREAFTER

BTU= 100 ms

Factor B: #GB-Seconds

Factor A: #Invocations

Key challenges

1. How do (public cloud) FaaS perform with respect to each other?

2. How to estimate the elusive GB-second?

FaaS microbenchmark

- (Micro)benchmarking an acceptable practice for comparing public cloud providers (Li et al. 2013)
- Existing benchmarks aimed at coarser granularity see for example Malawksi et al. 2018
- Publicly available https://github.com/timonback/faas-mubenchmark

Design & Implementation

- Functions with parameters ranging over discrete domains with known memory/processing demands
  - FFT
  - Matrix Multiplication (MM)
  - Sleep (S), and others

- Implemented in Node.js as LCD

- Builds on the *serverless* framework for instrumentation purposes

- Measured data as reported by provider-side event logs
  - Datasets available on the same Git repo as code

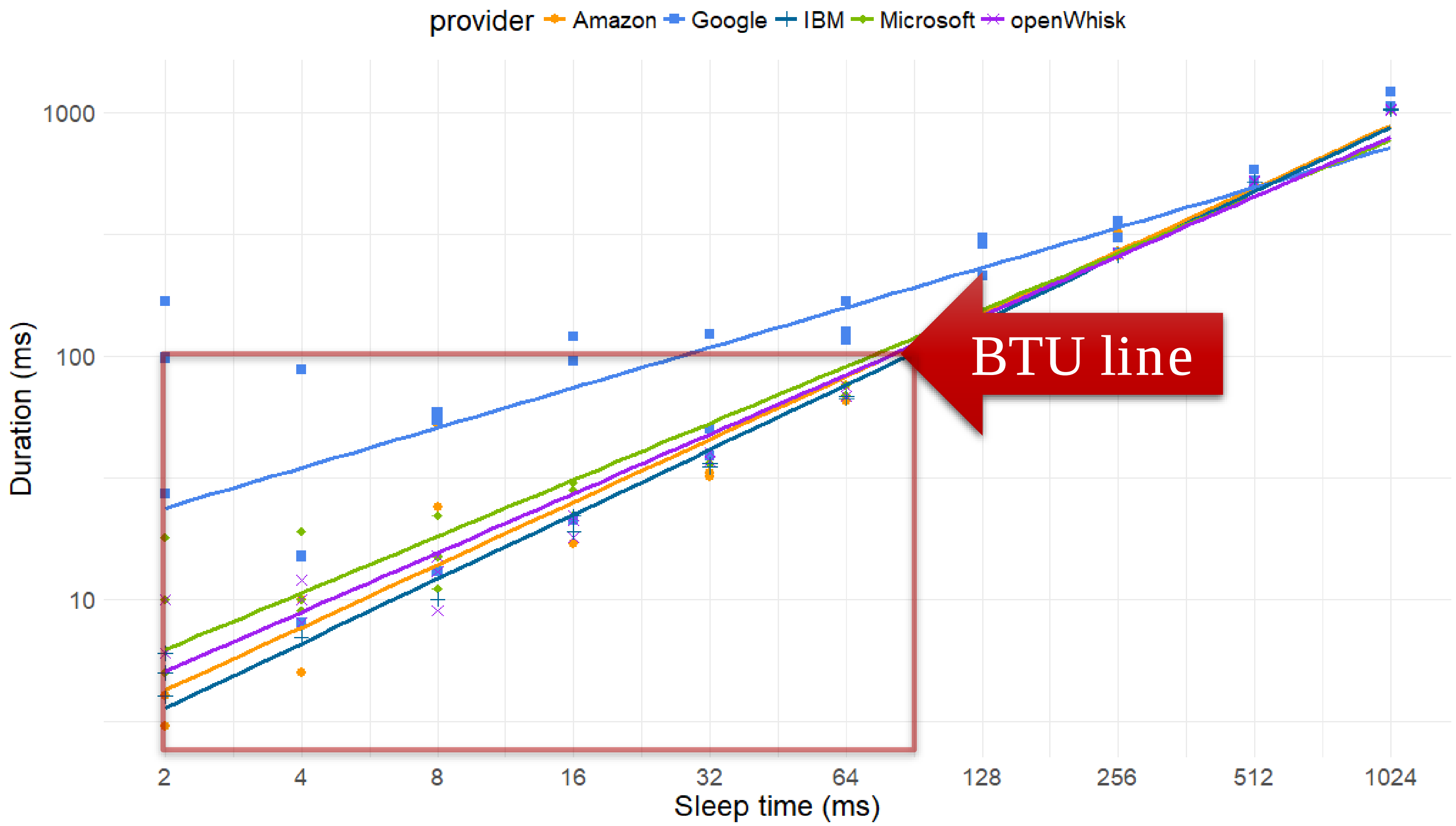
Service Comparison Setup

- Apache OpenWhisk (local deployment) as the baseline for comparisons
  - Forms the basis of IBM Cloud Functions

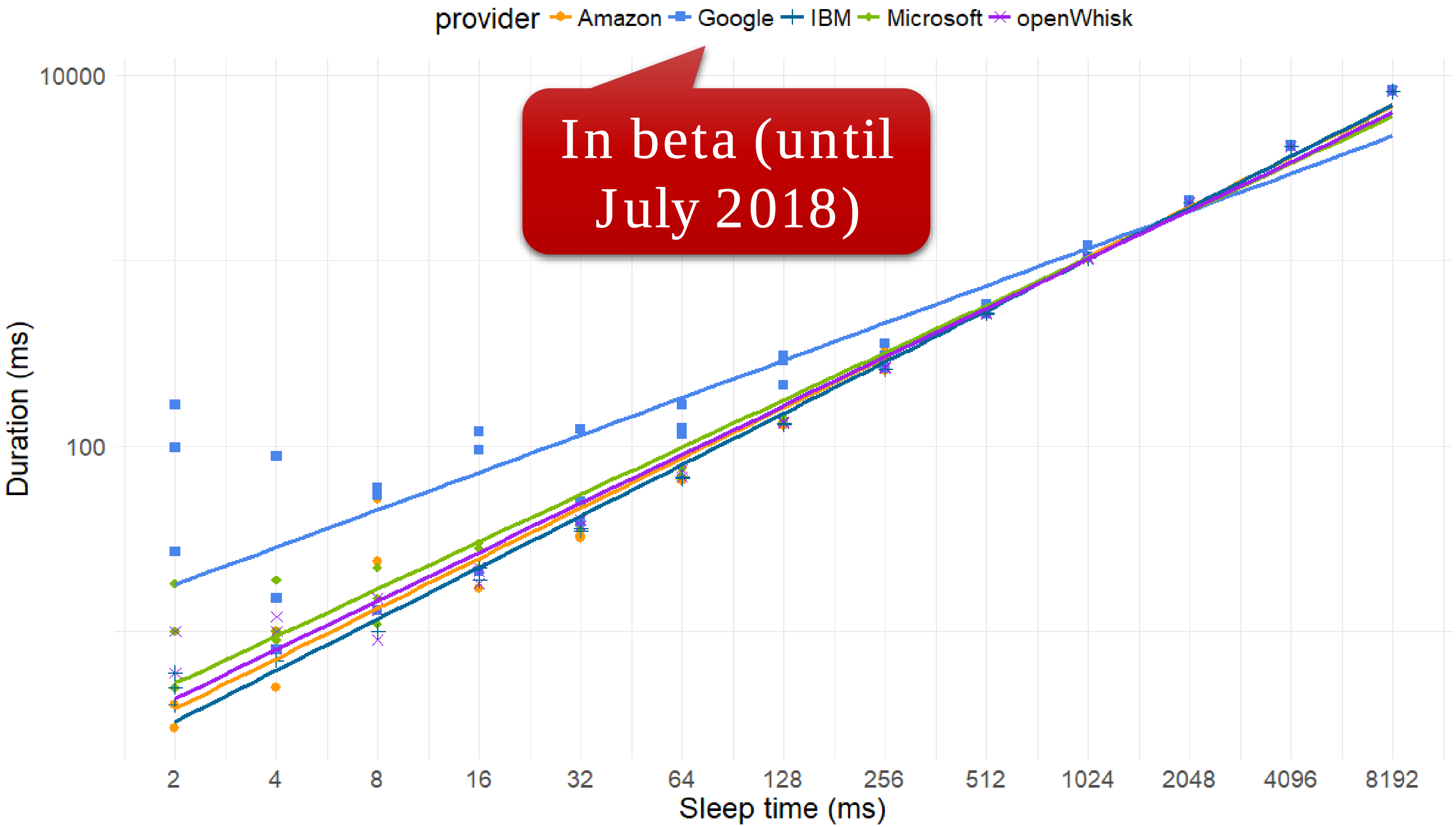
- Compared providers:
  - AWS Lambda, Google Cloud Functions (Beta), Microsoft Azure Functions, IBM Cloud Functions
  - Free tier services used only

- Allocated memory: 128, 256, 512, 1024, and 2048 MB for all functions and all* services

Finding 1: Beware the sub-BTU variability

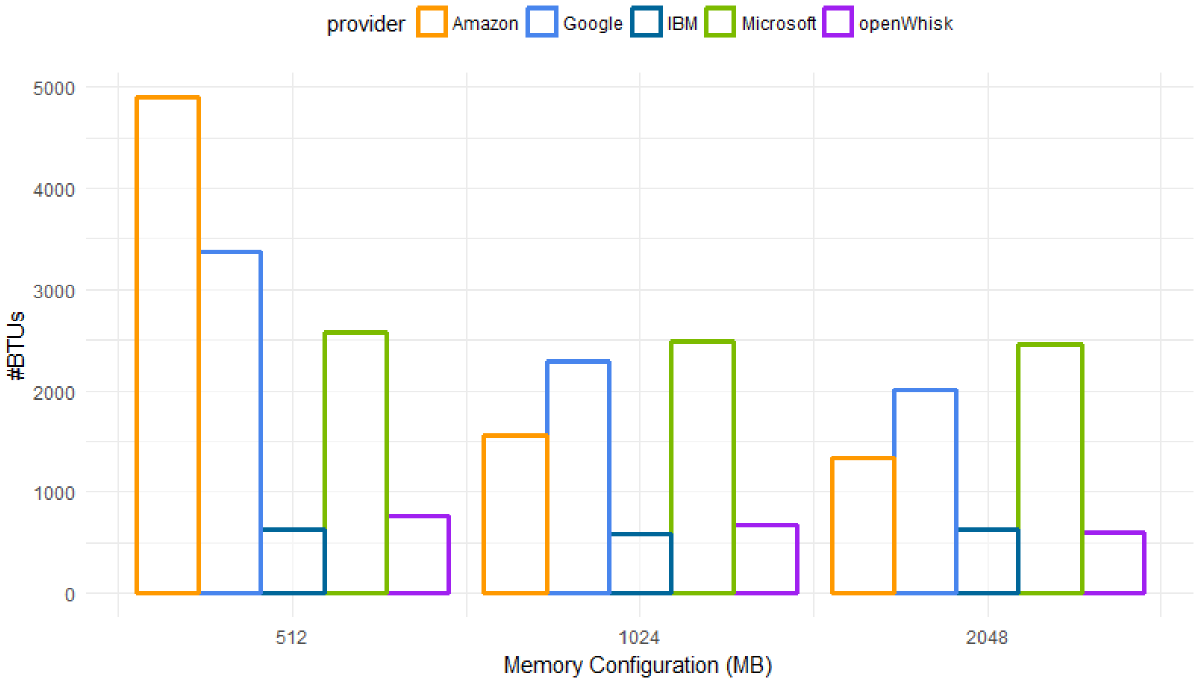


Finding 1: Beware the sub-BTU variability

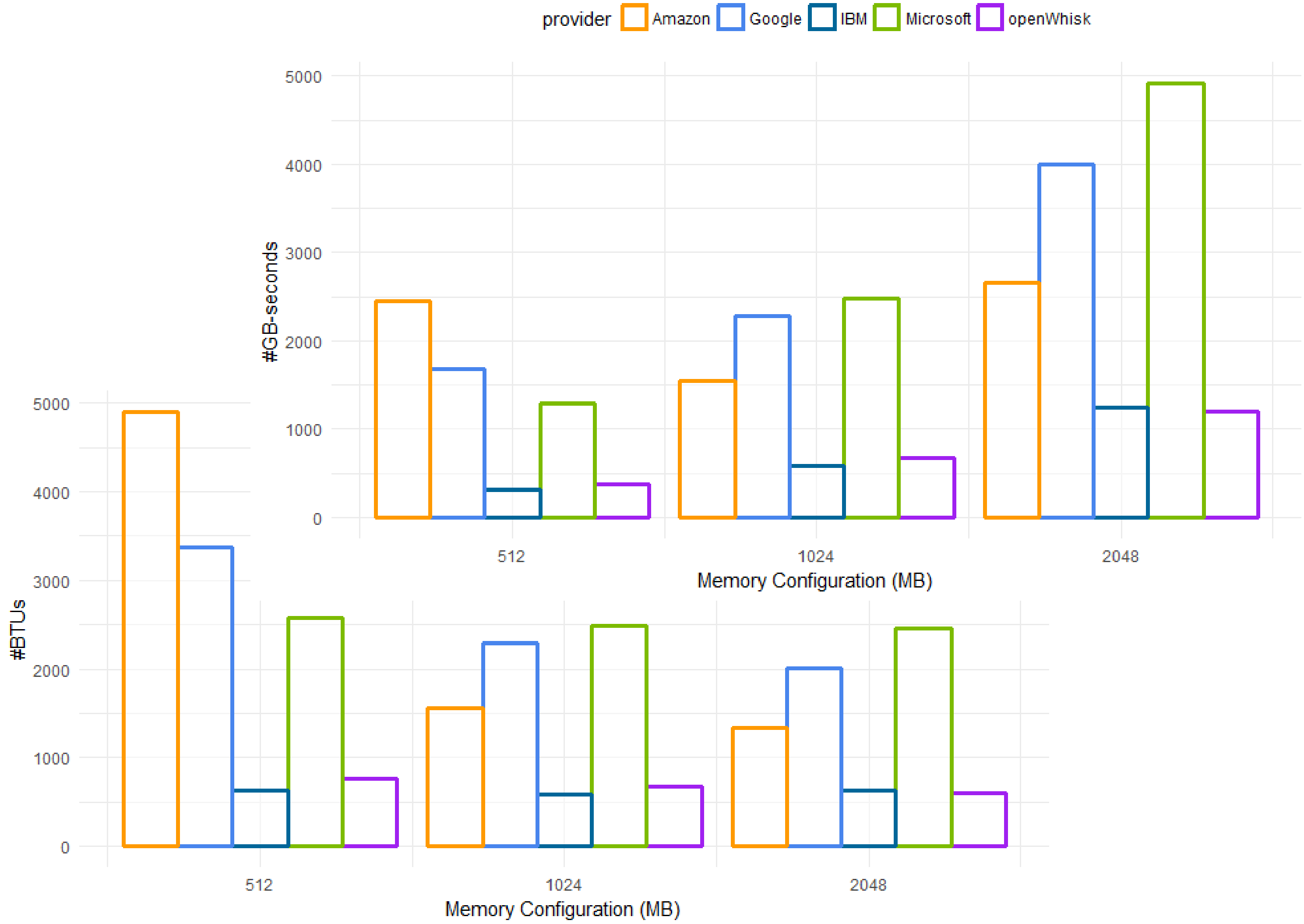


In beta (until July 2018)

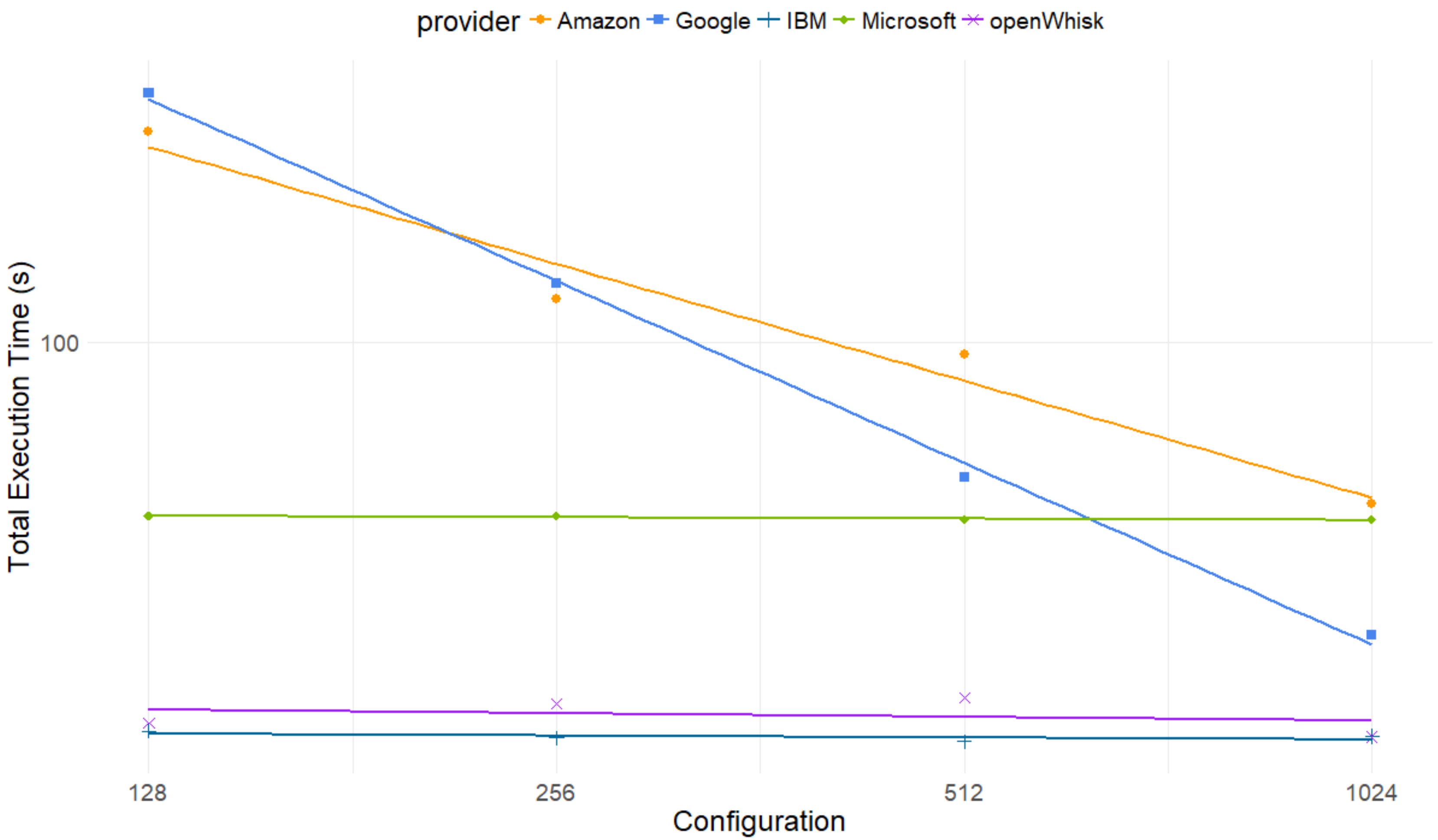
Finding 2: Your provider mileage may vary



Finding 2: Your provider mileage may vary

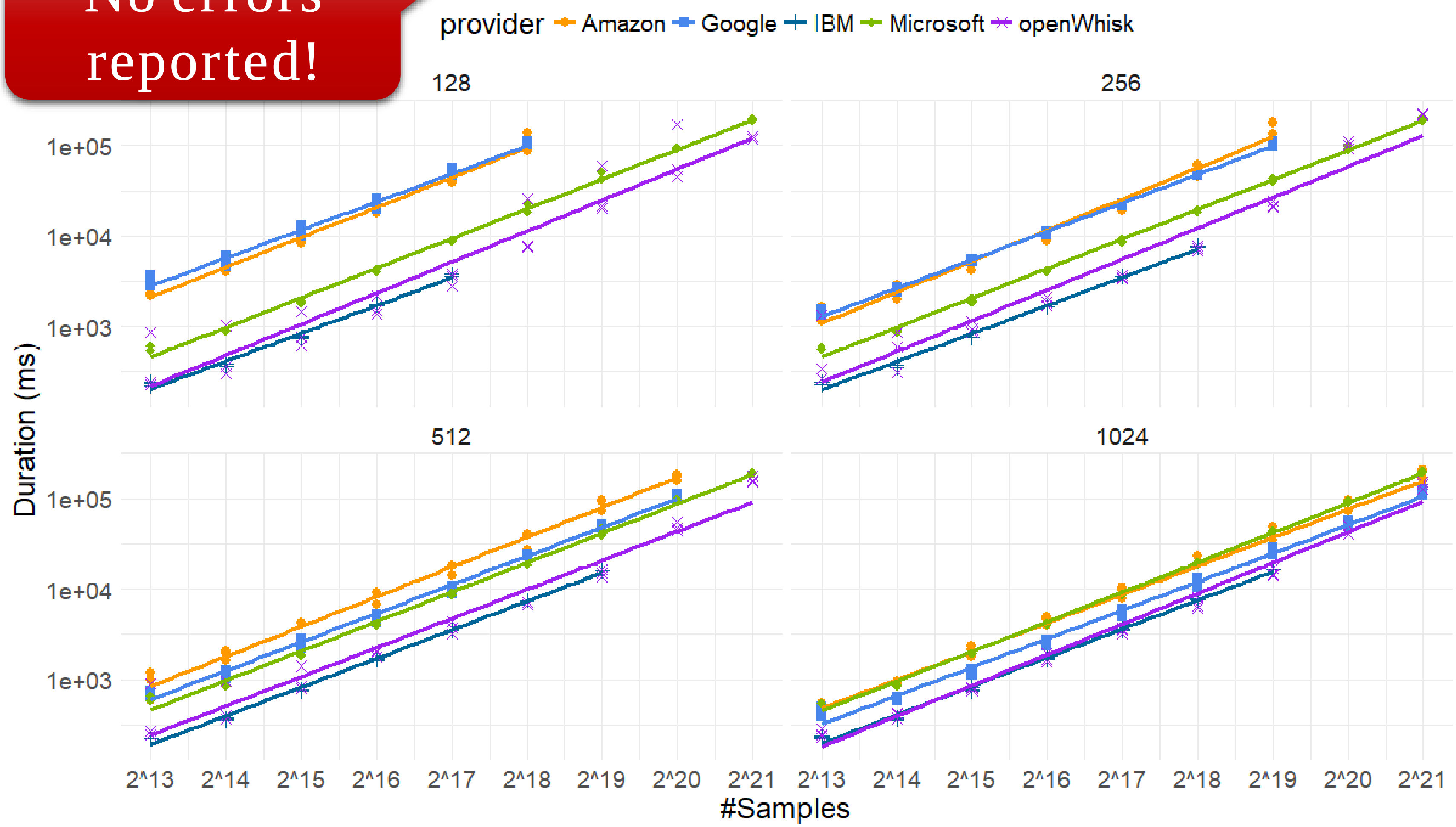


Finding 3: More memory, faster execution*

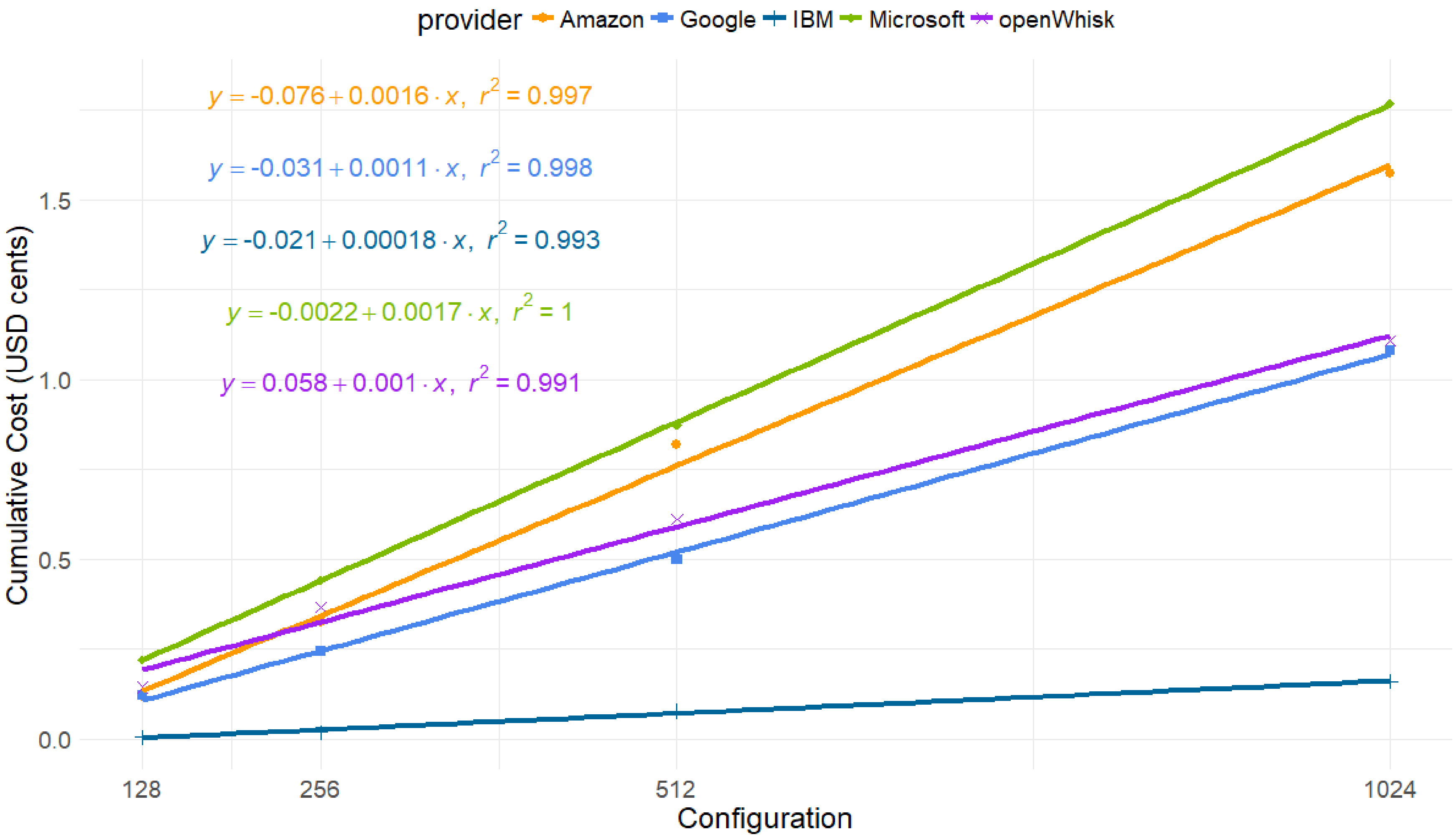


Finding 4: OoM causes abrupt termination

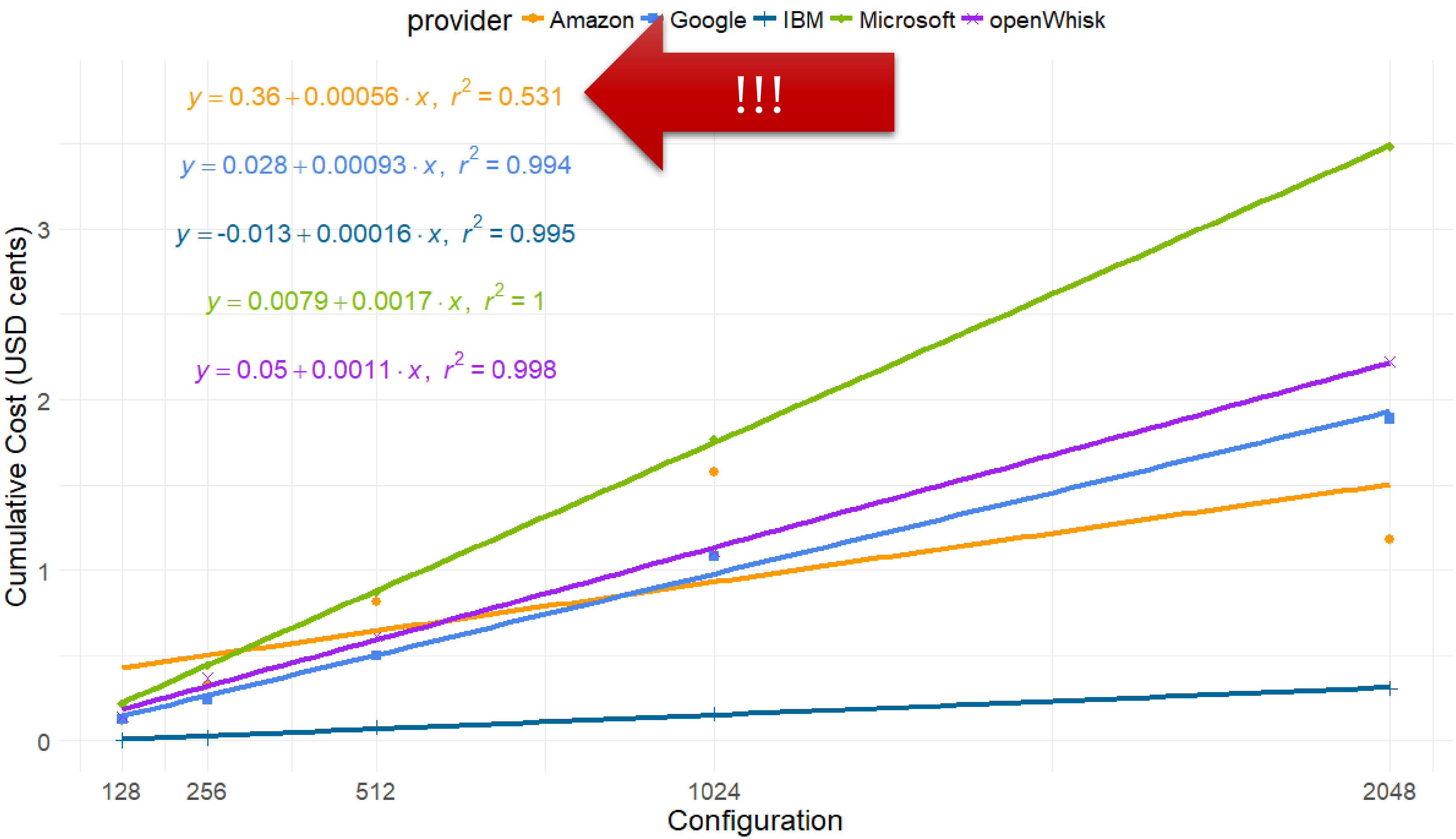
No errors reported!



Finding 5: The devil is in the coefficients



Finding 5: The devil is in the coefficients



!!!

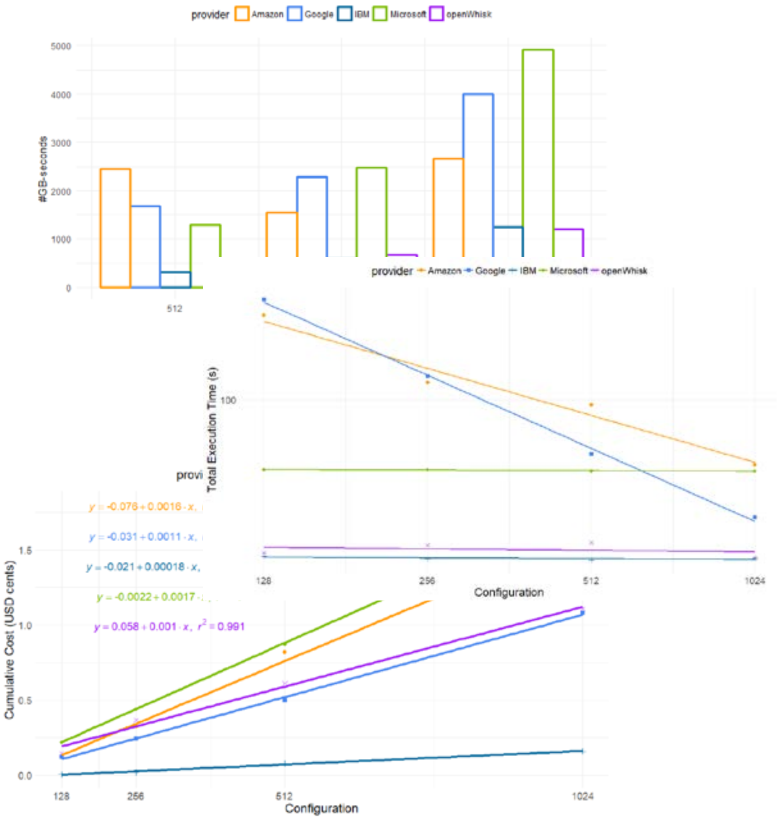
Some extra findings

- “Hockey stick” behavior for short living functions with CPU-biased load (π calculation)
  - CPU cycles/memory mapping only kicks in after enough stress to the function

- Dynamic allocation suffers under memory-biased loads (union-find algorithm)
  - Favors providers like Amazon & Google’s FaaS

Conclusion

- Microbenchmarking as a viable & efficient instrument
- Big differences between providers
- Function-specific benchmarking is required for “safe” results



- Future work
  - Decision model for FaaS adoption/bursting
  - Middleware implementing this model

Reach me at:

v.andrikopoulos@rug.nl

https://vandriko.github.io

@v_andrikopoulos