Profiling Functions as a Service (FaaS)

Andy Chen and Andrew Huang

Introduction

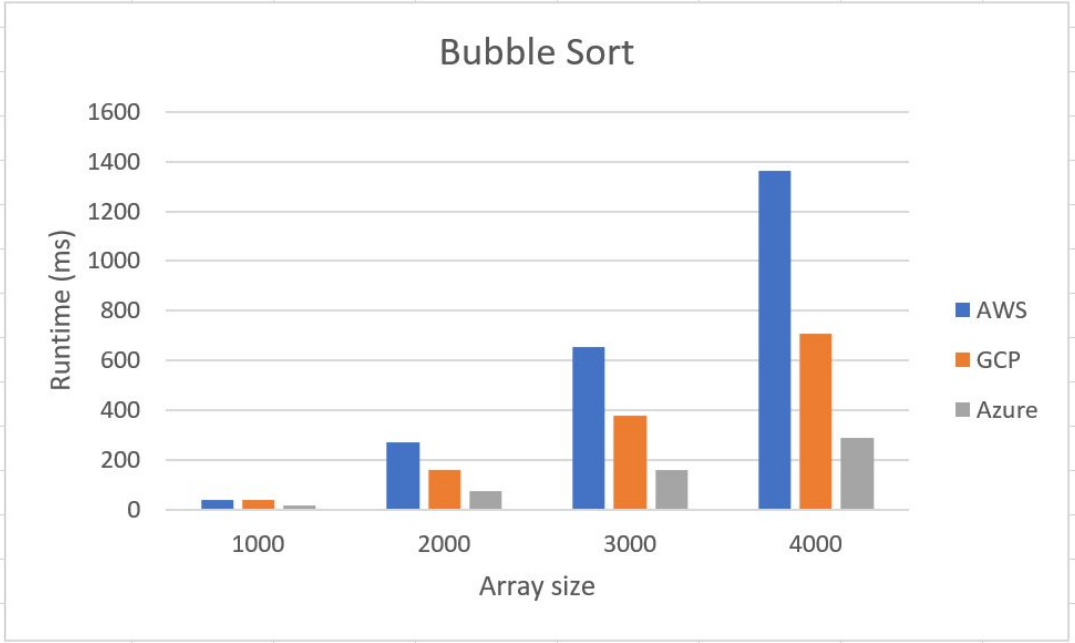
- Many FaaS platforms: AWS Lambdas, GCP Functions, and Azure Functions
- Benchmark payload sizes, computing capabilities, and performed load-testing

Contributions

- Compared AWS, GCP, and Azure
  - Measured **runtime of the method**, peak memory usage, average response time, and **request per seconds**
  - Created Node.js benchmarks: Spectral Norm, Bubble Sort, and Calculating Pi
- Analyzed performance of different ways to trigger AWS Lambda from Azure
  - Exploring options to prevent cloud lock in

Benchmark #1: **Bubble Sort**

- Send arrays of varying size encoded in json via POST request
- Testing how well platforms handle data load & memory usage



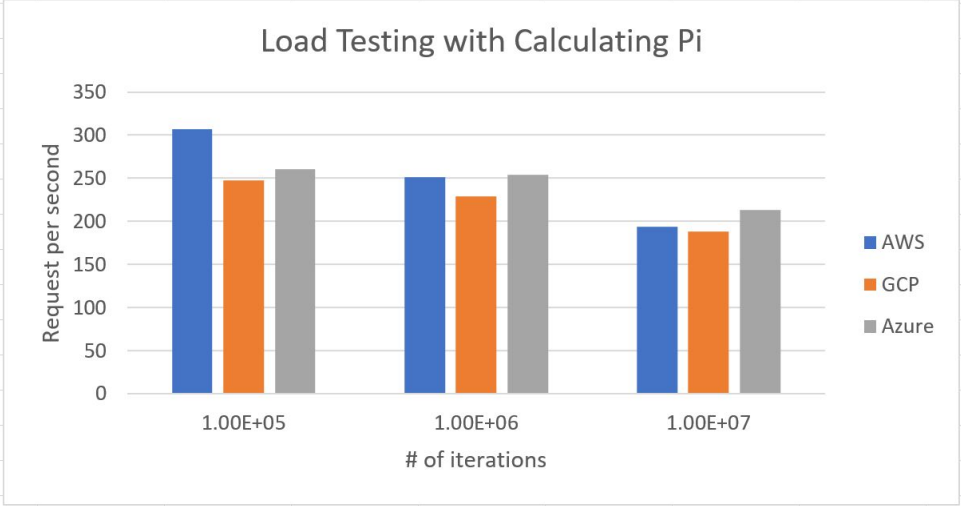
Benchmark #2: **Calculating Pi**

- Calculates Pi for a certain amount of for loop iterations using Leibniz formula
- Testing computation capabilities of platforms

Iterations	AWS	GCP	Azure
1.00E+05	0.6 ms	8.08 ms	2.22 ms
1.00E+06	24.6 ms	29.4 ms	15.9 ms
1.00E+07	603 ms	315 ms	167 ms
1.00E+08	6523 ms	2379 ms	1759 ms

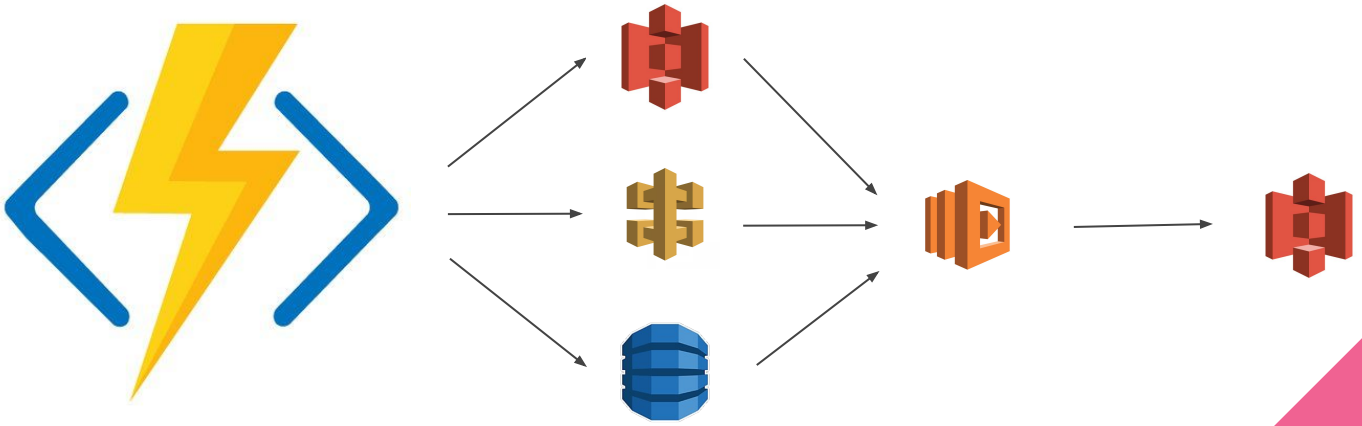
Benchmark #3: **Load testing**

- Gradual build up: 2 users per second, up to 500 users
- Tested with “Calculating Pi” benchmark



Cross Cloud Function Trigger

- S3 Trigger
- API Gateway Trigger
- DynamoDB Trigger



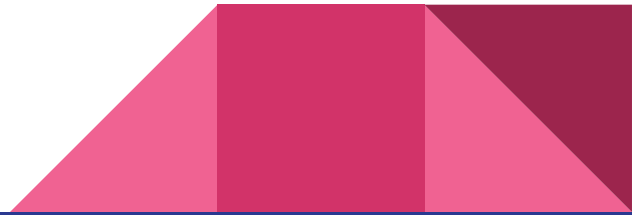
Results

- Test time (s) for single round trip
  - Average of 50 trips
- **Data:** small - text, large - picture
- DynamoDB performed the best

Trigger	Small	Large
S3	15.5 s	38.2 s
API Gateway	7.2 s	22.3 s
DynamoDB	6.8 s	-

Conclusion

- Azure outscales, best option for computationally heavy invocations
- AWS is a great option for small program because of its easy to use interface and performed best in load testing
- Triggering AWS lambda through API Gateway or DynamoDB
- **Future Work:** more extensive load testing and finding more metrics such as total memory usage or more accurate CPU utilization



Questions?