# AWS vs GCP vs Multi Cloud on Large I/O Workload

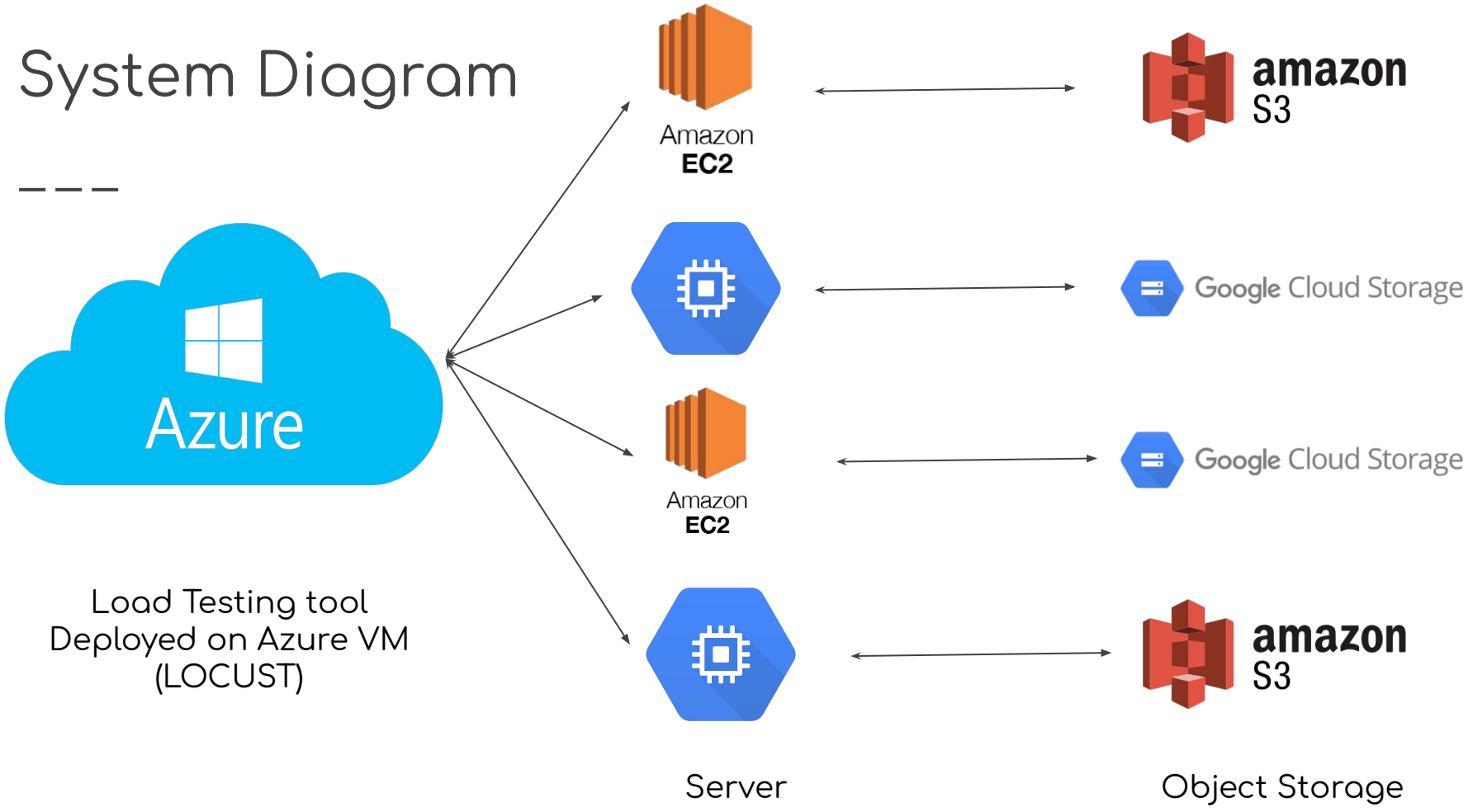
Group 15

# Problem

---

- Load Testing tool deployed on Azure (large and small workloads)
- Instance set 1: server on EC2 connect to S3
- Instance set 2: server on Google Compute Engine connect to Google Cloud Storage
- Instance set 3: server on EC2 connect to Google Cloud Storage
- Instance set 4: server on Google Compute Engine connect to S3
- Same OS, Same web server software, Same web app code (your own) on both clouds
- Comparing response time and throughput on client request/sec

# System Diagram



# Instance Detail

---

EC2:

Platform: Ubuntu 20.04 LTS, Type: t2.micro

vCPUs: 1, Memory(GiB): 1

GCP:

Platform: Ubuntu 20.04 LTS, Type: e2-micro

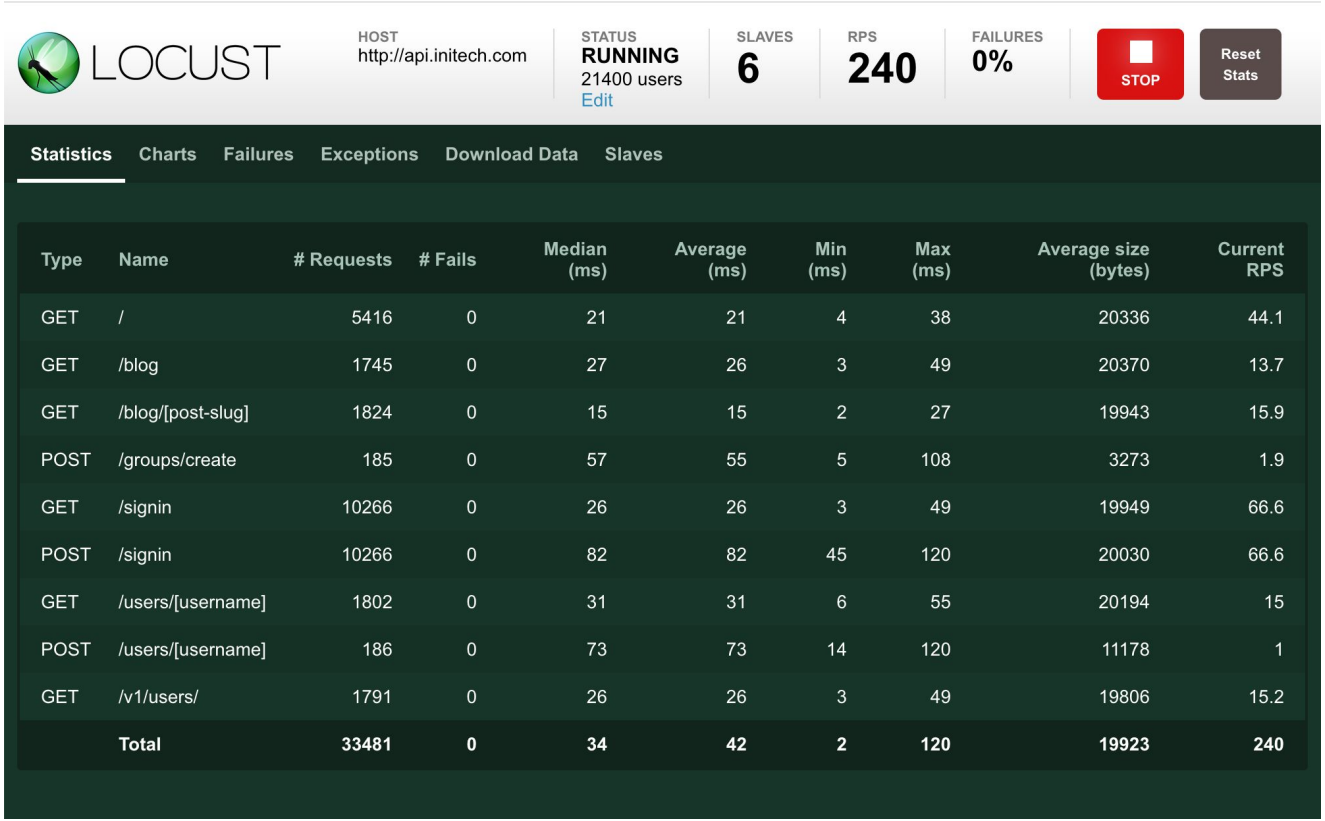
vCPUs: 2, Memory(GiB): 1

Azure:

Platform: Ubuntu 18.04, Type: Standard B1ls

vCPUs: 1, Memory(GiB): 0.5

# Load Testing Tool




| Type | Name | # Requests | # Fails | Median (ms) | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | Current RPS |
|---|---|---|---|---|---|---|---|---|---|
| GET | / | 5416 | 0 | 21 | 21 | 4 | 38 | 20336 | 44.1 |
| GET | /blog | 1745 | 0 | 27 | 26 | 3 | 49 | 20370 | 13.7 |
| GET | /blog/[post-slug] | 1824 | 0 | 15 | 15 | 2 | 27 | 19943 | 15.9 |
| POST | /groups/create | 185 | 0 | 57 | 55 | 5 | 108 | 3273 | 1.9 |
| GET | /signin | 10266 | 0 | 26 | 26 | 3 | 49 | 19949 | 66.6 |
| POST | /signin | 10266 | 0 | 82 | 82 | 45 | 120 | 20030 | 66.6 |
| GET | /users/[username] | 1802 | 0 | 31 | 31 | 6 | 55 | 20194 | 15 |
| POST | /users/[username] | 186 | 0 | 73 | 73 | 14 | 120 | 11178 | 1 |
| GET | /v1/users/ | 1791 | 0 | 26 | 26 | 3 | 49 | 19806 | 15.2 |
| | **Total** | **33481** | **0** | **34** | **42** | **2** | **120** | **19923** | **240** |

# Testing Scenario (Requests from Azure Client)

---

1. New file upload
    - File 10kB
    - File 10MB
2. Update file
    - File 10kB
    - File 10MB
3. Read file
    - File 10kB
    - File 10MB

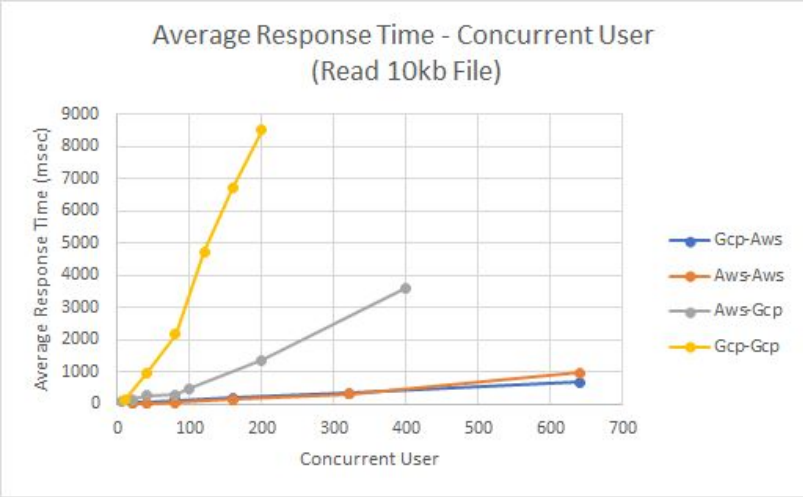
# Pre-Test Question

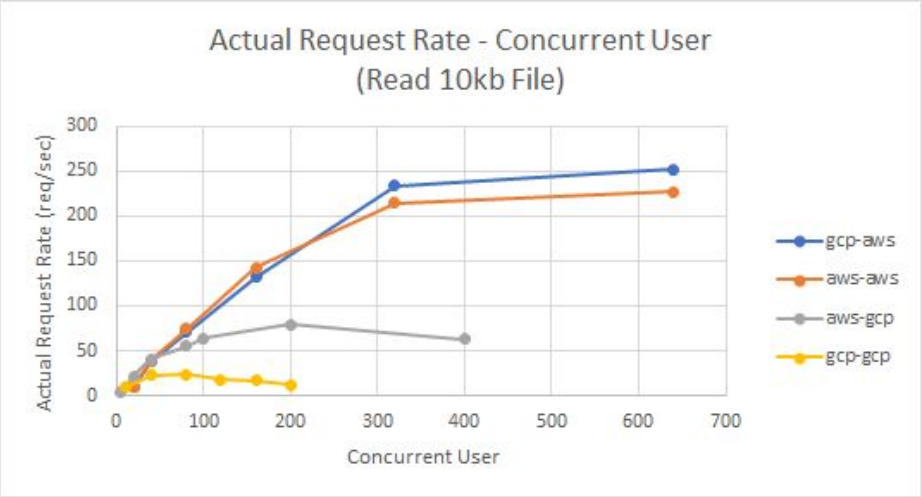
---

Which one do you think will perform better ?

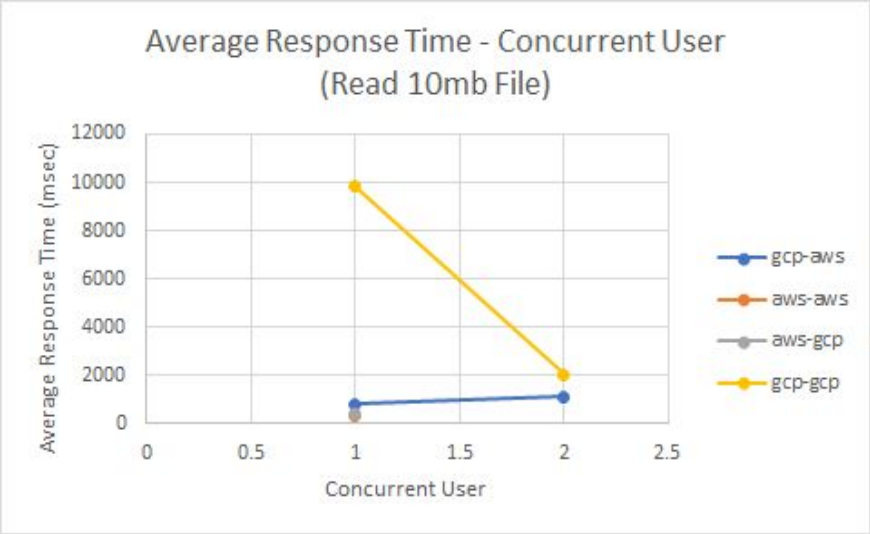
a. AWS EC2 connect to Google Cloud Storage
b. Google Cloud Engine connect to AWS S3

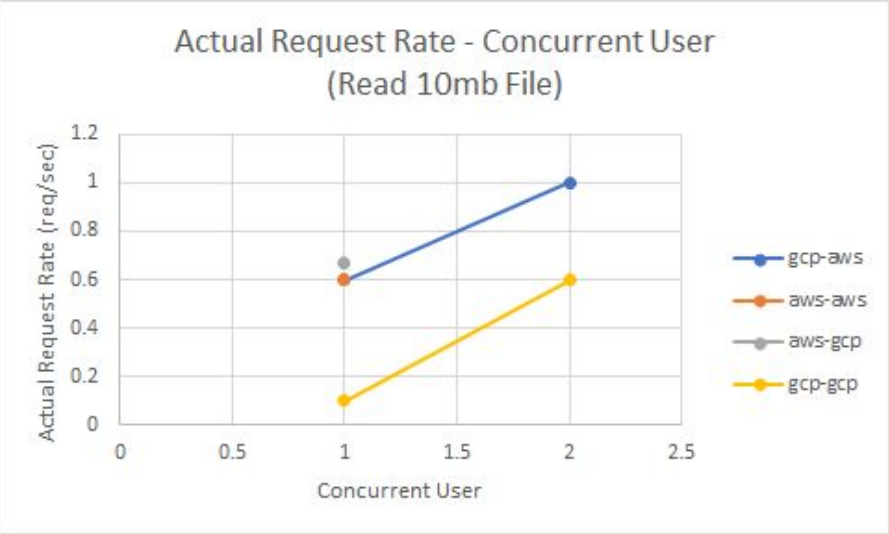
# Result


Average Response Time - Concurrent User
(Read 10kb File)
Average Response Time (msec)
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
0
100
200
300
400
500
600
700
Concurrent User
Gcp-Aws
Aws-Aws
Aws-Gcp
Gcp-Gcp

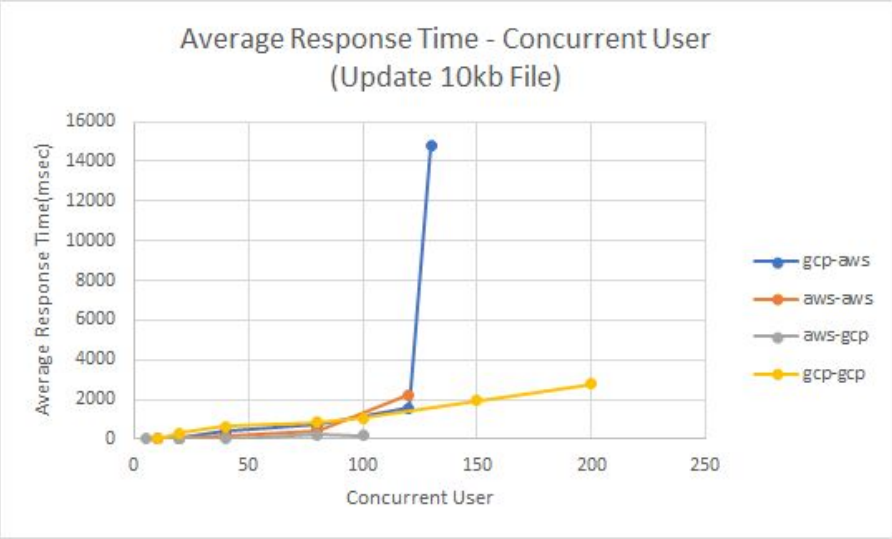

Actual Request Rate - Concurrent User
(Read 10kb File)
Actual Request Rate (req/sec)
300
250
200
150
100
50
0
0
100
200
300
400
500
600
700
Concurrent User
gcp-aws
aws-aws
aws-gcp
gcp-gcp

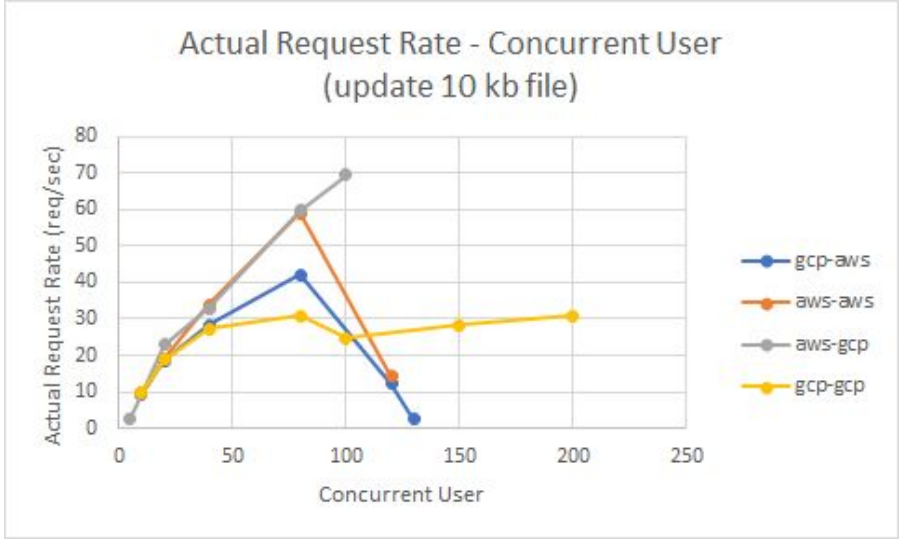
# Result


Average Response Time - Concurrent User
(Read 10mb File)
Average Response Time (msec)
12000
10000
8000
6000
4000
2000
0
0
0.5
1
1.5
2
2.5
Concurrent User
gcp-aws
aws-aws
aws-gcp
gcp-gcp

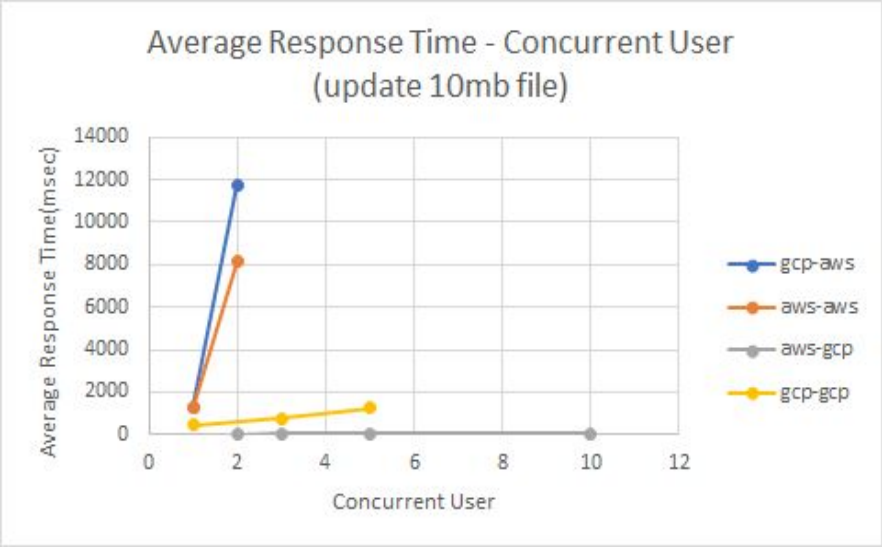

Actual Request Rate - Concurrent User
(Read 10mb File)
Actual Request Rate (req/sec)
1.2
1
0.8
0.6
0.4
0.2
0
0
0.5
1
1.5
2
2.5
Concurrent User
gcp-aws
aws-aws
aws-gcp
gcp-gcp

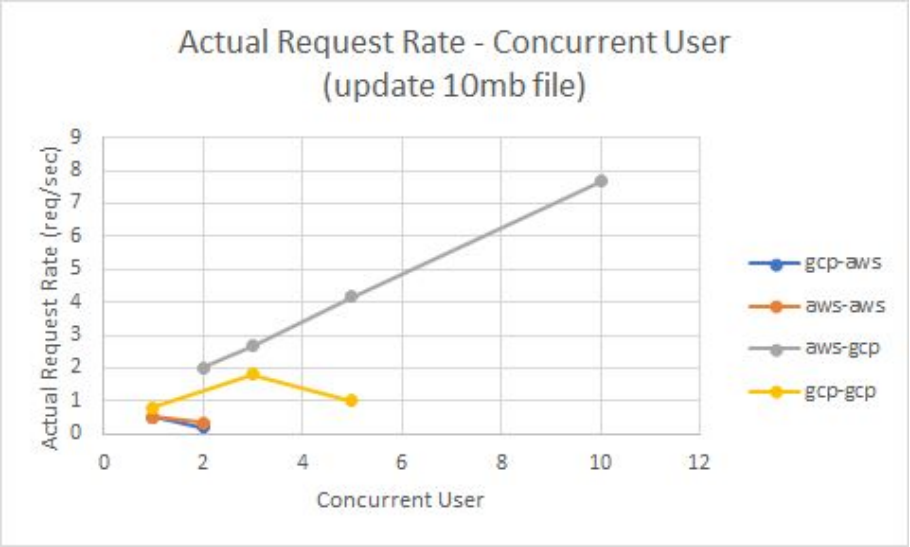
# Result


Average Response Time - Concurrent User
(Update 10kb File)
Average Response Time(msec)
16000
14000
12000
10000
8000
6000
4000
2000
0
0
50
100
150
200
250
Concurrent User
gcp-aws
aws-aws
aws-gcp
gcp-gcp

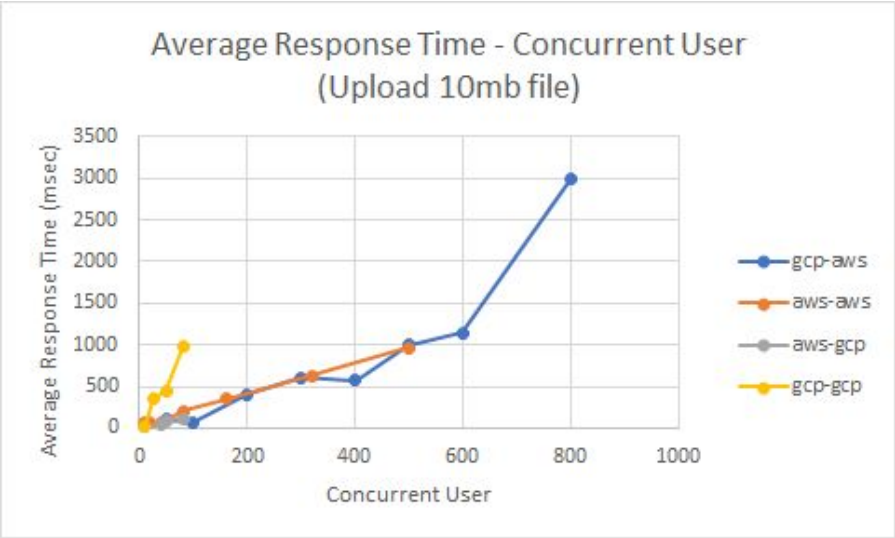

Actual Request Rate - Concurrent User
(update 10 kb file)
Actual Request Rate (req/sec)
80
70
60
50
40
30
20
10
0
0
50
100
150
200
250
Concurrent User
gcp-aws
aws-aws
aws-gcp
gcp-gcp

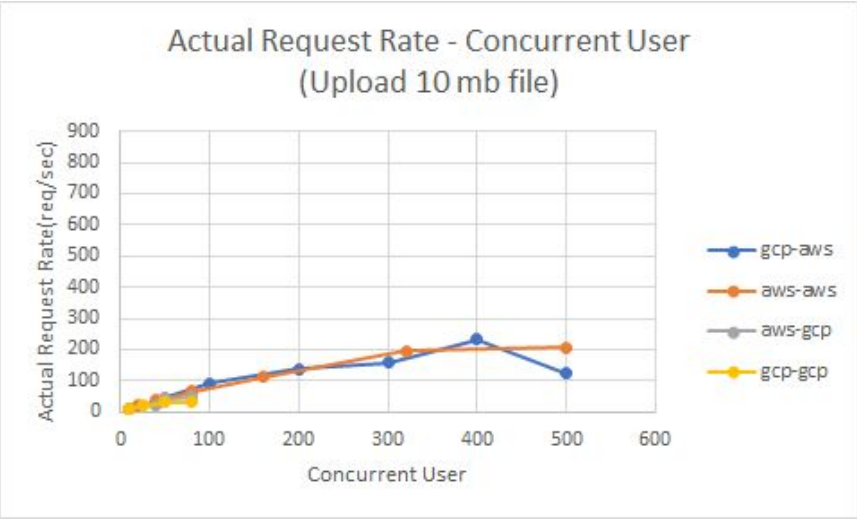
# Result


Average Response Time - Concurrent User
(update 10mb file)
Average Response Time(msec)
14000
12000
10000
8000
6000
4000
2000
0
0
2
4
6
8
10
12
Concurrent User
gcp-aws
aws-aws
aws-gcp
gcp-gcp

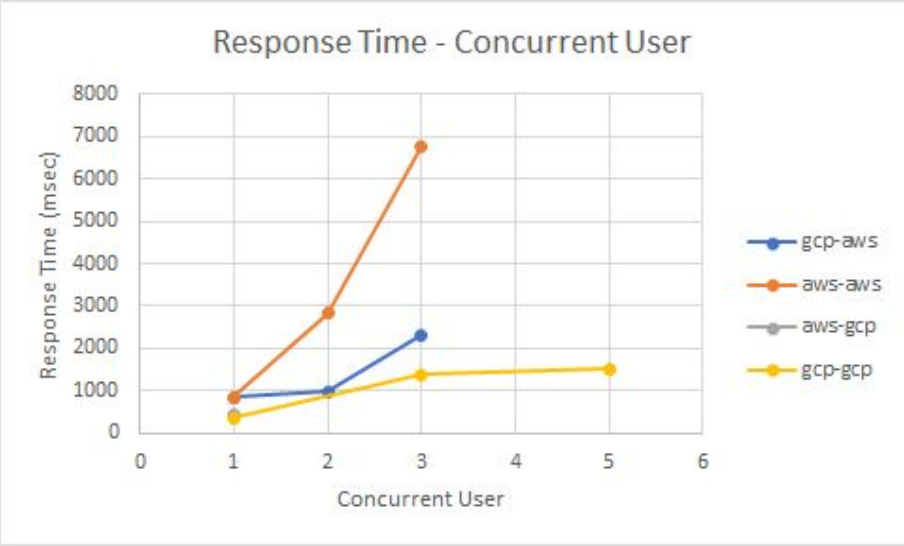

Actual Request Rate - Concurrent User
(update 10mb file)
Actual Request Rate (req/sec)
9
8
7
6
5
4
3
2
1
0
0
2
4
6
8
10
12
Concurrent User
gcp-aws
aws-aws
aws-gcp
gcp-gcp

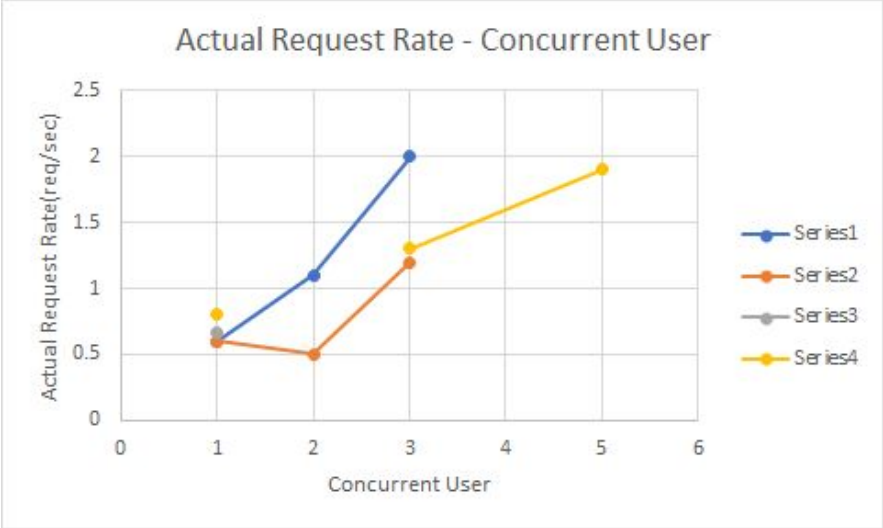
# Result


Average Response Time - Concurrent User
(Upload 10mb file)
Average Response Time (msec)
3500
3000
2500
2000
1500
1000
500
0
0
200
400
600
800
1000
Concurrent User
gcp-aws
aws-aws
aws-gcp
gcp-gcp


Actual Request Rate - Concurrent User
(Upload 10 mb file)
Actual Request Rate(req/sec)
900
800
700
600
500
400
300
200
100
0
0
100
200
300
400
500
600
Concurrent User
gcp-aws
aws-aws
aws-gcp
gcp-gcp

# Result


Response Time - Concurrent User
Response Time (msec)
8000
7000
6000
5000
4000
3000
2000
1000
0
0
1
2
3
4
5
6
Concurrent User
gcp-aws
aws-aws
aws-gcp
gcp-gcp


Actual Request Rate - Concurrent User
Actual Request Rate(req/sec)
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5
6
Concurrent User
Series1
Series2
Series3
Series4

# Conclusion and Recommendation

The full test results and graph are available on our slide. Comparing between AWS and GCP in a single cloud setting, AWS performs better in getting the file and with smaller file sizes, on the other hand, GCP wins on writing and updating. Despite the result shows GCP can handle a larger file size in a better performance, we need to take in consider that the smallest instance that use on GCP has double the vCPU of EC2 instance we were using.

Next, let's look from a bigger perspective. From the comparison between the multi-cloud settings and the vendor-native settings, we can conclude that s3 and EC2 tends to be better in handling concurrency. GCP services are vulnerable to high rates of concurrency.

If we compare multi-cloud settings alone, comparing between EC2 connect to GCS and GCE connect to S3, it is that EC2 to GCS has a better performance that GCE connect to S3, while the latter can handle more concurrency loads.

In conclusion, it is hard to compare these services with this low amount of data, but AWS performs better at a suitable rate of workloads. GCP then can perform better if the workload becomes overly high.

# Post-Test Question

---

Which one performed better ?

a. AWS EC2 connect to Google Cloud Storage
b. Google Cloud Engine connect to AWS S3