15-319 / 15-619 Cloud Computing

Overview 9

March 22nd, 2022

Reflection of Last Week

- **OLI, Unit 4: Cloud Storage**
  - Module 14: Cloud Storage
  - Module 15: Case Studies: Distributed File System
  - Module 16: Case Studies: NoSQL Databases
  - Module 17: Case Studies: Cloud Object Storage
- **Quiz 7 (OLI Module 14)**
- **Project 4 - Iterative processing with Spark**
- **Team Project Phase 1**

This Week

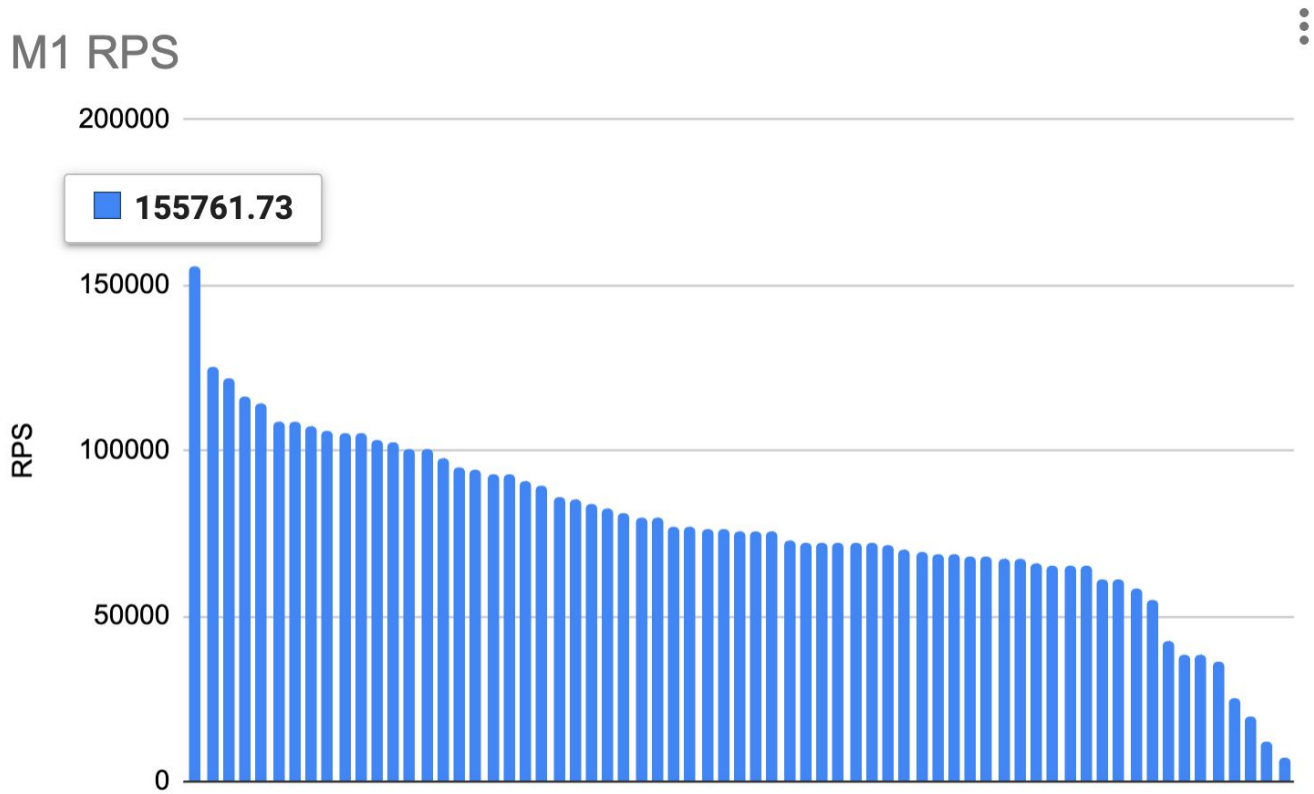
- **OLI, Unit 4: Cloud Storage**
  - Module 14: Cloud Storage
  - Module 15: Case Studies: Distributed File System
  - Module 16: Case Studies: NoSQL Databases
  - Module 17: Case Studies: Cloud Object Storage
- **Quiz 8 (OLI Module 15, 16 and 17)**
  - Due on **Friday**, October 29nd, 2021, 11:59PM ET

- **Team Project Phase 1 Report**
  - Due on **Tuesday**, October 26th, 2021, 11:59PM ET
- **Team Project Phase 2**
  - Started this week!

TEAM PROJECT
Twitter Data Analytics

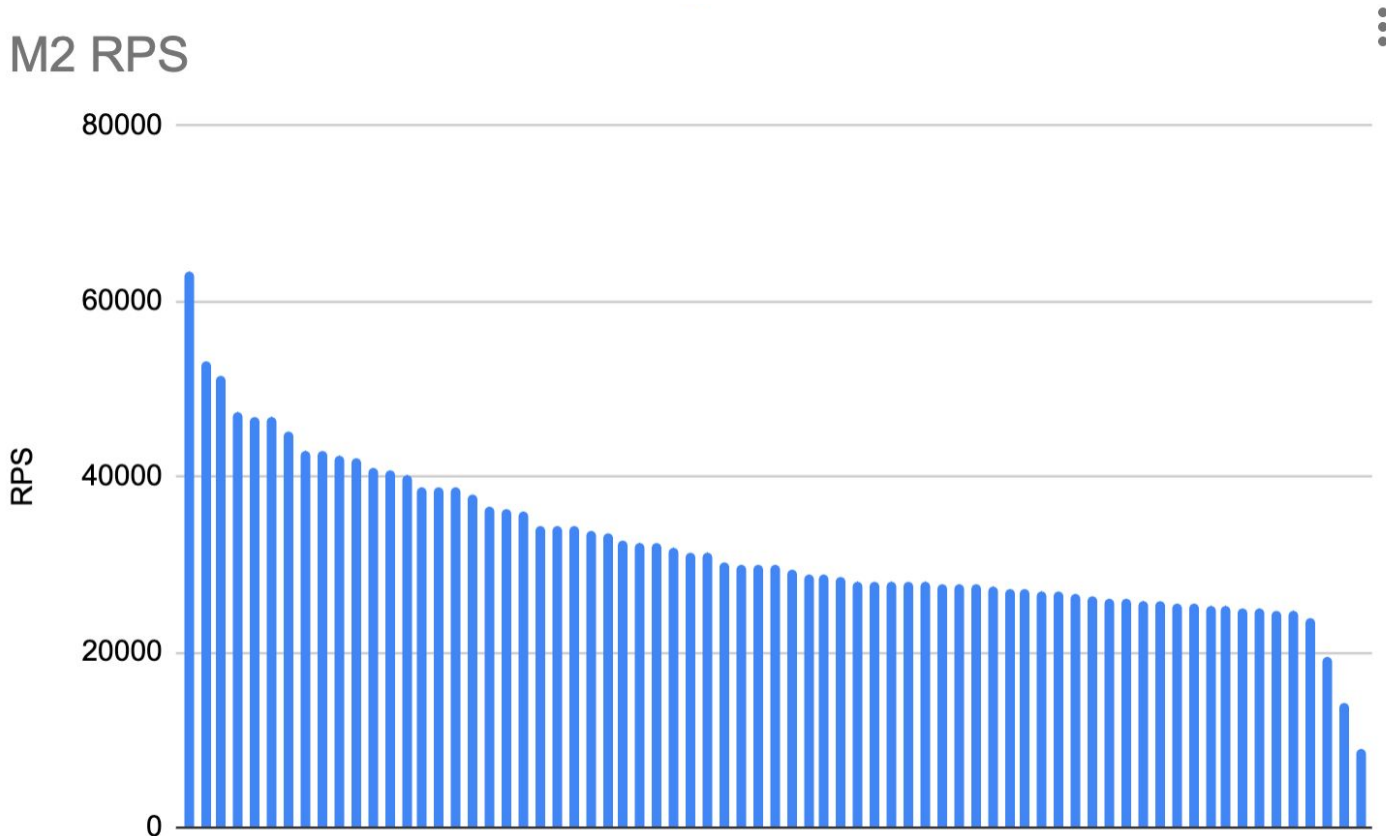
Reflection of Microservice 1

- QR Code Encoding/Decoding
- 54/68 Teams reached target RPS
- Team CloudWatchers reached 155,761 RPS!



Reflection of Microservice 2

- Blockchain Validation/Mining
- 62/68 Teams reached target RPS
- Team ThreeCobblers reached 63,287 RPS!



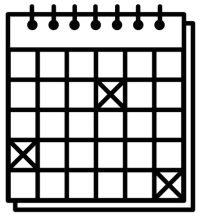
Reflection of Microservice 3

- 3/68 Teams received the checkpoint
- 2/68 Teams received the early bird bonus
- Please start early in Phase 2!

Team Project Time Table

Task	Timeline
Phase 2 - **M1** - **M2** - **M3**	● Submissions due: Sunday, 04/03/2022 3:59 PM EST
Phase 2 Live Test	● DNS Submission open: Sunday, 04/03/2022 3:00 PM EST ● DNS Submission due: Sunday, 04/03/2022 3:59 PM EST
Phase 2 Report	● Due: Tuesday, 04/05/2022 11:59 PM EST

Team Project Time Table



Phase	Deadline (<u>11:59PM EST</u>)
Phase 3 (50%) **- Managed Services for Microservice 1-3** **- Live Test!**	● Live Test on Sun, 04/17/2022
Phase 3 Report	● Due: Tuesday, 04/19/2022 23:59 PM EST

Live Test Schedule

Information

Time	Task	Description
4:00 pm	DNS	Submit your DNS for the Live Test before the deadline
5:33 pm - 5:34 pm	DNS Validation	We will validate your DNS. This is the last chance to update your DNS for the Live Test

Live Test

Information

Time (at 11:59PM EST)	Value	Target	Weight
6:00 pm - 6:30 pm	Warm-up (M1 only)	0	0%
6:30 pm - 7:00 pm	M1	65000	12%
7:00 pm - 7:30 pm	M2	25000	20%
7:30 pm - 8:00 pm	M3	10000	20%
8:00 pm - 8:30 pm	Mixed Reads (M1, M2, M3)	15000/10000/2000	8+10+10 = 28%

Task Reminder

- Throughput targets:
  - M1: 65,000 RPS
  - M2: 25,000 RPS
  - M3: 10,000 RPS
  - Mixed queries M1/M2/M3: 15,000/10,000/2,000 RPS
- Bonuses
  - Live Test Ranking Top 10
    → Bonus points ranging from 0.5% to 5%
  - Achieve 15,000 RPS for M3 during Live Test
    → Penalty waiver
  - Achieve >= 99% correctness for all Live Tests
    → Penalty waiver

Total Budget Reminder

- Budget limit $80, double budget $100

	No Penalty	-10% Penalty	-100% Penalty
Total cost	< $80	$80 - $100	$100+
Development cost	< $60	$60 - $80	$80+
Live Test cost	~ $20	~ $20	~ $20

- Use GCP and Azure for ETL
- Use Spot instances wisely

Hourly Budget Reminder

- Your web service should not cost more than **$0.70/hour (if using MySQL) and $1.10/hour (if using HBase)**
- This includes:
  - EC2 cost (Even if you use spot instances, we will calculate your cost using the **on-demand** instance price)
  - **EBS cost**
  - **ELB cost - excluding LCU-hour cost**
  - We will not consider the cost of data transfer and EMR software
  - See the write up for details

Resource Constraint Reminder

- Self-managed Kubernetes cluster + optional EMR, consisting of M family instances **only**, smaller than or equal to **large** type
- MySQL must be installed on the Kubernetes cluster
  - No standalone EC2 instance, no RDS
- Other types are allowed (e.g., t2.micro) **but only for testing**
  - Using these for the live test submission = 100% penalty
- Only General Purpose (gp2) SSDs are allowed for storage
  - e.g **m5d is not allowed** since it uses NVMe storage
- AWS endpoints only (EC2/ELB)

Loading data & Backup

- Refer to MySQL Primer and Project 3 for data loading
  - P3 YetAnotherImportTsv can be helpful
  - Be very careful about escape characters
  - Be very careful about encodings
  - You can use temporary EC2 instance or EMR clusters to load your data
- Backup
  - For MySQL, make EBS snapshots of your data directory and attach it to your Pod
  - For HBase, you can backup and restore HBase database on S3 using the HBase snapshot

Hints

- If you haven’t completed your ETL yet
  - Make sure to attach enough volume to your instances so that you can load and process the whole data set which is ~1TB.
  - MySQL and HBase have default limits on row/cell size. If you have large cells, the size limit probably needs to be tuned.
- How do I know whether my database schema is good enough to achieve the target throughput?
  - Use profiling to estimate the time your service used for processing one database query, and compare it with the time to process one request
  - Check how many tables you have in your database

Hints

- Iterations rank higher than parameter tuning
  - Do not waste time tuning parameters when you have only one tenth of the target RPS!
  - Are all database queries necessary? Can they be done in your ETL pipelines instead?
  - **A good schema can easily double or even triple the throughput with no parameter tuning!**

- To do performance tuning, you first need to identify which part of your system is the bottleneck
  - Profile and monitor your system
  - Read the Profile Primer for profiling tools

Hints

- Web Tier
  - Concurrency model?
  - Connection pooling?
  - Caching result? (no third-party cache library!)
  - Is every computation in the web tier necessary?
    - Can they be done in ETL instead?
  - Have you optimized your code?
    - StringBuilder vs ‘+’
    - Try different library (gson vs Jackson vs jsoniter)

Hints

- Storage Tier - MySQL
  - Different MySQL engines
  - EBS I/O Credits and Burst Performance
- Storage Tier - HBase
  - Locality and compaction, region server split, etc.
  - Scan can be really slow, try to avoid it if possible
    If you can’t, try to scan as few rows as possible
- Tune parameters ← Should be last thing to do!!
  - Check the official documentation
  - Search for performance tuning best practices

Suggested readings

Some suggested readings:

1. https://kubernetes.io/docs/tasks/run-application/run-replicated-stateful-application/ for a sample StatefulSet definition
2. https://kubernetes.io/docs/concepts/storage/volume-snapshots/#volume-snapshot-contents for defining snapshot with existing content
3. https://aws.amazon.com/blogs/containers/using-ebs-snapshots-for-persistent-storage-with-your-eks-cluster/ for provisioning EBS volumes from snapshots (the title says EKS but it also applies to our kOps cluster since we already installed the required controllers)

Best Wishes!!!