

15-319 / 15-619

Cloud Computing

Recitation 8

Mar 5, 2019

Overview

- **Last week's reflection**
 - Project 3.1
 - OLI Unit 3 - Module 13
 - Quiz 6
- **This week's schedule**
 - Project 3.2
 - OLI Unit 4 - Module 14
 - Quiz 7 (**Due Thursday 3/7**)
 - Online Programming Exercise for Multi-Threading
- **Team Project, Twitter Analytics**
 - Phase 1 is out! Q1 final due on 3/10.
 - Phase 1 due, Mar 31.

Last Week

- **Unit 3: Virtualizing Resources for the Cloud**
 - Module 13: Storage and network virtualization
- **Quiz 6**
- **Project 3.1**
 - Files v/s Databases (SQL & NoSQL)
 - Flat files
 - MySQL
 - HBase
 - Read the NoSQL and HBase basics primer

This Week

- **OLI : Unit 4 Module 14 - Cloud Storage**
- **Quiz 7 - Thursday, March 7**
- **Project 3.2 - Sunday, March 10**
 - Social Networking Timeline with Heterogenous Backends
 - MySQL
 - Neo4j
 - MongoDB
 - Choosing Databases, Storage Types & Tail Latency
 - MongoDB Primer
- **Online Programming Exercise for Multi-Threading on Cloud9**
 - This week
- **Team Project, Phase 1 released**

Conceptual Topics - OLI Content

- **OLI Unit 4 - Module 14: Cloud Storage**
 - File Systems and Databases
 - Scalability and Consistency
 - NoSQL, NewSQL and Object Storage
 - CAP theorem
- **Quiz 7**
 - **DUE on Thursday, March 07**
 - **Remember to click submit**
 - **Within 2 hours, and**
 - **Before the deadline!**

Individual Projects

- DONE
 - P3.1: Files vs Databases - comparison and Usage of flat files, MySQL, Redis, and HBase
 - NoSQL Primer
 - HBase Basics Primer
- **NOW**
 - P3.2: Social networking with heterogeneous backends
 - MongoDB Primer
- Coming Up
 - P3.3: Multi-threading Programming and Consistency

A Social Network Service



← → ↻ 🏠 🌐 www.facebook.com/zuck?sk=wall

facebook 🔍

Mark Zuckerberg

🏢 Works at Facebook 🎓 Studied Computer Science at Harvard University 🏠 Lives in Palo Alto, California 🗣️ Knows English, Mandarin Chinese 🏡 From Dobbs Ferry, New York 📅 Born on May 14, 1984

Wall

RECENT ACTIVITY

💬 "I like dangerous thoughts." on Samuel W. Lessin's status.

Mark Zuckerberg

Steve, you've done so much good for the world already. I hope you get better soon.

📱 January 17 at 11:43am via iPhone

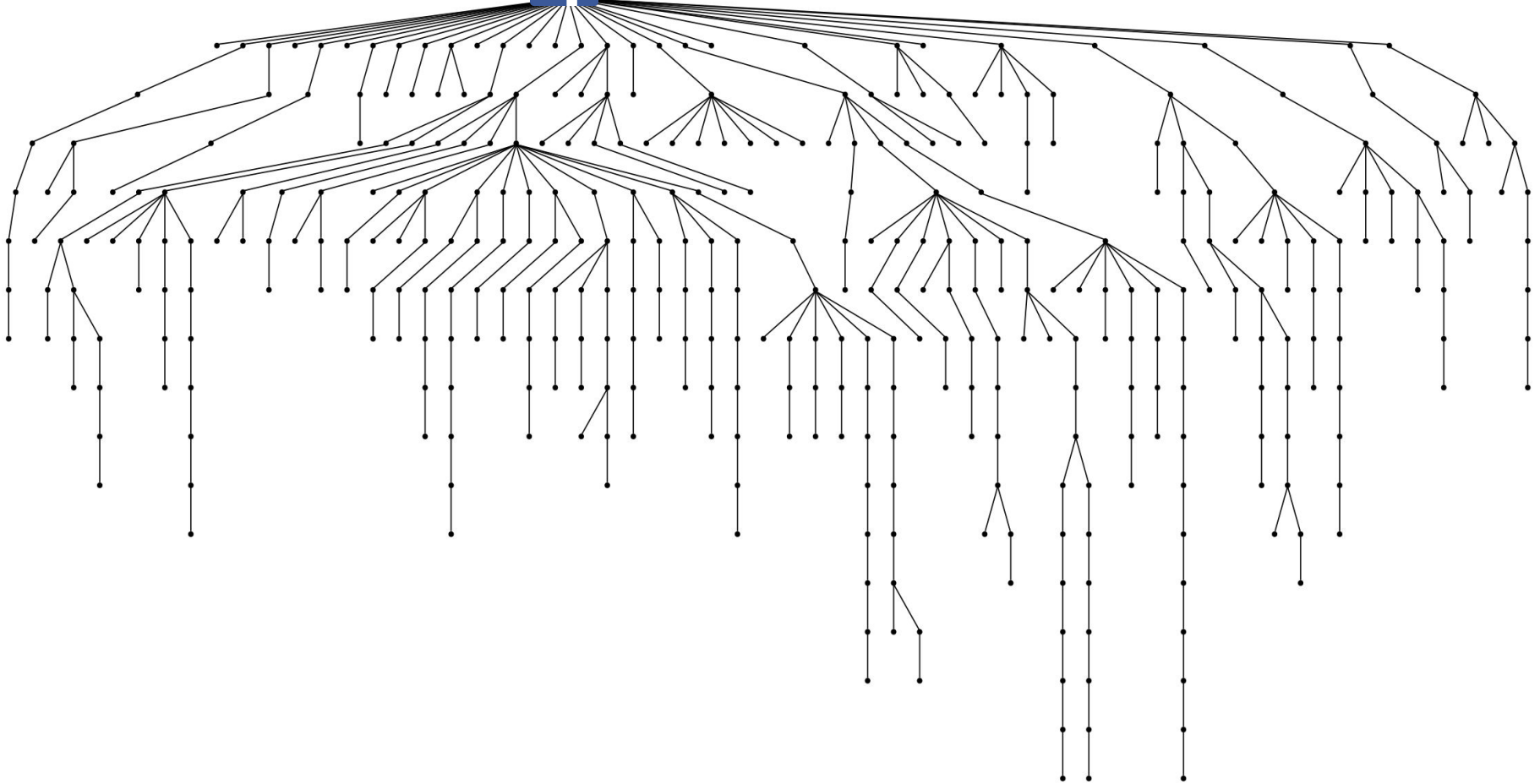
👍 150 people like this.

Wall
Info

Share Profile
Report/Block This Person

High Fanout in Data Fetching

A single  page, requires many data fetch operations



Nishtala, R., Fugal, H., Grimm, S., Kwiatkowski, M., Lee, H., Li, H. C., ... & Venkataramani, V. (2013, April). Scaling Memcache at Facebook. In *nsdi* (Vol. 13, pp. 385-398).

Graph Database Neo4j

- Designed to treat the relationships between data as equally important as the data
 - Relationships are very important in social graphs
- Property graph model
 - Nodes
 - Relationships
 - Properties
- Cypher query language
 - Declarative, SQL-inspired language for describing patterns in graphs visually

MongoDB

- Document Database
 - Schema-less model
- Highly Scalable
 - Automatically shards data among multiple servers
 - Does load-balancing
- Allows for Complex Queries
 - MapReduce style filter and aggregations
 - Geo-spatial queries

P3.2 - Overview

- Build a social network about Reddit comments
- Dataset generated from Reddit.com
 - **users.csv, links.csv, posts.json**
- Build a social network timeline on the Reddit.com data
 - **Task 1:** Basic login
 - **Task 2:** Social graph
 - **Task 3:** Rank user comments
 - **Task 4:** Generate User Timeline
- **Task 5: Understanding Tail Latency, BLOBs, Storage Types, and Selecting Databases**
 - Answer questions on relevant topics and choose the right database and storage type for a given scenario

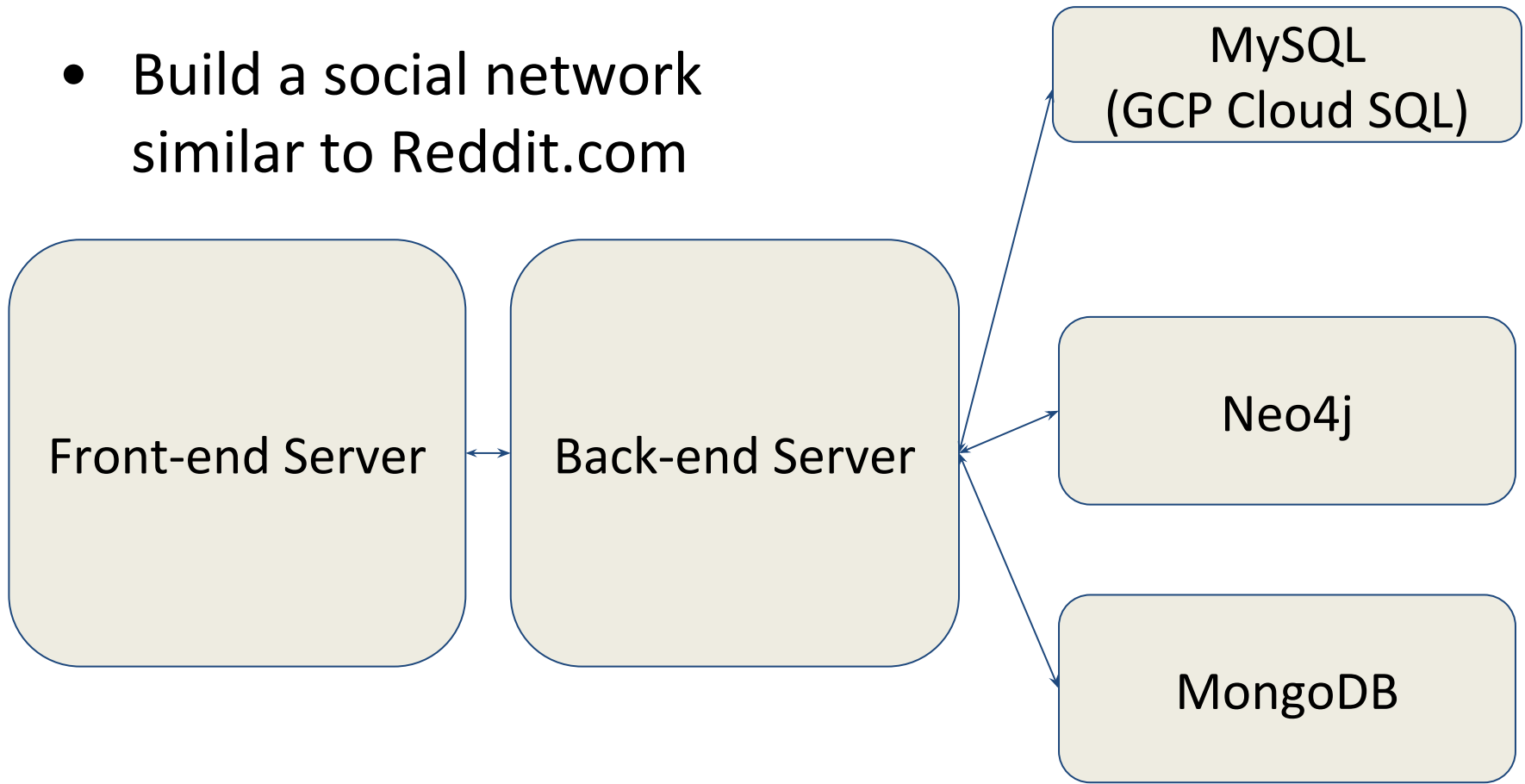
P3.2 - Reddit Dataset

- Task 1: User profiles
 - User authentication system : GCP Cloud SQL(**users.csv**)
 - User info / profile : GCP Cloud SQL
- Task 2: Social graph of the users
 - Follower, followee : Neo4j (**links.csv**)
- Task 3: User activity system
 - All user generated comments : MongoDB (**posts.json**)
- Task 4: User timeline
 - Put everything together



P3.2 - Architecture

- Build a social network similar to Reddit.com



Tasks, Datasets & Storage

Introduction

The Scenario: Build Your Own Social Network Website

Task 1: Implementing Basic Login with SQL

Task 2: Storing Social Graph using Neo4j

Task 3: Build Homepage using MongoDB

Task 4: Put Everything Together

Task 5: Choosing Databases

Dataset Name	Data Store Type
Login Information	RDBMS
Relation	Graph Database
Comments	Document Stores
Profile Images	S3

P3.2 - Task 5

- **Issues of dealing with Scale**
 - An overview of the systems issues that arise with scale and how they were addressed in the context of Facebook.
 - Tail Latency and Fanout
 - BLOBs and Storage Types
 - Cost and performance
 - Learn how popularity and freshness of data plays a role in designing efficient social networking backends.

P3.2 - Task 5

- **Choosing Databases & Storage Types**
 - Use your knowledge and experience gained working with the databases in the project to
 - Identify advantages and disadvantages of various DBs
 - Pick suitable DBs for particular application requirements
 - Provide reasons on why a certain DB is suitable under the given constraints
 - Instructions provided in **runner.sh**

Terraform

- Required in P3.2
- Required in the team project, get some practice
- Files provided
- Use **'terraform destroy'** to terminate resources
- This project is on GCP, so apply the following tag
 - The tag is "3-2" instead of "3.2" (for GCP only)

P3.2 - Reminders and Suggestions

- Set up a budget alarm on GCP
 - Suggested budget: \$15
 - No penalties
- Learn and practice using a standard JSON Library. This will prove to be valuable in the Team Project
 - **Google GSON** - Recommended for Java
- Set up Gcloud in your environment
- No AWS instances on your individual AWS account are allowed
 - Otherwise you will receive warning emails and penalties

P3.2 - Reminders and Suggestions

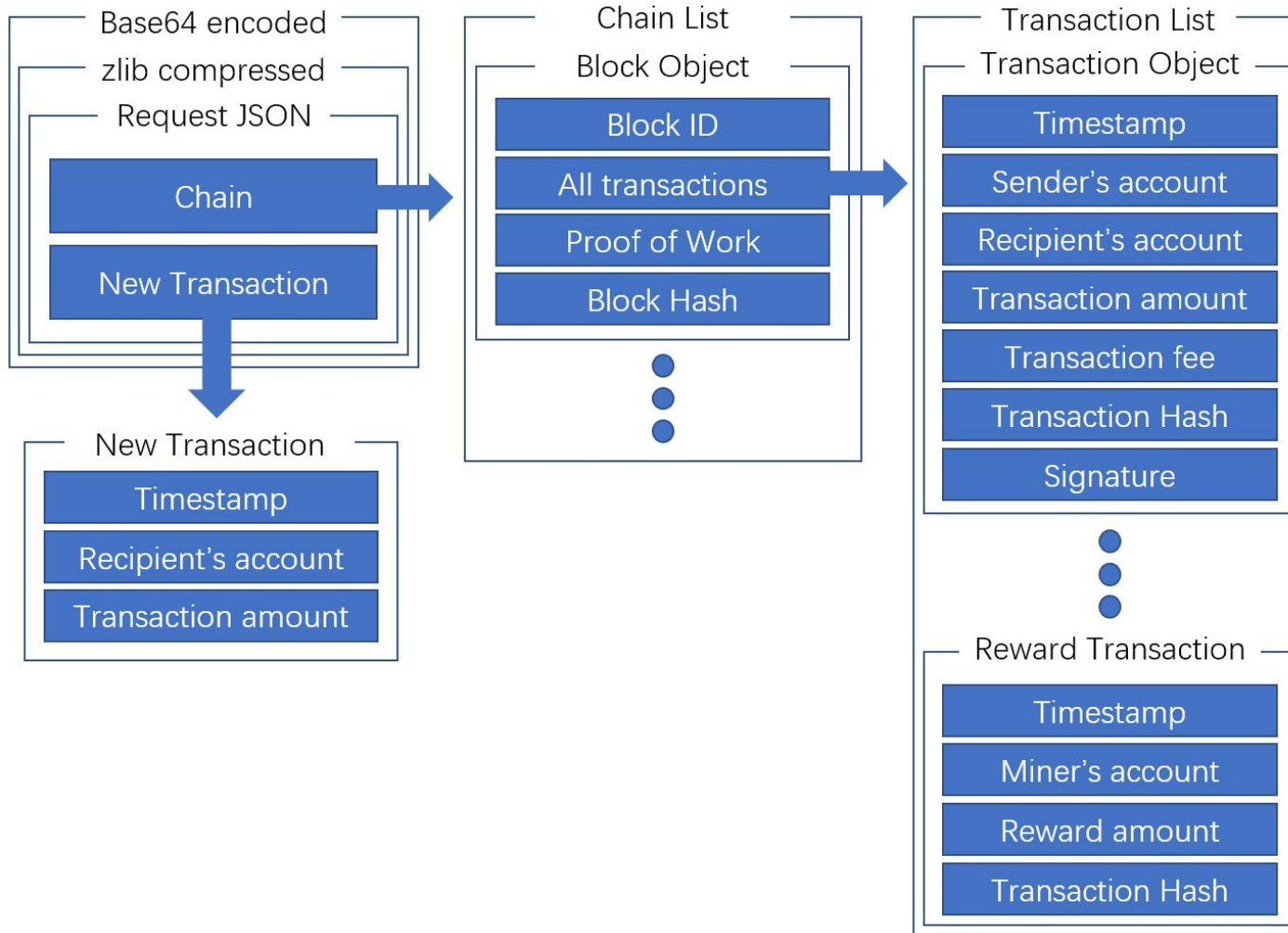
- In Task 4, you will use the databases from all previous tasks. Make sure to have **all** the databases loaded and ready when working on Task 4.
- You can submit one task at a time using the submitter. Remember to have your Back-end Server VM running when submitting.
- Make sure to terminate **all** resources using “terraform destroy” after the final submission. Double check on the GCP console that all resources were terminated.

TEAM PROJECT

Twitter Data Analytics



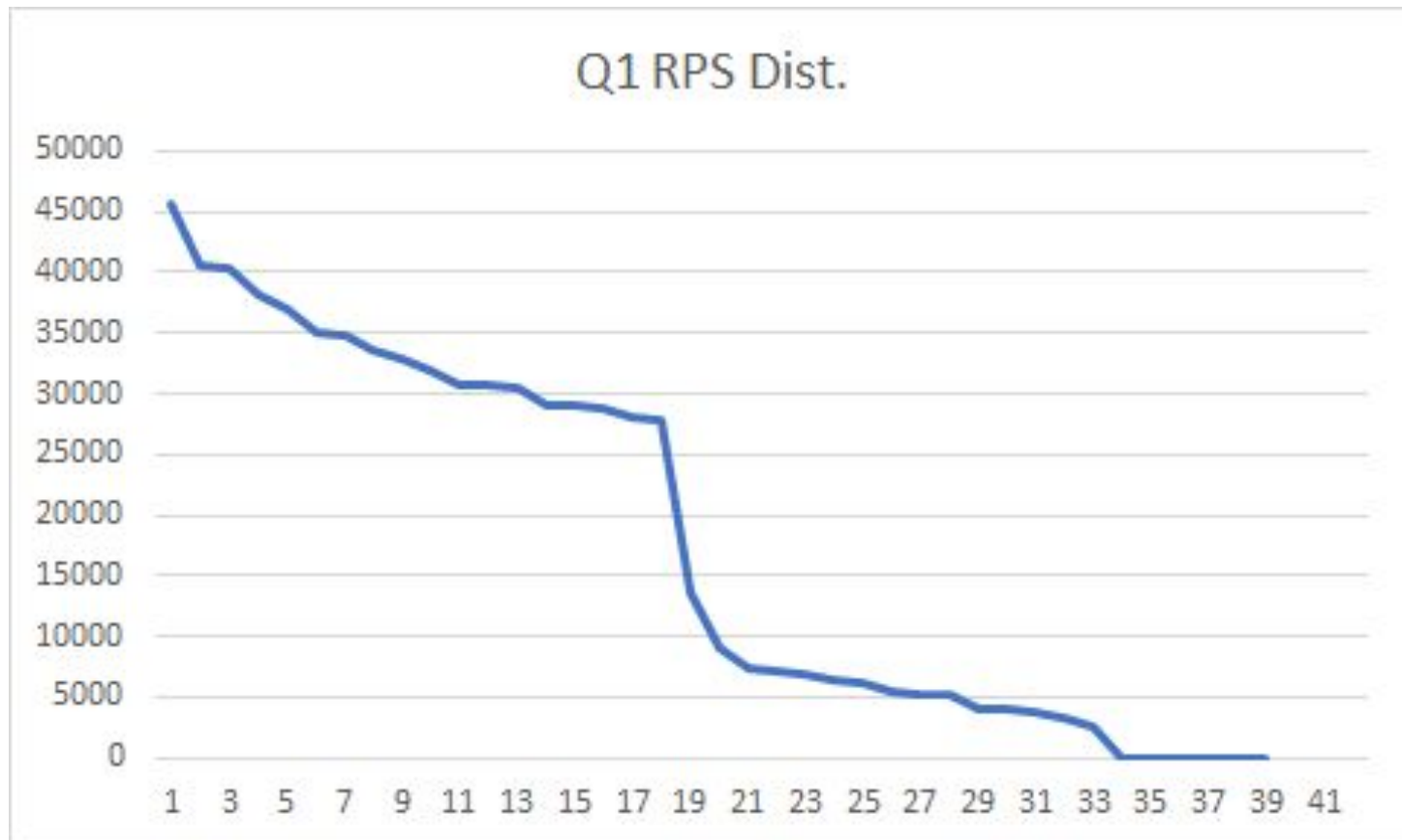
Query 1 Recap



```
{
  "chain": [
    {
      "all_tx": [
        {
          "recv": 509015179679,
          "amt": 500000000,
          "time": "155072196779362304",
          "hash": "d50e5266"
        }
      ],
      "id": 0,
      "hash": "02899b89",
      "pow": "postpone"
    },
    {
      "all_tx": [
        {
          "send": 509015179679,
          "recv": 484054352161,
          "amt": 126848946,
          "fee": 12488,
          "time": "155072196779391744",
          "hash": "5a2b4d71",
          "sig": 463884077351
        },
        {
          "recv": 1284110893049,
          "amt": 500000000,
          "time": "155072196779424000",
          "hash": "7924c55e"
        }
      ],
      "id": 1,
      "hash": "0fce51c1",
      "pow": "fountain"
    },
    {
      "all_tx": [
        {
          "send": 1284110893049,
          "recv": 484054352161,
          "amt": 58759591,
          "fee": 5048,
          "time": "155072196779447040",
          "hash": "b43737af",
          "sig": 1084970046728
        },
        {
          "recv": 34123506233,
          "amt": 500000000,
          "time": "155072196779474176",
          "hash": "d705e74e"
        }
      ],
      "id": 2,
      "hash": "03635f77",
      "pow": "jeans"
    }
  ],
  "new_tx": {
    "recv": 837939704897,
    "amt": 430642077,
    "time": "155072196779486720"
  }
}
```

Team Project - Q1 CKPT1

- 40 teams attempted a Query 1 submission.
- 33 teams got a 10-minute submission
- 17 teams reached 28000 RPS



Read about Query 2 Now. Start ETL Now.

Query 1 Final	28000	10%	Sunday, March 10
Query 2 Checkpoint	-	10%	Sunday, March 24
Query 2 Final	12000	50%	Sunday, March 31
Final Report + Code	-	20%	Tuesday, April 2

Read about Query 2 Now. Start ETL Now.

Query 1 Final	28000	10%	Sunday, March 10
Query 2 Checkpoint	-	10%	Sunday, March 24
Query 2 Final	12000	50%	Sunday, March 31
Final Report + Code	-	20%	Tuesday, April 2

After spring break, you got one week
to meet the Query 2 checkpoint.

Question: Is 1 week enough time for that?

Hint: No. Start now.

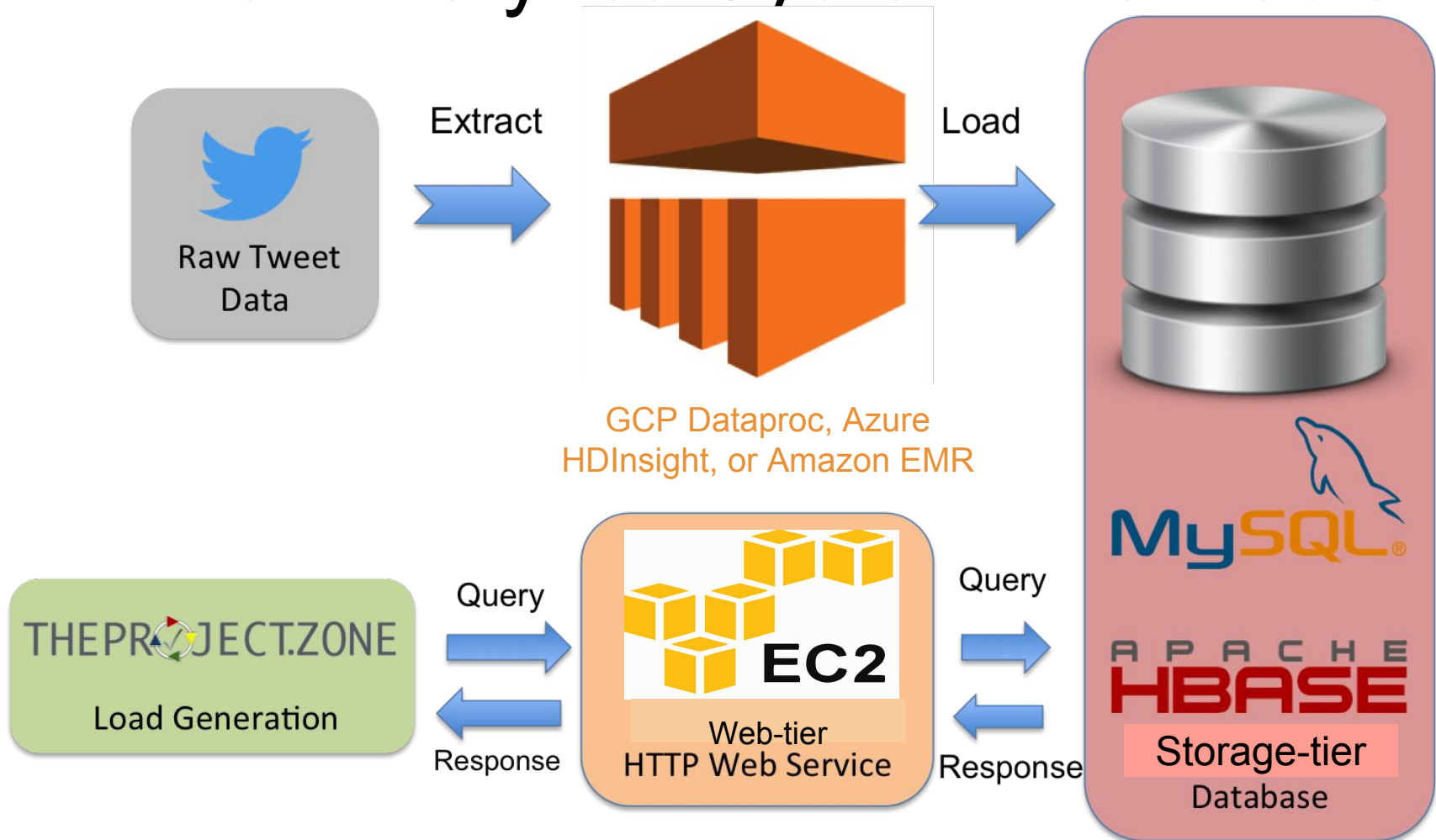
Team Project

Twitter Analytics Web Service

- Given ~1TB of Twitter data
- Build a performant web service to analyze tweets
- Explore web frameworks
- Explore and optimize database systems



Twitter Analytics System Architecture



- Web server architectures
- Dealing with large scale real world tweet data
- HBase and MySQL optimization



Query 2 - User Recommendation System

Use Case: When you follow someone on twitter, recommend close friends.

Three Scores:

- Interaction Score - closeness
- Hashtag Score - common interests
- Keywords Score - to match interests

Final Score: Interaction Score * Hashtag Score * Keywords Score

Query:

```
GET /q2?  
user_id=<ID>&  
type=<TYPE>&  
phrase=<PHRASE>&  
hashtag=<HASHTAG>
```

Response:

```
<TEAMNAME>,<AWSID>\n  
uid\tname\tdescription\ttweet\n  
uid\tname\tdescription\ttweet
```

Query 2 Example

```
GET /q2?  
user_id=100123&  
type=retweet&  
phrase=hello%20cc&  
hashtag=cmu
```

```
TeamCoolCloud,1234-0000-0001  
100124\tAlan\tScientist\tDo machines think?\n100125\tKnuth\tprogrammer\thello cc!
```

Reminders on penalties

- M family instances **only**, smaller than or equal to **large** type
- Only General Purpose (gp2) SSDs are allowed for storage
 - so **m5d is not allowed** since it uses NVMe storage
- Other types are allowed (e.g., t2.micro) **but only for testing**
 - Using these for any submissions = 100% penalty
- **\$0.85/hour** applies to every submission, not just the livetest
- AWS endpoints only (EC2/ELB).

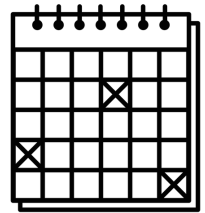
Phase 1 Budget

- AWS budget of \$45 for Phase 1
- Your web service should not cost more than **\$0.85 per hour** this includes (see write-up for details):
 - EC2 cost
 - **EBS cost**
 - ELB cost
 - We will not consider the cost of data transfer and EMR
- Even if you use spot instances, we will calculate your cost using the on-demand instance price
- Q2 target throughput: 12000 RPS for both MySQL and HBase

Tips

- Consider doing ETL on GCP/Azure to save your AWS budget
- Be careful about encoding 😊 (use utf8mb4 in MySQL)
- Pre-compute as much as possible
- ETL can be expensive, so read the write-up carefully

Suggested Tasks for Phase 1



Phase 1 weeks	Tasks	Deadline
Week 1 2/25	- Team meeting - Writeup - Complete Q1 code & achieve correctness - Q2 Schema, think about ETL	- Q1 Checkpoint due on 3/3 - Checkpoint Report due on 3/3
Week 2 3/4	- Q1 target reached - Q2 ETL & Initial schema design completed	Q1 final target due on 3/10
Week 3 Spring Break	Take a break or make progress (up to your team)	
Week 4 3/18	Achieve correctness for both Q2 MySQL, Q2 HBase & basic throughput	- Q2 MySQL Checkpoint due on 3/24 - Q2 HBase Checkpoint due on 3/24
Week 5 3/25	Optimizations to achieve target throughputs for Q2 MySQL and Q2 HBase	- Q2 MySQL final target due on 3/31 - Q2 HBase final target due on 3/31



This Week's Deadlines



- Quiz 7:
Due: **Thursday**, March 7th, 2019 11:59PM ET
- Complete Multi-Threading OPE task
Due: **This week** (date varies)
- Project 3.2: Social Networking Timeline
Due: **Sunday**, March 10th, 2019 11:59PM ET
- Team Project Phase 1 Q1 Final
Due: **Sunday**, March 10th, 2019 11:59PM ET

Q&A