15-319 / 15-619 Cloud Computing

Recitation 12

Tuesday, April 7, 2020

Overview

- **Last week's reflection**
  - Project 4.1
  - Quiz 10
  - Team Project Phase 2 released

- **This week's schedule**
  - Unit 5 - Modules 21 & 22
    - Quiz 11 (Last quiz)
  - Team Project, Phase 2, Queries, 1, 2, 3
  - Team Project, live test
    - HBase
    - MySQL

P4.1 Reflection

- Programming in Scala and Spark
- Understanding the differences between processing data with MapReduce and Spark
- Exploring Twitter social data with the RDD and DataFrame APIs
- Implementing an iterative processing algorithm - pagerank - on a large dataset
- Utilizing the Spark Web UI to monitor a Spark job and identify performance bottlenecks
- Tuning a Spark program to optimize for time
- Running the PageRank application on Azure Databricks to compare performance

P4.1 Reflection

- Common Issues
  - Handling dangling nodes in the graph
  - Tuning the cluster for better performance.
  - Long running jobs
    - Reduce the amount of data shuffling
- Takeaways
  - Some approaches to implementing pagerank are more efficient than others
  - The Spark Web UI is a useful visualization tool
  - Databricks offers optimized version of Spark providing better performance than HDInsight

Modules to Read

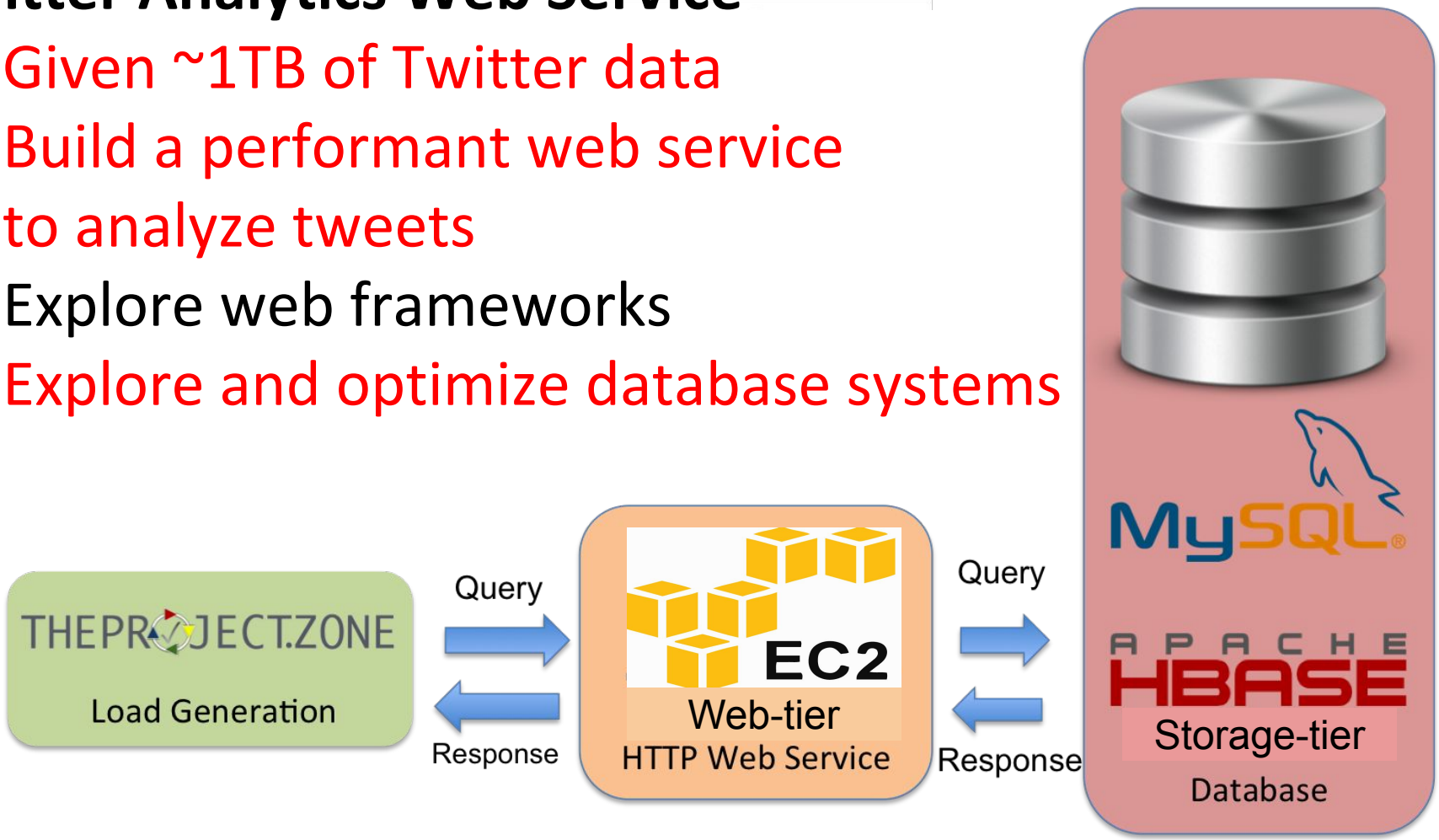
- UNIT 5: Distributed Programming and Analytics Engines for the Cloud
  - Module 18: Introduction to Distributed Programming for the Cloud
  - Module 19: Distributed Analytics Engines for the Cloud: MapReduce
  - Module 20: Distributed Analytics Engines for the Cloud: Spark
  - Module 21: Distributed Analytics Engines for the Cloud: GraphLab
  - Module 22: Message Queues and Stream Processing

TEAM PROJECT
Twitter Data Analytics

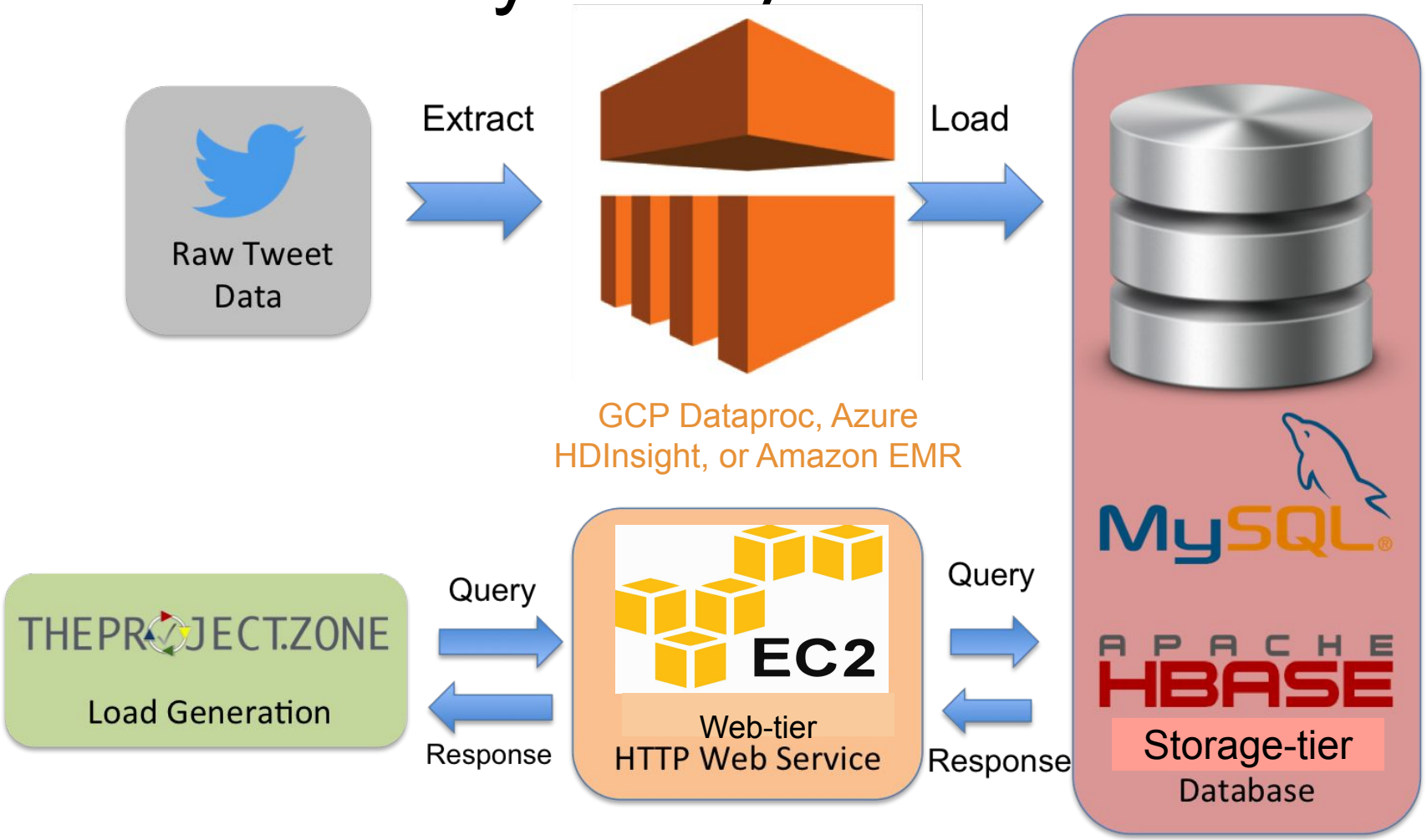
Team Project

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



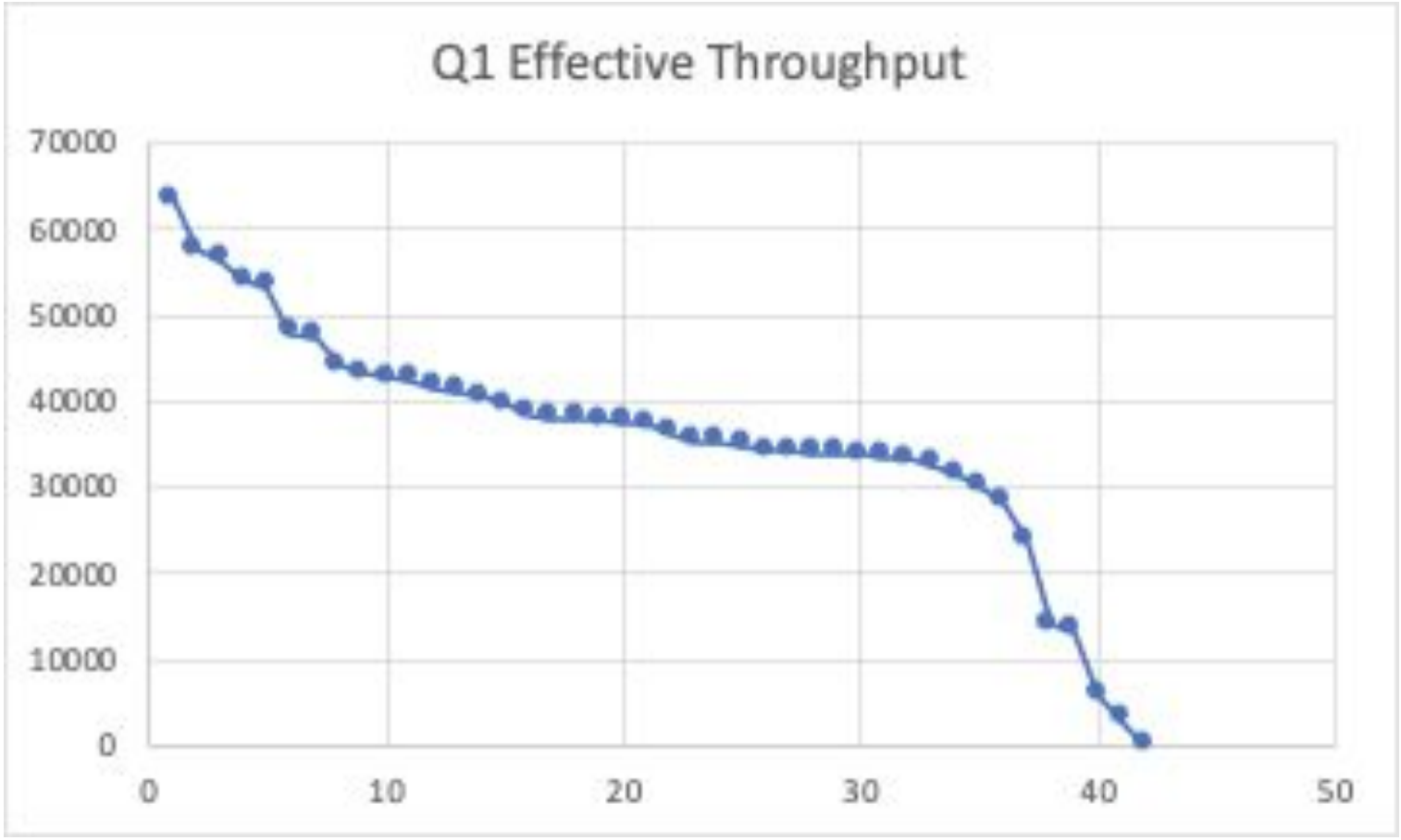
Twitter Analytics System Architecture



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

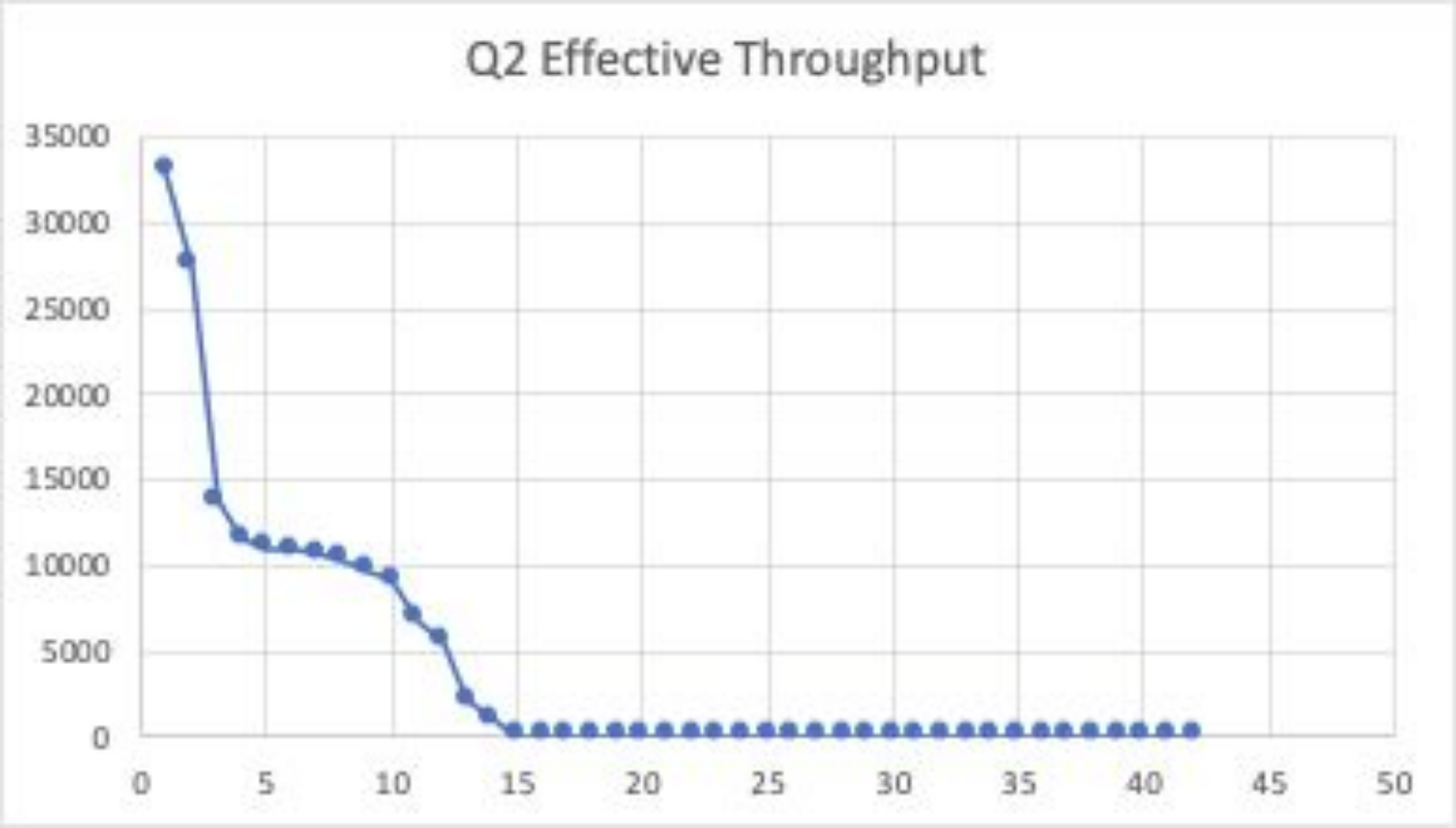
Team Project - Query 1

- 33 teams reached target RPS



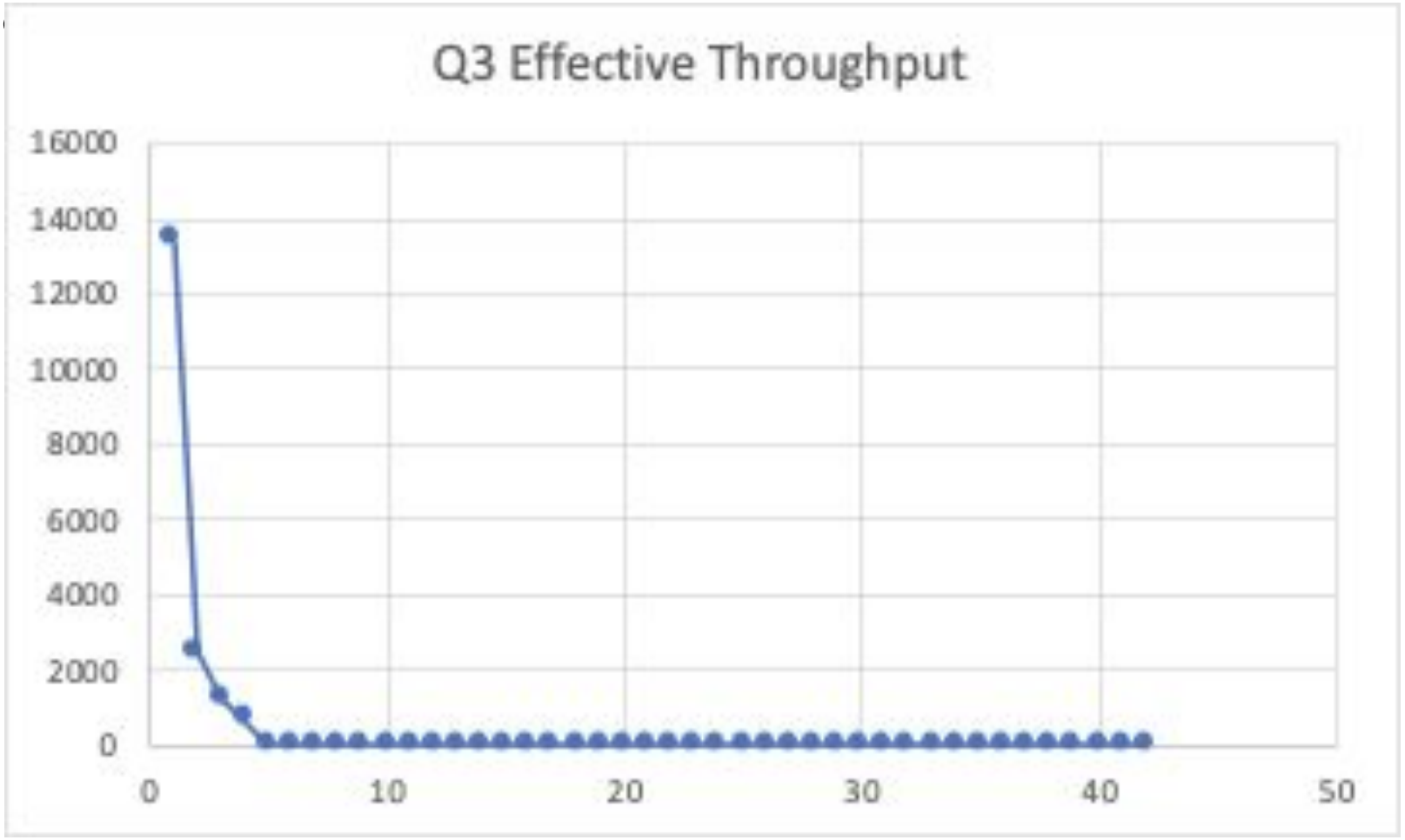
Team Project - Query 2

- 11 teams passed 30% RPS in both MySQL and HBase
- 4 teams reached target RPS in both databases



Team Project - Query 3

- 5 teams attempted Q3
- 2 teams passed 30% RPS in both MySQL and HBase



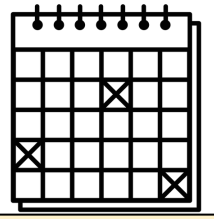
Team Project

- Phase 1:
  - Q1
  - Q2 (MySQL **AND** HBase)

- Phase 2
  - Q1
  - Q2 & Q3 (MySQL **AND** HBase)

- Phase 3
  - Q1
  - Q2 & Q3 (MySQL **OR** HBase)

Team Project Time Table



Phase (and query due)	Start	Deadlines	Code and Report Due
Phase 1 • Q1, Q2	Monday 02/24/2020 00:00:00 ET	Checkpoint 1, Report: Sunday 03/08/2020 23:59:59 ET Checkpoint 2, Q1: Sunday 03/22/2020 23:59:59 ET Phase 1, Q2: Sunday 03/29/2020 23:59:59 ET	Phase 1: Tuesday 03/31/2020 23:59:59 ET (upload PDF report and verify your submission)
Phase 2 • Q1, Q2,Q3	**Monday 03/30/2020 00:00:00 ET**	Q3 Early Bird Bonus: Sunday 04/05/2020 23:59:59 ET **Phase2 Due: Sunday 04/12/2020 15:59:59 ET**	
Phase 2 Live Test (Hbase **AND** MySQL) • Q1, Q2, Q3	**Sunday 04/12/2020 17:00:00 ET**	**Sunday 04/12/2020 23:59:59 ET**	**Tuesday 04/14/2020 23:59:59 ET (upload PDF report and verify your submission)**
Phase 3 • Q1, Q2, Q3 (Managed services)	Monday 04/13/2020 00:00:00 ET	Sunday 04/26/2020 15:59:59 ET	
Phase 3 Live Test • Q1, Q2, Q3 (Managed services)	Sunday 04/26/2020 17:00:00 ET	Sunday 04/26/2020 23:59:59 ET	Tuesday 04/28/2020 23:59:59 ET

Team Project Deadlines

- Phase 2 milestones:
  - Phase 2, Live test: on **Sunday, April 12**
    - HBase:
      - Q1/Q2/Q3/mixed
    - MySQL:
      - Q1/Q2/Q3/mixed
  - Phase 2, code, scripts and report:
    - due on **Tuesday, April 14**

Live Test Schedule - setup

Submit DNS for Live Test

Information

Time	Task	Description
4:00 pm	HBase	Submit your DNS for the HBase Live Test before the deadline
4:00 pm	MySQL	Submit your DNS for the MySQL Live Test before the deadline
5:30 pm - 5:31 pm	HBase DNS Validation	Validate your HBase DNS. This is the last chance to update your DNS for the HBase Live Test
5:33 pm - 5:34 pm	MySQL DNS Validation	Validate your MySQL DNS. This is the last chance to update your DNS for the MySQL Live Test

Live Test Schedule - HBase

HBase Live Test

Information

Time	Value	Target	Weight
6:00 pm - 6:25 pm	Warm-up (Q1 only)	0	0%
6:25 pm - 6:50 pm	Q1	32000	6%
6:50 pm - 7:15 pm	Q2	10000	10%
7:15 pm - 7:40 pm	Q3	1500	10%
7:40 pm - 8:05 pm	Mixed Reads(Q1,Q2,Q3)	10000/1500/500	4+5+5 = 14%

Half-time Break

Information

Time	Value
8:05 pm - 8:30 pm	Time to relax and prepare for the MySQL Live Test

Live Test Schedule - MySQL

MySQL Live Test

Information

Time	Value	Target	Weight
8:30 pm - 8:55 pm	Warm-up (Q1 only)	0	0%
8:55 pm - 9:20 pm	Q1	32000	6%
9:20 pm - 9:45 pm	Q2	10000	10%
9:45 pm - 10:10 pm	Q3	1500	10%
10:10 pm - 10:35 pm	Mixed Reads(Q1,Q2,Q3)	10000/1500/500	4+5+5 = 14%

AWS Budget Reminder

- Phase 2 budget is $60, with a double budget penalty at $100.

	No penalty	-10% grade penalty	-100% grade penalty
Total budget	$60	$60 - $100	$100+
Live Test budget	~$20	~$20	~$20
Development budget	~$40	~$40 - ~$80	~$80+

- Use GCP and Azure for ETL.
- Use spot instances to reduce spending during development.

Hourly Budget Reminder

- Your web service should cost ≤ **$0.89/hour**, including:
  - EC2
    - We evaluate your cost using the On-Demand Pricing towards **$0.89/hour** even if you use spot instances.
  - EBS & ELB
  - Ignore data transfer and EMR cost
- Phase 2 - Live Test Targets:
  - Query 1 - 32000 RPS
  - Query 2 - 10000 RPS (for both MySQL and HBase)
  - Query 3 - 1500 RPS (for both MySQL and HBase)
  - Mixed - 10000/1500/500 RPS (for both MySQL and HBase)

Phase 2, Query 3

- **Problem Statement**
  - Given a time range and a user id range, which tweets have the most **impact** and what are the **topic words**?

- Impact score and topic words (see the write up for details)
  - Impact of tweets: Which tweet is “important”? Calculate using the effective word count, favorite count, retweet count and follower count.
  - Topic words: In this given range, what words could be viewed as a “topic”? Done using TF-IDF.

- Request/Response Format
  - Request: Time range, uid range, #words, #tweets.
  - Response: List of topic words with their topic score, as well as a list of tweets (after censoring).

Phase 2, Query 3 FAQs

Question 1: How to calculate the topic score?

For word ***w*** in the given range of tweets, calculate:

- Calculate the Term Frequency of word *w* in tweet $t^{(i)}$
- Calculate Inverse Document Frequency for word *w*
- Calculate Impact Score of each tweet
- Topic Score for word w =

$$\sum_{i}^{n} TF(w, t^{(i)}) \cdot IDF(w) \cdot ln(Impact(t^{(i)}) + 1),$$

for n tweets in time and uid range

Phase 2, Query 3 FAQs

Question 2: When to censor? When to exclude stop words?

- Censor in the Web Tier or during ETL. It is your own choice.
  - If you censor in ETL, consider the problem it brings to calculating the topic word scores (two different words might look the same after censoring).
- You should count stop words when counting the total words for each tweet in order to calculate the topic score.
- Exclude stop words when calculating the impact score and selecting topic words.

General Hints

- Completely understand every AssessMe question.
- There are some useful tips for improving HBase performance in the writeup of the NoSQL primer, HBase primer and P3.1.
- Understand different metrics (e.g., locality, number of read requests) in HBase UI (port 16010) and HDFS UI (port 50070).

General Hints

- Remember that you can put the web-tier and storage-tier on the **same** instance.
- Profile your cloud service and think about which component is the bottleneck.
- Optimization is **time-consuming**. Before ETL, please
  - Think about your schema design (rowkey for HBase in particular).
  - Think about your database configuration.

Q2 Hints

- Consider replication and sharding in databases
- Identify latency between web server and database
- Design a suitable schema for a specific problem
  - Remember: Query 2 is a read-only problem
- Avoid using scan in HBase for Query 2
- Choose a suitable primary key in HBase
  - Which one can be used as key based on Query 2 request?
  - How to design a schema that use such a key?
- Balance workload between web server and database

Q3 Hints

- Completely understand the definition of a **word**. This is different for text censoring and calculating scores.
- A query contains two ranges. Log some requests to get an idea on the range of user_id and timestamps.
- Balance the requests through all the regions.
  - Presplitting
  - HBase Load Balancer (Monitor the HBase UI during writing)
- HBase data is local when it is written, but when a region is moved, it is not local until compaction.

Hints for the live test

- The request pattern will differ for Phase 2 submission test and the live test so your solution should handle all types of load.
- Lookup what commands you can use to learn about the aspects of your web service health.
- Monitor your system during the live test to recover in case of a system crash. Be prepared with your monitoring consoles setup.
- Understand and keep an eye on
  - EC2 CPU Credits and burstable performance
  - EBS volume I/O Credits and Burst Performance
- Take cloudwatch snapshots.

Warning

- NEVER open all ports to the public (0.0.0.0) when using instances on a public cloud.
- For your purposes, you likely only need to open port 80 to the public. Port 22 should be open only to your own machine.
- Port 3306 (for MySQL) and HBase ports should be open only to cluster members if necessary.

Upcoming Deadlines

- P4.1 Spark
  - **Code review due next week**
- Quiz 11
  - **Due: 04/10/2020 11:59 PM Pittsburgh**
- Team Project : Phase 2
  - **Live-test due: 04/12/2020 3:59 PM Pittsburgh**
  - **Code and report due: 04/14/2020 11:59 PM Pittsburgh**

Questions?