

TEAM PROJECT

Twitter Data Analytics



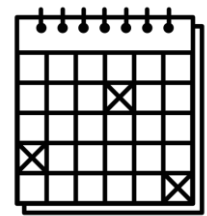
+



=

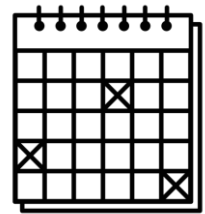


Team Project Time Table



Phase	Deadline (<u>11:59PM EST</u>)
Phase 1 (20%) <ul style="list-style-type: none">- Query 1- Query 2	<ul style="list-style-type: none">● Q1 CKPT (5%): Sun, 3/14● Q1 CKPT Report (5%): Sun, 3/14● Q1 FINAL (10%): Sun, 3/21● Q2 CKPT (10%): Sun, 3/21● Q2M & Q2H FINAL (50%): Sun, 3/28● Final Report (20%): Tue, 3/30
Phase 1 (20%) <ul style="list-style-type: none">- Query 1- Query 2	BONUSES: <ul style="list-style-type: none">● Q1 Early Bird Bonus (5%): Sun, 3/14● Q2M & Q2H Early Bird Bonus (5%): Sun, 3/21● Q2 Correctness Penalty Waiver: Sun, 3/21● Q2M & Q2H Performance Bonus (5%): Sun, 3/28

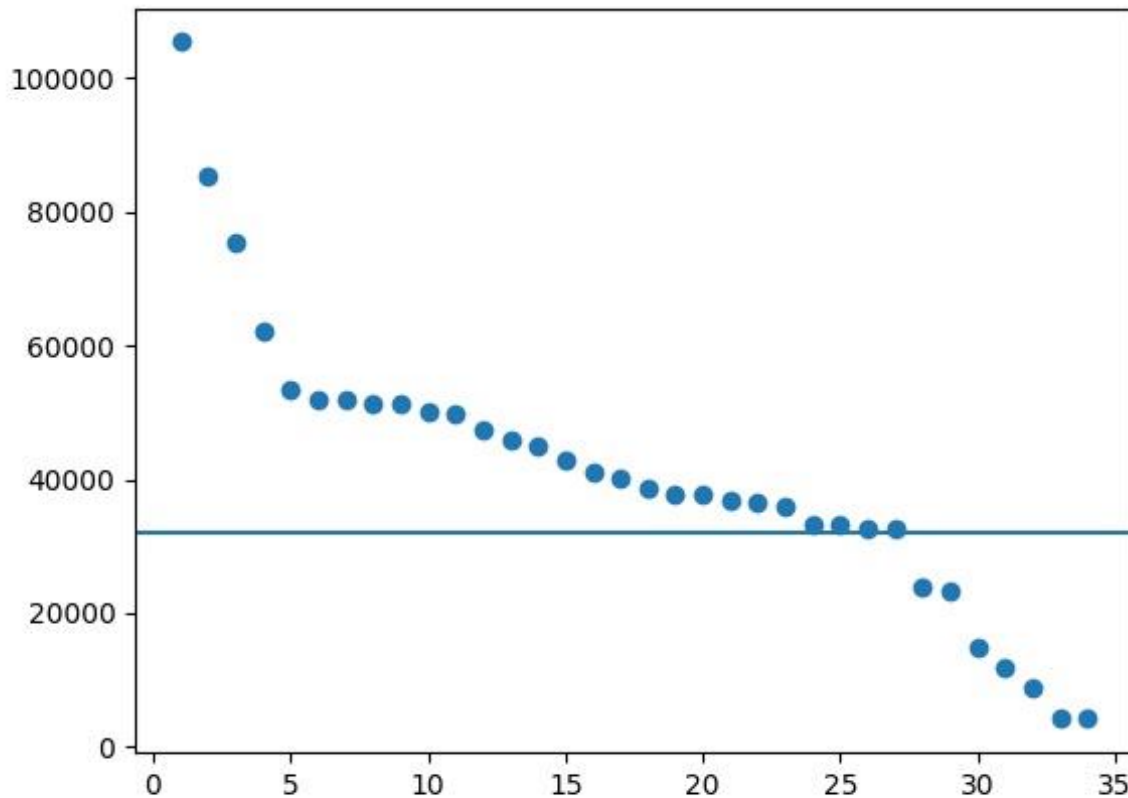
Team Project Time Table



Phase	Deadline (<u>11:59PM EST</u>)
Phase 2 (30%) - Add Query 3	● Live Test on Sun, 4/18
Phase 3 (50%) - Managed Services for Query 1-3	● Live Test on Sun, 5/2

Team Project - Query 1

- 27/34 teams reached 32,000 RPS.
- Team Hello(Yuan Gu, Junda An, Xuchen Zhang) got over 100k !



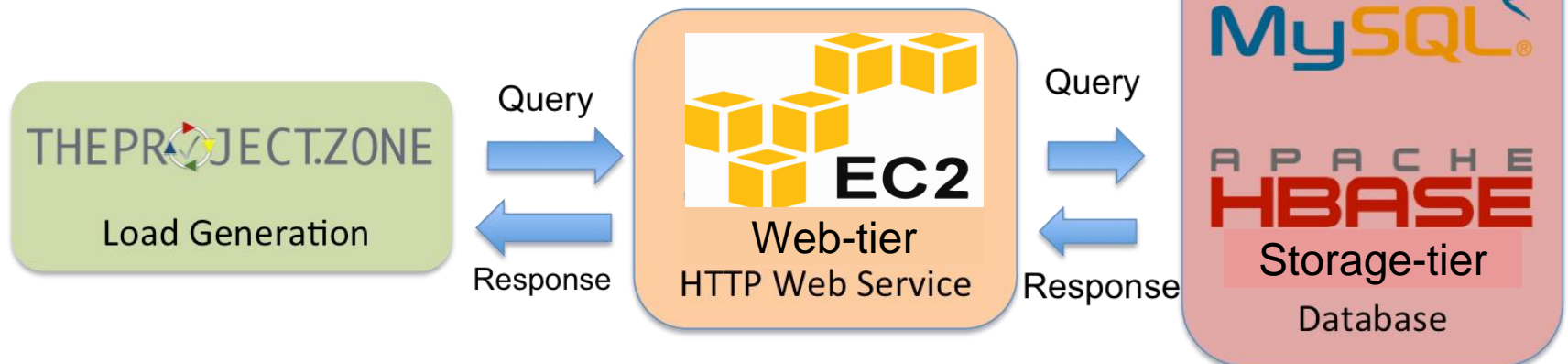
Team Project - Query 2

- Team CloudCrew (Aditya Shetty, Chih-Wei Fang and Pranav Dheer) got full score for MySQL and Hbase !
 - Also got Early Bird Bonus and Correctness Penalty Waiver

Query 2 - Recap

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



Query 2 - Recap

target throughput: 10,000 RPS for both MySQL and HBase

Use Case: When you follow someone on twitter, recommend users you may also be interested in

Query: GET

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

Response:

```
<TEAMNAME>,<AWSID>\nuid\tname\tdescription\ttweet\nuid\tname\tdescription\ttweet
```

Three Scores:

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

Final Score: Interaction Score * Hashtag Score * Keywords Score

Be prepared to reiterate your solution

- Your first solutions are very unlikely to be your last solution
- Design your ETL process carefully to make it reiterate-friendly
 - Do you have to redo the ETL from scratch?
- A typical iteration:
 - Implement your current solution
 - Identify bottlenecks with profiling tools
 - Web-tier, DB schema, Network, ...
 - Design a solution that would most likely improve your current bottleneck
 - Possibly rerun ETL to fit your new design
- It's a number game. Usually teams that went through the most iterations prevail

Load Data & Backup

- Refer to [MySQL Primer](#) and [HBase Primer](#) for dataloading
 - Experiment on both bulk and non-bulk loading for Hbase
 - Be very careful about escape characters.
 - Be very careful about encodings.
- Backup
 - There are various ways for you to backup your MySQL database, e.g., [mysqldump](#).
 - For HBase, you can backup and restore HBase database on S3 using the [HBase snapshot](#).

Performance Tuning Tips

- Reiteration ranks higher than parameter tuning
 - Do not waste time tuning parameters when you have only one tenth of the target RPS!
- 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
 - Use CloudWatch for resource utilization such as CPU, Network, Disk, etc
 - Use 'top' and 'iotop' to monitor your instance in real time

Performance Tuning Tips (cont.)

- Web Tier
 - Connection pooling?
 - Caching result?
 - Is the workload distributed evenly on multiple web servers?
 - 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)

Performance Tuning Tips (cont.)

- Database Tier - MySQL
 - Different MySQL engines
- Database 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
 - Check the official documentation
 - Search for performance tuning best practices

Reminders on penalties

- M family instances **only**, smaller than or equal to **large** type
- Other types are allowed (e.g., t2.micro) **but only for testing**
 - Using these for any submissions = 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).
- **\$0.70/hour (MySQL) and \$0.85/hour (HBase)** applies to every submission

Phase 1 Budget

- Your web service should not cost more than **\$0.70/hour (Q1 and Q2 MySQL)** and **\$0.85/hour (Q2 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**
 - We will not consider the cost of data transfer and EMR
 - See the writeup for details
- AWS total budget of \$55 for Phase 1

Piazza Team Project Hint Thread

We will keep posting hints and clarifications in this thread, please check it frequently

<https://piazza.com/class/kjvyqvk35s355?cid=1152>