15-319 / 15-619 Cloud Computing

Recitation 12

April 10th 2018

Overview

- **Last week's reflection**
  - Project 4.1
  - Unit 5 - Modules 19 and 20
  - Quiz 10
- **This week's schedule**
  - Team Project, Phase 2, Queries, 1, 2, 3
  - Live test!
  - Unit 5 - Modules 21 and 22
  - Quiz 11 (last quiz!)
- **Twitter Analytics: The Team Project**

Reminders

- Monitor AWS expenses regularly and tag all resources
  - Check your bill both on AWS and TPZ
- Piazza Guidelines
  - If you need us to debug a specific submission, please give your submission ID
  - Tag your question with the correct project
  - If you have a grading issue, please provide your andrew ID
- Utilize Office Hours
  - Take full advantage of the office hours

Apply to Become an F18 TA

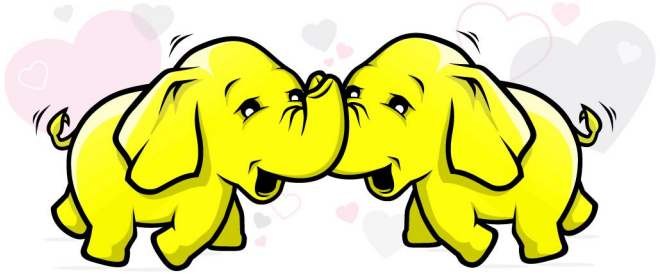
- Please apply using the following link:
  - https://goo.gl/forms/FX2x0t04zbrZBStf2
- The TAship will offer invaluable experience.
- TA team will be evaluating changes & improvements.
- Experience with infrastructure & microservices.
- We can scale the TA workload.
- The deadline to apply is:
  - Wednesday 04/11 at 11:59 pm ET.

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

Project 4.1 Reflection

- Designing a MapReduce job to maximize resource allocation and usage on a Hadoop cluster
- Exploring the usage of advanced MapReduce features such as the Partitioner class and Input/OutputFormat
- Utilizing the Hadoop UI to access logs and facilitate in locating bugs in complex MapReduce programs
- Creating MapReduce programs to write data HBase and Redis
- Building a text predictor from Wikipedia data and displaying the data on a web interface



Project 4.1 Reflection

- Please practice better time management
  - Project 4.2 involves big data analysis
  - Long running jobs and possibly many edge cases

- Common issues
  - Saving state in the Reducer class between different invocations
  - Dealing with small files on Hadoop MapReduce
  - Regexes that only worked on the sample dataset

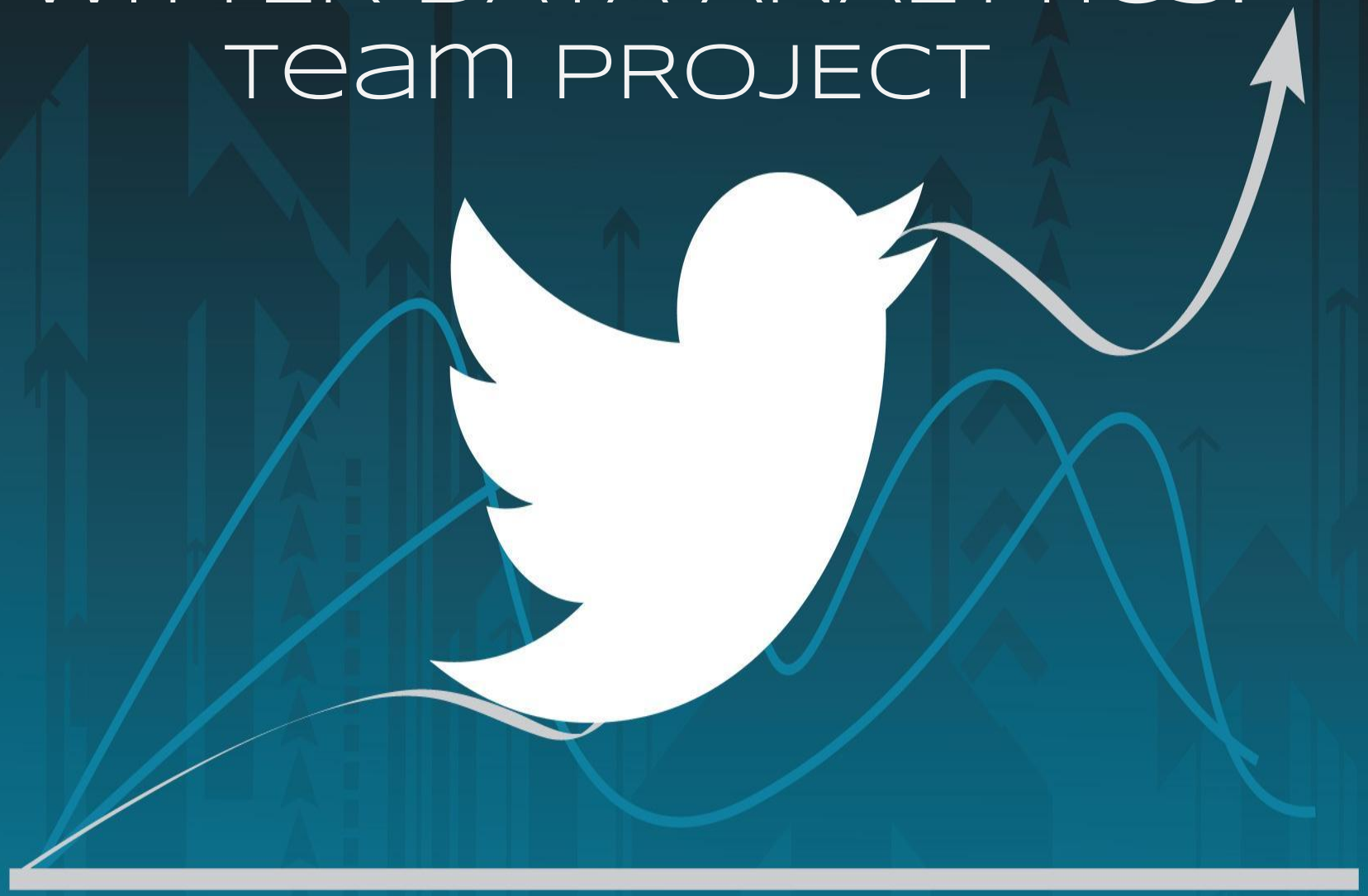
- Debugging best practices
  - Examine container logs from the UI

Phase 1 Report Feedback

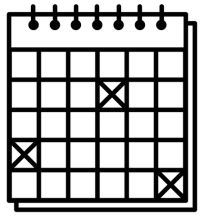
- Good job for most teams
- Reminders based on the Phase 1 Report:
  - Some teams exceeded the per-hour budget
  - Cache might harm your live-test performance
- The report helps you think about
  - The starting point for optimization: what to try, where is the bottleneck in our current implementation
  - How to further improve Q1 and Q2, and guide you to think about Q3
- If you explored enough for each question, you should have a lot to say in report.
- Hope you will have more “fancy tricks” or optimizations to share in future reports!!
  - Work hard to become one of the top teams!!!

Questions?

TWITTER DATA ANALYTICS: Team PROJECT



Team Project Time Table



Phase (and query due)	Start	Deadlines	Code and Report Due
Phase 1 • Q1, Q2	Monday 02/26/2018 00:00:00 ET	**Checkpoint 1, Report: Sunday 03/11/2018 23:59:59 ET** **Checkpoint 2, Q1: Sunday 03/25/2018 23:59:59 ET** **Phase 1, Q2: Sunday 04/01/2018 23:59:59 ET**	Phase 1: Tuesday 04/03/2018 23:59:59 ET
Phase 2 • Q1, Q2,Q3	Monday 04/02/2018 00:00:00 ET	**Sunday 04/15/2018 15:59:59 ET**	
Phase 2 Live Test (Hbase **AND** MySQL) • Q1, Q2, Q3	Sunday 04/15/2018 17:00:00 ET	**Sunday 04/15/2018 23:59:59 ET**	**Tuesday 04/17/2018 23:59:59 ET**
Phase 3 • Q1, Q2, Q3, Q4	Monday 04/16/2018 00:00:00 ET	Sunday 04/29/2018 15:59:59 ET	
Phase 3 Live Test (Hbase **OR** MySQL) • Q1, Q2, Q3, Q4	Sunday 04/29/2018 17:00:00 ET	Sunday 04/29/2018 23:59:59 ET	Tuesday 05/01/2018 23:59:59 ET

Team Project Phase 1 Review

- Q1 Task:
  - Front end (web tier) development.
  - QR code encoding and decoding algorithm

- Q2 Tasks:
  - ETL + web tier + database tier,
  - **SELECT** query on MySQL (Relational DBMS),
  - **GET** query on HBase (NoSQL)

Phase 1, Query 1 Tips

- Choose the web framework wisely based on quantitative testing as this will help you in the future queries.
- Set an appropriate number of threads to handle the incoming requests.
- Warm up the load balancer adequately before you begin testing.
- Use the allocated budget to the fullest.
- Lastly, optimize your code as much as possible.

Phase 1, Query 2 Implementation Tips

- Use regex “\p{L}+” to match words in Java
  - So that we only match the unicode letters.
- Treat all contents in the `text` field the same, including hashtag. Meaning if the text contains hashtags, these hashtags can be considered as a word and need to be included as a keyword.
- Keywords and hashtags are case insensitive.

Phase 1, Query 2 Performance Tips

- MySQL
  - Experiment with different MySQL database engines
  - Try out different schemas, a schema that appears better on paper may not give you the best performance.
  - It is possible to fetch the relevant data in one database query.
  - Use EXPLAIN to write better queries and analyze the query.
  - If you are using an AMI with the database already loaded, note that the disk is lazily loaded from S3.

Phase 1, Query 2 Performance Tips

- HBase
  - Avoid using a scan query when a **GET** query is possible.
  - Since the data is read-only, pre-splitting can be done to shard the data equally among all the HBase nodes.
  - Larger number of region servers will lead to better load balancing.
  - Play around with the Block Cache and Block Size.
  - Make use of the Web UI provided to analyze the performance of HBase.

Team Project Phases 2

- Q1
  - Building a heartbeat - QR code encoding and decoding algorithm
- Q2
  - Handling complex read-only queries.
  - Doing ETL, building, configuring and optimizing both the Web Tier and Database Tier.
  - Explore both MySQL and HBase.
- Q3
  - Handling range read requests
  - Try more optimizations in Web/DB tier
    - DB schema and configurations, explore optimizations given the type of the query. eg. point vs range read

Phase 2, Query 3

- **Problem Statement**
  - In 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.

Performance Tuning Tips

- To do performance tuning, you first need to identify which part of your system is the bottleneck.
  - Do profiling and monitoring on your system
    - Write a LG yourself to test your system performance
    - Use CloudWatch for resource utilization such as CPU, Network, Disk, etc.
- There is a detailed primer on profiling a cloud service which was designed mainly for the team project.

Performance Tuning Tips

- Think about the architecture of your system and what advantages and disadvantages it has compared to other settings
  - Sharding vs Replication
  - ELB vs Customized LB
  - Doing the calculation in Web Tier vs in ETL, etc.

Performance Tuning Tips

- Web Tier
  - Did you put too much computation at the Web Tier that could have been done beforehand?
  - If you have multiple Web Servers, is the workload distributed evenly?

Performance Tuning Tips

- Database Tier
  - Try to reduce the number of rows and the size of data retrieved in each request.
  - Remember that Q2 & Q3 are read-only.
    - You can choose schemas that are specifically optimized for Q2 & Q3.

Performance Tuning Tips

- Database Tier - MySQL
  - Tune the parameters
    - Storage engine
    - Buffer pool size
    - Many more params
- Database Tier - HBase
  - Tune certain parameters
    - Block size (4096/8192 etc.)
    - Block Cache size
    - Compression (snappy/gzip etc.)
  - Consider your data distribution and try to identify hotspots
  - Scans can be really slow, try to avoid them when possible
    - If not, try to scan as few rows as possible

Performance Tuning Tips

- Lastly, remember what we have learned in previous project modules
  - Scaling out
  - Load balancing
  - Replication and Sharding
- Ask on Piazza or go to office hours if you are stuck for too long!

Budget and Targets Reminder

- Your team has a total AWS budget of **$50** for Phase 2
- Your web service should cost ≤ **$0.83/hour**, including:
  - EC2
    - We evaluate your cost using the On-Demand Pricing towards **$0.83/hour** even if you use spot instances.
  - EBS & ELB
  - Ignore data transfer and EMR cost
- Live Test Targets:
  - Query 1 - 28000 rps
  - Query 2 - 8000 rps (for both MySQL and HBase)
  - Query 3 - 1500 rps (for both MySQL and HBase)
  - Mixed Reads (TBA)

Phase 2 Requirements

- Phase 2 accounts for 30% of the total score of the Team Project
  - Phase 1 only accounts for 20%
- You need to continue exploring MySQL and HBase in Phase 2
  - You will continue to work on Q1 & Q2
  - You will work on a new query, Q3
- **You must achieve over 80% correctness**, **AND at least 50% RPS in BOTH MySQL and HBase** in order to get points for each query.
- Your performance RPS is **SOLELY** determined by the Live-Test
  - Some students cached query results at front-end in phase 1
    - If not done wisely, this may lead to the web service crashing during the Phase 2 Live-Test
- As before, a report needs to be submitted for Phase 2 after the Live-Test
  - Check the schedule for deadlines

Notes for the Live Test

- During the live test, you must tag your HBase and MySQL cluster with Key: teambackend Value: hbase and Key: teambackend and Value: mysql.
- You must submit both your clusters' DNS before 4 pm, Sunday April 15th. Both of your clusters must be ready then.
- You must use the same cluster for all queries. So you can't launch different MySQL clusters for Q2 and Q3.
- Do not launch other testing instances during live test, or else we will count them towards your hourly budget.
- We encourage you to use on-demand instances for the live test or else you run the risk of your instances being shut down unexpectedly
- Leave enough budget for the Live Test. About $20 should be safe.

Hints for the live test

- The request pattern will differ for Phase 2 and the live test so your solution should handle all types of load.
- Monitor your system efficiently during the live test to recover in case of a system crash.
- Think about what could happen when your cluster should respond to mixed requests of Q1, Q2 and Q3. And try to convince yourself that your design and optimizations are not over aggressive towards some of the queries, and you have reasonable resources (CPU, Disk, Memory, DB&Web Tier) usage.
  - Think and try, but we won't release mixed requests before live test

Phase 2 Live Test - Hbase

Time	Value	Target	Weight
04/15 at 3:59pm	Deadline to submit the DNS (MySQL and HBase) of your Web Service		
5:30pm - 5.34pm	DNS validation for both MySQL and HBase		
6:00pm - 6.25pm	Warm-up (Q1 only)	0	0%
6:25pm - 6:50pm	Q1	28000	6%
6:50pm - 7:15pm	Q2	8000	10%
7:15pm - 7:40pm	Q3	1500	10%
7.40pm - 8:05pm	Mixed (Q1,Q2,Q3)	TBA/TBA/TBA	4+5+5=14%

- You need to achieve at least 50% of the target RPS and 80% correctness for BOTH MySQL and HBase to get a score for a query!

Phase 2 Live Test - MySQL

Time	Value	Target	Weight
8:30pm - 8.55pm	Warm-up (Q1 only)	0	0%
8:55pm - 9:20pm	Q1	28000	6%
9:20pm - 9:45pm	Q2	8000	10%
9:45pm - 10:10pm	Q3	1500	10%
10:10pm - 10:35pm	Mixed (Q1,Q2,Q3)	TBA/TBA/TBA	4+5+5=14%

- You need to achieve at least 50% of the target RPS and 80% correctness for BOTH MySQL and HBase to get a score for a query!

Upcoming Deadlines

- Quiz 11: Unit 5 - Modules 21 & 22

  Due: Friday, April 13 2018 11:59PM ET

- Team Project : Phase 2
  - Live-test due: Sunday, April 15, 2018 3:59 PM ET
  - Code & report due: Tuesday, April 17, 2018 3:59 PM ET

Questions?