15-319 / 15-619 Cloud Computing

Recitation 12

April 10th 2017

Overview

- **Last week's reflection**
  - Project 4.1
  - Quiz 10
- **This week's schedule**
  - Team Project, Phase 2
  - Quiz 11
- **Twitter Analytics: The Team Project**
  - Phase 2, Live Test

Reminders

- Monitor AWS expenses regularly and tag all resources
  - Check your bill (Cost Explorer > filter by tags)
- Piazza Guidelines
  - Please tag your questions appropriately
  - Search for an existing answer first
- Provide clean, modular and well documented code
  - Large penalties for not doing so
  - **Double check** that your code is submitted!! (verify by downloading it from TPZ from the submissions page)
- Utilize Office Hours
  - We are here to help (but not to give solutions)
- Use the team AWS account and tag the Team Project resources carefully

Conceptual Modules to Read on OLI

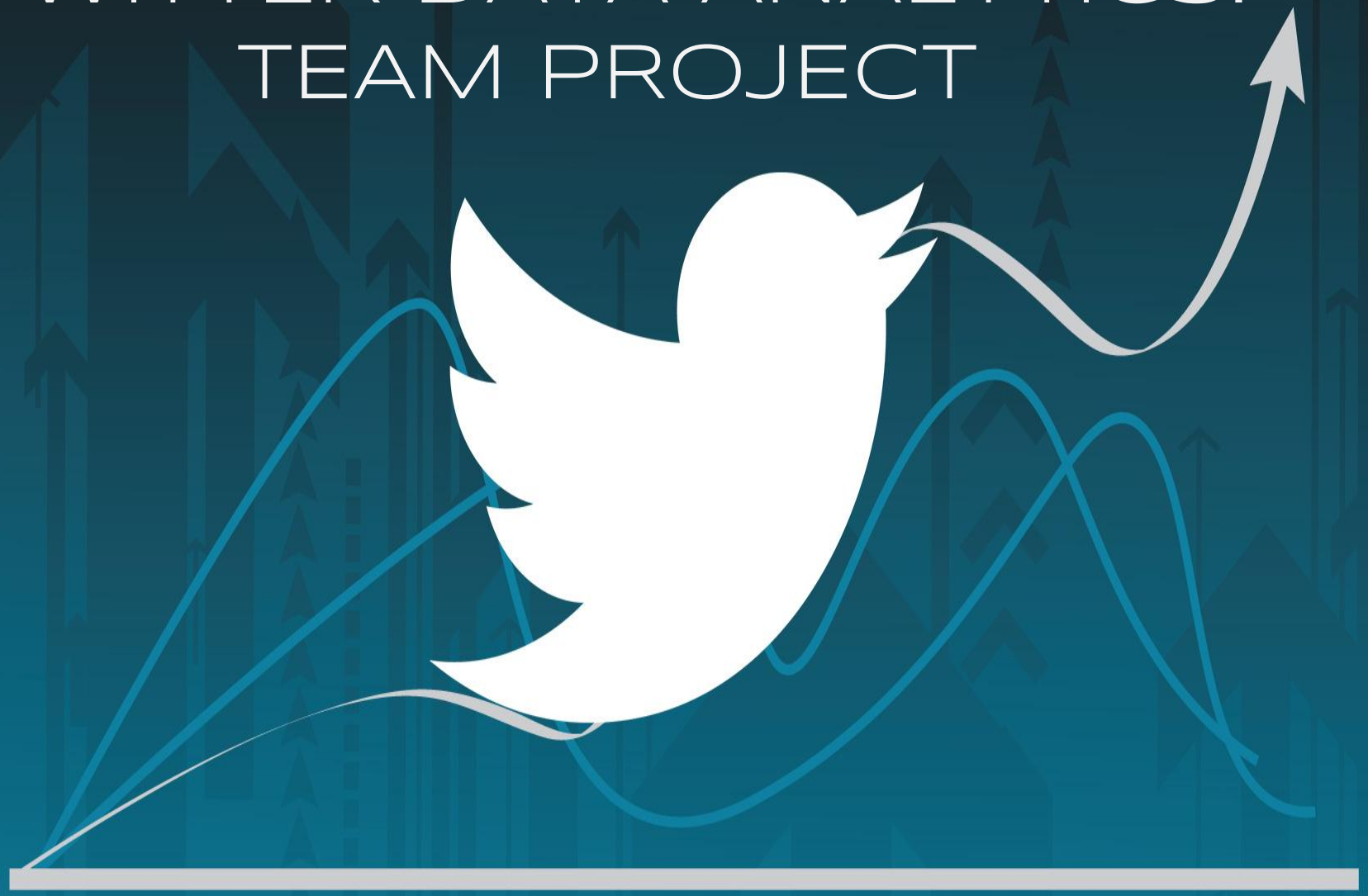
- UNIT 5: Distributed Programming and Analytics Engines for the Cloud
  - Module 18: Introduction to Distributed Programming for the Cloud
  - Module 19: Distributed Analytics Engines for the Cloud: MapReduce
  - Module 20: Distributed Analytics Engines for the Cloud: Spark
  - Module 21: Distributed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

P4.1 Reflection

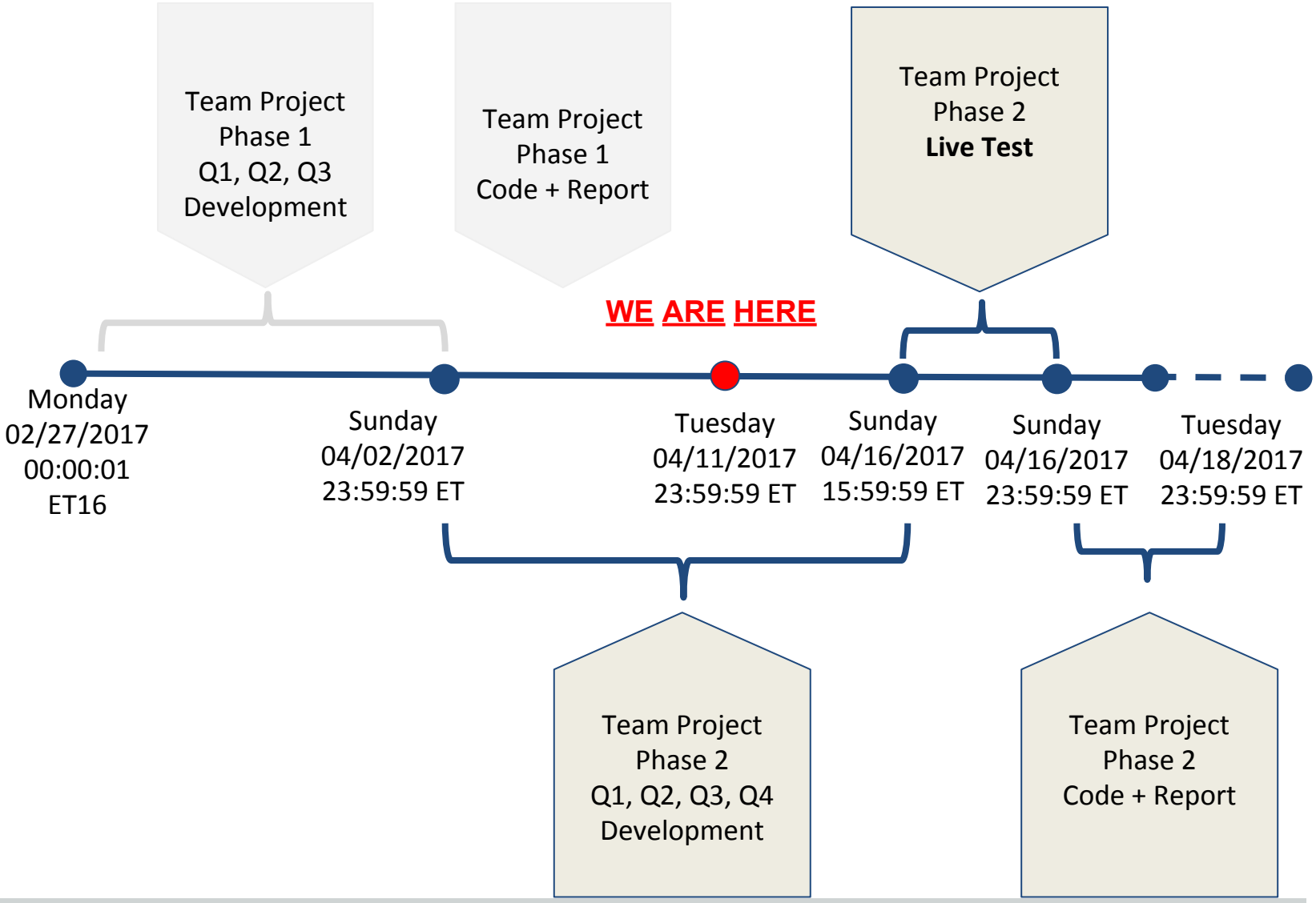
Takeaways from P4.1:

- Use the MapReduce Model and think like mappers and reducers. Understand the complete workflow and configurations to design and implement MapReduce applications.
- Work with an EMR cluster; use the UI to find logs and locate problems; and solve any dependency issues.
- Load data to HBase and Redis using a custom MapReduce program.
- Combiner, OutputFormat, etc.
- Start from a small demo data set, account for edge cases especially with the large dataset! Some bugs might look like memory or YARN scheduling issues, but they could be bugs in your program!
- Store probability rather than count!

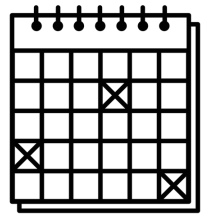
TWITTER DATA ANALYTICS: TEAM PROJECT



Team Project Phase 2 Deadlines



Team Project Time Table



Phase	Query	Start	Deadline	Code and Report Due
Phase 1	Q1	Monday 2/27/2017 00:00:01 EST	Sunday 3/12/2017 23:59:59 EST	-
	Q2 & Q3	Monday 2/27/2017 00:00:01 EST	Sunday 4/2/2017 23:59:59 EST	Tuesday 4/4/2017 23:59:59 EST
Phase 2	**Q1, Q2, Q3, Q4**	**Monday 4/3/2017 00:00:01 EST**	**Sunday 4/16/2017 15:59:59 EST**	**-**
	Live Test Q1, Q2, Q3, Q4	**Sunday 4/16/2017 18:00:01 ET**	**Sunday 4/16/2017 23:59:59 EST**	**Tuesday 4/18/2017 23:59:59 EST**
Phase 3	**Q1, Q2, Q3, Q4, Q5**	**Monday 4/17/2017 00:00:01 ET**	**Sunday 4/30/2017 15:59:59 EST**	**-**
	Live Test Q1, Q2, Q3, Q4, Q5	**Sunday 4/30/2017 18:00:01 ET**	**Sunday 4/30/2017 23:59:59 EST**	**Tuesday 5/2/2017 23:59:59 EST**

Honor board

Team “cc is my god” got full score for both Q4 MySQL and Q4 HBase before Sunday Apr 9th 11:59pm. Congratulations on getting the early bird bonus!

	Q4 MySQL	Q4 HBase
cc is my god	**8564.85**	**7721.81**

Phase 2

- 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 & Q3
  - You will work on a new query, Q4
- **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 front-end 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

Phase 2

- Budget on AWS for Phase 2 is $50 (including the Live Test), if you spend more than $75 you will receive a 100% penalty.
- You have a $0.88/hr budget for MySQL and HBase separately in the Live Test. You will receive (X-0.88)*2% penalty if you spend X dollars > $0.88. The hourly cost includes:
  - **EC2**
    - We evaluate your cost using the On-Demand Pricing towards **$0.88/hour** even if you use spot instances.
  - **EBS & ELB**
  - Ignore data transfer and EMR cost
- We encourage you to do ETL on Azure and GCP to save your budget on AWS

Phase 2

- 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 Apr 16th. Both of your clusters should be ready then.
- You must use the same cluster for all queries. So you can't launch different MySQL clusters for Q2, Q3 and Q4.
- 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 shutted down unexpectedly
- Leave enough budget for the Live Test. About $20 should be safe.

Phase 2 Live Test - Hbase

Time	Value	Target	Weight
04/16 at 3:59pm	Deadline to submit the DNS of your Web Service		
6:00pm - 6.25pm	Warm-up (Q1 only)	0	0%
6:25pm - 6:50pm	Q1	30000	4%
6:50pm - 7:15pm	Q2	11000	8%
7:15pm - 7:40pm	Q3	2500	8%
7:40pm - 8:05pm	Q4	7500	8%
8:05pm - 8:30pm	Mixed (Q1,Q2,Q3,Q4)	7500/2800/700/1900	3+3+3+3=12%

- 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!
- Report is worth 20% of the grade in this phase!

Phase 2 Live Test - MySQL

Time	Value	Target	Weight
9:00pm - 9.25pm	Warm-up (Q1 only)	0	0%
9:25pm - 9:50pm	Q1	30000	4%
9:50pm - 10:15pm	Q2	11000	8%
10:15pm - 10:40pm	Q3	2500	8%
10:40pm - 11:05pm	Q4	7500	8%
11:05pm - 11:30pm	Mixed (Q1,Q2,Q3,Q4)	7500/2800/700/1900	3+3+3+3=12%

- 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!
- Report is worth 20% of the grade in this phase!

Query 4: Interactive Tweet Server

There are 7 different parameters in the request URL for a request

/q4?**op**=<operation>&**field**=<field name>&**tid1**=<tweet id1>&**tid2**=<tweet id2>&**payload**=<value>&**uuid**=<unique id>&**seq**=<sequence number>

General Info:

1. Four operations:
   - write, set, read and delete
2. Operations under the same **uuid** should be executed in the order of the sequence number (Starting from 1).
3. Be aware of malformed queries.

Query 4: Interactive Tweet Server

field	type	example
tweetid	long int	15213
userid	long int	156190000001
username	string	CloudComputing
timestamp	string	Mon Feb 15 19:19:57 2017
text	string	Welcome to P4!#CC15619#P3
favorite_count	int	22
retweet_count	int	33

Query 4: Interactive Tweet Server

- Write Request

  `/q4?op=write&field=<empty>&tid1=<empty>&tid2=<empty>&payload=json_string&uuid=unique_id&seq=sequence_number`

- Response

  `TEAMID,TEAM_AWS_ACCOUNT_ID\n`

  `success\n`

- `payload` is the url-encoded json string; same structure as the original tweet json; only contains the seven fields needed. For tid and uid, don't get them from the "id_str" field, only get them from the "id" field.

Query 4: Interactive Tweet Server

- Read Request

  `/q4?op=read&field=<empty>&tid1=tweet_id1&tid2=tweet_id2&payload=<empty>&uuid=unique_id&seq=sequence_number`

- Response

  `TEAMID,TEAM_AWS_ACCOUNT_ID\n tid_n\ttimestamp_n\tuid_n\tusername_n\ttext_n\tfavorite_count_n\tretweet_count_n\n`

- Range read

Query 4: Tweet Server

- Delete Request

  `/q4?op=delete&field=<empty>&tid1=tweet_id&tid2=<empty>&payload=<empty>&uuid=unique_id&seq=sequence_number`

- Response

  `TEAMID,TEAM_AWS_ACCOUNT_ID\n`

  `success\n`

- Delete the whole tweet

Query 4: Tweet Server

- Set Request

  `/q4?op=set&field=field_to_set&tid1=tweet_id&tid2=<empty>&payload=string&uuid=unique_id&seq=sequence_number`

- Response

  `TEAMID,TEAM_AWS_ACCOUNT_ID\n`

  `success\n`

- Set one of the text, favorite_count, retweet_count of a particular tweet
- Payload is url-encoded

Query 4: Tweet Server

- Malformed Request

  /q4?op=set&field=field_to_set&tid1=tweet_id&tid2=<empty>&payload=0;**drop+tables+littlebobby**&uuid=unique_id&seq=sequence_number

- Response

  TEAMID,TEAM_AWS_ACCOUNT_ID\n

  success\n

Query 4: Tweet Server

- Part of the queries
  - To make debugging easier, there will always be a Write or Delete of a tweet before a Read
  - But some of the tweets are read only, we will only submit read requests on these tweets
  - For the Live Test, we will not follow the above rule so don't rely on the above to get a high RPS
  - Don't forget to **reload** the databases before the Live Test

Team Project General Hints

- Don’t blindly optimize for every component, identify the bottlenecks using fine-grained profiling.
- Use caches wisely: cache in HBase and MySQL is obviously important, storing everything in the frontend cache will lead to failure during the Live Test.
- Review what we have learned in previous project modules
  - Scale out
  - Load balancing
  - Replication and sharding
  - Strong consistency (correctness is very important in Q4)
- Look at the feedback of your Phase 1 report!

Team Project, Q4 Hints

- MySQL DBs behind an ELB may require a forwarding mechanism.
- Consider forwarding the requests but pay attention to latency.
- Consider batch writes/updates.
- Pay attention to blocking in the frontend.
- Log the requests and observe the pattern.

Upcoming Deadlines

- Quiz 11: Unit 5 - Module 19 20
  - Due: 04/14/2017 11:59 PM Pittsburgh 
- Team Project: Phase 2, Q1, Q2, Q3, Q4 (M&H)
  - Live-test DNS due: 04/16/2017 3:59 PM Pittsburgh 
  - Code and report due: 04/18/2017 11:59 PM Pittsburgh 

Questions?