15-319 / 15-619 Cloud Computing

Recitation 9

March 17th and 19th, 2015

Overview

- **Administrative issues**
  - Tagging, 15619Project
- **Last week's reflection**
  - Project 3.2
- **This week's schedule**
  - Project 3.3
  - Unit 4 - Module 13
- **Demo**
- **Twitter Analytics: The 15619Project**

Caution!

- Tag spot instances in the FIRST 59 mins.
  - Otherwise, it will be considered as an untagged instance for that hour.
- 15619Project is in progress!
  - Phase 1 report and code due on Thursday, 3/19
  - Phase 2 is released on Thursday, 3/19
  - Meet your TA mentor every week to get the token for the Query Reference Server.
  - Tag all resources used for 15619Project as
    Key: 15619project, Value: phase1

    Key: 15619backend, Value: hbase/mysql

Project 3.2 : FAQs - 1

Problem 1: I'm running the *update* YCSB benchmark against the replicated mysql cluster. Why am I not able to get a tick on the scoreboard?

- You might have forgotten to load the data before running the benchmark. If there isn't data in the table, the benchmark is not testing the actual throughput of the update YCSB benchmark.

Project 3.2 : FAQs - 2

Problem 2: My management node/API node does not start.

- Check log files for detailed error messages.
  - under /var/lib/mysql-cluster/ folder in management node.
  - under /usr/local/mysql/data/ folder in API node.
- Log files are very useful when you face issues when configuring, deploying and debugging applications.

Project 3.2 : FAQs - 3

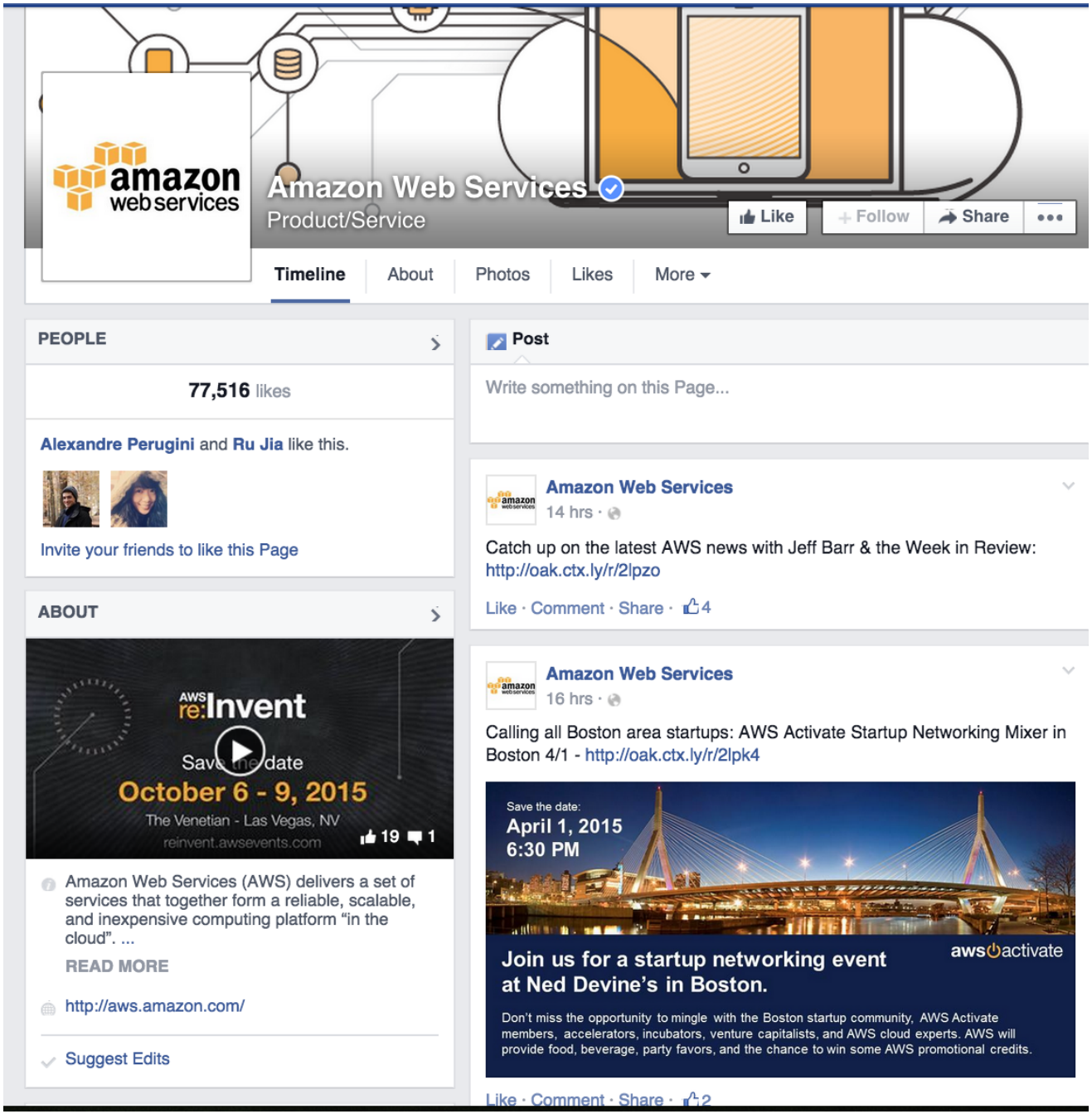
Problem 3: I don't understand the parameters and workloads in the YCSB benchmark.

- Here are some useful links that will help you understand the YCSB benchmark:
  - https://github.com/brianfrankcooper/YCSB/wiki/Papers-and-Presentations
  - https://www.cs.duke.edu/courses/fall13/compsci590.4/838-CloudPapers/ycsb.pdf
- Here is the description of running a workload:
  - https://github.com/brianfrankcooper/YCSB/wiki/Running-a-Workload

This week: Project

- P3.1 Files vs Databases
- P3.2 Partitioning and Replication
- P3.3 Database-as-a-Service
- P3.4 Cloud Data Warehousing
- P3.5 Consistency in Distributed Databases

Social Network

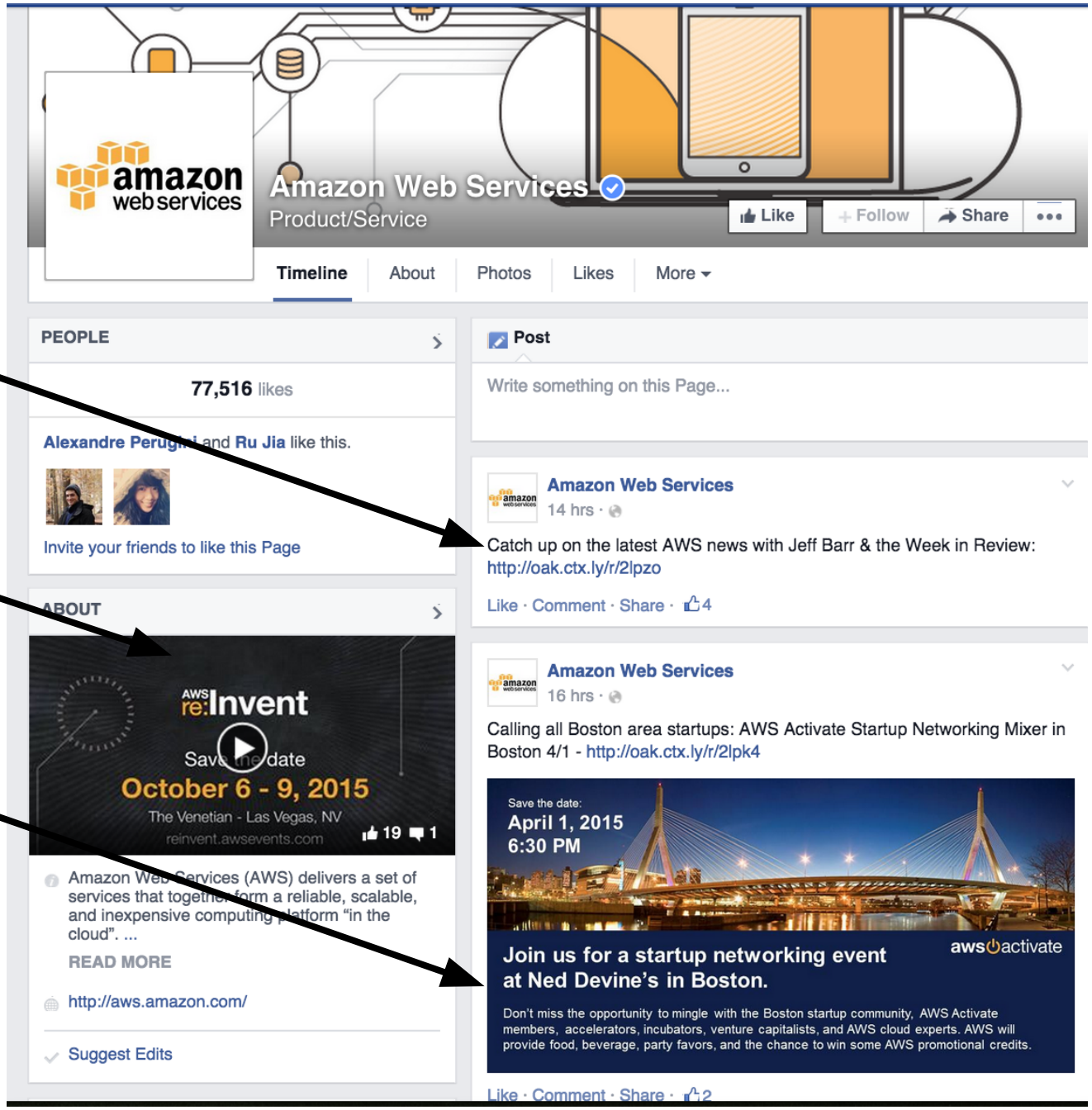


Social Network

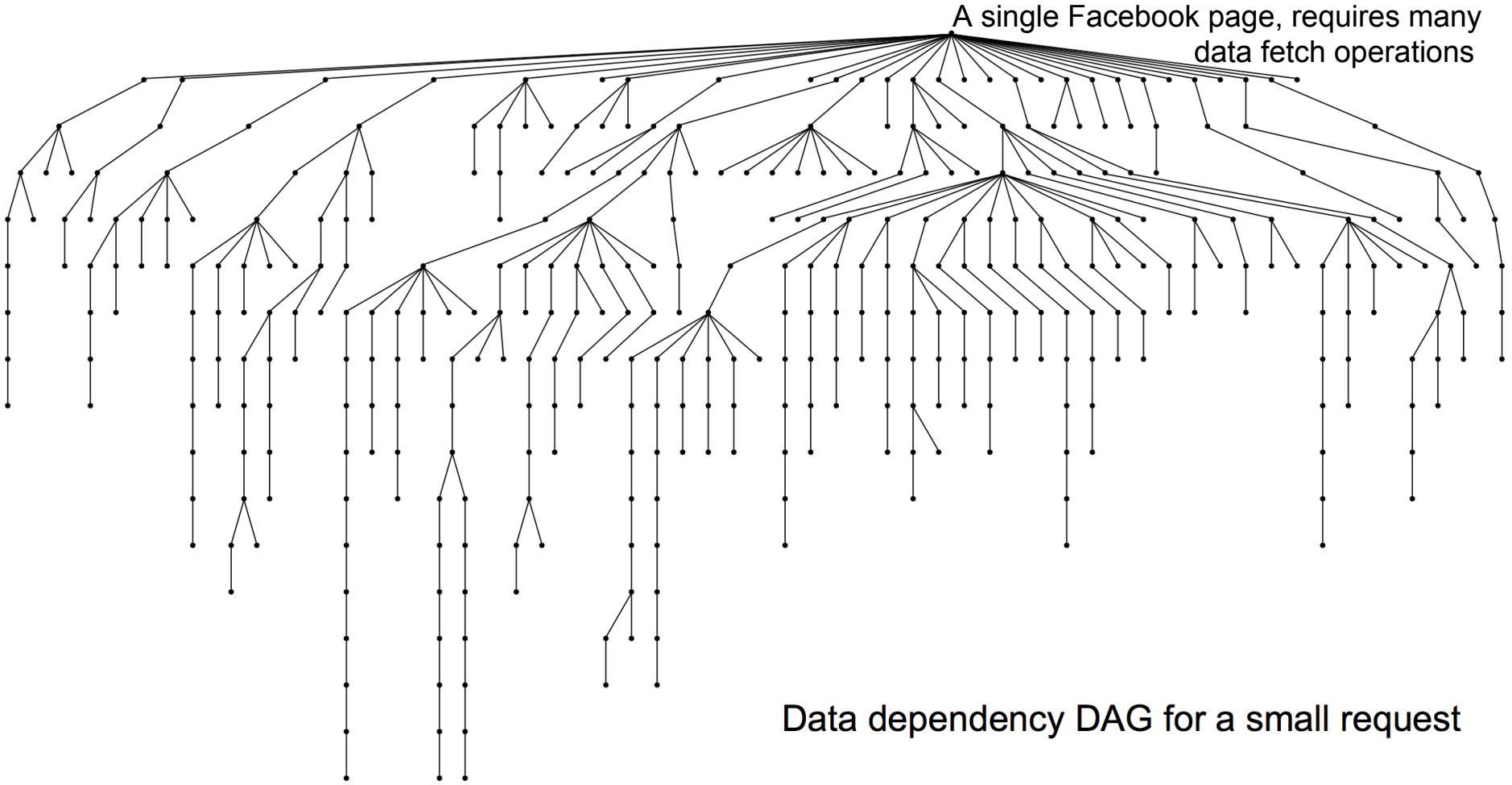
Text

Video

Image



High Fanout and Multiple Rounds of Data Fetching



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).

Database as a Service (DBaaS)

- Database-as-a-Service is provided by cloud operators
- Cloud operators are fully responsible for managing the databases that support applications.
- Application developers do not need to perform traditional database administration functions.
- The database can seamlessly scale and is maintained, upgraded, backed-up by the cloud provider.
- DBaaSes silently and transparently handle server failure, without impacting the application developer.
- The role of a database administrator becomes less essential in this scenario.

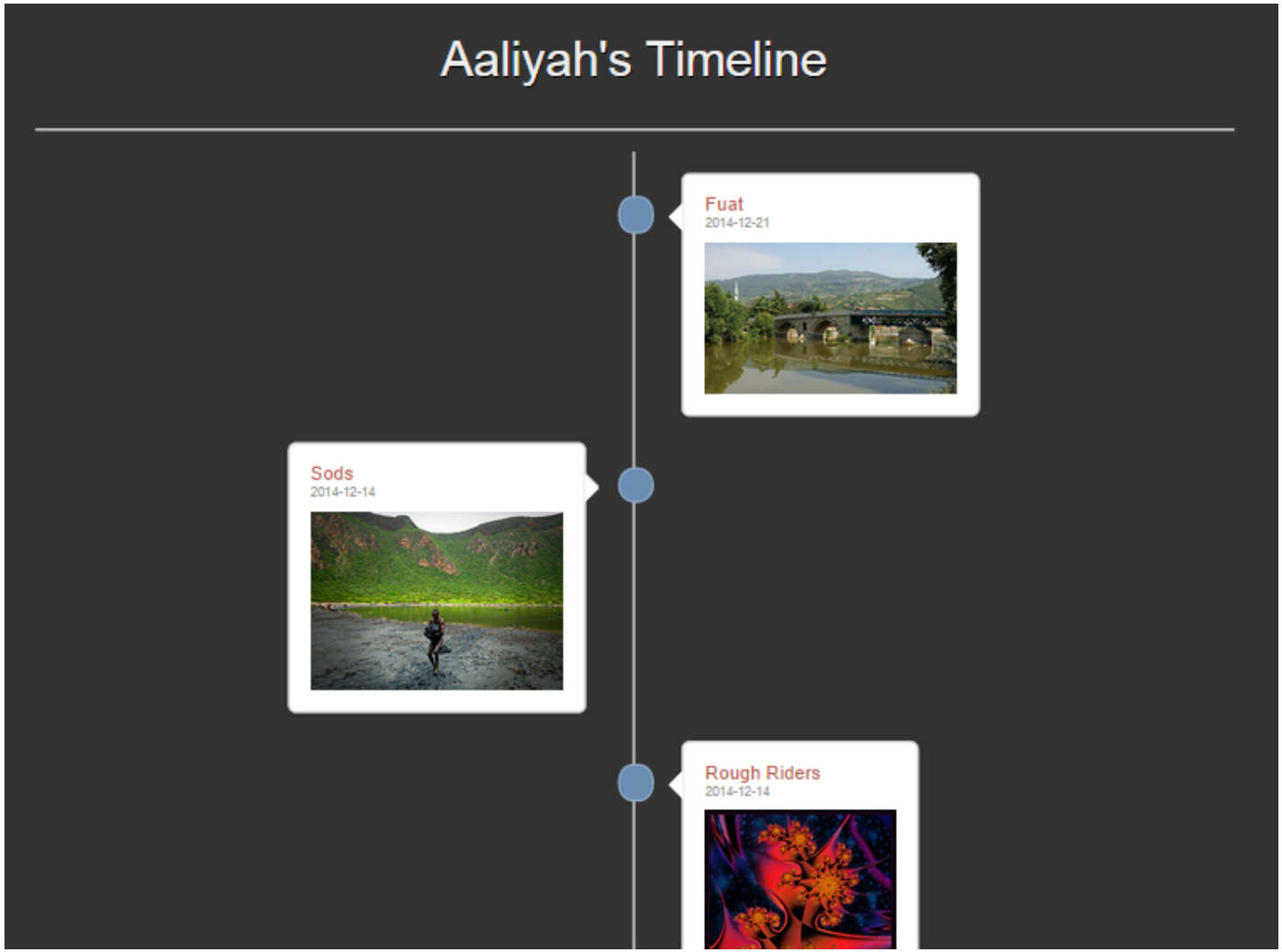
Amazon RDS

- Relational Database Service
- MySQL, Oracle, MSFT SQL server, other
- AWS manages the DB for you!

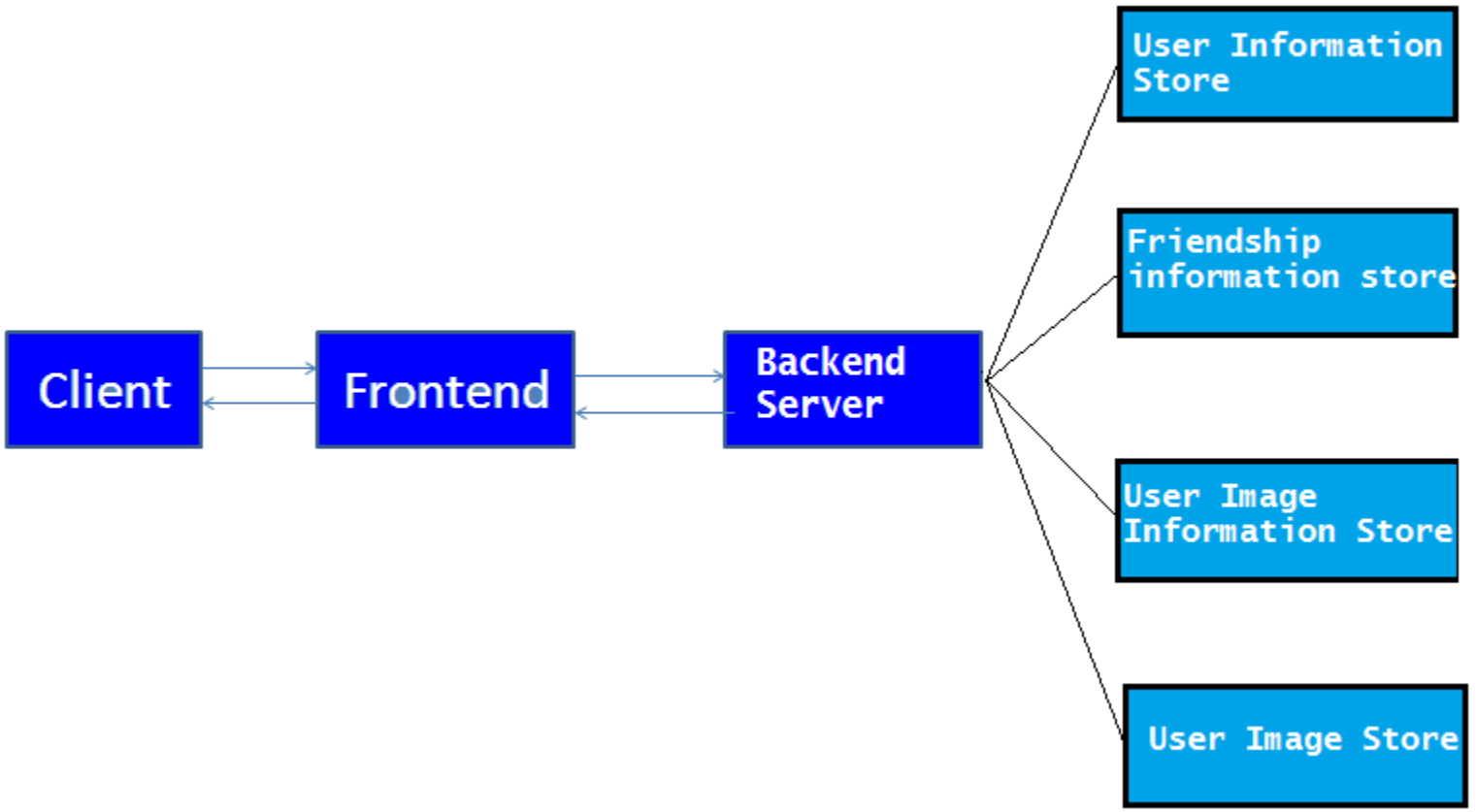
Amazon DynamoDB

- Managed NoSQL Database Service
- Limited functionality
- High performance at large scales
- Expensive!

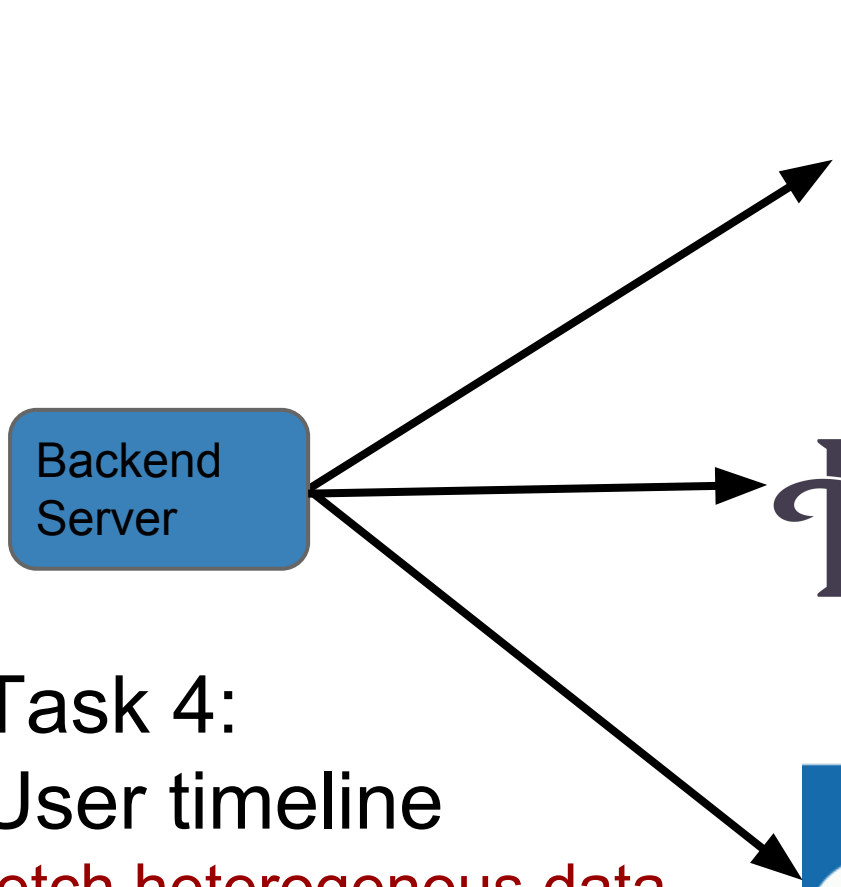
Project 3.3 - Build a Social Network Image Timeline



Project 3.3 - Architecture



Project 3.3 Tasks



Task 1:
Authentication

user information

Task 2:
User relationship

social graph

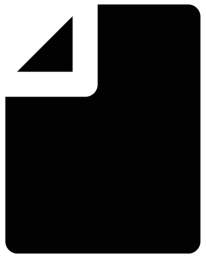
Task 4:
User timeline

fetch heterogenous data

Task 3:
User images

links to images on S3

Unit 4 Module 13



HDFS vs Ceph FS

UNIT 4: Cloud Storage			
	Module 12: Cloud Storage (Gradebook) (Learning Dashboard)		
	Module 13: Case Studies: Distributed File Systems (Gradebook) (Learning Dashboard)		Opens on 3/16/15 12:01 AM
	Module 14: Case Studies: NoSQL Databases (Gradebook) (Learning Dashboard)		Opens on 3/23/15 12:01 AM
	Module 15: Case Studies: Cloud Object Storage (Gradebook) (Learning Dashboard)		Opens on 3/23/15 12:01 AM
	Quiz 4: Cloud Storage	Checkpoint	Not yet available

Upcoming Deadlines

- 15619Project Phase 1 Report
  - Due: 11:59PM ET Mar 19th (Thursday)
- P3.3
  - Due: 11:59PM ET Mar 22nd (Sunday)
- Module 13
  - Due: 11:59PM ET Mar 22nd (Sunday)
- 15619Project Phase2
  - Due: 16:59PM ET Apr 1st (Wednesday)

P3.3 Demo

Demo 1. Maven

Build and manage P3.3 project using Maven

Demo 2. DynamoDB

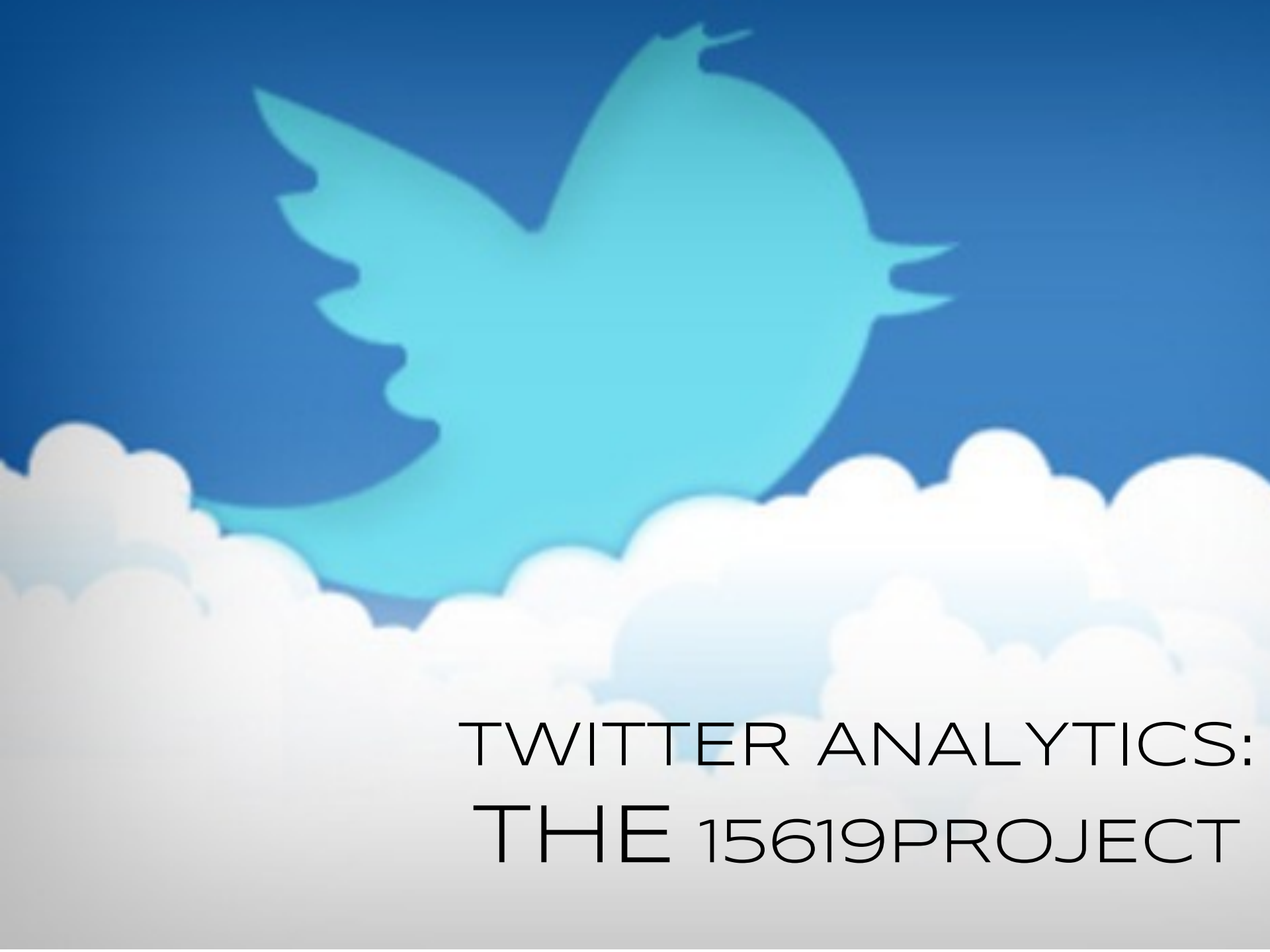
Import data into DynamoDB using Data Pipeline

Demo 1. Maven

- Build the project
- Launch the Undertow server
- Install a new dependency

Demo 2. DynamoDB

- Create a DynamoDB table
- Create an IAM role
- Import data using Data Pipeline



TWITTER ANALYTICS:
THE 15619PROJECT

What's due soon?

- Phase 1 Report & Code Deadline
  - **[11.59 PM Pitt Thursday 3/19]**
  - Upload to TheProject.Zone
  - No code ⇒ ZERO POINTS FOR PHASE 1
  - Missing files ⇒ ZERO POINTS FOR PHASE 1

- Very High Standard Expected in Report
  - Make sure you highlight failures and learning
  - If you didn't do well, explain why
  - If you did, explain how

What to watch out for in Q2...

- Encoding issues
  - If you have *???s* in your output
  - Figure out where you lost the encoding information
  - Restart ETL process beyond that point
- Container memory issues
  - YARN kills containers that occupy too much memory
  - Solution 1 (recommended): Debug YARN logs
  - Solution 2: Brute force
- HBase Remote Connection fails
  - Solution 1 (recomd): Read Apache Docs, HBase book
  - Solution 2: Don’t use a remote connection

What to watch out for in Phase 2...

- Two more queries (Q3 and Q4)
  - More ETL
  - Multiple tables and queries

- **Live Test!!!**
  - **For HBase and MySQL**
  - **Includes Mixed-Load**
  - **No more pre-caching of known requests**

What to watch out for in Query 2

- Loading into MySQL
  - Think about indexes and PKs
  - If using a cluster, think about capacity
- Money
  - Remember EMR Costs
  - Remember EBS Costs v/s IOPS
  - Do not use another region (even accidentally)

Query 3: Retweet Buddies

Q. What's a retweet and how do I find it?

Read https://dev.twitter.com/docs/platform-objects/tweets

Query 3: Retweet Buddies

- A retweeted B twice
- B retweeted A once
- C retweeted A once
- A retweeted D once

GET /q3?userid=A

- *,3, B
- +,1, C
- -,1, D

Query 4: Trending Hashtag

- Use the hashtag entity

  GET /q4?hashtag=SamSmith&start=2014-06-23&end=2014-06-24

  …

  481298397299630080,57299114,2014-06-24+04:50:54

  …

Query 4: Trending Hashtags (how it fits in)

- GET /q2?userid=57299114&tweet_time2014-06-24+04:50:54
- 481298397299630080:0:tapi gak papa deh, doi Taurus juga #SamSmith

Thank You

Any Questions?