

# 15-319 / 15-619

## Cloud Computing

Weekly Overview 7

October 11th, 2022

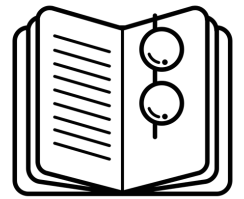
# Recap of Last Week's Activities

- Project 2 Discussion
- Project 3 - Part 1 Released - SQL and NoSQL
- OLI Unit 3 - Modules 10, 11, 12
- Quiz 5
- Team Project, Phase 1
  - Microservice 1 Checkpoint
  - Microservice 1 Early Bird Bonus

# This Week's Activities

- **Project 3, Part 1 Discussion**
  - Due on 16th October 11:59 PM ET
- **OLI Unit 3: Virtualizing Resources for the Cloud**
  - Module 13: Storage and Network Virtualization
- **Quiz 6**
  - Due on 14th October 11:59 PM ET
- **Project 3 - Part 2: Cloud Storage - Heterogeneous Storage in the Cloud**
  - Due on 16th October 11:59 PM ET
- **Team Project, Phase 1**
  - Microservice 1 Final Due on 16th October 11:59PM ET
  - Microservice 2 Checkpoint Due on 16th October 11:59 PM ET

# This Week: Conceptual Content



- **Unit 3: Virtualizing Resources for the Cloud**
  - Module 7: Introduction and Motivation
  - Module 8: Virtualization
  - Module 9: Resource Virtualization - CPU
  - Module 10: Resource Virtualization - Memory
  - Module 11: Resource Virtualization – I/O
  - Module 12: Case Study
  - **Module 13: Storage and Network Virtualization**

# OLI Module 13 - Storage and network Virtualization

- **Unit 3 - Module 13: Storage and network virtualization**

- Software Defined Data Center (SDDC)
- Software Defined Networking (SDN)
  - Device virtualization
  - Link virtualization
- Software Defined Storage (SDS)
  - IOFlow

- **Quiz 6**

- **Due on October 14th**

# Project 3 Part 2: Heterogeneous Storage on the Cloud



A screenshot of a web browser displaying Mark Zuckerberg's Facebook profile. The browser's address bar shows the URL `www.facebook.com/zuck?sk=wall`. The Facebook interface includes a top navigation bar with the logo and a search bar. On the left side, there is a large profile picture of Mark Zuckerberg and a sidebar with navigation links for 'Wall' and 'Info', along with options to 'Share Profile' and 'Report/Block This Person'. The main content area on the right shows the profile name 'Mark Zuckerberg', his work and education details, and a section titled 'Wall' with a post from January 17, 2011, which reads: 'Steve, you've done so much good for the world already. I hope you get better soon.'

← → ↻ 🏠 `www.facebook.com/zuck?sk=wall`

facebook 🔍 Search

**Mark Zuckerberg**  
🏢 Works at Facebook 🎓 Studied Computer Science at Harvard University 🏠 Lives in Palo Alto, California 🌐 Knows English, Mandarin Chinese 🏠 From Dobbs Ferry, New York 📅 Born on May 14, 1984

**Wall**

RECENT ACTIVITY

💬 "I like dangerous thoughts." on Samuel W. Lessin's status.

 **Mark Zuckerberg**  
Steve, you've done so much good for the world already. I hope you get better soon.  
📱 January 17 at 11:43am via iPhone  
👍 150 people like this.

💬 Wall  
📄 Info

Share Profile  
Report/Block This Person

# Project 3, Part 2 - Reminder!

- By Sunday, October 16th at 11:59 PM
  - Heterogeneous Storage on the Cloud (Part 2) -- Complete the tasks and make a reflection post
- By Sunday, October 16th at 11:59PM
  - SQL and NoSQL (Part 1) -- Complete discussion tasks
- By Sunday, October 23rd at 11:59 PM
  - Heterogeneous Storage on the Cloud (Part 2) -- Finish discussion tasks

# Primers for Project 3, Part 2

- Neo4j Primer
- MongoDB Primer
- MySQL Primer

# Neo4j Primer

- **Introduction to Graph Databases**

- Need for Graph Databases

- We usually require joins between entities in RDBMSs, and they are expensive to compute. Graph databases store connections alongside data in the model

- Cypher Query Language (CQL)

- Create: CREATE, Read: MATCH, Update: SET, Delete: DELETE

- **Table indexing**

- Single Property vs Composite indexing

# MongoDB Primer

- **Compare MongoDB and MySQL**
- **MongoDB Features**
- **MongoDB Technicalities**
  - Documents
  - Collections
- **MongoDB Tutorial to practice:**
  - How to import data into MongoDB
  - Some basic queries with Mongo Shell
  - How to Build index to speed up your query

# Project 3, Part 2 Overview

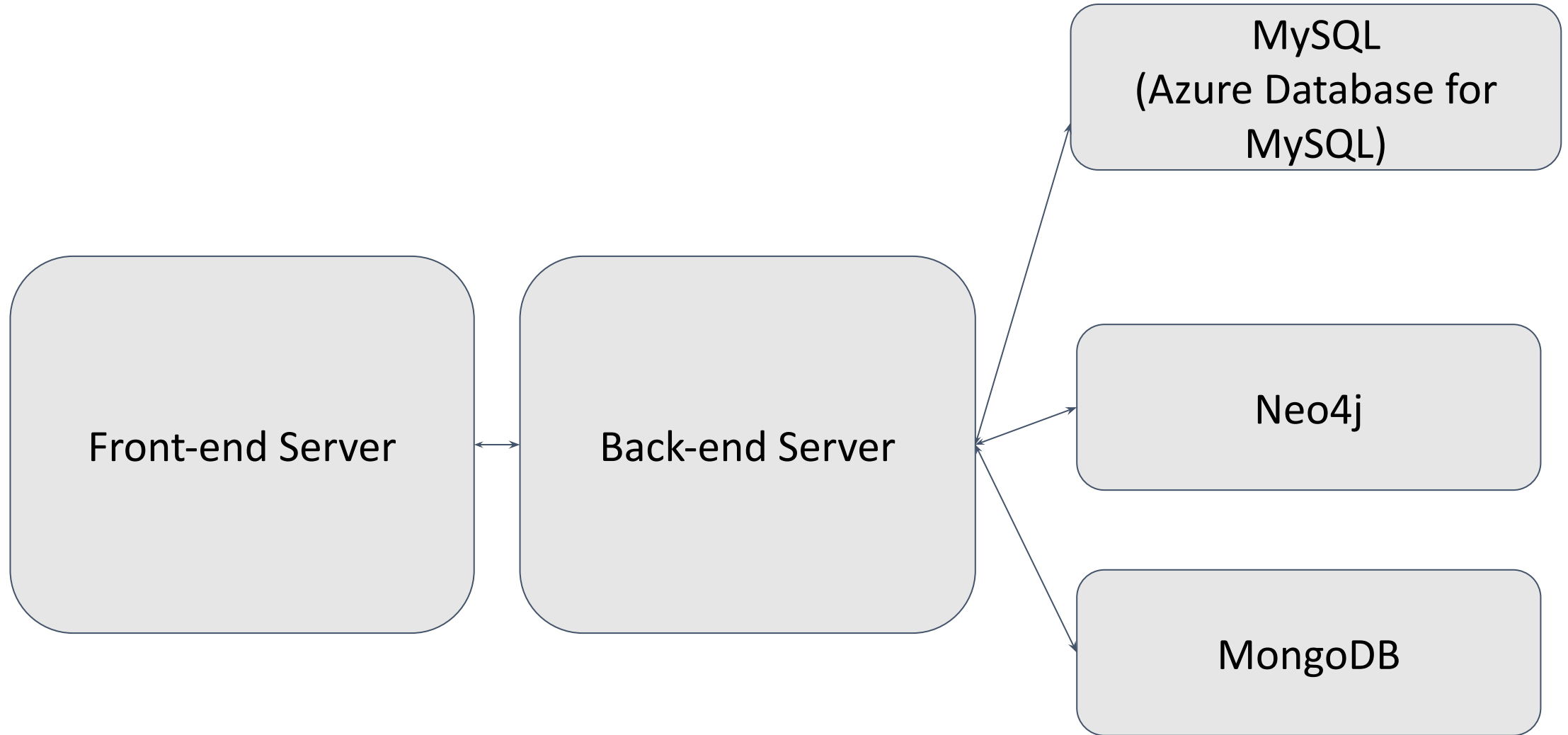
Scenario: Build Your Own Social Network Website using datasets from Reddit.com: **users.csv**, **links.csv**, **posts.json**

- Task 1: Implementing Basic Login with SQL
  - User authentication system : Azure Database for MySQL (**users.csv**)
  - User info / profile : Azure Database for MySQL
- Task 2: Storing Social Graph using Neo4j
  - Follower, followee : Neo4j (**links.csv**)
- Task 3: Build Homepage using MongoDB
  - All user generated comments: MongoDB (**posts.json**)

# Project 3, Part 2 Overview (contd.)

- Task 4: Put Everything Together
  - User Timeline: Fanout
- Task 5: Caching
  - Cache the requests with high frequency

# Social Network Architecture



# Task 1: Implementing Basic Login with MySQL

- Designed to manage highly structured data.
  - Authentication data
- Database-as-a-Service (DBaaS)
  - Azure Managed MySQL database
    - Cloud vendor is responsible for administrative tasks
    - Users are responsible for optimizing applications that use database resources

# TDD with Mockito

- Mockito is an open-source testing framework that allows the creation of test double objects (mock objects).
- It is used to mock interfaces so that the specific functionality of an application can be tested without using real resources such as databases, expensive API calls, etc.
- You are required to understand the given implementation, and may use it to quickly debug your solution for Task 1.

# Task 2: Storing Social Graph using Neo4j

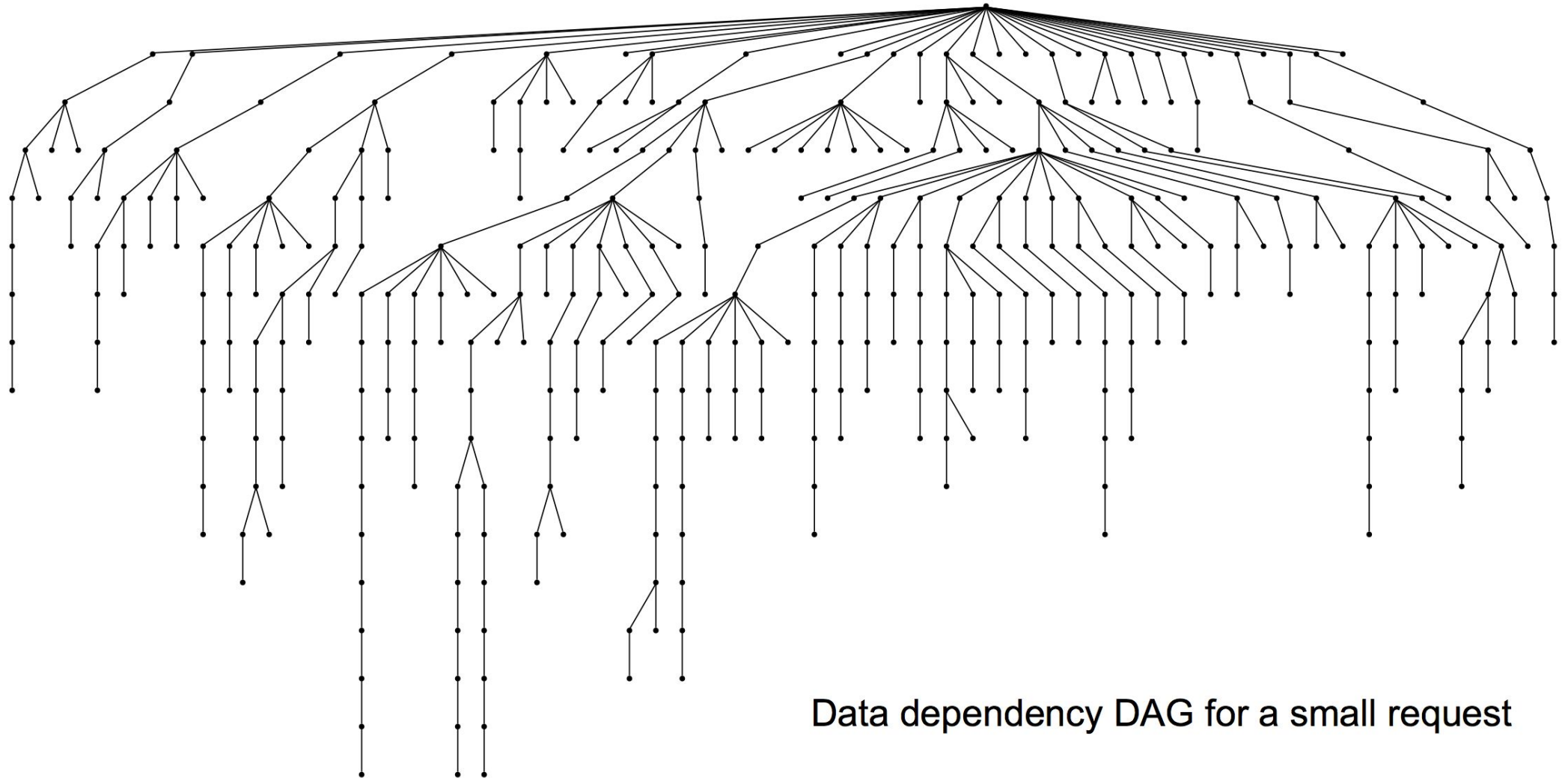
- Designed to treat the relationships between data as equally important as the data
  - Relationships are very important in social graphs
- Property graph model
  - Nodes
  - Relationships
  - Properties
- Cypher query language
  - Declarative, SQL-inspired language for describing patterns in graphs visually

# Task 3: Build Homepage using MongoDB

- Document Database
  - Schema-less model
- Highly Scalable
  - Automatically shards data among multiple servers
  - Does load-balancing
- Allows for Complex Queries
  - MapReduce style filter and aggregations
  - Geospatial queries

# Task 4: Social Network Timeline

## High Fanout in Data Fetching



Data dependency DAG for a small request

# Task 4: Social Network Timeline

## High Fanout in Data Fetching

- Practice writing complex fan-out queries that span multiple databases:
  - MySQL
  - Neo4j
  - MongoDB

# Task 5: Social Network Timeline with Cache

- Fanout and Caching
  - Practice writing complex fan-out queries that span multiple databases.
  - Also practice using a caching mechanism to boost your backend!

## P3 - Reminders and Suggestions

- In Task 4 and 5, you will use the databases from tasks 1 to 3. Make sure to have **all** the databases loaded and ready when working on Task 4 & 5.
- Make sure that you have written **Modular code** for tasks 1 to 3, since you will need to re-use the code in tasks 4 and 5.
- You can submit one task at a time using the submitter. Remember to have your back-end server running when submitting.
- Make sure to terminate **all** resources using “terraform destroy” after the final submission. Double check on the Azure console that all resources were terminated.

Best Wishes!!!

