CS15-319 / 15-619
Cloud Computing

Recitation 7

February 25th and 27th, 2014

Announcements

- Piazza posts
  - Encounter a general bug:
    - Post Publically
  - Encounter a grading bug:
    - Post Privately
  - Do not ask if my answer is correct
  - Do not post code on Piazza
  - Search before posting
- Give feedback on OLI

Announcements

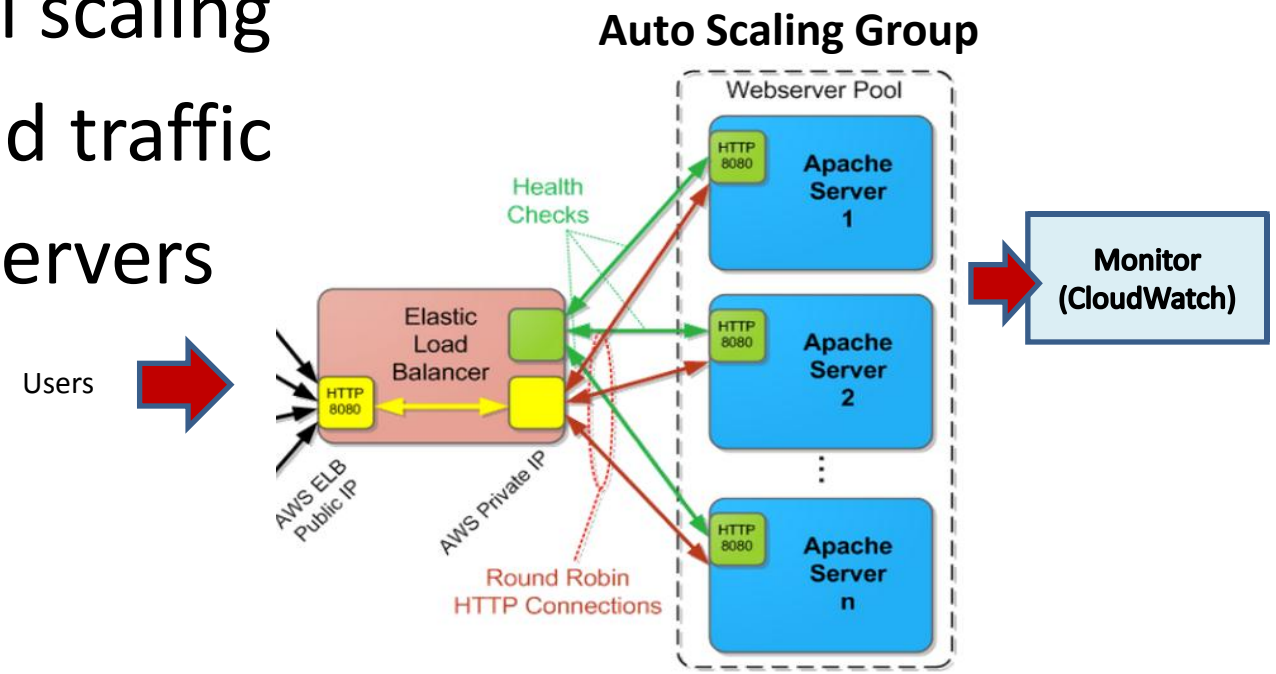
- Manual grading
  - For project checkpoints
    - Will be done by **Tuesday**, one week after the deadline
- Budget control
- Code submission
  - Submit a single ZIP file
  - Check Piazza @616

15-619 Project

- For students who **enrolled in 15-619**
  - Since 12 units, an extra project worth 3-units
  - Project will be out next week
- A **team project** for a team no more than 3 students.
  - Fill in the team information online https://docs.google.com/forms/d/18fpdRlC8A7CCcpvcZ8WKCmV2m2KOsknqFGaJCLfQdu4/viewform
  - No team, advertise your skills on Piazza
  - The team registration is due on 03/02 (This Sunday)
  - Keep record of the team name, you will need it!

Project 2 Reflections

- Find the best suitable instance as server
- Vertical scaling
- Horizontal scaling
- Distributed traffic
- Monitor servers



Project 2 Reflections

- Read project description more than once
- Think about workflow before starting
- Look up API references
  - Read overview first, then details
  - Use samples to go over simple APIs
  - Use the Internet to debug
- Check every step carefully
  - Debug with AWS console

Piazza Questions

- Socket & Timeout Error on Autoscaling
  - Due to the large amount of concurrent connections
  - Make instances in the same availability zone
  - Try different availability zones
  - Increase failure threshold

This Week

- UNIT 3: Virtualizing Resources for the Cloud
  - Module 6: Introduction and Motivation
  - Module 7: Virtualization
  - Module 8: Resource Virtualization - CPU
  - Module 9: Resource Virtualization - Memory
  - Module 10: Resource Virtualization – I/O
  - Module 11: Case Study
  - Quiz 3: Virtualizing Resources for the Cloud

Project 3

- Files vs. Databases
  - File vs. Database
- Vertical Scaling in Databases
  - Vertical Scaling
- Horizontal Scaling in Databases
  - Horizontal Scaling
- Working with NoSQL: DynamoDB / Hbase
  - Amazon DynamoDB
  - DynamoDB vs. HBase

Files vs. Databases

- Example: Student course registration
  - Fields
    - First name
    - Last name
    - Course number
    - Section
    - …

Flat Files

- Computer-based flat files
  - Data accessed by programs
- Student course format
  - A comma-separated ‘csv’ file
  - Content:

  Xiaokang, Zhang, 15619, A
  Ying, Gao, 15319, A
  Guangcheng, Lu, 15619, A

- What if billions of records?

Flat Files

- Accessing data is inconvenient
- Lacking knowledge of file-layout
- Data redundancy
- Concurrent-access
- …

Databases

- Database
  - Organized collection of data
  - Refers to the data themselves and supporting data structures
- Database management system (DBMS)
  - A software application
  - Interface between user and databases
  - Capture and analyze data

Databases

- Relational databases
  - Data is stored as type-defined fields in normalized tables
  - MySQL is an example
  - SQL (Structured Query Language)

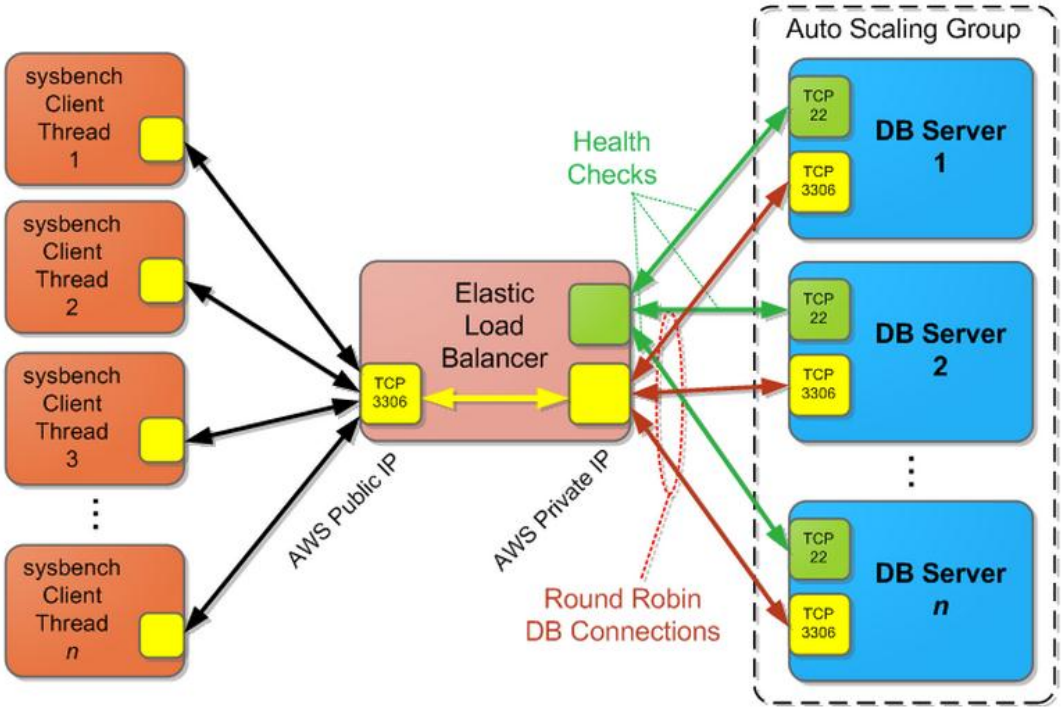
ID	FirstName	LastName	Course	Section
1	Xiaokang	Zhang	15619	A
2	Ying	Gao	15319	A
3	Guangcheng	Lu	15619	A

Files vs. Databases

- Compare a flat file vs. MySQL
- Answer:
  - What are the advantages and disadvantages using flat files versus databases?
  - In what situation would you use a flat file or a database?
  - How to build your own databases? How to manipulate it?

Project 3 Overview

- Storage types
  - Vertical scaling
  - Horizontal scaling
- Data abstraction
  - MySQL
  - NoSQL databases
    - DynamoDB
    - HBase



- Number of posts in 2013: 40 million
- Number of comments in 2013: 404 million
- Number of visitors: 100 million monthly unique
- Number of employees: 28

- How to store data?

reddit

ID	UPS	DOWNS	TITLE	URL
12345	120	34	Buffins Create Zombie Dog!	www.someaussiesite.co.au/dog.html
12346	3	24	Check out my new blog!	noobspamer.blogspot.com
12347	509	167	Pee in a sink if you've ever voted up.	self

*Steve Huffman - Lessons learned while at reddit.com

reddit

ID	UPS	DOWNS	TYPE
12345	120	34	Link
12346	3	24	Link

THING_ID	KEY	VALUE
12345	Title	Boffins Create Zombie Dog!
12345	URL	www.someaussiesite.com.au/zombiedog.html
12346	Title	Pee in a sink if you've ever voted up.
12346	URL	self

*Steve Huffman - Lessons learned while at reddit.com

Upcoming Deadlines

- Project 3:

Project 3		
Files vs. Databases		
File vs. Database	Checkpoint	Available 2/24/14 12:01 AM Due 3/1/14 11:59 PM

- Unit 3:

UNIT 3: Virtualizing Resources for the Cloud		
Module 6: Introduction and Motivation Module 7: Virtualization Module 8: Resource Virtualization - CPU Module 9: Resource Virtualization - Memory Module 10: Resource Virtualization – I/O Module 11: Case Study		
Quiz 3: Virtualizing Resources for the Cloud	Checkpoint	**Available Now**. Due 2/27/14 11:59PM Pittsburgh

Due on Thursday!
Pay attention to the duration change!

Upcoming Deadlines

- Quiz 3 Due THIS WEEK!
  - Thursday 2/27/2014 11:59PM Pittsburgh
  - Late submissions are NOT accepted
- Timed
  - **180mins** once started
  - Remember to click SUBMIT before deadline
- Reminder
  - Tag your instance: Key: **Project**, Value: **3.1**

Demo Outline

- Simple demo with flat files
  - Count the number of lines in a file
  - Filter certain words
- Introduction to databases
  - SQL DB versus NoSQL DB
  - Introduction to MySQL
- Simple demo of using MySQL
  - Connect with mysql server
  - Create table and import data into table
  - Sample queries

Flat File Processing Demo

- Tools
  - grep, awk and etc.
- Working principle
  - Read as input stream
  - Process line by line
- SQL Program Sample
  - create_table.sql
- Demo
  - Count number of line
  - Match key world

SQL Database Introduction

- SQL History
  - Structured Query Language
  - ANSI Standard in 1986, ISO in 1987
- Relation Model
  - Structure data model

ID	Name	Email	Age
45637	Mike	mike@example.com	23
12456	Jake	jake@example.com	25

- Relational language
  - DDL: Database Definition Language
  - DML: Database Manipulation Language

MySQL Introduction

- Data Definition Language
  - Table definition:
    - column definition(name, length, data type), constraints(key)
  - Data type:
    - INT, CHAR,NUMERIC, DATE etc
- Example: create a table “students”
  - How many columns do we need?
  - What’s the name of the column?
  - What datatype should be applied?
  - What’s their relationship?

MySQL Example

- Create a table

```
CREATE TABLE students
(
ID int,
Name varchar(255),
email varchar(255)
);
```

id	name	email

MySQL Example

- Insert into a table

```
INSERT INTO students
VALUES ('1','Mike','mike@example.com');
```

id	name	email
1	Mike	mike@example.com

MySQL Example

- Insert into a table

```
INSERT INTO Persons
VALUES ('1','Mike','mike@example.com');

INSERT INTO Persons
VALUES ('2','Jake','jake@example.com');
```

id	name	email
1	Mike	mike@example.com
2	Jake	jake@example.com

MySQL Example

- Insert into a table

```
INSERT INTO Persons
VALUES ('1','Mike','mike@example.com');

INSERT INTO Persons
VALUES ('2','jake','jake@example.com    ');

INSERT INTO Persons
VALUES ('3','Bob','bob@example.com');
```

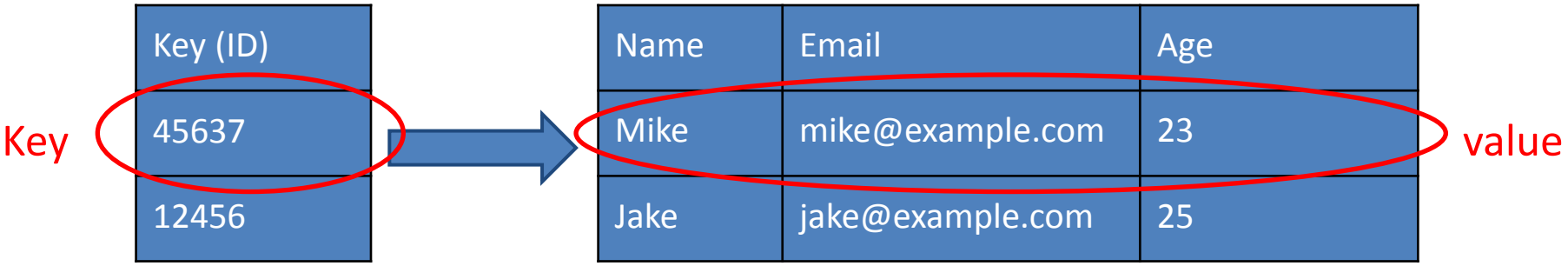
id	name	email
1	Mike	mike@example.com
2	Jake	jake@example.com
3	Bob	bob@example.com

MySQL Manipulation

- Data Manipulation Language
  - select, from, where, renaming
  - set operation
  - ordering
  - aggregate function
  - nested subqueries

NoSQL Databases

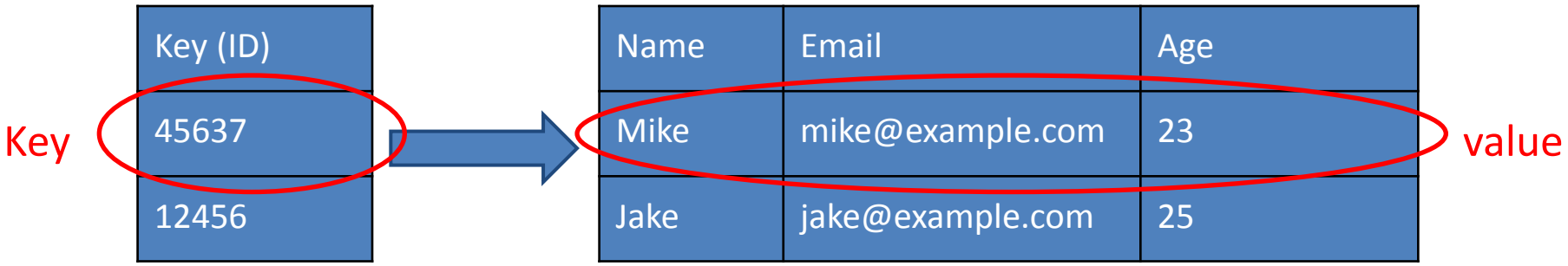
- NoSQL means non-relational
  - Many NoSQL DBs are organized as Key-Value pairs



  - Release the guarantee of relational model to gain better horizontal scalability
    - Typically contain billions of records
    - Google: 91 million searches/day
      - => 33 trillion database entries in one year

NoSQL Databases

- NoSQL means non-relational
  - Many NoSQL DBs are organized as Key-Value pairs



  - Release the guarantee of relational model to gain better horizontal scalability
    - Typically contain billions of records

MySQL Demo

- Connect with SQL server
- Create Database
- Create Table
- Load data into table
- Sample queries
  - count number of records
  - find certain column