15-319 / 15-619
Cloud Computing

Recitation 5

February 9th, 2016

Overview

- **Administrative issues**
  Office Hours, Piazza guidelines
- **Last week's reflection**
  Project 2.1, OLI Unit 2 module 5 and 6
- **This week's schedule**
  - Project 2.2 - February 14, 2016
  - Quiz 4 - February 12, 2016 (Modules 7, 8, 9)
  - Finalize 3 person teams for the 15619Project

Announcements

- Monitor your expenses regularly
  - Check your bill frequently on TPZ
  - Check on AWS, use Cost Explorer filter by tags
  - Check on the Azure portal since only $100/mo

- Terminate your resources when not in use
  - **Stop still costs EBS money ($0.1/GB/Month)**
  - Amazon EC2 and Amazon Cloudwatch fees for monitoring, ELB
  - Autoscaling group - no additional fees

- Use spot instances

Announcements

- Protect your credentials
  - A Student had his credentials stolen.
    - $1.1K was spent in a few hours.
  - Crawlers are looking for AWS credentials on public repos!

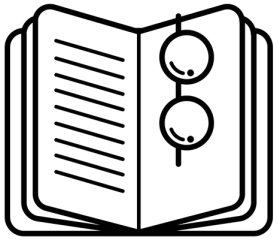
Last Week's Reflection

- Content
  - Unit 2 - Modules 5 and 6:
    - Cloud Management & Software Deployment Considerations
  - Quiz 3 completed
- Azure and AWS EC2 APIs
  - CLI, Java, Python
- Load Balancing and AutoScaling
  - Experience horizontal scaling
  - Manage cloud resources and deal with failures using programs
  - Initial experience with load balancing

Project 2.1

- Manual grading of submitted code
  - 10% of the total Azure points
  - 20% of the total AWS points
  - Always make sure that your code is readable
    - Use the Google Code Style guidelines
  - Always add comments especially for complex parts

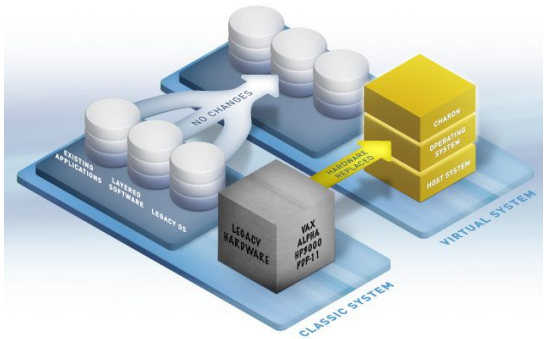
This Week: Content



- 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: Network and Storage Virtualization

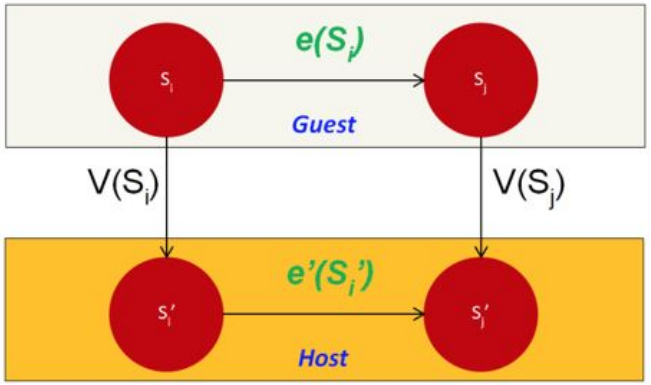
OLI Module 7 - Virtualization Introduction and Motivation

- Why Virtualization
  - Enabling the cloud computing system model
  - Elasticity
  - Resource sandboxing
  - Limitation of General-Purpose OS
  - Mixed OS environment
  - Resource Sharing
    - Time
    - Space
  - Improved system utilization and reduce costs



OLI Module 8 - Virtualization

- What is Virtualization
  - Involves the construction of an isomorphism that maps a virtual guest system to a real (or physical) host system
  - Sequence of operations *e* modify guest state
  - Mapping function *V(Si)*



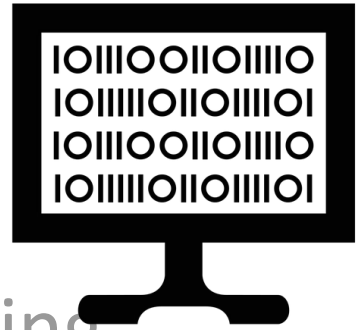
- Virtual Machine Types
  - Process Virtual Machines
  - System Virtual Machines

OLI Module 9
Resource Virtualization - CPU

- Steps of CPU Virtualization
  - Multiplexing a physical CPU among virtual CPUs
  - Virtualizing the ISA (Instruction Set Architecture) of a CPU

- Code Patch, Full Virtualization and Para virtualization
- Emulation (Interpretation & Binary Translation)
- Virtual CPU

This Week: Project



- P2.1: Horizontal Scaling and Autoscaling
  - MSB First Round Interview
- P2.2: Load Balancing Strategies
  - MSB Second Round Interview
- P2.3: Caching Strategies
  - MSB Final Round Interview

Load Balancing

- What is Load Balancing
  - Efficiently dividing incoming network traffic among a pool of back-end servers
- Motivations
  - Improved Quality of Service (QoS)
    - Increased throughput
    - Decreased latency
  - High Availability (HA)
  - Health check and fault tolerance

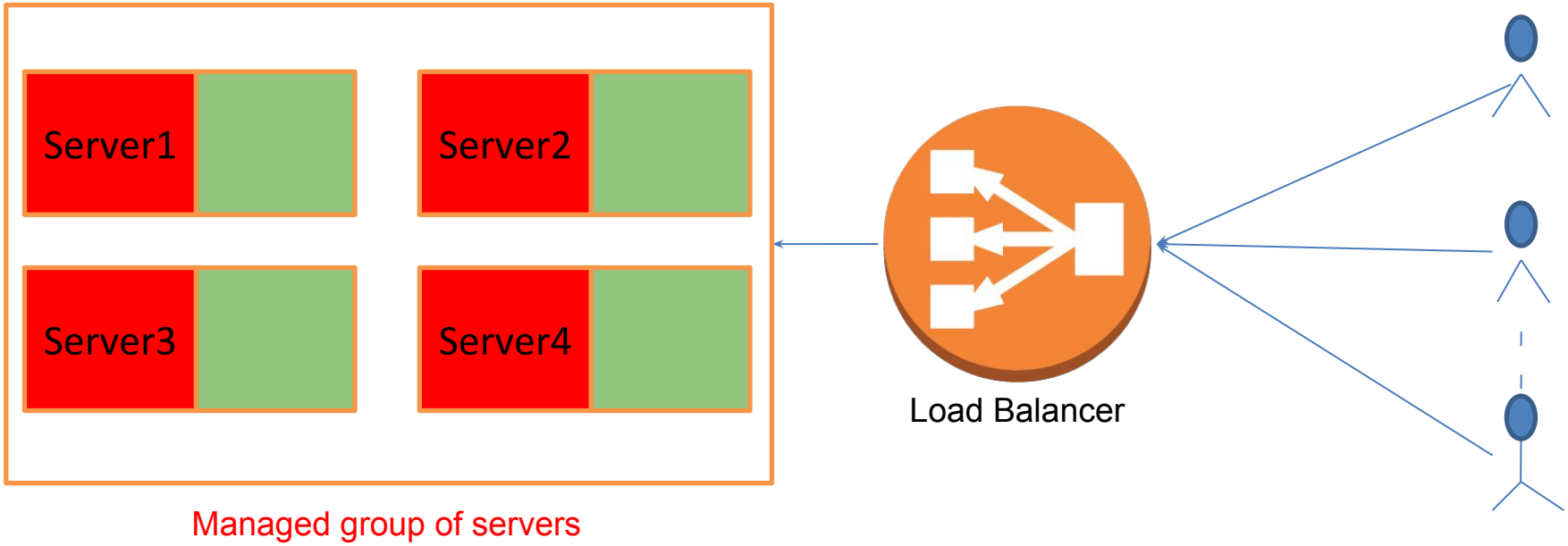
Load Balancer - Distribution Strategy

Simple load distribution strategy:

- Random scheduling
- Round Robin
  - Works well for a homogeneous load
  - Might not work so well for a heterogeneous load

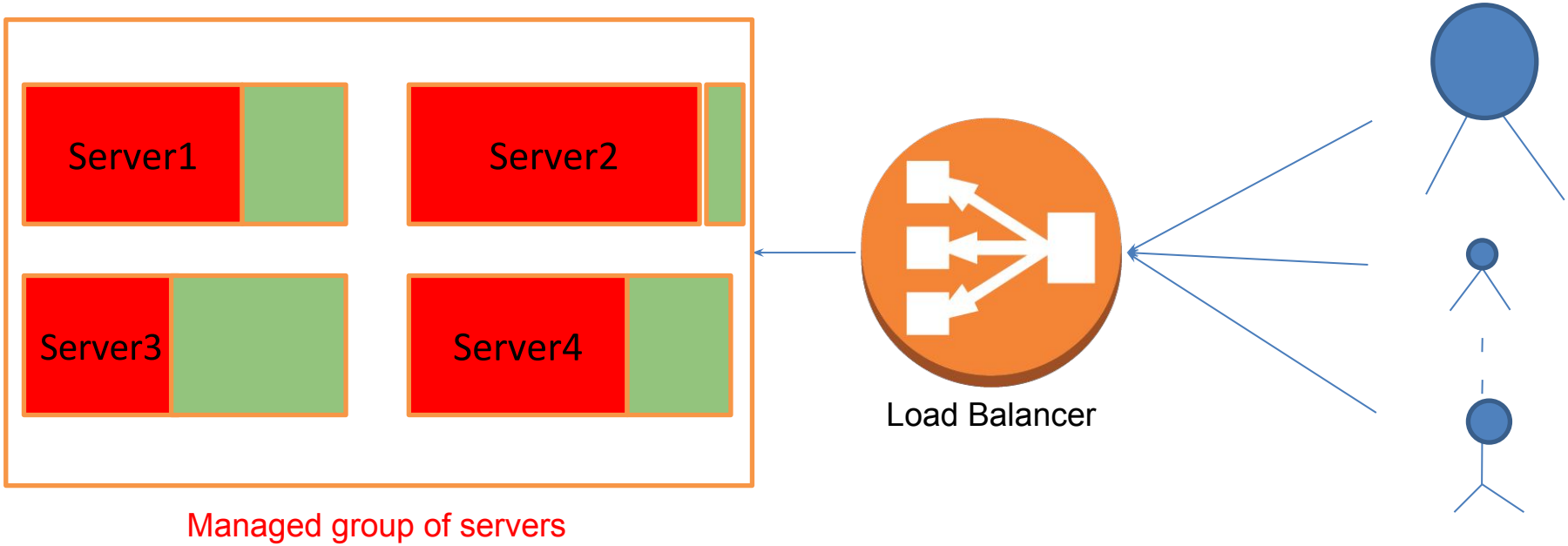
Homogenous Load

- The requests issued in a homogenous workload put the same demands on the back-end resources.
- Simply distributing the requests to the available servers could lead to a balanced load.



Heterogenous Load

- Could lead to an uneven workload because the resource demands of heterogeneous requests might be different.
- One of the machines could receive several requests that overwhelm its resources compared to others.



Load Balancer - Evaluation

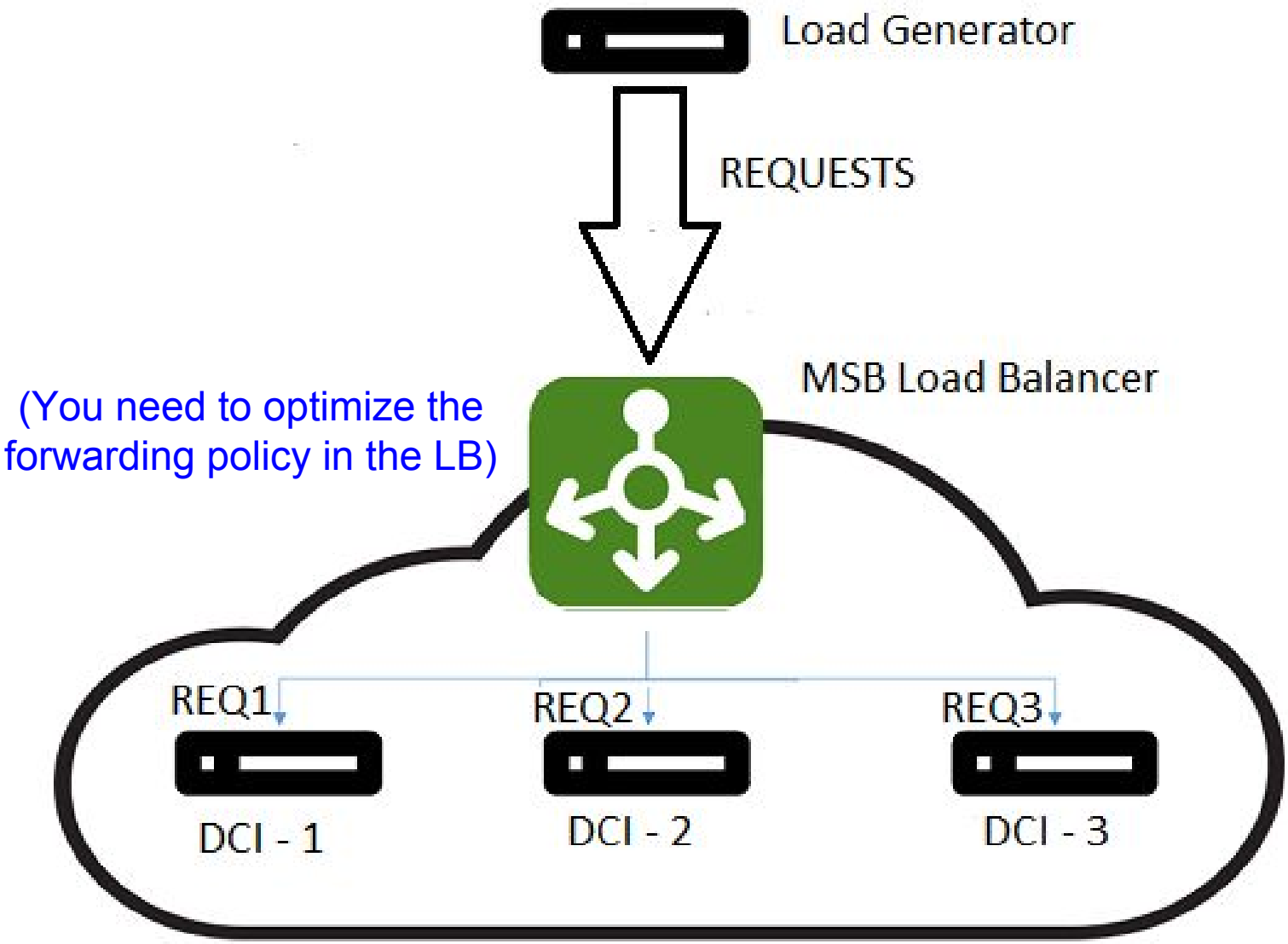
Load distribution strategies:

- How about something more intelligent?
  - Based on request execution time?
  - Based on resource utilization?

P2.2 - Tasks

- Load balancer
  - Write code for round-robin scheduling
  - Implement an effective load distribution strategy
  - Implement a health check
- All code to be written in the load balancer
- Skeleton code given
- We provide an API to retrieve a Data Center instance’s CPU utilization

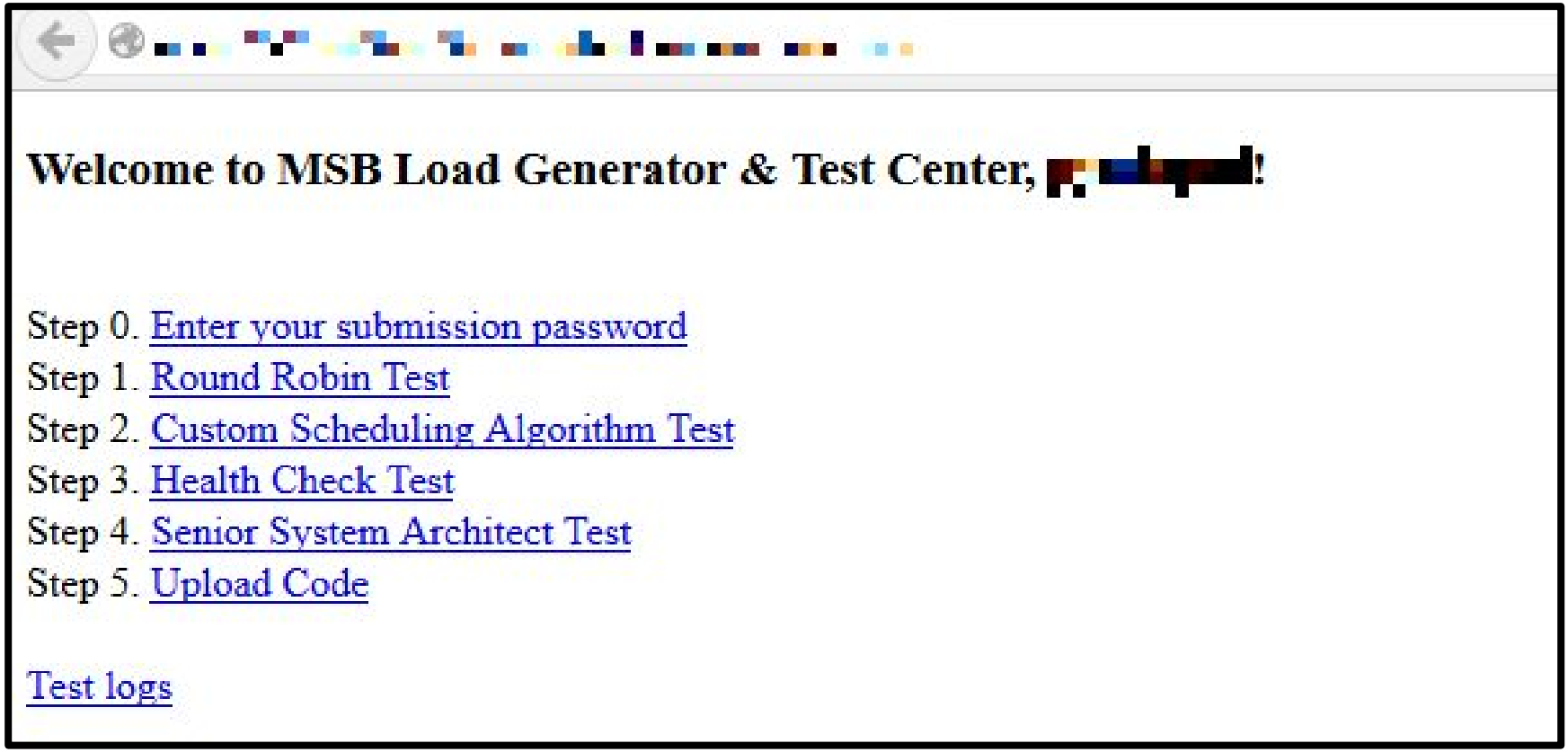
P2.2 Architecture



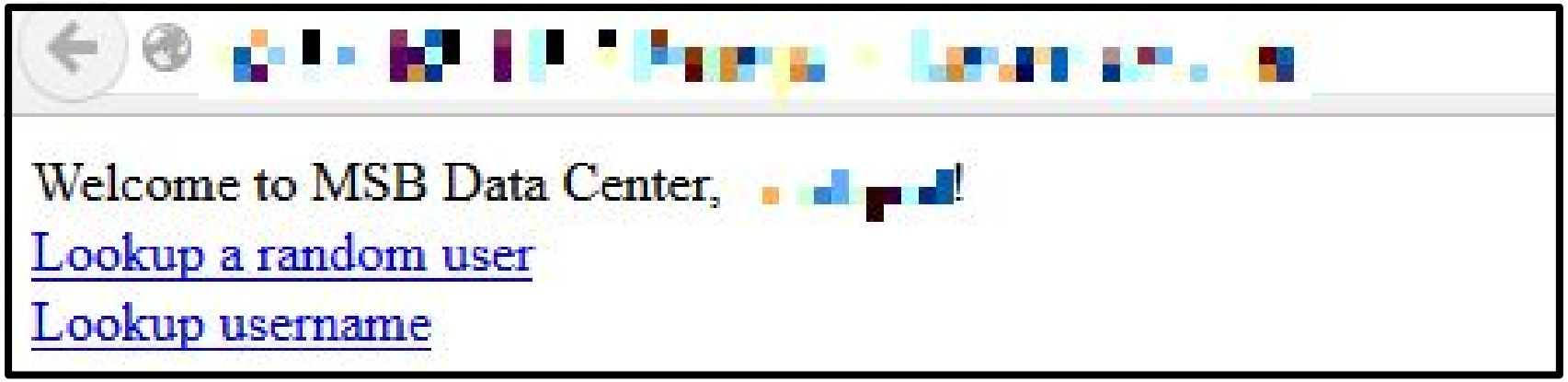
P2.2 - Load Balancer Functionality

- Distribute incoming requests evenly among the available data center instances
- Health check
  - Measure health of instances
    - Faulty instances will bring down performance
  - Handle the case of instance failure
    - Stop sending requests to failed instance
    - Launch a new instance and add to LB

Project 2.2: Load Generator UI



Project 2.2: Data Center UI



Skeleton Code Provided

Skeleton code in Java (**`loadbalancer`**)

- Implements “the plumbing” of a load balancer
  - Setting up of sockets to the client and server
  - API for forwarding requests and receiving responses

- Pending tasks:
  - Implement a simple round-robin request router
  - Implement a health-check that detects and recovers from failure
  - Understand the different types of requests and learn to monitor web servers in real-time
  - Forward requests based on observed load on each web server [`in start() method`]

Project 2.2 Penalties

Project Grading Penalties

Besides the penalties mentioned in recitation and/or on Piazza, penalties accrue for the following:

Violation	Penalty of the project grade
Spending more than $5 for this project phase on AWS	-10%
Spending more than $10 for this project phase on AWS	-100%
Failing to tag all your AWS resources in either parts (EC2 instances, etc) for this project.	-10%
Submitting your AWS credentials in your code for grading	-100%
Submitting your Azure credentials in your code for grading	-100%
Submitting the Azure part with AWS instances or the AWS part with Azure VMs.	-100%
Attempting to hack/tamper the autograder in any way	-100%
Using virtual machines other than Standard_A1(DC) and Standard_D1(LG and LB) in the Azure part	-100%

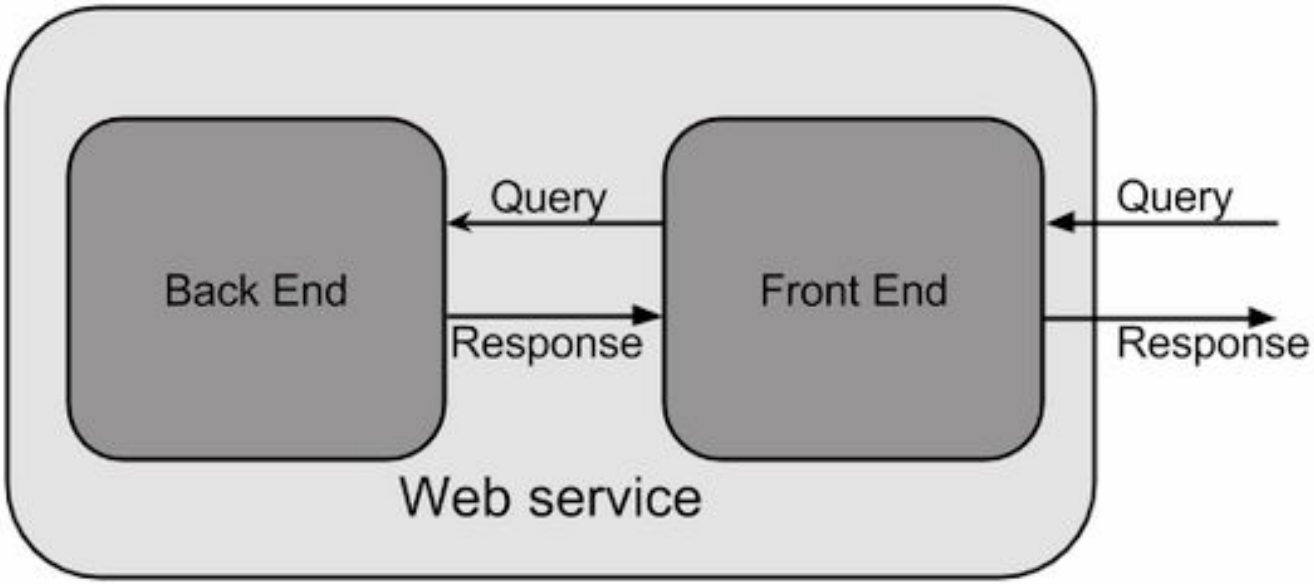
Upcoming Deadlines

- Project 2.2: Load Balancer Internals
  Due: Sunday 02/14/2016 11:59PM Pittsburgh

- Quiz 4: Modules 7, 8 and 9:
  Due: Friday 02/12/2016 11:59PM Pittsburgh

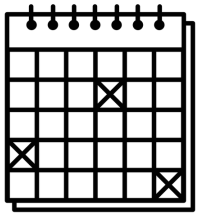
- 15619Project Team Formation (@1336)
  Due: Monday 02/15/2016 11:59PM Pittsburgh

15619Project Architecture



- Writeup and Queries will be released on Thursday, February 25th, 2016
- We can have more discussions in subsequent recitations
- For now, ensure 3-person teams you decide have experience with web frameworks and database, storage principles and infra setup/hacking

15619 Project Time Table



Phase (and query due)	Start	Deadline	Code and Report Due
Phase 1 Part 1 ● Q1, Q2	Thursday 02/25/2016 00:00:01 EST	Wednesday 03/16/2016 23:59:59 E**D**T	Thursday 03/17/2016 23:59:59 E**D**T
Phase 2 ● Q1, Q2, Q3	Thursday 03/17/2016 00:00:01 E**D**T	Wednesday 03/30/2016 15:59:59 E**D**T	
Phase 2 Live Test ● Q1, Q2, Q3	Wednesday 03/30/2016 18:00:01 E**D**T	Wednesday 03/30/2016 23:59:59 E**D**T	Thursday 03/31/2016 23:59:59 E**D**T
Phase 3 ● Q1, Q2, Q3, Q4	Thursday 03/31/2016 00:00:01 E**D**T	Wednesday 04/13/2016 15:59:59 E**D**T	
Phase 3 Live Test ● Q1, Q2, Q3, Q4	Wednesday 04/13/2016 18:00:01 E**D**T	Wednesday 04/13/2016 23:59:59 E**D**T	Thursday 04/13/2016 23:59:59 E**D**T

The End