Team 2:

Leduan Camarero  Requirement Analyst
Reynier Ortiz    Interface Designer
Qais Mazhar      System Architect
Austin von Nehring  Project Manager

Mentor – Dr. Ming Zhao
Outline

Introduction
- Problem Definition & Background
- Current System
- Project Management

System Design
- Design Objectives
- Architecture Overview
- System Decomposition
- System Deployment
- Persistent Data Design

System Implementation
- Technologies & Techniques Used
- Main Algorithm Used

System Evaluation
- Experiment Setup
- Evaluation Results
Outline

Introduction
- Problem Definition & Background
- Current System
- Project Management

System Design
- Design Objectives
- Architecture Overview
- System Decomposition
- System Deployment
- Persistent Data Design

System Implementation
- Technologies & Techniques Used
- Main Algorithm Used

System Evaluation
- Experiment Setup
- Evaluation Results
Problem Definition & Background

- vMoodle – provides online access to Virtual Machines (VMs) hosted on local servers or a public cloud.

- Phase III goal - manage resources more efficiently to support larger numbers of users.

- Previous system did not monitor resource usage or load balance.
Outline

Introduction
 Problem Definition & Background
 Current System
 Project Management

System Design
 Design Objectives
 Architecture Overview
 System Decomposition
 System Deployment
 Persistent Data Design

System Implementation
 Technologies & Techniques Used
 Main Algorithm Used

System Evaluation
 Experiment Setup
 Evaluation Results
Previous System

- Students could make VMs remotely from any computer that has an internet connection.
- The student could start and stop the VM.
- System allowed students to host their VM on a local server or in the Amazon EC2 cloud.
- System chose arbitrarily which host on the local server to use to run VMs.
Outline

Introduction
- Problem Definition & Background
- Current System
- **Project Management**

System Design
- Design Objectives
- Architecture Overview
- System Decomposition
- System Deployment
- Persistent Data Design

System Implementation
- Technologies & Techniques Used
- Main Algorithm Used

System Evaluation
- Experiment Setup
- Evaluation Results
## Project Management - Schedule

<table>
<thead>
<tr>
<th>ID</th>
<th>Tasks</th>
<th>Deliverable</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Problem Definition</td>
<td>D1</td>
</tr>
<tr>
<td>4</td>
<td>Current System Limitations</td>
<td>D1</td>
</tr>
<tr>
<td>5</td>
<td>High-level User Requirements</td>
<td>D1</td>
</tr>
<tr>
<td>6</td>
<td>Alternative Solutions</td>
<td>D1</td>
</tr>
<tr>
<td>8</td>
<td>Determine Personnel Organization</td>
<td>D1</td>
</tr>
<tr>
<td>9</td>
<td>Identify Software and Hardware</td>
<td>D1</td>
</tr>
<tr>
<td>10</td>
<td>Identify Tasks, Milestones, Deliverables</td>
<td>D1</td>
</tr>
<tr>
<td>11</td>
<td>Determine Estimated Cost</td>
<td>D1</td>
</tr>
<tr>
<td>12</td>
<td>Submit Document</td>
<td>D1</td>
</tr>
<tr>
<td>13</td>
<td>Presentation 1</td>
<td>D1</td>
</tr>
<tr>
<td>16</td>
<td>Identify Functional Requirements</td>
<td>D2</td>
</tr>
<tr>
<td>17</td>
<td>Identify Non-functional Requirements</td>
<td>D2</td>
</tr>
<tr>
<td>18</td>
<td>Formulate Use Cases</td>
<td>D2</td>
</tr>
<tr>
<td>20</td>
<td>Create UML Diagrams</td>
<td>D2</td>
</tr>
<tr>
<td>21</td>
<td>Submit SRD</td>
<td>D2</td>
</tr>
<tr>
<td>22</td>
<td>Presentation 2</td>
<td>D2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Tasks</th>
<th>Deliverable</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>High-level System Architecture</td>
<td>D3</td>
</tr>
<tr>
<td>26</td>
<td>Subsystem Decomposition</td>
<td>D3</td>
</tr>
<tr>
<td>28</td>
<td>Design GUI</td>
<td>D3</td>
</tr>
<tr>
<td>29</td>
<td>Code Generation</td>
<td>D3</td>
</tr>
<tr>
<td>30</td>
<td>Submit SDD</td>
<td>D3</td>
</tr>
<tr>
<td>31</td>
<td>Presentation 3</td>
<td>D3</td>
</tr>
<tr>
<td>34</td>
<td>Complete Implementation</td>
<td>D4</td>
</tr>
<tr>
<td>35</td>
<td>Merge modules</td>
<td>D4</td>
</tr>
<tr>
<td>36</td>
<td>Functional Testing</td>
<td>D4</td>
</tr>
<tr>
<td>38</td>
<td>Record benchmarks of Phase 2</td>
<td>D4</td>
</tr>
<tr>
<td>39</td>
<td>Record benchmarks of Current Phase</td>
<td>D4</td>
</tr>
<tr>
<td>40</td>
<td>Evaluate Benchmarks Results</td>
<td>D4</td>
</tr>
<tr>
<td>41</td>
<td>Submit Final Deliverable</td>
<td>D4</td>
</tr>
<tr>
<td>42</td>
<td>Final Presentation</td>
<td>D4</td>
</tr>
</tbody>
</table>
Outline

Introduction
- Problem Definition & Background
- Current System
- Project Management

System Design
- Design Objectives
- Architecture Overview
- System Decomposition
- System Deployment
- Persistent Data Design

System Implementation
- Technologies & Techniques Used
- Main Algorithm Used

System Evaluation
- Experiment Setup
- Evaluation Results
Design Objectives

- Allow 300 concurrent virtual machines.
- Efficiently balance use of resources to avoid overloading any single server.
- Provide enough resources per virtual machine to ensure reasonable performance.
- Lock virtual machines after the due date of the assignment.
Outline

Introduction
- Problem Definition & Background
- Current System
- Project Management

System Design
- Design Objectives
- Architecture Overview
- System Decomposition
- System Deployment
- Persistent Data Design

System Implementation
- Technologies & Techniques Used
- Main Algorithm Used

System Evaluation
- Experiment Setup
- Evaluation Results
Architecture Overview

- Moodle is implemented as a three-tier architecture: Interface, Logic, Storage.
- Interface and Logic are implemented in PHP, supported by a SQL Database.
- vMoodle links two new subsystems to the existing Moodle architecture: private cluster and public cloud for VM hosting.
Architecture Overview
Outline

Introduction
- Problem Definition & Background
- Current System
- Project Management

System Design
- Design Objectives
- Architecture Overview
- System Decomposition
- System Deployment
- Persistent Data Design

System Implementation
- Technologies & Techniques Used
- Main Algorithm Used

System Evaluation
- Experiment Setup
- Evaluation Results
Subsystem Decomposition

- The majority of our focus will be on the following three subsystems:

  - **vMoodle Web Interface**: provides users with a graphical web interface to interact the vMoodle system.
  
  - **Logical Infrastructure**: Links all the other subsystems together, processes user requests, and enforces load balancing.
  
  - **MySQL Database**: Responsible for storing persistent data, and enforcing its consistency.
Outline

Introduction
- Problem Definition & Background
- Current System
- Project Management

System Design
- Design Objectives
- Architecture Overview
- System Decomposition
- System Deployment
- Persistent Data Design

System Implementation
- Technologies & Techniques Used
- Main Algorithm Used

System Evaluation
- Experiment Setup
- Evaluation Results
Outline

Introduction
- Problem Definition & Background
- Current System
- Project Management

System Design
- Design Objectives
- Architecture Overview
- System Decomposition
- System Deployment
- Persistent Data Design

System Implementation
- Technologies & Techniques Used
- Main Algorithm Used

System Evaluation
- Experiment Setup
- Evaluation Results
Persistent Data Design
Outline

Introduction
- Problem Definition & Background
- Current System
- Project Management

System Design
- Design Objectives
- Architecture Overview
- System Decomposition
- System Deployment
- Persistent Data Design

System Implementation
- Technologies & Techniques Used
- Main Algorithm Used

System Evaluation
- Experiment Setup
- Evaluation Results
Technologies & Techniques

- Server run by LAMPP (Linux package of Apache webserver, MySQL database, PHP and Perl)
- Server run on a Linux (Ubuntu 10.04) Virtual Machine, which is in turn hosted by Oracle VirtualBox
- Software used for database setup and administration: phpMyAdmin, MySQL Workbench
- Server connects to host machines via SSH (Secure Shell) over a VPN (Virtual Private Network)
- All code created in PHP and Linux shell scripts (.sh) using gedit (a text editor).
Outline

Introduction
- Problem Definition & Background
- Current System
- Project Management

System Design
- Design Objectives
- Architecture Overview
- System Decomposition
- System Deployment
- Persistent Data Design

System Implementation
- Technologies & Techniques Used
- Main Algorithm Used

System Evaluation
- Experiment Setup
- Evaluation Results
Algorithm: Load Balance Analyzer

1: Retrieve host list and maximum resource settings
2: For each host:
   Retrieve current resource usage (average of last 10 entries)
   Set availability flag = True
3: For each (source) host:
   3a: If source host’s CPU usage > CPU maximum setting: critical resource = CPU
   3b: Else if source host’s RAM usage > RAM maximum setting: critical resource = RAM
   3c: Else continue to next host, this one is not overloaded
   3d: Initialize BestTarget = source host
   3e: For each (target) host:
      If target != source AND..
      target’s availability flag == True AND..
      target’s critical resource use < maximum setting AND..
      target’s critical resource use < BestTarget’s critical resource use AND..
      target’s noncritical resource use < maximum setting:
         Set BestTarget = target
   3f: If BestTarget != source host:
      VM = choose a VM from source host according to VM’s resource use (separate algorithm)
      If a valid VM was found:
         Set target’s availability flag = False
         Teleport VM from source to target (separate algorithm)

Running time: 1: O(n) 2: O(n) 3: O(n^2 + n*m)
Where: n = number of host machines, m = number of VMs per host
Outline

Introduction
- Problem Definition & Background
- Current System
- Project Management

System Design
- Design Objectives
- Architecture Overview
- System Decomposition
- System Deployment
- Persistent Data Design

System Implementation
- Technologies & Techniques Used
- Main Algorithm Used

System Evaluation
- Experiment Setup
- Evaluation Results
Experiment Setup

- Server runs on an Ubuntu VM supported by VirtualBox
- Server VM is hosted on a Windows Laptop
- Connects to actual private cluster host machines via VPN

Evaluations Completed:
- Web interface responsiveness,
- VM performance on private cluster vs. Server VM

Problems with host resource use evaluation:
- Difficulty duplicating VMs, hard coded template limitations
- No internet access or file sharing with VMs
- Inconsistent VM performance, particularly latency
- VM resource usage capped, negligible impact on host machine
Outline

Introduction
- Problem Definition & Background
- Current System
- Project Management

System Design
- Design Objectives
- Architecture Overview
- System Decomposition
- System Deployment
- Persistent Data Design

System Implementation
- Technologies & Techniques Used
- Main Algorithm Used

System Evaluation
- Experiment Setup
- Evaluation Results
### Evaluation Results

#### Web Responsiveness

<table>
<thead>
<tr>
<th>Interface Action</th>
<th>Response Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create VM</td>
<td>5.5</td>
</tr>
<tr>
<td>View VM</td>
<td>4.2</td>
</tr>
<tr>
<td>Stop VM</td>
<td>3.2</td>
</tr>
<tr>
<td>Run VM</td>
<td>2.4</td>
</tr>
<tr>
<td>Delete VM</td>
<td>2.8</td>
</tr>
<tr>
<td>Start Monitor Script</td>
<td>1.8</td>
</tr>
<tr>
<td>Start Analyzer Script</td>
<td>2.1</td>
</tr>
<tr>
<td>All Others Actions</td>
<td>&lt; 1.0</td>
</tr>
</tbody>
</table>

#### VM Performance

<table>
<thead>
<tr>
<th>Testbed</th>
<th>Average time per spinlock:</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM hosted on private cluster</td>
<td>6.40 seconds</td>
</tr>
<tr>
<td>VM running the local server (hosted on team member laptop)</td>
<td>5.46 seconds</td>
</tr>
</tbody>
</table>

**Analysis:**
- Reasonable web responsiveness, well under 30 second maximum
- Server VM performs significantly (17%) faster than VMs on private cluster host, despite lower hardware specifications: host may be limiting resource use somehow