CS 6343: CLOUD COMPUTING
Term Project (draft)

- Project: PaaS/SaaS exploration
    - Develop a SaaS application on a PaaS platform
- Project Steps
    - Study the PaaS platforms and summarize their APIs
        - GAE
        - Azure
        - Heroko
        - Cloud Foundry
        - Cloudify
        - Any other you are interested in
    - Design an application
        - Convert the standalone RoboCode to a web based application
        - Enhance it to become a Web based gaming system
    - Install PaaS: Cloud Foundry
        - Preliminary: IaaS environment supported by CF (OpenStack recommended)
    - Deploy the SaaS application on PaaS
        - The application should use supported language, DB, etc.
    - Enhance the security support in PaaS
        - To support a true role based model within each tenant
        - To support secure cross-tenant information sharing
        - To support information flow control (optional)
    - SaaS RoboCode functionalities
        - Support RoboCode editing, compilation, and playing
        - Support scoring and rewards for individual users
        - Use cloud data store to store data related to individual users, including their RoboCodes, scores and rewards they obtained, their specific configurations in the SaaS, etc.
        - Enforce access control
            - Consider system level data
            - Consider user specific data
            - Consider user groups, support group based sharing (can be considered as tenants)
        - Test your access control mechanism thoroughly to ensure the system is secure
    - Explore the impact of various load
        - Develop a client program to issue http requests to the SaaS application
            - Use multiple threads to simulate multiple clients
        - Varying the load issued by the clients and observe the behavior of the PaaS
            - Load should be dynamically changeable, and controllable through some interface
        - Deploy multiple instances of the SaaS application in different names (as different SaaS)
            - Group clients to send different loads to different SaaSs
        - Observe how the PaaS adapt to load changes
    - Study the performance of the application system on different PaaS
        - Install other PaaSs
            - At least one, more will be better
            - Compare the performance of different PaaS for this specific SaaS under different load patterns

Submissions

- Submission guideline
  - Each team only needs one submission through e-learning
  - Report submission
    - Attach the doc file during e-learning submission
    - The file name has to be <team-label>-report.doc or <team-label>-report.docx
    - Do not submit pdf
    - We will let you know your team label, e.g., A1, A2, …
  - Code submission
    - Zip or tar your source code and the deployable microservices
    - Zip file name has to be <team-label>-code.zip
    - Attach the zip file during e-learning submission
  - The workload distribution report
    - Just collect the logs you prepared week by week and combine them into one pdf file
      - Should be ordered by week and dated for each week
    - Principles in preparing the weekly detailed workload log
      - The items about the workload for each member should be distinct
      - When multiple members are together performing a task, the role of each member should be clearly stated (e.g., if multiple members are together performing a certain installation, then state who actually typed in the commands, who simply made observation, and who provided the guidance on what to do)
      - When multiple members are attempting to resolve the same problem encountered when performing a task, state clearly the problem, the resolution approach(es) each member came up with (even if they were not successful), and the references used for the resolution approach
      - When multiple members are helping implement the same component, state the role of each member (coding, testing, debugging, giving guidance, etc.)
      - Whenever multiple members are performing the same task (in the same or different roles), clearly state the percentage of contribution to the task
        - If the percentage cannot be agreed upon by the team members, state the believed percentage of each and let the TA and the instructor know about the issue
      - Provide any other information that can help us understand your contribution and efforts toward the project
    - Pdf file name has to be <team-label>-work.pdf
    - Attach the zip file during e-learning submission

Required information in the report (can include more than listed)

- In the cover page, provide team-label, title, and team members
- Introduction
  - Goal of the project
    - What you would offer in your system and why what you offered is important
    - What you should show in the report and/or in a demo of your project (at an upper level, based on the goal)
- Study of related work
  - Summary of related works (in paper or similar products available)
  - How your project is different from or is similar to some existing works
- Approach
  - System architecture

    - Activity diagram (workflow)
    - Architecture diagrams (from high level to low level decomposition of the system)
    - Description of the nodes in the architecture
  - Detailed design
    - For each component in the architecture, provide the list of code files for the component
    - Discuss the APIs of the important components and the algorithms used in some components, etc.
    - Indicate which components have been implemented and which have not been
      - You can simply do color coding or something alike and explain the indicator
  - Implementation details
    - Description of each code file (what it does) and the relations between the code files
    - Description of the system environment
      - Major components in the system environment
      - Your steps for installing these components (No need to give details, just refer to the online resources you used for your installation)
      - Problems encountered during installation of the system environment and how they are resolved
  - Experimental setup
    - Architecture of the experimentation system
      - The system you plan to build and explore in the experimental study can be a black box (or a few black boxes)
    - List of experiments and the goal of each experiment
      - What you plan to learn from each experiment
      - The data to be collected
      - The metrics you plan to use
      - The control parameters you plan to use
  - ...
- Experimental results
  - Clearly state the control parameters and the metrics for measurement in the results
  - Experimentation needs to be thorough and results need to be easy to read (e.g., use graphs and table)
- Installation manual
  - Focus on how to set up your program
  - The url and version number of the open source components needed to set up your system
    - No need to give installation guide for open sources
- User manual, discussing how to use your system
- Team member contribution
  - A high-level summary
  - Can be done based on the detailed system architecture, experimentation architecture and system environment
    - Indicate who have contributed to each component, what type of contribution (installation, design, coding, testing, debugging, experimental data collection, reporting), and at what percentage
- …

Project logistics

- Weekly documentation

  - Document the issues during converting standalone software to SaaS
  - Document the issues during PaaS installations
  - Document the tasks assigned, performed, and accomplished by each team member
  - Your exploration and experimentation results
- Regular meetings
  - There will be weekly meetings with the TA and bi-weekly meetings with the Professor
  - Discuss the weekly work progress by each team member
  - Schedule and milestones will be defined during the meetings
- Subgoals
  - PaaS study, first PaaS installation, and sample program deployment
    - If you wish, you can further divide the task into single machine and multiple maches
  - SaaSification, improvement, deployment
  - SaaS implementation, deployment
  - Client program for load simulation
  - Performance exploration and PaaS behavior observation
  - Installation of additional PaaSs
  - Performance and PaaS feature comparisons
- Some early scheduling
  - 1st week
    - Large team formation, schedule exploration
    - Read PaaS documents, especially Cloud Foundry
    - Try to install Cloud Foundry on your personal computers
  - 2nd week
    - Email large team formation to TA before class, schedule setting in class
  - 4th week, at least
    - PaaS installation item done
    - SaaSification item done