Leah Chatkeonopadol
Smruthi Manjunath
CS 263
4.8.13

Vision Statement

Technology Description:

(Leah) For my technology tutorial, I will demonstrate how to use Google App Engine (GAE). Specifically, I will explain how to develop and host Python web apps on GAE, from downloading the App Engine SDK to deploying the application and configuring it to run on the web server. This explanation will include an overview of the various capabilities that GAE offers, such as data storage and data processing, as well as a closer look at several of the services and tools available. I also plan to create a demo web app that uses these APIs so that in the tutorial, I can use the app to show how the APIs work in detail.

(Smruthi) For my technology tutorial, I will be going over the working of python virtual machine internals. It mainly involves the study of the python runtime execution model, the interpreter, its optimization and garbage collection in CPython, which is an implementation of the Python programming language in C (generally considered same as python). CPython is a bytecode interpreter. I am planning to write a simple python code and step through the code to demonstrate how the code gets executed internally and how the garbage collector works.

Project Description:

For our project, we will implement an API Description Generator in Python for Python web apps on Google App Engine. The goal of this work is to enable automatic generation of a more readable form of documentation for a web app.

The first part of the project will consist of an API parser and validator for Python. This will be implemented using the JSON library support available in Python. The parser will identify different tokens, such as data types and methods, and semantic information, including comments. It will also check the syntax for invalid statements. If the code is valid, we will use the output from the parser to generate a description of the input API. For example, the comments in the code may be translated into descriptions of methods. The API description will be in the form of a JSON file that is easier for a user to read and understand. It will identify the different services the app provides, as well as the inputs and outputs of those services. Ultimately, our API Description Generator will be an automated tool that can be run on given web apps. A similar tool has already been implemented in Java; our project will be based on this earlier work.

We expect description generation to be slightly simpler in Python as opposed to Java, since Python decorators will be helpful in identifying functions, arguments, and other relevant information. However, we do anticipate facing several challenges. One such challenge is a result of the fact that Python is dynamically typed. The inputs and outputs will be difficult to identify, especially if the code is not properly documented. At this time, in order to simplify the project, we will assume that the needed information about data types is present in the comments. Another challenge is obtaining other information that may not be present in the code of the API, such as the URL and path information. We will pull this extra data from the configuration file app.yaml, which Google App Engine requires to be supplied with the API. In general, any necessary information that is not provided will have its field left blank.

We will start the project by learning how to use pythong and Google App Engine. First, we will build our API Description Generation in pythong. Then, the same will be deployed on the Google App Engine as an automated tool. We also plan to study Python VM internals to better understand how our parser/validator should function. Hopefully, all of this will provide enough background for our project.