**CS 497/597: Programming Assignment 2**

**Load Balancing**

**250 points**

**Due date: 10/07/2014**

Course: Cloud Computing

Term: Fall 2014

Instructor: Vijay Dialani

# Introduction

Social networks such as Facebook, LinkedIn, Twitter and Google+ allow a community of users to connect and interact by exchanging messages amongst them. These messages can refer to other users using ‘@’ prefixed to the username or they could be referencing a hashtag topic prefixed by a ‘#’. Each user could reference a group of users using ‘@@’ sign that references a group. Recipients of the message need not be online to receive the message, messages held for them at the server and are delivered at subsequent login. Most of these social networks provide an API to access the contents, one such example of Twitter API can be found at <https://dev.twitter.com/docs/api/1.1>

As the number of users grow the services need to be implemented in such a manner that they can scale horizontally. Load balancing techniques are used to improve the performance of the web services.

# Load Balancing

As a part of the first assignment two web services were developed: Friend and Follower Service, and Tweet Service. In this assignment, the goal is to make tweet service scale horizontally, based on the load experienced by its service instances. You can use open source HTTP load balancers like HAProxy <http://www.haproxy.org/> to dynamically route the requests to multiple tweet service instances. In addition develop a load monitoring service that is collocated with the tweet service instances and support the following functionality:

|  |  |
| --- | --- |
| **Resource** | **Description** |
| GET monitor/processingtime | Returns a collection of processing time for buckets in percentile increment of 10%, since start of the service. For example, Fastest 10% took 30 microseconds/request, next 10% took 45 microseconds/request. |
| GET monitor/queuedepth | Returns the number of pending html requests being handled by the service. |
| GET monitor/qps:resolution | Returns the collection of number of messages processed with the specified resolution. The collection size is restricted to top 100. The resolution can be in minutes/hours/days/months only. |
| GET monitor/errors:type | Returns the collection of messages that resulted in an error. The collection should aggregate error counts for valid HTTP error codes. |

# Load Monitoring Client

Develop a simple web client using your language/framework of choice to demonstrate the functionality of your web service. The client should support the set of function calls mentioned earlier. The name of the web client should be *Monitor* and it should be packages as a war for deployment in apache tomcat 8.0 or higher. The client should allow monitoring of all the instances of tweet service.

# Extra Credit (20 points)

* Develop a Web Client that uses CSS and whose appearance could be easily modified.

OR

* Demo your project and design in the class

# Submitting the Assignment

* There must be a README that mentions the name of the student, course number, and assignment number. It should provide details about any assumptions made, configurations and comments.
* The assignment must have a pom file that generates war files that can be deployed in Apache tomcat server.
* Assuming all your assignment files are in your directory ~/cs597/p1 on onyx, change the directory to p1 and type the following command.

submit vijaydialani cs597 p2

or

submit vijaydialani cs497 p2

The submit command will provide you with a timestamp and directory path name to confirm your submission.