**INFO 6205, Summer 2017**

**Homework 2 Due June 2**

The purpose of this homework problem is to learn how use linked lists and queues support elastic websites that must respond to fluctuating demand. When the number of requests coming in for processing is low then one server will be sufficient to handle the loud. As the demand increases, a slower response time will be experienced by the users due to the long wait time for processing.

You are to write a simulator to study the user requests for services in relation to an elastic pool of virtual machines that are available on demand. In other words, more machines are added from the pool when load gets heavy and returned when the traffic slows down.

The data structure will consist of a point of entry to the website which will receive all requests for services, a dispatcher of the arriving requests, as well as, an array of services handling the execution of incoming requests. Each service has a queue of waiting requests. Each service handles a max of n requests to be in its waiting queue. In addition, there is a pool of available virtual machines that can be employed to reduce the load on the other machines. For example, the dispatcher could sense that all running services are maxed out and thus requests an additional machine to be employed.

The steps will be follows:

1. Requests arrive at a given rate r at the website entry point. The requests are immediately inserted into the incoming queue.
2. The dispatcher picks a request for the incoming queue and adds it to one of the available services (running on one of the virtual machines). If all machines are maxed out then the dispatcher will request the addition of a new virtual machine to be added from the pool.
3. A service will retrieve requests from the queue of waiting requests and process it. The processing time will be p.

Your job is to simulate the load on the website and learn how many machines you need to handle a fluctuating load. You must design a swing app that will allow you to vary the various parameters and see how the system behaves under different request arrival and processing rates. Also, you must incorporate total response time and its impact on the best scheduling strategy by the dispatcher (response time is time from when the user submits the request to the time the request is completed by the service).

You will need to learn how to use timers to solve this problem. Also, read more about elasticity in cloud computing.

You can work on this problem as a team of 1, 2, or 3 of you and your peers.

