**CS423**

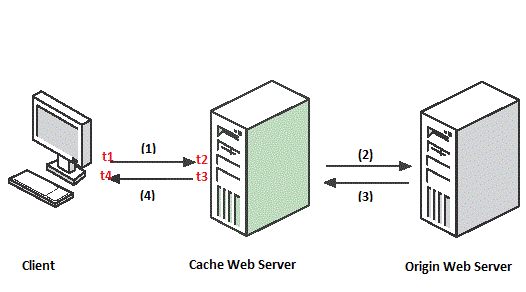
**Spring 2017**

**Instructor: Sheyda Kiani Mehr**

**Project: Client-Server web application**

Design a web client-server application which the server entity applies a cache server to serve a client. The client and the proxy cache can be coded with any of these programming languages: python, Java, C, C++ in any Platform. In this project, you need three different machines Client, Cache server and Origin server which the processes are explained below:

The Origin Server can use Apache2 in one of the GENI machines applied in the slice you made before. Cache server code should be in one local machine, Client code should be in another local machine (your teammate). The client code asks for a file from cache server. With the request that the client asks it will put the Epoch time of the machine in the header and sends it to the caches server. The cache code receives the request for the file and start checking in the directory if the file exists, cache sends the file to the client, but if the file does not exist in cache directory, then the cache will ask origin server for the file. Origin server will send the file to the cache and cache will save it in the directory and send it to the client. The cache webserver gets the Epoch time of the cache machine at the moment it receives a request from the client and at the moment it sends the file to the client and put it in the HTTP header and send it to the client.



Client after receiving the file will calculate Request time, Process time, Transfer time, and Total time as shown below.

1. Client HTTP Request with the Epoch time t1
2. Cache estimates Epoch time t2 and send an HTTP Request to the server if the requested file is not in the cache. If file exists, you the code will skip number (3.)
3. Response from Origin Server to Cache Server which contains the file.
4. Cache server estimate t3 and Response to Client which contains the file and Epoch time t2 and t3
5. The next step is when the Client received the whole file, it will estimate Epoch time t4. Then it will calculate the request time (t2-t1), Process time at the cache (t3-t2), transfer time (t4-t3) and the total time which is (t4-t1.) The last step in to show how much was the throughput in MBps.

\*Please Print HTTP headers in both machines and use comments to make your code clear.

\*(t4-t1) = (t2-t1) + (t3-t2) + (t4-t3), Throughput = Size / (t4-t1)

\* You may want to use functions related to HTTP Client-Server Application in your code instead of TCP socket functions.