**Problem Statement**

The task is to design, implement and test a load balancing algorithm that can load balance video packets across servers. Each server has a fixed processing capability but dynamically changing load. The server processing capability is in units/sec. Two files will serve as inputs: one with server processing capabilities - "server\_capabilities", and the other with load on each server - "server\_load". A server executable will be provided to read the server loads from "server\_load" file. The video stream has an incoming rate of 1 packet/sec and each packet requires processing worth of 5 units on the server. The algorithm needs to consider the packet processing time, load read from the server, and server's processing capability while load balancing the packets across servers.

**Server Capability**

For example, the processing capability for 4 servers can be defined as:

Server1 Server2 Server3 Server4

30 25 50 40

where the server processing capability is in units/sec. The values in the file are tab delimited.

**Server Load**

The load on the server can be defined as:

Server1 Server2 Server3 Server4

-1 160 150 200

50 130 80 150

120 90 175 160

130 140 95 190

The numbers represent each server's current load in units. -1 represents that the server is currently down and should not be considered for processing any packets. The values in the file are tab delimited.

**Algorithm requirements**

Given these inputs the algorithm/solution should be able to:

1. Read in the server capability file and determine the number of servers from the file. Also, read in the number of packets that you are considering for the packet stream.

2. A server executable will be provided to get the server loads. On connecting to the server, the client needs to send a message saying 'Current load on the server?' and the server will respond with current load in units. You will need to spawn n instances of these servers running at ports 8001, 8002, 8003 and so on.

Each server can be executed as:

./server 8001 server\_load &

...

./server 8003 server\_load &

where server\_load is the file containing server load values. The file should be in the format specified above.

3. All video packets are "similar" and have similar processing times. There is no packet stream provided. Assume incoming packets.

4. Load balance the packets across servers based on the processing capability and the current load on the server. This would mean assigning the packets to a particular server.

5. The load returned by each server is defined as the cumulative load on the server at that time. After every 5th packet, the next set of loads on the server MUST be read.

6. The output should be a text file showing the load balanced output on the server after every packet in the following format:

Packet Current Server Loads Chosen Server

1 <server loads> Server1

2 <server loads> Server2

...

20 <server loads> Server4

Where 1st column is packet index

next will be server loads when the packet arrives

last column is the server that was picked for processing the packet

7. For testing you should be able to generate 3 input files for server capability, 3 input files for server load and 3 output files with the result of the algorithm. The files should have names: server\_capabilities[1,2,3], server\_load[1,2,3], load\_balanced\_output[1,2,3].

8. Basic error handling is required.

9. Include a README with your submission describing the overall design, details of the files submitted and how to execute your code. Also, include any assumptions that you might have made.

**Further information that may be useful while designing the algorithm**

a). Initial server load determination and 1st packet arrival is at the same time.

b). If two servers are able to process packets at same time, then the packet should be sent to server with greater processing capability.

c). Server load is read from the file every 5 packets and the new value is inclusive of all previous loads.