CSSE 492 Advanced Computer Networks

Project 1: Proxy Cache Replacement Simulator

1 Background

A web proxy manages web objects that were previously requested by its local users. The cached objects are stored in the proxy's cache (i.e., memory and disk storage) and will service the future requests of its local users. Therefore, the proxy cache can reduce bandwidth consumption, load on web servers, and user-perceived latency (i.e., response time). However, cache size is finite at the proxy. As a result, the proxy needs to employ a cache replacement algorithm to maximize the benefits of caching. This project will allow students to evaluate different cache replacement algorithms and compare their performance in terms of hit ratio and response time.

2 Requirements

You are required to evaluate the following cache replacement algorithms:

- LRU, which evicts the object that was requested the least recently.
- LFU, which evicts the object that was requested the least frequently.
- GD (GreedyDual-Size), which is described in Figure 5 of the paper "WC3".

The simulator called **ProxySim**, can be configured with various cache sizes. The cache entry for each object is allocated/deallocated in a multiple of 1KB, depending on the object size. The minimum allocation/deallocation unit is 1KB. The simulator starts with an empty cache, which means the first access of each object will be a cache miss.

The command to run the simulator is:

./ProxySim cache_size LRU[/LFU/GD] requests.txt

Where cache_size is an integer, representing the cache size in KB, and requests.txt is a file for user requests sorted in the arrival order. requests.txt is 3-column file: the first column represents the object ID (integer type), the second column denotes the request object size in bytes, and the third column is the time (in ms) to fetch the object from its web server. The file is ready to be downloaded.

The simulator computes and outputs the average response time for the request stream in requests.txt. If the object is in cache (cache hit), the response time for this object is 5ms; otherwise, the penalty for a cache miss is also 5ms. We also assume that the cached objects are always up-to-date.

For each replacement algorithm, you need to get the average response time and hit ratio (# of cache hits /# of requests) for various cache sizes. The cache sizes include 100KB, 500KB, 1000KB, 5000KB, 10000KB, 20000KB, 30000KB, and 32000KB.

Assume running the command below:

./ProxySim 100 LRU requests.txt

The output should be:

```
---------------------------------------------------------------------------
The proxy cache uses LRU replacement, with the size of 100KB.
The hit ratio = 34%.
The average response time = 76 ms.
---------------------------------------------------------------------------
```

For each replacement algorithm, you need to collect the results for all the aforementioned cache sizes. Then you need to plot two graphs for the three replacement algorithms (in Excel). One plots the hit ratio with respect to the cache size for the three algorithms, and the other depicts the average response time with respect to the cache size for the three algorithms. From the graphs, we can compare the performance of the three replacement algorithms.

3 Submissions

You are required to submit the source file, Makefile and a graph file, contained in a single .tar file. You need to submit the .tar file to zhuy@seattleu.edu, before 10:00AM, April 22, Tuesday.

4 Grading

The total score for this project is 20 points. In order to receive a good score, you must ensure: (1) the simulator is compiled and executed successfully; (2) the graphs generated from the simulation results closely characterizes the performance of different replacement algorithms under various cache sizes; and (3) the submission meets the deadline.