CIS6930 Network Data Streaming

Project I Individual Project

Implementation of Hash Tables

1. Description

In this project, you will implement multi-hashing table, Cuckoo hash table, and d-left hash table.

Multi-hashing table
Input: number of table entries, number of flows, number of hashes – for demo, they are 1000, 1000 and 3, respectively
Function: generate flow IDs randomly, assume each flow has one packet, record one flow at a time into the hash table, and ignore the flows that cannot be placed into the hash table.
Output: number of flows in the hash table, and the list of table entries (print out the flow ID if an entry has a flow or zero otherwise)

Cuckoo hash table
Input: number of table entries, number of flows, number of hashes, number of Cuckoo steps – for demo, they are 1000, 1000, 3, and 2, respectively
Function: generate flow IDs randomly, assume each flow has one packet, record one flow at a time into the hash table, and ignore the flows that cannot be placed into the hash table.
Output: number of flows in the hash table, and the list of table entries (print out the flow ID if an entry has a flow or zero otherwise)

d-left hash table
Input: total number of table entries, number of flows, number of segments (hashes) – for demo, they are 1000, 1000, and 4 respectively; each segment has 250 table entries.
Function: generate flow IDs randomly, assume each flow has one packet, record one flow at a time into the hash table, and ignore the flows that cannot be placed into the hash table.
Output: number of flows in the hash table, and the list of table entries (print out the flow ID if an entry has a flow or zero otherwise)

How to implement multiple hash functions?

For k hash functions, you may define an integer array s[k] and initialize the array with randomly generated numbers. When computing Hi(f), you may compute H(f XOR s[i]), where XOR is a bitwise operator with 0 XOR 0 = 0, 1 XOR 0 = 1, 0 XOR 1 = 1, and 1 XOR 1 = 0.

2. Dates

Handout: 9/10/2020
Due in Canvas: 10/10/2020

3. Programming Environment

Programming language: Java, C, C++, C#, Python
Operating System: Windows, Mac OS or Linux
Programming Tool: Eclipse, IntelliJ, Jcreator, Kawa, Netbeans, … whatever you like.

To use Eclipse, please go through the following list:

1. Download JDK from: https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

2. Download Eclipse from: http://www.eclipse.org/downloads/

3. Here is a link for eclipse tutorial: http://eclipsetutorial.sourceforge.net/totalbeginner.html

4. Here is a tutorial for socket programming in Java: http://java.sun.com/docs/books/tutorial/networking/sockets/

4. Code Submission

You must submit the source code and one output file for each hash table using the demo input parameters given earlier. The first number in the output file must be the number of flows in the table, and it must be shown in the first line by itself in the output file. Include readme.txt to explain your files.

Submit the project through Canvas:

1) Go to https://elearning.ufl.edu/
2) Click “Login to e-Learning”
3) Login with your gator link username/password
4) Go in CIS6930 Network Data Streaming
5) Click “Assignments” and submit your project

This is an **individual** project. We will run an automatic tool to catch submissions with identical or similar code. There will be no late submissions.