Assignment2

Due Date: 02/13/2013

Berkeley DB

You should use Berkeley DB for this assignment. The latest version of Berkeley DB is installed on the flip servers and you can find it at /usr/local/apps/berkeleydb/berkeleydb.5.3. The Berkeley DB library Jar file (db.jar) is at /usr/local/apps/berkeleydb/berkeleydb.5.3/lib/db.jar.

In order to use Berkeley DB, you should update the LD_LIBRARY_PATH variable in your session using:
export LD_LIBRARY_PATH=/usr/local/apps/berkeleydb/berkeleydb.5.3/lib

and place its library Jar file in your class path.

Data Set

The data set contains the information about various movies downloaded from IMDb (www.imdb.com) web site, where the information about each movie is stored in a separate xml file. The data set can be found at: /scratch/cs440/imdb and its tar file is at /scratch/cs440/imdb.tar. You can access the data set from all three flip machines (flip1, flip2, flip3) from on and off campus.

What you need to do

For this assignment you are required to write a program in Java that stores all of the xml files in a table in Berkeley DB and answers some types of queries. Each tuple should contain a file's name (a string value), its size (in bytes, an integer value), and its content (a string value). You should use the appropriate access path for each type of query. Your program should implement the following types of queries over the IMDb data set:

1- [20 points] Inserting the information about all files in the data set into a Berkeley DB table.
2- [10 points] Answering point queries for the file name over the table. For example, finding a file whose name is "282441.xml".
3- [15 points] Answering range queries over file name. For example, finding all files whose names are from "20000.xml" to "30000.xml".
4- [10 points] Answering point queries over file size. For example, finding all files with size equal to 2000.

5- [15 points] Answering range queries over file size. For example, finding all files whose sizes are from 2000 to 3000.

6- [20 points] Answering range queries over both file name and file size. For example, finding all files whose names are from "20000.xml" to "30000.xml"and their sizes range from 2000 to 3000.

7- [10 points] Finding all files whose contents contain an input string value. For example, finding all files that contain a specific string such as "Arnold".

Note: Performance matters! The programs that do not choose the efficient access paths or take too long to complete will lose 50% of the scores.

Input Format

The first argument of your program will be the task switch (1 for the insertion, 2 for the first query type, 3 for the second query type, …), then the first parameter of the query, and so on. For example, for the queries of type 6, the input will be (6 20000.xml 30000.xml 2000 3000)

Output Format

You should store the answers for each type of query in a separate output file. The outputs for all types of queries contain only the name and the size of the files (no content). The format for the name of each output file should be "FirstNameLastName_TypeNumber" where FirstNameLastName are the first and last names of the contacting group member (See the next part).

What to turn in?

Since we will use flip servers to test and grade your program, you should make sure that your program runs on these servers You should include the source code and the executable jar file of your program in a zipped file and turn-in the zipped file via TEACH. The name of the zipped file should have the names of all the group members and be based on the following format:

CS440_HW2_FirstnameLastname1_ FirstnameLastname2_ FirstnameLastname3

, where FirstnameLastname1 is the first and last name of the group member who submits the assignment, i.e. contacting group member. Each group should have only one contacting member, i.e. only one person per group should submit this assignment.