CSCI 620/Section 02/Mior, Introduction to Big Data, Spring 2235

Assignment 4 – Document databases

Description

Create the Posts and the Users collection in MongoDB with appropriate types and populate it with Stack Exchange data from Assignment 2. You may need to install a local copy of MongoDB to complete this assignment.

Posts:
```
{       _id : _,
        OwnerUserId: _,
        PostTypeId : _,
        CreationDate : _,
        Title : _,
        Body : _,
        ViewCount : _,
        Score : _,
        Tags : [iptables, ekiga, …],
        Comments : [ {
                UserId : _, Text: _,
                CreationDate: _, Score: _
        }, … ]
}
```

Users:
```
{       _id : _,
        DisplayName : _,
        CreationDate : _,
        LastAccessDate : _,
        Views: _,
        Badges: [ {
                Name: Student,
                Date: 2010-07-28},
        … ]
}
```

Note that Owners.UserId and each UserId in Comments in the Posts collection refers to _id in the Users collection. When you have null attributes, the documents must not contain those fields, i.e., do not insert null or dummy values in MongoDB.

Your tasks

1. Provide a program to load the data from your relational database or the original data files into MongoDB. Your program needs to load the whole database in approximately one hour using commodity hardware.

   Hint: You may find the JSON support in Postgres helpful.
   https://www.postgresql.org/docs/current/functions-json.html

   (25 points)

(continued on next page)

2. Provide a program issuing the following queries over the MongoDB database. Each should consist of a single query (find and/or aggregation pipeline) and you need to report evidence that you retrieved what was expected. Report the time your queries took to run. **(10 points per query)**

    2.1. Names of the top 10 most popular badges earned by users within a year of creating their accounts.
    2.2. Display names of users who have never posted but have a reputation greater than 1,000.
    2.3. Display name and reputation of users who have answered more than one question with the tag “postgresql”.
    2.4. Display name of users who posted comments with a score greater than 10 within the first week of creating their accounts.
    2.5. The tag names of the tags most commonly used on posts along with the tag “postgresql” and the count of each tag.

3. Provide a brief explanation of the execution plan for each of the previous queries.

    (Hint: https://docs.mongodb.com/manual/reference/explain-results)
    (10 points)

4. Taking the previous queries into account, create appropriate indexes where they are required. Document your decisions, provide the code to generate them, and analyze the performance differences using a program. That is, for each query, show your times with and without indexes.

    (Hint: See https://docs.mongodb.com/manual/indexes)
    (15 points)