CS 5513 - Spring 2019 - Homework 3
Assigned: 4/2/2019
Due: 4/16/2019 by 3:00 PM to the class website
Maximum Points: 75 points (without the bonus problem) or 85 points (with the bonus problem)

Notes:

- **Homework answers must be typed and submitted by 3:00 PM to the class website; do not put your work under my/TA's door or in my/TA's physical/email mailbox; otherwise it will not be graded.**
- **Homework is individual work; it must be done by you only, no collaboration with anyone else is allowed.**
- **Late homework will be accepted until 11:59 PM on the date following the due date with 5% (of the maximum points) penalty. Late homework submitted after this time will not be graded.**

REQUIRED PROBLEM: Suppose that the popular student newspaper, The Daily, has decided to obtain an online presence by posting the latest news and feature articles online and allowing readers to post comments on these news and feature articles. For this purpose, The Daily obtained 3 sites (computer nodes): gpel9.cs.ou.edu, gpel10.cs.ou.edu, and gpel11.cs.ou.edu. The editor of this newspaper wants you to create the database backend for a news article and feature article archiving system (using MongoDB) using the three sites, and ensure that the news archiving system is highly available and flexible so that news and features can be inserted at any time without having to bring the system down, and so that even in the face of network failures, readers can still access portions of the news archive. The editor of this newspaper tells you that the following queries will be frequently issued:

1. Insert an (news or feature) article into the archive.
2. Retrieve the latest news from the archive.
3. Retrieve the top 10 most read news in the past month.
4. Add a comment to an article.
5. Add a story to the 'similar stories' section of all articles that contain a given word and that have been published within the past 2 years.

Every news or feature article has a unique ID (article_id). For each article, its attributes that must be available in the archive at the time of implementation are the following, where attributes of the form attribute_name(attr_1, attr_2,…) denote composite attributes, and attribute_name{…} denote multi-value attributes:

- **Article**: article_type, article_section, article_id, post_on_date(day, month, year, time), reporter_name, similar_stories{…}, num_times_read, article_text, comments{(comment_id, article_id, user_id, posted_on_date(day, month, year, time), comment_text, score)}

  An example entry in the archive:

  ```
  'news article', 'Sports', 75, (31,12,2017,(23:59:59)) ,
  'Bridgette Nolan',{'thedaily.com/sports/oklahoma-
  football-sooners-land-kentucky-graduate-transfer.html',
  ```

```
'thedaily.com/sports/oklahoma-football-sooners-look-to-
fill-void.html'}, 11357, 'Despite the Cowboys' last
loss last season, they have several options to replace
their linebacker…', {(12, 75, 'dtillman@gmail.com',
(31,12,2017,(09:29:31)), 'Keep up the work with your
fake news!', -99 )}
```

where the format of the date is (31,12,2017,(23:59:59)) meaning December 12th, 2017 at 23:59:59.

Using the above scenario, perform the following tasks (a-g) using MongoDB on the GPEL machines:

a) Produce a dataset (real or synthetic) for the task (or use case) discussed above and write it to a file named *articleData.txt*. You can choose whatever human-readable format for your data. You can obtain this dataset directly from another source, or you can generate it yourself with a program. This dataset should have a size of at least 50 entries. Once you generate this dataset, print out its first 5 entries. Do not print out the whole dataset.

b) Using the programming language of your preference (Java, Python, etc.), write a program '*txt_parser*' that takes your dataset file *articleData.txt* as input, and then generates a MongoDB script named '*dataInsertion.js*' that contains the "MongoDB commands" to create the collection(s) that implement the database, and then inserts every data entry contained in *articleData.txt* into the database.

c) Run the '*txt_parser*' program and generate your 'dataInsertion.js' script. Then populate your MongoDB database by running 'dataInsertion.js' at the MongoDB shell.

d) Create the necessary indexes to support your queries. Justify your decision for creating them.

e) Implement the five queries (1-5) described above <u>using only the MongoDB query language (not Java, Python or any other programming language)</u>. Run each of your queries one time. Provide screenshots of the output of each run.

f) Design a suitable replication scheme to answer your queries. This replication scheme must use at least three GPEL machines (for example: gpel9.cs.ou.edu, gpel10.cs.ou.edu and gpel11.cs.ou.edu). <u>You need to describe and justify your replication design **in detail**</u>. Then, implement your design in MongoDB. Your implementation must include the code to set up the replication scheme. For this task, you only need to implement replication; you do not need to implement sharding. You also need to provide a screenshot of the output of the command rs.status() when you run it in your replica set. This is to ensure that your replica set has been properly configured.

g) In this task, you will do a simple simulation of the failure of the primary node. To do this, perform the following sub-tasks:

   i. Connect to the machine running the primary of your replica set and stop the MongoDB process running on that machine, using kill instrustion. Do not attempt to shut down or restart the entire GPEL machine (you do not have user privileges to do so).

   ii. Immediately connect to another active MongoDB server in your replica set and run the command rs.status(). Take a screenshot that shows that no primary has yet been elected.

iii. Wait until a primary has been elected, and then run your queries one time each again with the remaining nodes. Provide screenshots for the output of each run.

BONUS PROBLEM (optional) (10 points) your answer to this bonus problem will be graded only if you also completed ALL the tasks of the required problem in this homework assignment):

Select a commercial or open-source ***graph*** NoSQL system of your choice. Read its documentation and perform the following two tasks:

1) Describe the system's data model **in detail**.
2) Describe **in detail** an appropriate use case for that system and justify it based on your description of the system. Explain **in detail** why that use case is not appropriate for a relational database system.

Note that the intention of this problem is for you to understand the fundamental and/or innovative database concepts implemented by that NoSQL system. The intention of the problem is not for you to only report the query syntax of the system. You need to provide references for your answers to this problem. The PDF files for the references must be submitted as a part of your submission (submit them as separate PDF files; do not submit them as one ZIP file).

Important notes:

- To use MongoDB in OU's system, you need an account in the gpel machines in the School of Computer Science (CS). If you do not have an account on these machines, contact the CS system administrator, Mr. John Muller (jmuller@ou.edu).
- For launching a MongoDB server in a machine, you need to ensure that the port is free and that it is not being used by another process (of possibly another user) at the moment you want to start the MongoDB server. To avoid conflicts with other users, it is recommended that you use the ports 2xxxx to 2xxxx + 3, where 'xxxx' are the last 4 digits of your OU ID.
- All of your files must be well documented. Your work will be graded based on correctness as well as documentation.
- Before working on this homework assignment, you should have reviewed the Lecture Topic 7 "NoSQL Systems" slides posted on the course website and the references listed below.
- Submit all your files and results.
- Keep a soft copy of all your files and results in your directory. To check if your program works correctly, we will have online demonstration. The exact time and date for the online demonstration will be announced later. You MUST NOT modify your files or results after you turned them in. You will get a zero grade for this homework assignment if you violated this rule.

References:

1. David Hows, Eelco Plugge, Peter Membrey, and Tim Hawkins. **The Definitive Guide to MongoDB. A complete guide to dealing with Big Data using MongoDB**. Appress, 2nd edition, 2015. *Available at the OU Digital Library*.
2. **The MongoDB manual** https://docs.mongodb.org/manual/

3. **MongoDB use cases description** https://docs.mongodb.org/ecosystem/use-cases/product-catalog/