**Loel Nelson**

**ICS 490: Big Data Storage**

**Homework Assignment #2**

**Due: See Syllabus**

**Total: 50 Points**

**What to submit?** submit a **copy** to D2L class on the due date. If you are going to hand-write, please make sure you write neatly, take a picture and upload picture.

**Question 1 (10 Points): Parallel databases:**

Consider two relations R(a,b) and S(c,d) that are stored in a parallel database management systems where each relation is horizontally partitioned across N = 3 nodes. That mean that each node of the parallel database locally stores approximately 1/3 of the tuples in R and 1/3 of the tuples in S. Assume further that the tuples of R are randomly organized across machines (i.e., R is round-robin partitioned across machines) while the tuples of S are hash-partitioned on the key S.c.

Given the below query, draw a query plan to show how the query is executed on a centralized database management system. Then explain how the query will be executed in the 3 nodes of the parallel DBMS. Make sure to clarify **when and how** data re-shuffling is needed.

SELECT a, avg(d)

FROM R, S

WHERE R.b = S.c AND S.d > 0

GROUP BY a

If more than one relation on a machine, then scan S, scan R. Partitioning R and S on (a,b) for join will require repartitioning for the aggregation, but partitioning R and S on ‘a’ for join will allow aggregation with no further repartitioning. The coordinator obtains the necessary number of servers. If we hash on R(b) before executing the same query it would avoid the re-shuffling phase and it would compute the aggregates locally.

**Diagram

Description automatically generated**

**Question 2 (22 Points): Hadoop Distributed File System (HDFS):**

2.1) (10 Points) Read the white paper “10 Hadoopable problems” by Cloudera. The paper discusses 10 real-world Hadoop use cases. Choose one of the cases discussed in the paper and answer the following questions.

2.1.a) Which use case did you choose? I chose Recommendation Engine and the dating network problem.

2.1.b) What are the main challenges that led the organization to think about using Hadoop? Why do you think these challenges cannot be addressed using a traditional computing system or a DBMS? The main challenge of the dating network was the company growing and getting new users and the user wanting to be “matched” to more users and widen the search for a potential match. As stated, they originally did use a traditional system but as compatibility criteria grew and also the customers the DB needed to grow as well and fast. This was not possible with the old or legacy system but is exactly a key thing about Hadoop “proven at scale” which is able to grow with the company and increase the storage of data as needed.

2.1.c) Explain in your own words how the organization used Hadoop to improve their business. The dating network used Hadoop to help connect more users on more criteria and was able to provide a bigger and able to grow database to store the ongoing data that keeps coming in to the system. Seeing as how this online dating is really the new way to do it this is much needed with more users everyday turning to dating services versus the traditional meeting a bar or whatever.

2.2) (8 Points) “*Failures are the norm other than the exception*” is one of the main motivations behind introducing the Google File System and the Hadoop Distributed File System (HDFS).

2.2.a) Explain the rationale of this motivation. Reminds me of the scene from the movie Big Daddy where at the end the father says, “I don’t think you will screw up I know you will”. With this statement saying the same thing to me, its not a matter of if the systems will fail but when. Knowing this we can change the ways things are taken care of, which is what is a main focus of the 2 file systems.

2.2.b) Describe, in your own words, **three protocols** that are implemented by HDFS to accommodate this motivation. The HDSF uses the heartbeat and DataNodes which are located on each server and because of this heartbeat signal the NameNode knows that the DataNode is online. If the DataNode fails to issue the heartbeat than the NameNode knows to take that DataNode offline and reroute the data blocks to different nodes.

The NameNode recovers fast by loading the namespace image into memory then replays the edit logs then receives the enough data blocks to get back at it.(from slides)

2.3) (4 Points) Explain the advantages and disadvantages of having a single master node in HDFS. First, a disadvantage of a single master node or NameNode is the block of data depends on the ram of the host namenode, favoring small number of large files vs. large number of small files. If the namenode runs out of ram and crashes the HDFS will be temporarily out of commission until a new NameNode becomes active.

**Question 3 (18 Points): Map-Reduce:**

For each of the following questions, explain the map and reduce functions you will use to produce the required output. Make sure to clarify the input and output keys and values for both the mapper and reducer. Also for the given input examples, show the mapper outputs and the reducer inputs.

3.1) (6 points) **Input:** a collection of text files.

**Required output:** For each character of the alphabet, fine the average length of all words that start with that character. For example, for input:

*No now is definitely not the time*

The case-sensitive output should be:

*N 2.0 -- 2/1*

*n 3.0 -- (3+3)/2*

*d 10.0 –- 10/1*

*i 2.0 –- 2/1*

*t 3.5 (3+4)/2*

|  |  |  |
| --- | --- | --- |
|  | Key | Value |
| Mapper input | Record\_id | A collection of text files |
| Mapper output | Char | Sum of lengths for starting letter |
| Reducer input | Char | Length of word and count |
| Reducer Function | Avg = (sum of length for each starting letter) /( total count of words with that starting letter) |  |
| Reducer Output | Average length of words containing that starting letter | Avg() |

Look at the word grab the first letter and count the letter count of the word. Go to next word if different starting letter grab that and count length. If we come across a word starting with a letter we already have add the length to the sum of lengths for that starting letter and increase the divisor by +1.

Continues on next page……

3.2) ( 6 points) **Input:** a collection of text files.

**Required output:** how many words are there of length 1, 2, 3 and so on. For example, if the input is: ‘Hello there how are you doing today !’, the output should be:

Length number of words

1 1

3 3

5 4

|  |  |  |
| --- | --- | --- |
|  | Key | Value |
| Mapper input | Record\_id | Collection of text files |
| Mapper Output | Word | Length and count |
| Reducer input | word | (1,1)(3,3)(5,4) |
| Reducer Output | Word | Length and number of words matching the lengths |

3.3) (6 Points) **Input:** a text file containing flightdata where each line in file has the following format:

flight\_id, carrier\_id, departure\_city, destination\_city, delay

**Required output**: for each city, find the number of flights that either departed from or landed in the city.

For example, if the input is as follows:

1,AA,New York, Los Angeles, 10

2,DL,Los Angeles, New York10

3,SW,Minneapolis, Atlanta, 10

4,AA,Minneapolis, New York, 10

5,DL,New York, Atlanta, 10

6,AA,New York, Minneapolis, 10

The output will be:

Atlanta 2

New York 5

Minneapolis 3

Los Angeles 2

|  |  |  |
| --- | --- | --- |
|  | Key | Value |
| Mapper Input | Record\_id | Flight id, carrier id, departure city, destination city, delay |
| Mapper Filter | None |  |
| Mapper Output | City | Sum of cities |
| Reducer Input | City | (2), (5), (3), (2) |
| Reducer Output | City | Count |