Ref. No. : Ex/IT/PE/H/T/325B/2023

B.E. INFORMATION TECHNOLOGY THIRD YEAR SECOND SEMESTEREXAM 2023

BIG DATA(HONS.)

Time: 3 hours Full Marks: 100

ANSWER ALL QUESTIONS

1. Answer the following questions [CO1]
    a. GFS makes the design decision to use fixed sized chunks of 64 MB? What factors argue for large chunk sizes? What factors argue for small chunk sizes? How does 64 MB interact with Map-Reduce applications? Does 64 MB seem reasonable?
    b. GFS specializes its consistency model to its application domain. What does it mean for replicas to be *consistent*? What does it mean to be *defined*? What is the difference between a *write*? And a *record append*? How do the different states occur? Why or why not are all of these states acceptable?
    c. Is it ever possible for a client to read an inconsistent (i.e., stale) replica? Do you think this is acceptable?
    d. What happens when a client wants to do a write? Why is it helpful to have a primary? Is the use of leases appropriate here? How does the replica-update protocol achieve decent performance while ensuring that replicas are kept consistent?
    e. Why might the write protocol lead to inconsistent regions? Why undefined regions?
    f. How is the protocol for record appends different than ordinary writes? (Why must the primary sometimes pad the previous chunk?) Why might this protocol lead to some inconsistent entries? How do applications deal with this model?
    g. Imagine a Google file system instance configured to use the default of 3 replicas for each block. After a server fails, there will only be two replicas of many blocks. Joe claims that the Google file system can restore full redundancy of all blocks faster than would be the case with traditional 3-way replication (in which sets of three servers contain identical sets of blocks). Is he right? Explain your answer.
    h. The GFS master keeps chunk location information only in main memory, even though it uses writeahead logging and checkpoints on persistent storage for other file system metadata (like the namespace and inodes). So, the master does not retain the chunk location information across reboots. After a reboot, how can the GFS master figure out where a given chunk is stored?

7x4

2. Answer the following questions [CO2]
    a. How are input splits and output part files related with number of map and reduce jobs?
    b. What is the output of a reduce task?
    c. What happens when the Namenode is down?
    d. Name the services running in slave part of hadoop architecture

4x2

    e. Answer the following questions.
    Provided the list of purchases made, provide a mapreduce solution to find the most correlated pair of items (items bought together). Provide pseudo code for the map and reduce functions along with necessary explanations.

    Example input
    C1, {a, b}
    C2, {a, d, e}
    C1, {b, c}
    C3, {a, b, c}

    Expected output
    (a,b)
    (b,c)

10

    f. Consider that a number of files contains words, one word in each line. You are also given another word.
    Provide a map reduce solution to print the line numbers and the file names where the given word is present in these files (if any). Provide pseudo code for the map and reduce functions along with necessary explanations.

6

3. Answer the following questions [CO3]
    a. State a possible scenario where column family database is not suitable. Mention the suitable category of database that is applicable in such a scenario.
    b. What is the JOIN equivalent operator in MongoDB? Describe its parameters.
    c. With reference to Cassandra's tuneable consistency feature, describe strict and casual consistencies.
    d. Mention the advantage of peer to peer architecture over master slave architecture.

4x3

e. Consider the following database.

Assume, in a MongoDB database named MyDatabase, there exists a collection named Collection1, Collection2 and Collection3 containing the following Student details:

Collection1

… {Name:”Andrew Mark”, Department:”I.T.”, Roll:”1”, Marks:”79”},
... {Name:”John Mathews”, Department:”I.T.”, Roll:”34”, Marks:”98”},
... {Name:”Lisa Brown”, Department:”I.T.”, Roll:”3”, Marks:”28”},
... {Name:”Elma Woods”, Department:”I.T.”, Roll:”10”, Marks:”100”},
... {Name:” John Brown”, Department:”I.T.”, Roll:”19”, Marks:”59”}

Collection2:

… {Name:”Andrew Mark”, Department:”I.T.”, Roll: 1, Marks: 79},
... {Name:”John Mathews”, Department:”I.T.”, Roll: 34, Marks: 98},
... {Name:”Lisa Brown”, Department:”I.T.”, Roll:3, Marks: 28},
... {Name:”Elma Woods”, Department:”I.T.”, Roll: 10, Marks: 100},
... {Name:” John Brown”, Department:”I.T.”, Roll: 19, Marks: 59}

Collection3

… {Roll:34, Status: [“Regular”,”Attentive”]},
... {Roll:1, Status:[“Average”, “Improving”]},
... {Roll:10, Status:[“Excellent”, “Honest”]},
... {Roll:3, Status:[“Expelled”]},
... {Roll:19, Status:[“Below Average”]}

where, each row is a document in the collection.

(i) Write a query to find the details of all documents where Marks is between 70 and 99 from Collection2. Explain, with justification, if the same query would return the same result for Collection1.

(ii) The 5th document of Collection2 is named as Rec5. Add a field named “address” in Rec5 that stores an embedded document containing the following: { “addressLine1” : “XY Street”, “addressLine2” : “California” } Write a query to update Collection2 with the modified Rec5.

(iii) Write a query to enlist the details of those documents in Collection3, whose Status doesn’t contain any of the following tags: “Regular”,”Attentive”,”Excellent”.

(iv) From Collection2, find the details of the document with the 3rd highest Marks.

(v) Create an aggregation pipeline in MongoDB, that displays the autogeneratedObjectID, Name and Marks of the students from Collection2, in descending order of Marks.

5x3

4. [CO4] Answer either (a) or (b) in the following question:
   a. Answer the following questions.
      (i) Explain with an example the HIVE data model (that is, the way data are store in HIVE).
      (ii) Why are Serde's needed?
      (iii) Explain the components and sub-components of HIVE.

8+3+10

   b. Answer the following questions.
      (i) Consider the following tables.

page_view

pageid	**userid**	time
1	**111**	9:08:01
2	**111**	9:08:13
1	**222**	9:08:14

user

userid	age	gender
111	25	female
222	32	male

What will be the output id the following quesry is executed?
INSERT INTO TABLE pv_users
SELECT pv.pageid, u.age
FROM page_viewpv JOIN user u ON (pv.userid = u.userid);
Show the join in map-reduce framework

3+8

(ii) Consider the following table.

pv_users

pageid	age
1	25
2	25
1	32
2	25

What will be the output if the following query is executed?

```
INSERT INTO TABLE pageid_age_sum
SELECT pageid, age, count(1)
FROM pv_users
GROUP BY pageid, age;
```

Show the join in map-reduce framework

3+7