

Birla Institute of Technology & Science, Pilani
Work Integrated Learning Programmes Division
Second Semester 2024-2025

Mid-Semester Test
(EC-2 Regular)

Course No. : DSECLZG522
Course Title : Big Data Systems
Nature of Exam : Closed Book
Weightage : 30%
Duration : 2 Hours +20 Mins
Date of Exam : 28-06-2025 (EN)

No. of Pages	= 3
No. of Questions	= 8

Note to Students:

1. Please follow all the *Instructions to Candidates* given on the cover page of the answer book.
2. All parts of a question should be answered consecutively. Each answer should start from a fresh page.
3. Assumptions made if any, should be stated clearly at the beginning of your answer.

Q.1. In each of the following scenarios, point out and give a brief reason for the type of multi-processor computer one would use as per Flynn's taxonomy. The choices are SIMD, SISD, MIMD, MISD.

- (a) A scientific computing application does $\sin(a) + \tan(a)$ transformation for every variable a .
- (b) An image processing application where multiple images are merged at corresponding pixel level with a certain operator F to create the result image. E.g. $\text{image3} = F(\text{image1}(i, j), \text{image2}(i, j), \dots)$ for all pixels (i, j) in the input images.
- (c) A Hadoop system for big data analysis.

[Marks 3]

Q.2. A 3-tier application uses a 3 Node web front-end cluster, a 2 Node application server cluster and a 2 Node DB cluster. The application works only when all tiers are available. The web and application tiers are individually, (within the tier nodes) configured in an active-active or load-balanced HA mode. But the database tier uses an active-passive cold standby configuration where it takes 6 hours to switch from an active to a passive node. If nodes of web front-end, application server and DB tiers have failure rates of once in every 5 days, 2 days, and 10 days respectively, find the following:

- (a) MTTF of the web and the application server tiers respectively.
- (b) MTTF of the database tier
- (c) Availability of the database tier
- (d) Overall availability of the application, assuming MTTR of the web and application server tiers are negligible.

[Marks 4]

Q.3. In the following application scenarios, point out what is most important - Consistency or Availability, when a system failure results in a network partition in the backend distributed DB. Explain briefly the reason behind your answer.

- (a) A limited quantity discount offer on a product for 100 items at an online retail store is almost 98% claimed.
- (b) An online survey application records inputs from millions of users across the globe.
- (c) A travel reservation website selling rooms at a destination that is seeing very few bookings.
- (d) A multi-player game with virtual avatars and users from across the world needs a set of sequential steps between team members to progress across game milestones.

[Marks 4]

Q.4. What are the trade-offs between consistency, availability, and partition tolerance in a distributed NoSQL database system for data acquisition and online analysis. How might these trade-offs impact the design of a big data analytics lifecycle?

[Marks 4]

Q.5. Assume that you have a NoSQL database with 3 nodes and a configurable replication factor (RF). R is the number of replicas that participate to return a Read request. W is the number of replicas that need to be updated to acknowledge a Write request. In each of the cases below explain why data is consistent or in-consistent for read requests.

(1). RF=1, R=1, W=1.

(2). RF=3, R=2, W=Majority/Quorum.

(3). RF=3, R=Majority/Quorum, W=3.

[Marks 3]

Q.6. Write a MongoDB aggregation pipeline to calculate the average order value for each customer in a collection named "orders". The collection has the following fields: `_id`, `cust_id`, `order_date`, `total_price`. The pipeline should group the orders by customer ID, calculate the average total price, and sort the results in descending order by average order value.

Input:

```
[
  { "_id" : ObjectId("..."), "cust_id" : 1, "order_date" : ISODate("2022-01-01T00:00:00.000Z"),
    "total_price" : 100 },
  { "_id" : ObjectId("..."), "cust_id" : 1, "order_date" : ISODate("2022-01-15T00:00:00.000Z"),
    "total_price" : 200 },
  { "_id" : ObjectId("..."), "cust_id" : 2, "order_date" : ISODate("2022-02-01T00:00:00.000Z"),
    "total_price" : 50 },
  { "_id" : ObjectId("..."), "cust_id" : 2, "order_date" : ISODate("2022-03-01T00:00:00.000Z"),
    "total_price" : 75 }
]
```

Output:

```
[
  { "_id" : 1, "average_order_value" : 150 },
  { "_id" : 2, "average_order_value" : 62.5 }
]
```

Provide the aggregation pipeline in the format of the MongoDB aggregation framework.

[Marks 4]

Q.7. You are running a map-reduce program on a Hadoop cluster. If you run the program on a single node, it takes total 345 sec and 5% of the overall time is spent in a sequential reduce operation. The rest of the map reduce application code and runtime can be parallelized.

(a) If you had a 10 Node cluster with similar nodes for the same program and data set, how much time would you theoretically expect the program to take?

(b) What are the factors contributing to the higher run time observed on the 10 Node cluster than the expected runtime calculated theoretically?

(c) Suppose you ran this program, with necessary modifications, but with a larger data set trying to accomplish more work with the 10 Node cluster. What is a theoretical speed up you could target?

[Marks 4]

- Q.8. An online store is analyzing sales data to announce discounts on purchases made using the most popular Credit card. The data for the year 2023 is having a volume of 1 TB and schema of the data is given below:

Transaction_date	Product_ID	Price	Card_Type	Name	City	State	Country	Account_created
------------------	------------	-------	-----------	------	------	-------	---------	-----------------

You need to find out (1) The type of the Card used for maximum total payments received in the year and (2) The highest amount paid through each type of the cards. Write pseudo code for a MapReduce program to extract these two parameters from the sales data.

[Marks 4]
