Big Data Ques/Ans

---

10 marks

➔ Fragmentation ques. 3 cases….some cities

1. Veritcal Fragmentation
2. Horizontal Fragmentation
3. Hybrid Fragmentation

City problem:

```
Consider the distributed database for a company in California having three sites at
San Francisco (site 1), Fresno (site 2), and Los Angeles (site 3); Fresno is located
about halfway between San Francisco and Los Angeles. There are 30 departments,
physically grouped as follows:

• the first 10 are close to San Francisco,

• departments between 11 and 20 are close to Fresno, and

• departments over 20 are close to Los Angeles.

Suppliers of the company are all either in the city of San Francisco or in the city of
Los Angeles; there is also the notion of areas into which the company is divided; the
area "North" includes San Francisco, the area "South" includes Los Angeles, and Fresno
falls exactly on the border, with some departments close to Fresno in the northern
area and some in the southern.

The global schema is that of EXAMPLE_DDB including relations EMP, DEPT, SUPPLIER, and
SUPPLY.
```

➔ Algorithm for best fit, all beneficial

To measure cost and benefits of allocation we use 3 algorithms, namely

1. Best Fit
2. All site Beneficial
3. Additional Replication

For non redundant allocation best fit algorithms is used. And for redundant allocation both additional replication along with all beneficial site is used.

- **i is the fragment index**
- **J is the site index**
- **K is the application index**
- **f_{kj} is the number of frequency of application k at site j**
- **R_{ki} is the number of retrieval references of application k to fragment i**
- **U_{ki} is the number of update references of application k to fragment i**
- **$n_{ki} = R_{ki} - u_{ki}$**

BEST-FIT

Using the best-fit approach for a non-replicated allocation, we replace Ri at the site where the number of references to Ri is maximum. The number of local references Ri at site j is:

$$B_{ij} = \sum_k (f_{kj} n_{ki})$$

R_i is allocated at site J* such that B_{ij} is maximum.

ALL BENEFICIAL SITE

Using the "all beneficial sites" method for replicated allocation, we replace Ri at all sites j where the cost of the retrieval references of applications is larger than the cost of the update references to Ri from applications at any other site. B_{ij} is evaluated as the differences:

$$B_{ij} = \sum_k (f_{kj} n_{ki}) - C \sum_k \sum_{j' \neq j} f_{kj'} u_{ki}$$

ADDITIONAL REPLICATION

Using the "additional replication" method for replicated allocation, we can measure the benefit of placing a new copy of Ri in terms of increased reliability and availability of the system. As already stated, this benefit does not grow proportionally to the degree of redundancy of Ri.

Let di denote the degree of redundancy of Ri, and let Fi denote the benefit of having Ri fully replicated at each site. The following function β(di) is introduced to measure this benefit:

$$\beta(d_i) = \left(1 - 2^{1-d_i}\right) F_i$$

Note that β(1) = 0, β(2) = Fi /2, β(3) = 3Fi/4, and so on. Then, we evaluate the benefit of introducing a new copy of Ri at site j by modifying the previous formula:

$$B_{ij} = \sum_k (f_{kj} r_{ki}) - C \sum_k \sum_{j' \neq j} f_{kj'} u_{ki} + \beta(d_i)$$

Slide 43

ALL BENEFICIAL SITES

The set of all sites where the benefit of allocation one copy of the fragment is higher than the cost, and allocate a copy of the fragment to each element of this site.

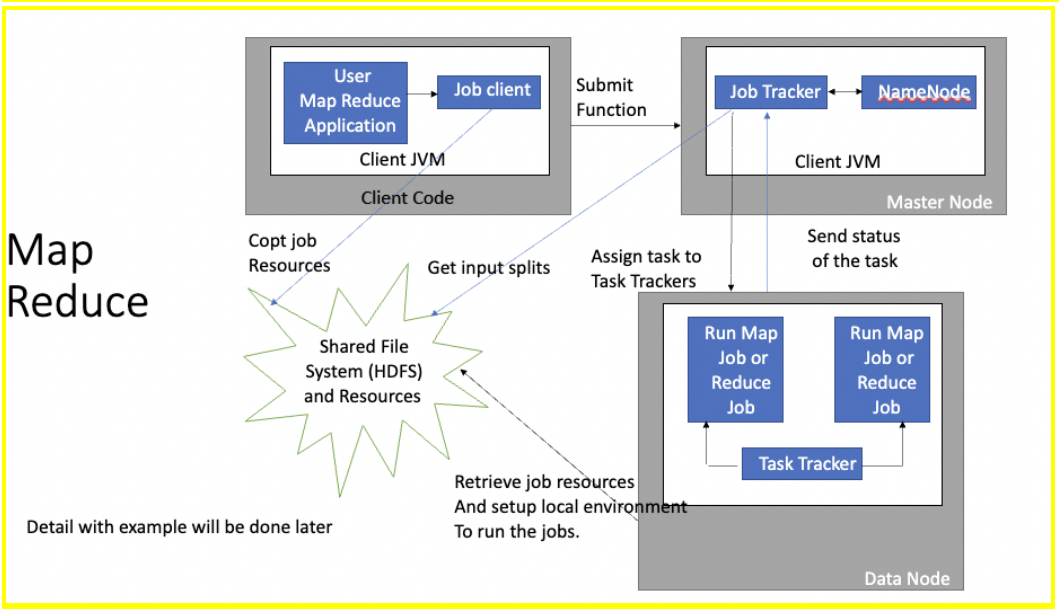
CSS Framework

ADDITIONAL REPLICATION

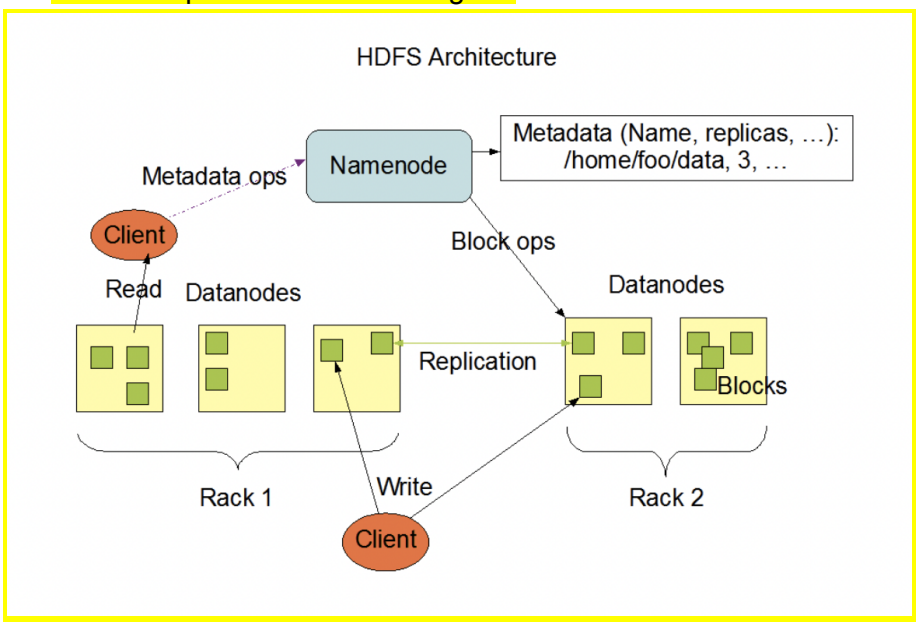
First the solution of the non-replicated problem, and then progressively introduce replicated copies starting from the most beneficial; the process is terminated when no additional replication is beneficial.

➔ Map-reduce example with steps and diagram

The basic principle of operation behind MapReduce is that the "Map" job sends a query for processing to various nodes in a Hadoop cluster and the "Reduce" job collects all the results to output into a single value. Map Task in the Hadoop ecosystem takes input data and splits into independent chunks and output of this task will be the input for Reduce Task. In The same Hadoop ecosystem Reduce task combines Mapped data tuples into smaller set of tuples. Meanwhile, both input and output of tasks are stored in a file system. MapReduce takes care of scheduling jobs, monitoring jobs and re-executes the failed task.

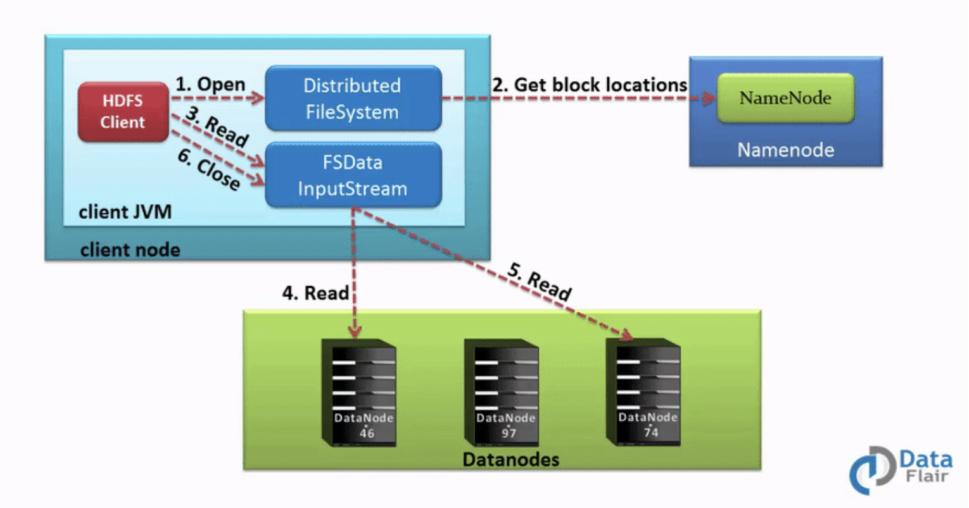


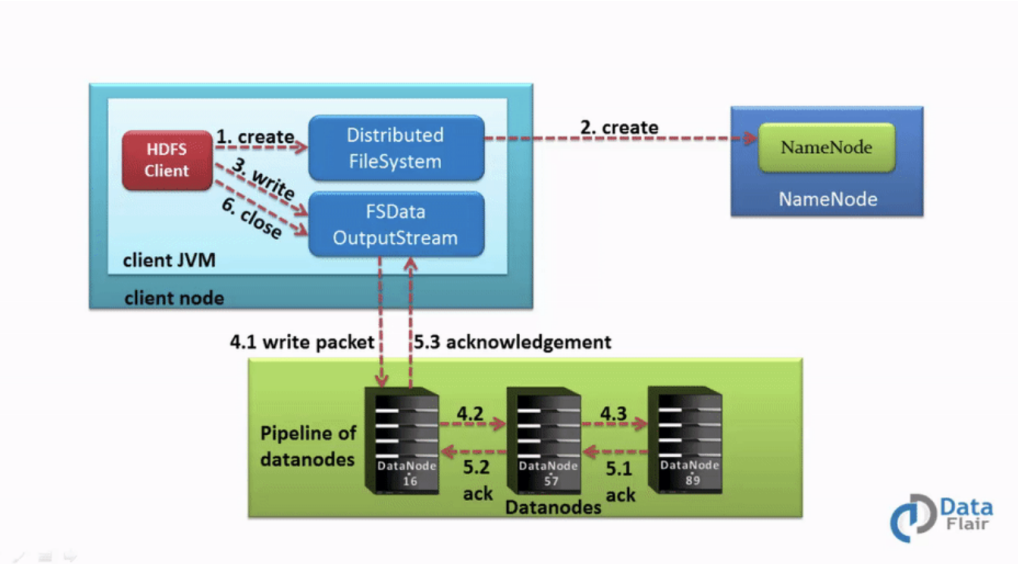
➔ Hadoop architecture—>diagram



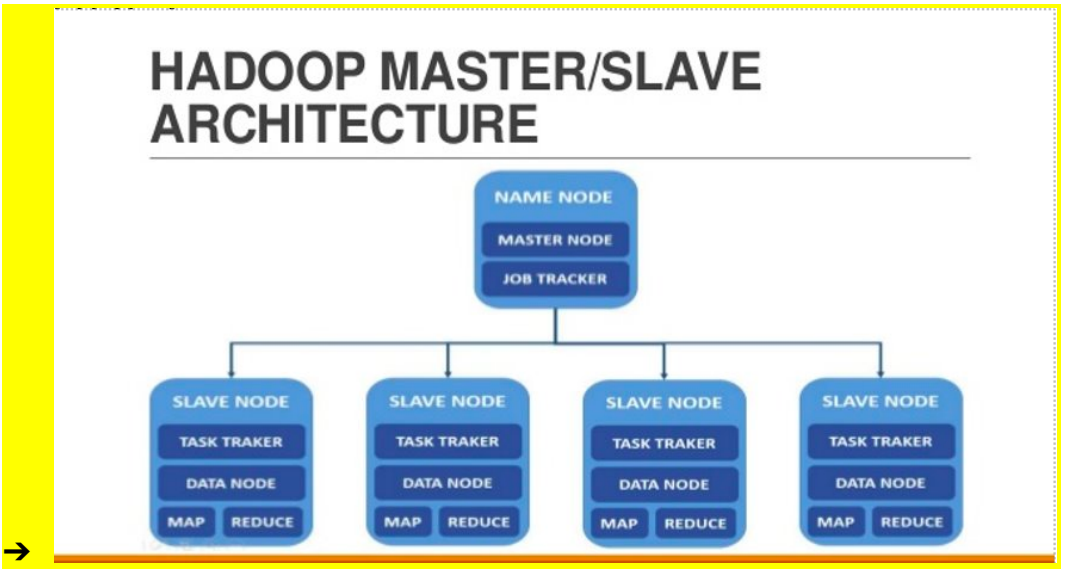
➔ Additional replication is used to improve reliability and availability. Suggest: Additional replication does not provide the degree of benefit. Discuss how more replication has less degree of benefit?

➔ Read/write operations in correct steps in Hadoop.

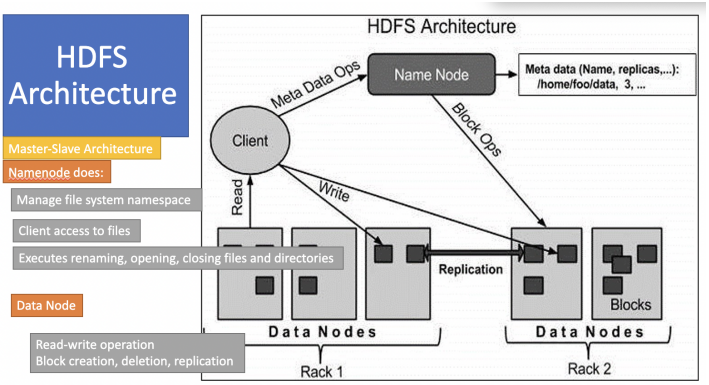




➔ Hadoop master-slave architecture of Hadoop and Hadoop cluster (with diagram)



➔



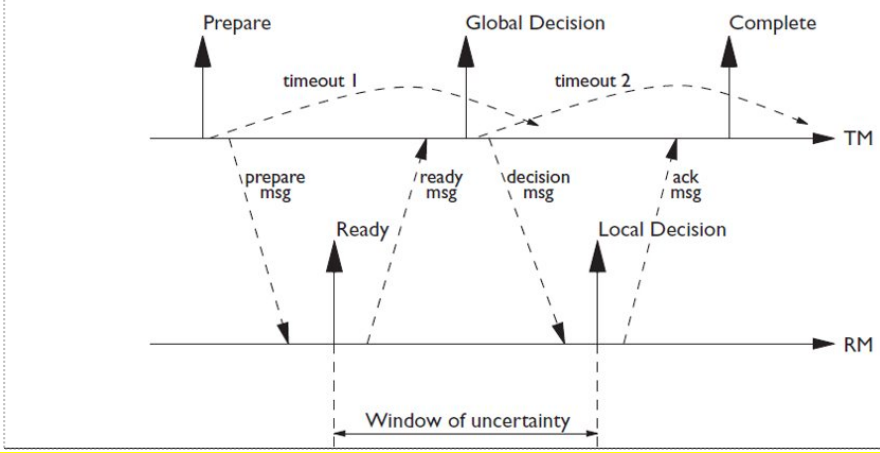
- A Hadoop cluster is a special type of computational cluster designed specifically for storing and analyzing huge amounts of unstructured data in a distributed computing environment.

➔ Comment critically——> for and against

5 marks

➔ Update operation in DDBMS

Distributed Transaction – 2PC



➔

➔ In and out operation in Hadoop

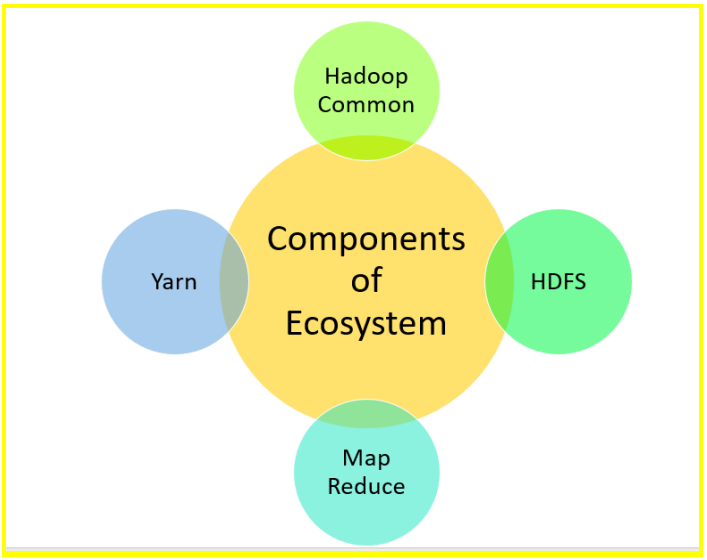
➔ Correct directory path for formatting /checking the hadoop env

/usr/local/cellar/hadoop/3.3.1/libexec

bin	hdfs-config.sh	sbin
etc	libexec	share
hadoop-config.sh	logs	yarn-config.sh

➔ Describe in brief the complexity managing the data quality —> go with the closest match/average

➔ Ecosystem of Hadoop



Hadoop Common

- Apache Foundation has pre-defined set of utilities and libraries that can be used by other modules within the Hadoop ecosystem. For example, if HBase and Hive want to access HDFS they need to make of Java archives (JAR files) that are stored in Hadoop Common.

➔ Diff types of transparency

Data Transparency: The query writer is unaware of the source of data

Distribution transparency is the property of distributed databases by the virtue of which the internal details of the distribution are hidden from the users. The DDBMS designer may choose to fragment tables, replicate the fragments and store them at different sites. However, since users are oblivious of these details, they find the distributed database easy to use like any centralized database.

The three dimensions of distribution transparency are −

Location transparency:

• Location transparency ensures that the user can query on any table(s) or fragment(s) of a table as if they were stored locally in the user's site.

• The fact that the table or its fragments are stored at remote site in the distributed database system, should be completely oblivious to the end user.

• The address of the remote site(s) and the access mechanisms are completely hidden.

Fragmentation transparency:

• Fragmentation transparency enables users to query upon any table as if it were unfragmented.

• Thus, it hides the fact that the table the user is querying on is actually a fragment or union of some fragments.

• It also conceals the fact that the fragments are located at diverse sites.

Replication transparency:

Replication transparency ensures that replication of databases are hidden from the users. It enables users to query upon a table as if only a single copy of the table exists.

Replication transparency is associated with concurrency transparency and failure transparency. Whenever a user updates a data item, the update is reflected in all the copies of the table. However, this operation should not be known to the user. This is concurrency transparency. Also, in case of failure of a site, the user can still proceed with his queries using replicated copies without any knowledge of failure. This is failure transparency.

➔ Hadoop commands
➔ equation of —>MSE, RMSE
➔ strategies to convert the big data into insights

* Decision trees can be used for the following type of dataset: 1) attributes are categorical 2) continuous numerical value 3) discrete value

Why 2PC is blocking and 3pc is non-blocking protocol?

Failure in transaction management in the first phase of 2PC leads to an uncertain state, thus 2 PC is called blocking protocol.

In 3PC the addition of a phase protocol allows the reaction to a failure of the transaction manager by electing one participant to the substitute the transaction manager. The new coordinator can decide the results of the transaction by looking at the logs. This leads to the transaction being fault-tolerant.