

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TEST -2 EXAMINATION- 2023

B.Tech-V Semester (CSE/IT/ECM)

COURSE CODE (CREDITS): 20B1WCI532 (02)

MAX. MARKS: 25

COURSE NAME: Cloud Computing: Concepts, Technology & Architecture

COURSE INSTRUCTORS: ARV

MAX. TIME: 1 Hour 30 Minutes

Note:

(a) All questions are compulsory.

(b) Marks are indicated against each question in square brackets.

(c) The candidate is allowed to make Suitable numeric assumptions wherever required for solving problems

1. Explain the concept of virtualization and how does it differ from traditional computing? Discuss the benefits of server virtualization in data centers? **[4 Marks][CO-2]**

2. Write the difference between Type 1 and Type 2 hypervisors. When would you use each type? What is the purpose of virtual machine (VM) isolation in hypervisors?

[4 Marks][CO-2]

3. Discuss the Hyper-V Architecture with suitable diagram. Write down the advantages and disadvantages of Hyper-V architecture. **[5 Marks][CO-2]**

4. Explain the concept of access control in cloud computing. Discuss the various access control models used in cloud computing? **[5 Marks][CO-3]**

5. Consider the following use case:

Facebook has a list of friends (note that friends are a bi-directional thing on Facebook. If A is friend of B then B will be friend of A). We've decided to pre-compute calculations when we can to reduce the processing time of requests. One common processing request can be "A and B have 20 friends in common" feature. When you visit someone's profile, you see a list of friends that you have in common. This list doesn't change frequently so it'd be wasteful to recalculate it every time you visited the profile.

Consider, given a social network like Facebook which is having tens of millions of users, using MapReduce find common friends of Facebook data to identify "common friends" among all pairs of users . **[7 Marks][CO-3]**

Hint:

Assume the friends are stored as Person->[List of Friends], our friends list is then:

A -> B C D

B -> A C D E

C -> A B D E

D -> A B C E

E -> B C D

Each line will be an argument to a mapper. For every friend in the list of friends, generate the output using mapper as a key-value pair (The key will be a friend along with the person. The value will be the list of friends). Sort the pairs according the keys so that the friends are in order, causing all pairs of friends to go to the same reducer. After all the mappers are done running, generate a list like this:

For map(A -> B C D) :

(A B) -> B C D

(A C) -> B C D

(A D) -> B C D

Send the outputs of each mapper to the reducers and group them according to their keys.

The reduce function will simply intersect the lists of values and output the same key with the result of the intersection.

For example, reduce((A B) -> (A C D E) (B C D)) will output (A B) : (C D) and means that friends A and B have C and D as common friends.