



Name of the Subject: DISTRIBUTED COMPUTER Subject Code: IT-717

Seat No: IT076 Student ID: 18ITUBN116 Branch/Sem: IT-VII

Q2 (a)

### Microservices

- Service Startup is relatively quick.
- All microservices should be loosely coupled so that changes made in one does not affect the other.
- Businesses can deploy more resources to services that are generating higher ROI.
- Always remains consistent & continuously available.
- Change in the data model of one microservice does not affect other microservices.
- The Principle that focuses on product, not projects.
- Small focus Teams, Parallel & faster development.
- Interacts with other microservices by using well defined interface.

### Monolithic

- Service Startup takes more time.
- Monolithic arch. is tightly coupled. Changes in one module of code affect the other.
- Since services are not isolated, individual resource allocation not possible.
- Development tools get overburdened as the process to start from the scratch.
- Change in data model affects the entire database.
- Put emphasis on the entire project.
- Large Team & considerable team management effort required.
- Not applicable.



Name of the Subject: DC Subject Code: IT-717

Seat No: IT076 Student ID: 18ITUBW116 Branch/Sem: IT-VII

Q2

(b)

(i)

- As, we are also doing many Input/output operations just like Map-Reduce.
- But, the only Input/output operation happens at first stage (the file gets executed).
- After that our data is always sitting in memory.
- So, it gives faster output.
- Apache Spark stores data in random access memory, while Hadoop MapReduce persists data block to disk after a map.

(ii) Advantages of Microservices.

(i) Small in Size:- It is an implementation of SOA design pattern. It will be obviously small in size & easy to maintain.

(ii) Autonomous:- Each microservice should be an autonomous business unit of the entire application. Hence, the application becomes more loosely coupled.

(iii) Technology heterogeneity:- Different technologies to communicate with each other in one business unit. By implementation.





Name of the Subject: DC Subject Code: IT-717  
Seat No: IT076 Student ID: 18ITUBN116 Branch/Sem: IT-VII

(12) Resilience:- Resilience is a property of isolation a system unit.

Q3 (1)

- HDFS is highly fault-tolerant. Before Hadoop 3 it handles faults by the process of replica creation.
- HDFS also maintain the replication factor by creating a replica of data on other available machines in the cluster if suddenly one machine fails.
- Hadoop 3 introduces Erasure coding to provide Fault Tolerance.

(2)

- DataNodes are the commodity hardware only as it can store data like laptops & personal computers, these are required in large number.
- Instead, NameNode is the master node, it stores metadata about all the blocks stored in HDFS.
- It needs high memory space, thus works as a high-end machine.



Name of the Subject: \_\_\_\_\_

DC

Subject Code: IT-717

Seat No: IT076

Student ID: 18ITUBN116

Branch/Sem: IT-VII

(3)

- As soon as the data nodes is declared dead all the data blocks it hosts are transferred to the other data nodes, with which the blocks are replicated.
- This is how NameNode handles datanode failures.
- HDFS ~~just~~ works on Master/Slave mode where NameNode act as Master & Datanode act as Slave.

(4)

- A Record Reader converts the byte-oriented view of the input to a record-oriented view for the Mapper & Reducer tasks for processing.
- To understand Hadoop Record Reader, we need to understand MapReduce Dataflow.
- Let The Data flow: MapReduce is a Simple Model of data processing.