DC Theory Assignment No 1

Name- Rohit Saini
Roll no- PC41
erp- 1032200897
Panel- C

Q1. Case Study on grid Computing.

Ans- Introduction :- Distributed computing has emerged as a pivotal technology in overcoming the challenges possed by large-scale data processing. In this case study, we delve into the application of distributed computing, particularly grid computing, and its transformative impact on a data-intensive project.

Objective: The primary objective is to illustrate how distributed computing, specifically grid computing, optimize data processing by distributed tasks across a network of interconnected computers.

Scenario: Imagine a financial institution handling vast volumes of transactional data daily. The traditional centralized computing infrastructure struggles to cope with the increasing data load, resulting in processing delays and inefficiencies. The institution seeks a solution to enhance its data processing capabilities and reduce turnaround times.

Implementation of Distributed Computing: The institution adopts a distributed computing model, leveraging grid computing to distribute data processing tasks across multiple nodes in the network. Each node collaboratively processes a portion of the data, enabling parallel execution and significantly improving overall processing speed.

Advantages:

1. Parallel Processing: Distributed computing allows the financial institution to process multiple transactions concurrently, overcoming the limitations of sequential processing. This process approach boosts efficiency and reduces the time required for data processing.

2. Scalability: The institution can easily scale its computing resources by adding or removing nodes from the distributed network based on demand. This scalability ensures that the system can adapt to fluctuations in data volume, maintaining optimal performance.

3. Fault Tolerance: Distributed computing enhances system resilience. In the event of a hardware failure or system crash on one node, the workload is automatically redirected to functionsing nodes, ensuring continuous data processing without significant disruptions.

4. Cost Efficiency: By utilizing existing computing resources within the network, the institution minimizes the need for expensive infrastructure investments. Distributed computing optimizes resources utilization, resulting in a cost-effective solution for handling large-scale data processing.

Conclusions:

The adoption of distributed computing, specifically grid computing, proves instrumental in revolutionizing the financial institution's data processing capablities. The parallel processing, scalability, fault tolerance, and cost efficiency offered by this approach collectively contribute to a more agile and responsive data processing infrastructure.

Q2. Case study on Byzantine Problem and its solution.

Ans. Title: Tackling the Byzantine Generals Problems in Distributed Systems.

Introduction: The Byzantine Generals Problem is a classic challenge in distributed computing where a group of generals commanding differents divisions of a Byzantine army, must coordinate their attack or retreat. The problem arise when some generals may be traitors, sending conflicting or deceptive messages. This case study explores a scenario in which a solution to the Byzantine Generals Problem is implemented to ensure reliable communication in a distributed system.

Scenario: Consider a network of interconnected servers responsible for managing a critical financial transaction system. Each server must agree on a common decision regarding the approval or rejection of transactions. However, due to potential server failures or malicious attacks, some servers may behave dishonestly, providing confilicting information to compromise the integrity of the system.

The Byzantine Generals Problems:

In this context, the Byzantine Generals Problems manifiests as the challenge of achieving consensus among the servers despite the presence of fault or malicious nodes. Traditional consensus algorithms, such as majority voting, fails when faced with Byzantine faults as they assume a trustworthy majority.

Solution: Practical Byzantine Fault Tolerance (PBFT)

The implementation of the Practical Byzantine Fault Tolerance (PBFT) algorithm serves as the solution to address the challenges posed by the Byzantine Generals Problem.

1. Replication and Redundancy: The critical components of the financial transaction system are replicated across multiple servers to ensure redundancy.
- Each server maintains an identical copy of the transaction log and state.

2. Voting-Based Consensus: PBFT relies on voting-based consensus mechanism where each server communicates with others to reach an agreement on the validity of transactions.
- Nodes exchange message to propose and vote on transactions with a two-thirds majority required for a decision to be accepted.

3. Cryptographic Signatures: Message exchanged between nodes are digitally signed, ensuring the authenticity and integrity of the communications.
- Cryptographic techniques are employed to verify the origin of messages, preventing malicious nodes from forging messages.