QP - DC - 03 evaluation.

VIT | Vidyalankar Institute of Technology
(Accredited A+ by NAAC)
(Autonomous Institute Affiliated to University of Mumbai)

Mid Semester Examination

Branch	Date	Sem.	Roll No. / Exam Seat No.	Subject	Student's Signature	Junior Supervisor's Name and Sign

Question No.	A	B	C	D	E	F	G	H	Total	Total out of (20 / 30 / 40)
1										
2										
3										
4										

Examiners Signature	Student's Sign (After receiving the assessed answer sheet)

1 A- Discuss the desirable features of Global scheduling algorithm.

→ Global scheduling algorithm in distributed computing environment efficiently allocate resources and tasks execution across multiple processing units. The desirable features of global scheduling algorithm play a critical role in optimizing resource utilization, system performance, and ensuring fairness and responsiveness. Below are the key features of global scheduling algorithm.

1. fairness
2. Efficiency

④ ~~Productivity~~ Adaptability
⑤ Load balancing

⇒ After scheduling algorithm prevents resource starvation and ensures that no task ~~gets preferential treatment~~ monopolizes system resources to that determine of other. Fairness ensures that all task or processor receives a responsible share of and treated equally in the scheduling process.

⇒ Efficiency :- Efficiency refers to the ability of the scheduling algorithm to maximum maximum utilization & minimum response time.

⇒ Scability :- scability is critical in distributed system with ~~heterogeneity~~ varying no. of processing unit and changing workload demands.

⇒ Adaptability :- Allows the scheduling algorithm to dynamically adjust its behaviour base on workload characteristics, system conditions and user defined policies.

Load balancing :- load balancing ensure that computation load is evenly distributed across processing units to avoid resource bottlenecks and minimize response time variability.

to maintain coherent and consistency across multiple memory modules. These challenges impact system performance, scability, fault tolerance and overall reliability.

Below are some key design challenges in DSM systems.

1. coherent & consistency.
2. Scability
3. communication overhead.
4. fault tolerance
5. performance optimization

The design challenges in DSM systems revolves around the maintaining coherence, consistency, ensuring scability, managing communication overhead, providing fault tolerance and optimizating performance. Addressing these challenges requires careful consideration of system architecture algorithm design and resource management techniques. to build reliable, scalable and efficient distributed memory system

1c) Code migration explain :—

→ Code migration, also known as code relocation, refers to the process transferring executable code or software components from one computing environment to another. This concept enables dynamic flexible development of software. across distributed systems and flexible deployment of software across distributed systems

flexible deployment of software across distributed systems and facilitates efficient resource utilization load balancing, fault tolerance, and adaptive system behavior. Below are key aspects of code migration.

1) Dynamic Code Transfer - Code migration involves dynamically transferring executable code or software components from one location to another at runtime.

2) Purpose and Benefits - code migration serves various purpose, including load balancing, resource optimization, fault tolerance, and adaptive system behavior.

3) Types of Code migration - Can be classified into several types based on the direction of migration and the granularity of code transfer

4) Mechanisms and Techniques - Various mechanisms and techniques are employed to implement code migration effectively.

5) Applications and Use cases - code migration finds applications in diverse domains, including cloud computing, edge computing, mobile computing and distributed systems.

Code migration enables dynamic and flexible deployment of software across distributed system, providing benefits such as load balancing, resaurce optimization, fault tolerance and adaptive system behavior. By facilitating the movement of executable code or software mponents across networked nodes or devices, de migration enhances system flexibility, alability and efficiency.

2 a) Discuss the benefits of load sharing in improving system performance and resource utilization.

→ Load sharing, a fundamental concept in distributed computing involves redistributing computational tasks or data dynamically across multiple processing units to balance the workload and optimize resource utilization. This strategy offers several benefits that significantly enhance system performance and resource efficiency. Below are key advantages of load sharing:

1) Optimized Resource Utilization - Load sharing ensures the computational tasks are evenly distributed across available processing units, preventing resource underutilization and maximizing system throughput.

2) Improved system Responsiveness - Load sharing reduces response time and improves System responsiveness by preventing resource bottlenecks and avoiding overburden Processing units.

3) Enhanced scalability - Load sharing facilitates system scalibility by enabling the efficient utilization of additional processing units as the workload increases.

4) Fault Tolerance and Reliability - Load sharing enhances system fault tolerance and reliability by mitigating the impact of node failures or performance degradation.

5) Balanced Workload Distribution - Load sharing ensures balanced workload distribution across processing units, preventing resource contention and hotspots. Load sharing plays a crucial role in improving system performance and resource utilization by optimizing resource allocation, enhancing system responsiveness, and maintain

balancing workload distributor across distributed computing enviornments. By dynamically redistributing tasks or data, load sharing maximizes the efficiency and ~~so~~ ~~red~~ effectiveness of distributed systems, ultimately leading to better user experience and higher system throughput.

2 B) Discuss the load sharing approach in context with of process management practices

=> Load sharing is a crucial aspect of process managment in distributed computing enviornments where computational tasks need to be efficiently allocated and executed across multiple processing units. In the context of process managment load sharing refers to the dynamic redistribution of processes or tasks among available processing nodes to balance workload and optimize resource utilization. Below are the key aspects of the load sharing approach and its significance in process management.

1) Dynamic Task Redistribution - Load sharing involves dynamically redistributing computational tasks or processes across available processing nodes based on current system conditions and workload distribution.

2) Resource Utilization Optimization - Load sharing optimizes resource ~~et~~ utilization by effectively leveraging available processing resources to execute computational tasks.

3) Response Time Improvement - Load sharing enhances system responsiveness and reduces response time by preventing overloading of processing nodes and avoiding resource contention.

4) Scalability and Adaptability - Load sharing

enhances system scalability and adaptability by facilitating the efficient utilization of additional processing nodes as the workload changes.

5> Fault tolerance and Reliability - Load sharing improves system fault tolerance and reliability by mitigating the impact of node failures or performance degradation.

Load sharing in the context of process management plays a critical role in optimizing resource utilization, improving system responsiveness, enabling scalability, ensuring fault tolerance, and maintaining balanced workload distribution across distributed computing environments. By dynamically reallocating tasks among processing nodes, load sharing enhances the efficiency and effectiveness of process management, ultimately leading to better overall system performance and user satisfaction.

6> Identify and discuss the types of distributed deadlock.

→ Deadlock, a critical issue in distributed systems occurs when two or more processes are unable to proceed because each is waiting for a resource held by the other(s). In distributed systems, deadlocks can manifest in various forms due to the distributed nature of resources and processes. Below are the types of distributed deadlocks:

1> Resource Deadlock - In resource deadlock, processes in a distributed system contend for shared resources such as files, databases, or communication channels.

2> Communication Deadlock - occurs when processes in a distributed system are waiting for messages or acknowledgments from each other to

3) Transaction Deadlock - Transaction deadlock arise in distributed database systems when multiple transactions hold locks on data items and are waiting for locks held by other transactions
4) Control Deadlock - occurs in distributed systems when processes contend for control over shared resources, such as access to critical sections for code or synchronization primitives
5) Message Deadlock - arises in distributed systems where process may block while waiting for a response or acknowledgment from another process

13 A) Explain Raymonds Tree Algorithm in detail
→ Raymond's Tree Algorithm is a distributed algorithm used mutual exclusion in distributed systems. It allows processes to enter critical sections in a mutually exclusive manner without encountering deadlocks or starvation. The algorithm is based on a hierarchical tree structure, where processes are organized in a logical tree topology.

1) Tree Structure - In Raymond's Tree Algorithm, processes are organized in a logical tree structure where each process in a node in the tree
2) Requesting Access - To request access to a critical section, a process sends a request message to its parent process in the tree.
3) Propagation of Requests - upon receiving a request message from a child process, a parent process compares the timestamp of the received request with its own timestamp. receives
Granting Access - When a process requires equest message from all its child proless ar

has no pending requests of its own, it grants access to the requesting process.
By carefully managing the propagation of request through the tree, the algorithm prevents deadlocks and ensures that processes can enter critical sections in a mutually exclusive manner.

Q3B) Explain client-centric model of replica managment.

→

⇒ The client-centric model of replication management in a distributed system architecture approach that optimize the needs and requirments of client in accessing replica of data. In this model replicas are strategically placed and managed to minimize client latency, performance, availability, and reliability. The key characteristics of client-centric model are,

1. Replica procurement based on client proximity.
2. Dynamic replica Allocation
3. client aware data access
4. load balancing and fault Tolerance
5. client-centric caching and prefetching

The client-centric model of replication manages among preferences in accessing data. By strategically placing

closer to clients
replica based - dynamically allocating
optimized - data on client demand, and
enhance system performance, availability,
and reliability. Below is a detailed
explanation of client-centric model
along with its key characteristics:

(1) Replica Placement Based on Client Proximity

- In the client-centric model, replicas are strategically placed in locations that are geographically close to the clients they serve.

- This proximity minimizes network latency and improves data access speeds for clients, enhancing overall system performances and user experience.

(2) Dynamic Replica Allocation:

- The client-centric model employs dynamic replica allocation strategies to adapt to changing clients demands and workload patterns.

- Replica may be dynamically created, migrated or removed based on factor such as client access pattern, network conditions and resources availability.

③ Client-Aware Data Access:

- The client-centric model prioritizes client aware data access by directing client to nearby replicas for read and write operations.

- clients are providing with information about the availability and location of replicas, allowing them to select the most suitable based on factors such as proximity, load, and reliability.

④ Load Balancing and fault Tolerance:

- load Balancing mechanisms are employed to evenly distributed clients requests across replicas and prevents overloading of individual replicas.
- Fault tolerance is achived through the replica of data across multiple replicas, ensuri data availability and resilience to replica failures or network partitions.

⑤ client-centric Caching and Prefete

- The client-centric model utilizes client caching and prefetching techniques to fur improve data access performance and r latency.

Q Identify and discuss the type of distributed deadlock

→ Deadlock, a critical issue in distributed system, occurs when two or more processes are unable to proceed because each is waiting for a resource held by the other. In distributed system, deadlocks can manifest in various forms due to the distributed nature of resources and process. Below are its types of distributed deadlock

1) Resource Deadlock - In resource deadlock, processes in a distributed system contend for shared resources such as files, database or communication channels.

2) Communication Deadlock - occurs when processes in a distributed system are waiting for message or acknowledgments from each other to proceed

3) Transaction Deadlock - arises in distributed database systems when multiple transaction hold locks on data items and are waiting for locks held by other transaction

4) Control Deadlock - occurs in distributed systems when processes contend for control of shared resources, such as access to critical sections of code or synchronization primitives

5) Message Deadlock - arises in distributed systems when process exchange message asynchronously and a process may block while waiting for a response or acknowledgment from another process

Considering detection, prevention & resolution strategies by identifying can enhance the distributed system