Type: MCQ

Q1. Node which can be either a hardware device or a software process is called as (0.5)

1. **Computing Element
2. Simulation
3. Clock
4. None of the above

Q2. Single Coherent system in a distributed system should have (0.5)

1. **Single System View.
2. No data.
3. No Graphical User Interface.
4. Shared Memory.

Q3. Access transperency is (0.5)

1. Not a type of Distribution Transparency.
2. **Type of Distribution Transparency.
3. Non defined term in Distribution Transparency.
4. none of the above.

Q4. Names that are assigned to resources which do not have any relation with its physical location are called as (0.5)

1. Singular Names.
2. Multiple Names.
3. **Logical Names.
4. None of the above.

Q5. A system that offers components that can be easily used by , or integrated into other systems is calles as (0.5)

1. Non Open Distributed System
2. Closed Distributed System
3. **Open Distributed System.
4. None of the above.

Q6. A process where a component is split into smaller parts and distributed across the system is called as (0.5)

1. **Partitioning and Distribution .
2. Un complete .
3. Not Defined
4. None of the above

Q7. Distributed System that can be viewed as a huge collection of resources that are individually managed by components is called as (0.5)

1. Layered Architecture.
2. **Resource Based Architecture.
3. Object Based Architecture .
4. None of the above.

Q8. Centralized Components which handles all the accesses between different applications are (0.5)

1. Wrappers .
2. **Brokers.
3. Interceptors.
4. None of the above

Q9. One of the Mutual Exclusion based Algorithm in Distributed System is (0.5)

1. Stand Alone Algorithm.
2. **Token Based Algorithm.
3. Not an existing algorithm.
4. None of the above.

Q10. Berkley Algorithm uses (0.5)

1. **Physical Clock.
2. Logical Clock.
3. Vector Clock.
4. None of the above.

Type: DES

Q11. Giving the basic points of Object Based Architecture, show organization of a remote object with client side proxy , with a diagram. (2)

Q12. Consider a distributed implementation of an algorithm for accessing a shared resource. Apply the algorithm on the scenario, as shown in Fig. 1, in which two processes simultaneously try to access a single shared resource and describe the steps involved in resolving the possible conflict.

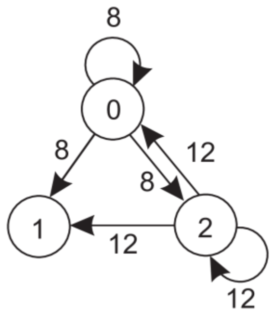


Fig. 1

Q13. What are the three dimensions of scalability in Distributed Systems? Analyse with examples. (3)

Q14. Answer the following.

I. With diagram analyse the Berkeley algorithm for averaging of physical clocks across machines.

II. Causality among processes in distributed systems is essential to know the relationship between them. Apply the vector clock algorithm on Fig. 2 and give the logical time at process P3 after receiving m2.

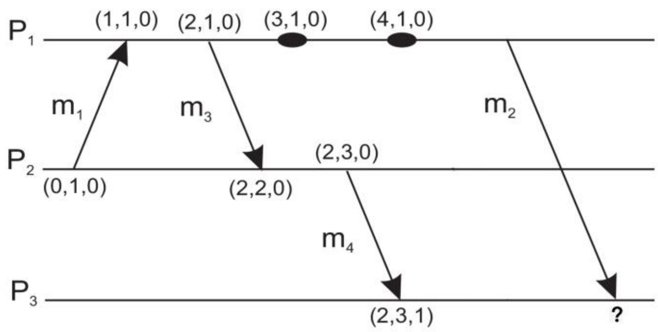


Fig. 2 (3)

Solutions for Q 12 and Q 14

Q. 12 ➔ 2M

Consider a distributed implementation of an algorithm for accessing a shared resource. Apply the algorithm on the scenario, as shown in Fig. 12, in which two processes, P0 and P2, simultaneously try to access a single shared resource and describe the steps involved in resolving the possible conflict.

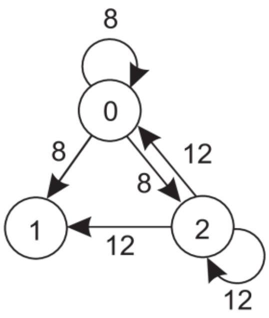


Fig. 12

Scheme:

Explanation – 1M

Necessary Figures – 1M

Solution:

- Process P0 sends everyone a request with timestamp 8, while at the same time, process P2 sends everyone a request with timestamp 12. P1 is not interested in the resource, so it sends OK to both senders. This is shown in Fig. 12 (b).
- Processes P0 and P2 both see the conflict and compare timestamps. P2 sees that it has lost, so it grants permission to P0 by sending OK. (As shown in Fig. 12 (b))
- Process P0 n

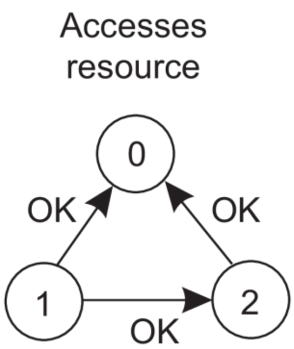


Fig. 12 (b)

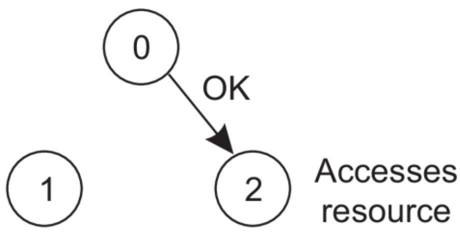


Fig. 12 (c)

ow queues the request from P2 for later processing and accesses the resource, as shown in Fig. 12 (b).

- When it is finished, it removes the request from P2 from its queue and sends an OK message to P2, allowing the latter to go ahead, as shown in Fig. 12 (c).
- The algorithm works because in the case of a conflict, the lowest timestamp wins and everyone agrees on the ordering of the timestamps.

Q. 14 ➔ 3M

Answer the following.

I. With diagram analyse the Berkeley algorithm for averaging of physical clocks across machines.

II. Causality among processes in distributed systems is essential to know the relationship between them. Apply the vector clock algorithm on Fig. 14 and give the logical time at process P3 after receiving m2.

Scheme:

Berkeley algorithm – 1.5 M Logical time at P3 – 1.5 M

Solution:

I. Berkeley Algorithm

- Here the time server (actually, a time daemon) is active, polling every machine from time to time to ask what time it is there.
- Based on the answers, it computes an average time and tells all the other machines to advance their clocks to the new time or slow their clocks down until some specified reduction has been achieved.
- This method is suitable for a system in which no machine has a UTC receiver. The time daemon's time must be set manually by the operator periodically.
- In Figure 6.6(a) at 3:00, the time daemon tells the other machines its time and asks for theirs. In Figure 6.6(b) they respond with how far ahead or behind the time daemon they are. Armed with these numbers, the time daemon computes the average and tells each machine how to adjust its clock [see Figure 6.6(c)].

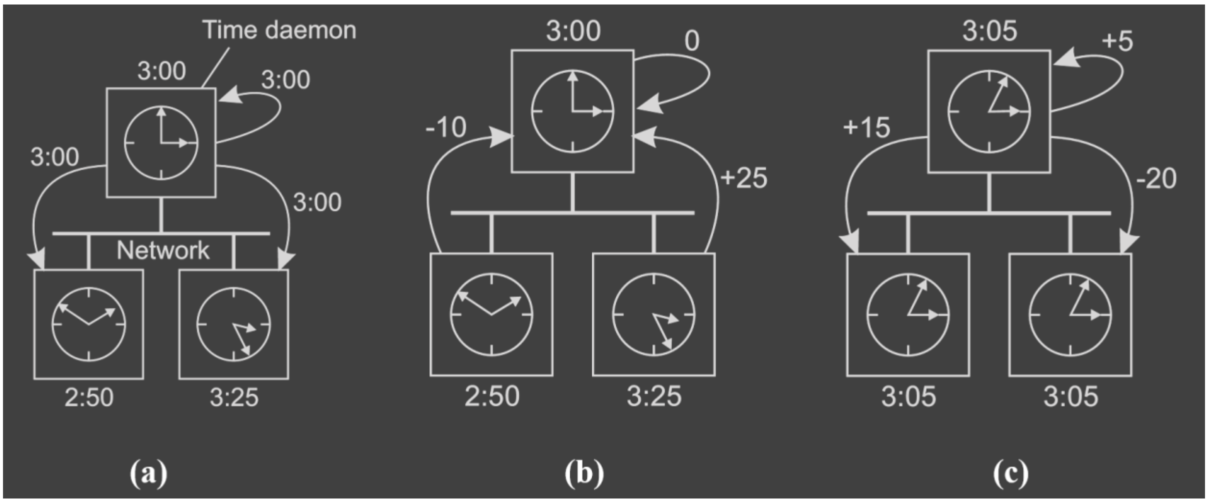


Fig. 6.6

II. Logical time at P3

The logical time at P3 is (5,3,2) as shown in Fig. 14s

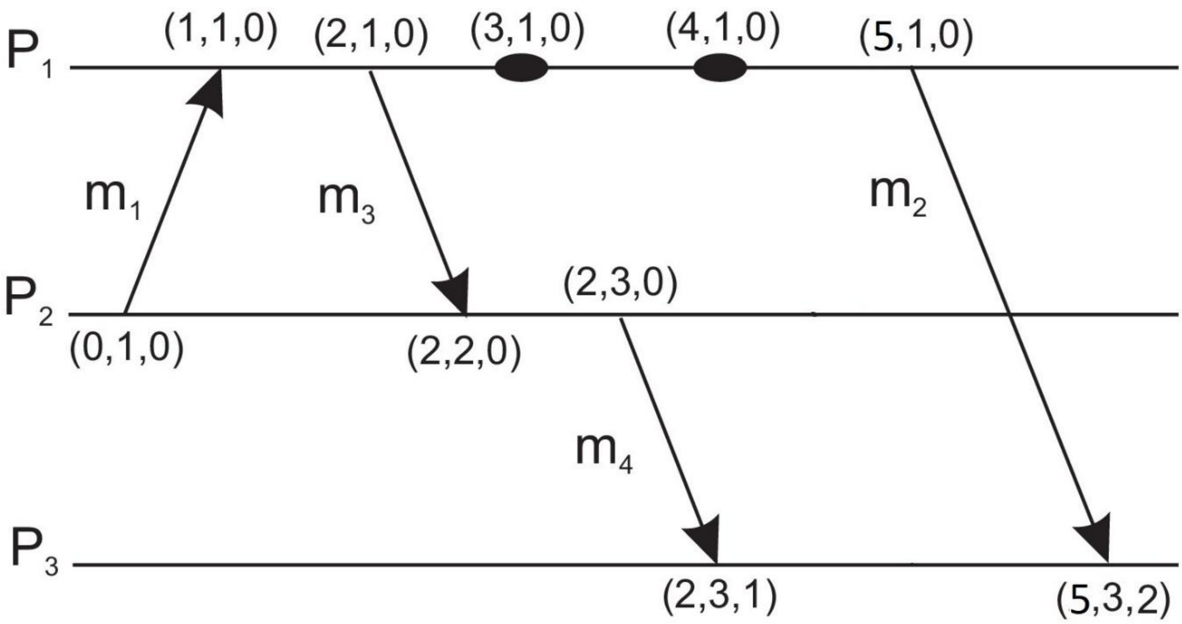
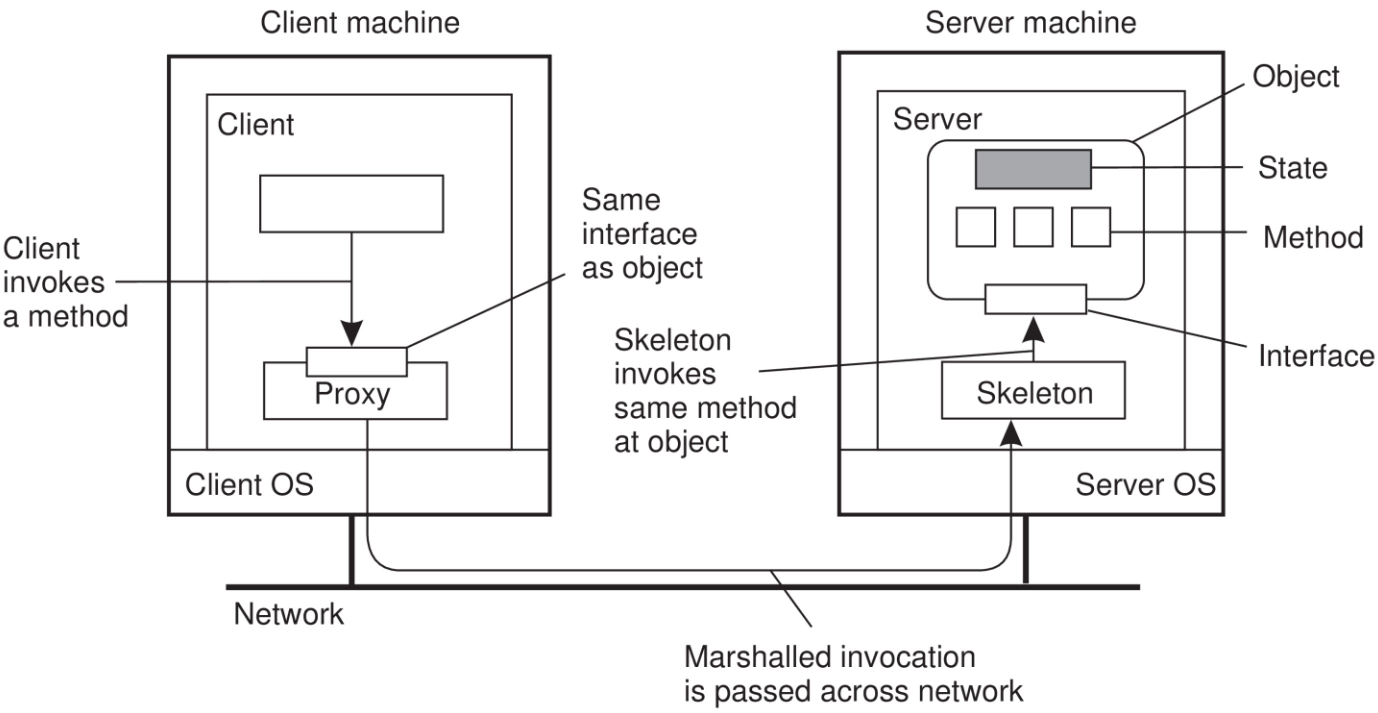


Fig. 14s

Scheme for Q 11 and Q 13

Q 11



(1M for diagram +1 M for explanation)

Q 13

Size Scalability, Geographical Scalability and Administrtive Scalability

(1.5 M for points + 1.5 M for explanation)