Handling Deadlock for Distributed Systems - I

Different Approaches

1

Outline

- Introduction to Distributed Deadlock Detection
- Control Models for DDD algorithm
- Centralized DDD algorithm
- Diffusion Computation-based DDD algorithm
- Mitchell-Merritt Edge Chasing DDD algorithm

7 June 2024

2

Outline

- Introduction to Distributed Deadlock Detection
- Control Models for DDD algorithm
- Centralized DDD algorithm
- Diffusion Computation-based DDD algorithm
- Mitchell-Merritt Edge Chasing DDD algorithm

Definition

- Deadlock is a state of contention in the system involving two or more blocked processes that can never be resolved unless there is some external intervention.
- Deadlocks are always stable.

Conditions for Deadlock

- Mutual Exclusion: Resource is held by one and only one process at a time
- Hold and Wait: A process is allowed to hold on allocated resources while it's waiting to acquire other resources
- No Preemption
- Circular Wait

7 June 2024

5

Handling Deadlocks

- There are three broad approaches towards mitigating deadlocks either for centralized system or for a distributed system. These are:
  - Deadlock Prevention
  - Deadlock Avoidance
  - Deadlock Detection and Recovery

7 June 2024

6

Deadlock Prevention

- Prioritize processes and assign resources accordingly
- Make prior rules to deny one of the 4 necessary conditions
- May lead to starvation and affect concurrency

7 June 2024

7

Deadlock Avoidance

- Only fulfill those resource requests that won't cause deadlock
- Simulate resource allocation and check if resultant state is safe or not.
- Requires Prior resource requirement information for all processes.
- High cost for scalability

7 June 2024

8

Deadlock Detection and Recovery

- Periodically examine process status and check if one or more processes are in deadlock.
- Select the process to be killed such that it affects least
- Roll back on one or more processes and break the circular wait

7 June 2024

9

DD Detection Requirements

- Progress Condition
  - No undetected deadlocks
    - All deadlocks found
    - Deadlocks found in finite time
- Safety Condition
  - No false deadlock detection
    - Phantom deadlocks caused by network latencies
    - Principal problem in building correct DS deadlock detection algorithms

7 June 2024

10

Models for Requests

- The AND model requires all resources to be granted to un-block a computation
  - **A cycle is sufficient to declare a deadlock with this model**
- The OR model allows a computation making multiple resource requests to un-block as soon as any one is granted
  - **A cycle is a necessary condition**
  - **A knot is a sufficient condition**

7 June 2024

11

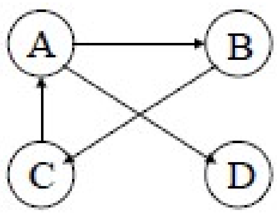
What is a Knot?

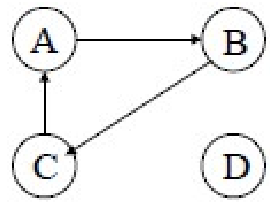
- A knot of a graph is a subset K of nodes such that the reachable set of each node in K is exactly K.
- A knot is a cycle with no non-cycle outgoing path from any node.
- In presence of a knot, there will be no active processes to release resources

7 June 2024

12

Cycle and Knot





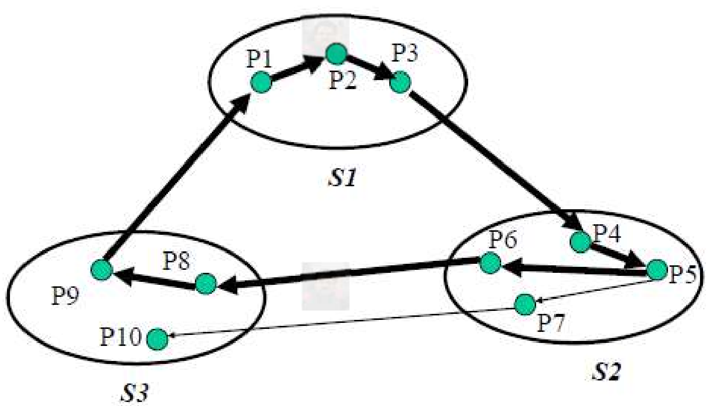
On left A, B and C are in a Cycle, but not in a Knot
On right A, B and C are in a Knot

In presence of a knot, there will be no active processes to release resources

7 June 2024

13

Cycle and Knot

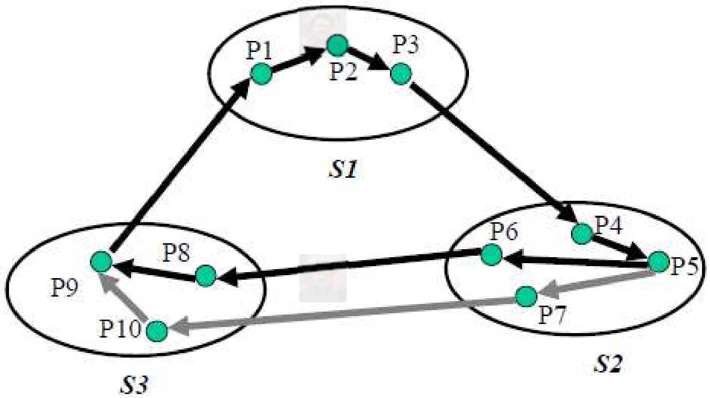


Not a deadlock in the OR model – deadlock in the AND model

7 June 2024

14

Cycle and Knot



A deadlock for both OR and AND model for a Knot

7 June 2024

15

Outline

- Introduction to Distributed Deadlock Detection
- Control Models for DDD algorithm
- Centralized DDD algorithm
- Diffusion Computation-based DDD algorithm
- Mitchell-Merritt Edge Chasing DDD algorithm

7 June 2024

16

Control Models for DDD

- Centralized Control
  - A single control site constructs wait-for graphs (WFGs) and checks for directed cycles.
  - WFG can be maintained continuously (or) built on-demand by requesting WFGs from individual sites.

7 June 2024

17

Control Models for DDD

- Distributed Control
  - WFG is spread over different sites. Any site can initiate the deadlock detection process.
- Hierarchical Control
  - Sites are arranged in a hierarchy.
  - A site checks for cycles only in descendents.

7 June 2024

18

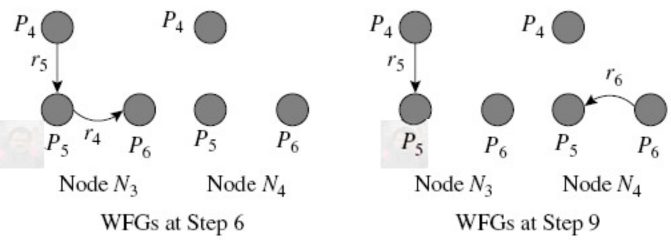
Outline

- Introduction to Distributed Deadlock Detection
- Control Models for DDD algorithm
- Centralized DDD algorithm
- Diffusion Computation-based DDD algorithm
- Mitchell-Merritt Edge Chasing DDD algorithm

Ho-Ramamurthy DDD Algorithm

- Each site maintains 2 status tables: resource status table and process status table.
  - Resource status table: Resources locked by or requested by processes.
  - Process status table: Processes that are locked or are waiting for resources.
- Controller periodically collects these tables from each site.

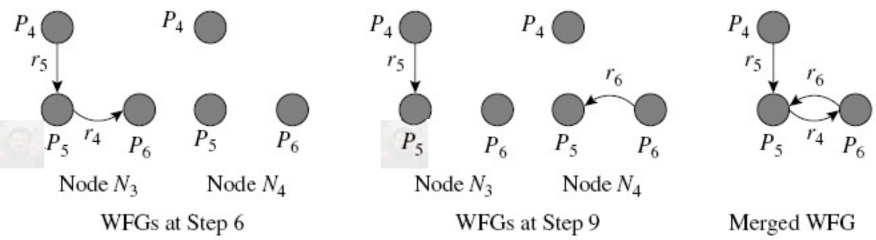
Problem with Centralized Approach



Ho-Ramamurthy DDD Algorithm

- Controller constructs a WFG from transactions common to both the tables.
- If there is no cycle, then no deadlock is detected.
- A cycle means a deadlock.

Problem with Centralized Approach



7 June 2024

23

Ho-Ramamurthy DDD Algorithm

- Single point of failure
- Network congestion issues
- False deadlock detection

7 June 2024

24

Ho-Ramamurthy 2-phase algorithm

- Each site maintains a status table of all processes initiated at that site.
- It includes all resources locked and all resources being waited on.
- Controller site requests (periodically) the status table from each site.
- Controller then constructs WFG from these tables and looks for cycle(s).

Ho-Ramamurthy 2-phase algorithm

- If no cycle exists then there is no deadlock.
- If cycle exists, then request for status tables again.
- Construct WFG based only on common transactions in the 2 tables.
- If the same cycle is detected again, system is in deadlock.

Ho-Ramamurthy 2-phase algorithm

- Is this approach free from phantom deadlock?

7 June 2024

27

Outline

- Introduction to Distributed Deadlock Detection
- Control Models for DDD algorithm
- Centralized DDD algorithm
- Diffusion Computation-based DDD algorithm
- Mitchell-Merritt Edge Chasing DDD algorithm

7 June 2024

28

Chandy, Misra, Haas Algorithm

- Initiation: Any process, say A, blocked for a long time may initiate the diffusion process as:
  - Send query to all outgoing edges on the WFG
  - Wait for that many replies

7 June 2024

29

Chandy, Misra, Haas Algorithm

- On receipt of an engaging query, a blocked process B does the following:
  - Send query to all outgoing edges on the WFG
  - Wait for that many replies
  - If all the replies arrive and B is blocked continuously since it received the engaging query, then B sends a reply to its parent

7 June 2024

30

Chandy, Misra, Haas Algorithm

- On receipt of a non-engaging query, a blocked process B does the following:
  - If process B is blocked continuously since it received the engaging query, then B sends a dummy non-engaging reply to its parent

7 June 2024

31

Chandy, Misra, Haas Algorithm

- Deadlock Detection: If the initiator receives all the replies and it is blocked continuously since it initiated the diffusion process, then a deadlock is detected

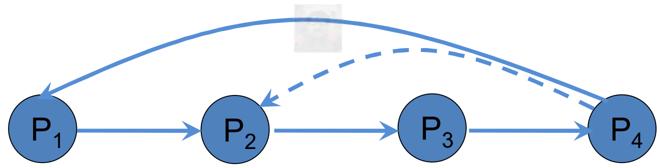
7 June 2024

32

Diffusion-Computation approach

- P_2, P_3 are blocked.
- Hence, P_1 is blocked and sends a query but does not receive a reply because P_4 is not blocked

Diffusion-Computation approach



- Now, P_4 requests for a resource held by P_1 or by P_2.
- In this case, the reply would reach the initiator and a deadlock will be detected.

Thanks for your kind attention

Questions??

35