Dead lock Detection Algorithm in Distributed Systems

30/08/22 9:10 PM

So there are four types of algorithms which are avaliable in Distributed systems

Types of Deadlock Detection algorithms in Distributed system
Path Pushing
Edge Chasing
Diffusion Computation
Global sate detection

What is a Deadlock
Dead lock is a state where a process may ask for resources which are held by other processes

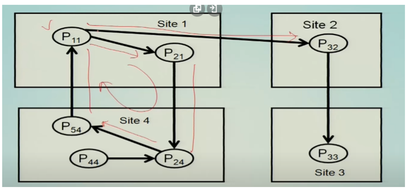
Model For Detecting Deadlock in a Distribuated Systems
Before we talk about models we need to make few assumptions regarding Distribuated systems

Assumptions for a Distribuated systems

- The systems have only reusable resources.
- Processes are allowed to make only exclusive access resources.
- There is only one copy of each resource.
- A process can be in two states: running or blocked.
- In the running state (also called active state), a process has al needed resources and is either executing or is ready for execution
- In the blocked state, a process is waiting to acquire some resources

Wait For Graph Data structure
Now we can represent a state of a processor with a Graph data structure called wait for graph

What is wait For Graph Data structure
In wait for graph data structure we are having all the processors as represent as node and there is a edge between node 01 --> node 02 if noted 01 is waiting for a node 02 to release some resources and we can say that the system is in dead lock if there is a cycle which is forming in the graph



Strategies of Deadlock Detection in DC

There are Four strategies for Dead lock Detection
Dead lock Avoidance
Dead lock handling
Dead lock prevention
Dead lock Detection

Dead lock Handling

In the Deadlock Handling is a very complex procress because Distribuated systems don't have a full knowledge of the global system and systems don't have a common physical clock so not good for DS

Dead lock prevention

In Dead lock prevention what we are trying to do is a procress will take all the resources before it starts executing all preempt any process who is holding a resource which is needed by the process and this is not possible

Dead lock avoidance

The resource will only be allocated if the final state will not result in a deadlock

Dead lock Detection

Deadlock detection requires examination of the status of process resource interactions for presence of cyclic wait.

Dead lock Detection is the best method

Requirements that ever algorithm should follow

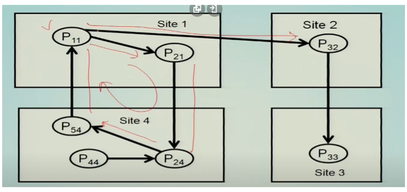
Progress
All the dead locks should be detected by any algoritham

Safety
No false deadlock should be detected by an algoritham
A false deadlock detected by the algoritham is known as phantom dead lock

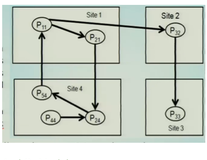
Solving the dead lock
We can solve any dead lock by roll backing one or more dead locked procress to it's previous state

One resource model
In this model each process is allowed to have only one outstanding resource request since the maximum degree of node in 01 then WSG will identify a dead lock as soon as it arises

And model
In and model an procress can request for multipal number of resources and all the request are satisfier only when all the requested resources are avaliable at the same time



Or Model
In OR model an procress can request for multipal number of resources and all the request are satisfier when any one resource is avaliable the presence of a cycle don't represent a dead lock in a or model



And Or Model

And or model is a generalization of the 2 models and or model and or model a request may specify any combination of AND or in the request resource. For example, in AND OR model a request for multiple resources can be of the form x, y or z to detect the presence of a deadlock in such a model there is no familiar construct of a graph theory for that use, the wait for graph hence the deadlock is detected using its stable property. So, a deadlock in AND OR model can be directed by repeated application of the test for the or model

P out of Q model
Now, another model is called P out of Q modeless, another form of and or model is called P out of Q model which allows the process to request any k available resources from a pool of n resources it has same expressive power as at as and or model we have seen earlier; however, P out of Q model lends itself to a much more compact formation of a request. So, every request in a P out of Q model can be expressed in the form of AND OR graph and vice versa note that and requests for P resources can be stated as P out of P; that means, all P resources are required that is the AND model and the OR model request for the P resources can be stated as 1 out of P that is an OR model. So, P out of Q can be expressed in these 2 forms of OR and n

Classification of deadlock detection algorithm
So there are four types of dead lock detection algoritham which are there in a distributed system
Path pushing
Edge chasseing
Diffusion computation
And global state detection based algoritham

Path pushing
In this algoritham a global WFG is maitened and when any procress ask for a resource at that time it will send it's current state to all the node and this procress is continues until any one node don't have the full picture of a Distributed system and once a full picture is established at that time one node will asses the scenario and decide weather dead lock is present in the system or not

Edge Chasseing
In the Edge chasseing algorithm a probe is sent to all the system and if any system which is currently exciting will reject the probe and continue it's execution hence if any process will accept the probe then it is decided that a cycle is there in the DS the benefit of this system is that the size of a probe is smaller so the over head on the system is less