ACS Theory Assignment 3

ndp689 mnp553

January 2022

1 Reliability

1. There are 3 nodes (buildings) in the Daisy Chain network topology. So, the network has 2 links.

   And we assume each link's failure is independent with probability p. Then, the probability that the daisy chain network is connecting all the buildings is:

   $$(1-p)^2$$

2. The fully connected network is connecting if there are at least two connected links. Thus,

   $$\binom{3}{2}p(1-p)^2 + \binom{3}{3}(1-p)^3 = 3p(1-p)^2 + (1-p)^3$$

3. For the Daisy Chain network, the probability of connecting all the buildings is

   $$(1-p)^2 = (1-0.000001)^2 = 0.999998$$

   For the fully connected network, the probability of connecting all the buildings is

   $$\begin{aligned} &3p(1-p)^2 + (1-p)^3 \\ &= 3 \times 0.0001 \times (1-0.0001)^2 + (1-0.0001)^3 \\ &= 0.99999997 \end{aligned}$$

   Obviously, the fully connected network is more reliable, so they should buy a fully connected network with three low-reliability links.

2 Vector Clock

For an event, if the value of *Vector Clock at Process Before the Event* is larger than or equal to the value of *Vector Clock in the Message*, the event will take Action **A1**.

For an event, if the value of *Vector Clock at Process Before the Event* is less than the value of *Vector Clock in the Message*, the event will take Action **A2**.

For an event, if the value of *Vector Clock at Process Before the Event* and the value of *Vector Clock in the Message* cannot be compared, the event will take Action **A3**.

Event	Vector Clock Before	the Message	Action	Vector Clock After
A	(0,0,0)	(1,0,0)	A2	(1,0,0)
B	(0,0,0)	(1,0,0)	A2	(1,0,0)
C	(0,0,0)	(1,0,0)	A2	(1,0,0)
D	(1,0,0)	(1,1,0)	A2	(1,1,0)
E	(1,0,0)	(1,1,0)	A2	(1,1,0)
F	(1,1,0)	(1,1,1)	A2	(1,1,1)
G	(1,0,0)	(1,1,1)	A2	(1,1,1)
H	(1,1,1)	(1,1,0)	A1	(1,1,1)
I	(1,1,0)	(1,2,0)	A2	(1,2,0)
J	(1,1,1)	(1,2,0)	A3	(1,2,1)
K	(1,1,1)	(1,2,0)	A3	(1,2,1)
L	(1,2,0)	(1,1,1)	A3	(1,2,1)

3 Distributed Transactions

1. Is there a local deadlock on any of the nodes?

Node 1: There are no deadlocks.

Transaction 2 wants to get an exclusive lock on A, but it needs to wait until T1 is finished. And transaction 3 wants to get an exclusive lock on B, but it needs to wait until T2 is finished. Then, as the Figure 1 shows, there are no cycles in the graph. So, there are no deadlocks.

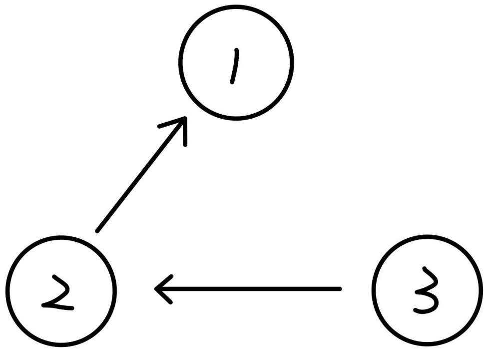

Figure 1: the local waits-for graph on Node 1

Node 2: There are no deadlocks.

Transaction 4 wants to get an exclusive lock on D, but it needs to wait until T2 is finished. And transaction 2 wants to get an exclusive lock on C, but it needs to wait until T1 is finished. Then, as the Figure 2 shows, there are no cycles in the graph. So, there are no deadlocks.

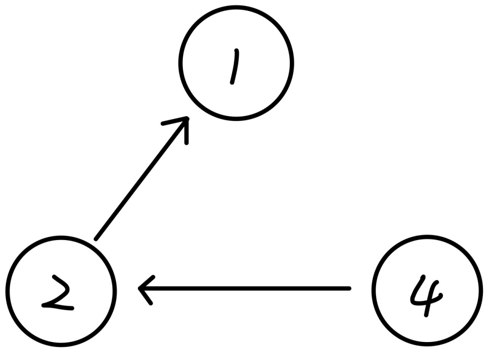

Figure 2: the local waits-for graph on Node 2

Node 3: There are no deadlocks.

Transaction 1 wants to get an exclusive lock on E, but it needs to wait until T3 is finished. And transaction 2 wants to get an exclusive lock on G, but it needs to wait until T1 is finished. Then, as the Figure 3 shows, there are no cycles in the graph. So, there are no deadlocks.

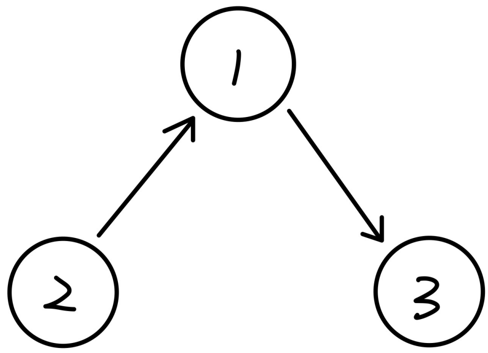

Figure 3: the local waits-for graph on Node 3

2. Is there a global deadlock?

   There is a global deadlock.

   Putting all the local waits-for graphs (Figure 1, 2, 3) together, we get the following global waits-for graph. (Because T4 is a non-distributed transaction, we did not draw it in the graph). Then, as the Figure 4 shows, there is a cycle in the graph. So, there is a global deadlock.

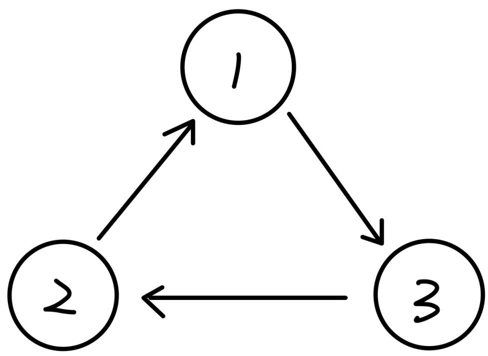

Figure 4: the global waits-for graph

3. Because there is a deadlock in the global waits-for graph. If we want to commit T3, we need to abort T1 or T2, then T3 is allowed to commit.

   The steps and messages needed by T3 to commit are shown in Figure 5.

   The decisions r_1, r_2, r_3, and commits should be recorded to survive failures.

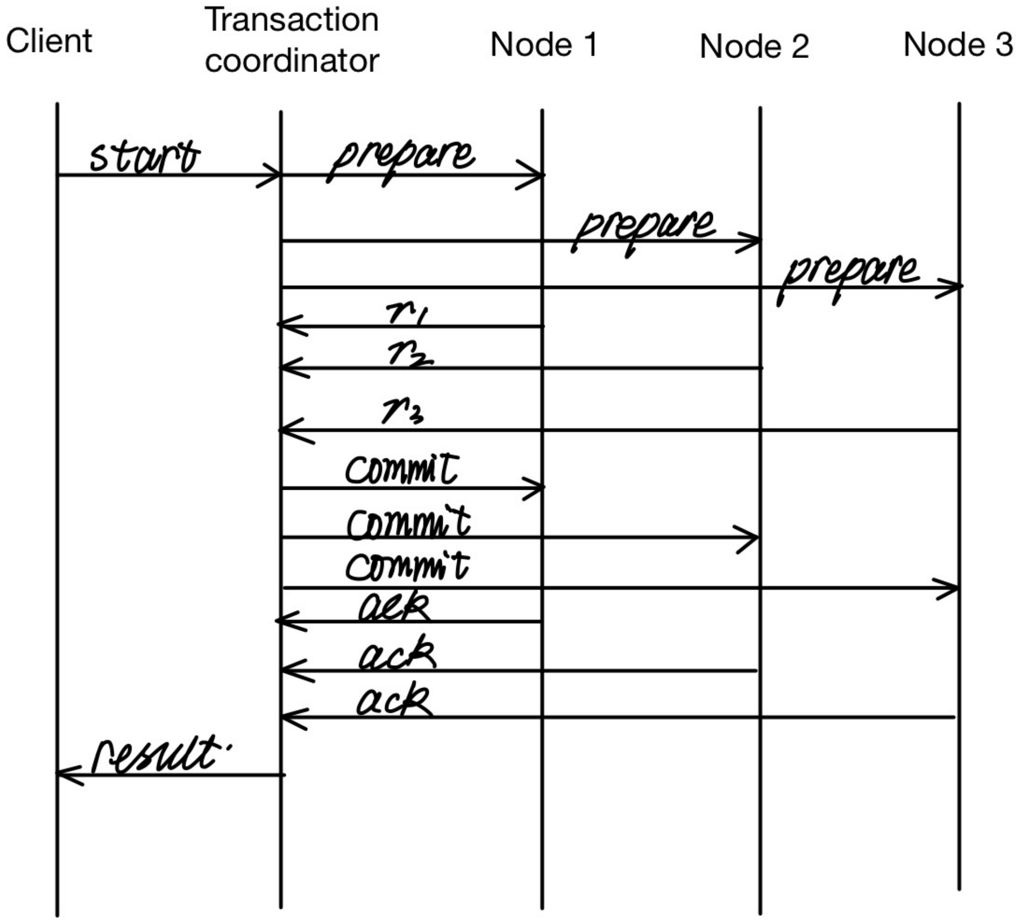

Figure 5: two phase commit protocol