answer – 25.1

1605023

Let us consider the transaction T_{41} at site 4. The transaction displays of balance of account A. There is single lock manager at site 1.

i. Write the transaction with lock.

ii. Write the steps to obtain the lock by T_{41}.

Solution

(i) Transaction

i. LOCK-S(A)

ii. READ(A)

iii. DISPLAY(A)

iv. COMMIT(A)

v. UNLOCK(A)

A transaction does not remain serializable after commit operation. So, commit precedes unlock here.

(ii) Here, TC4 initiates the transaction T_{41} at site 4 and sends lock request for data item A to single lock manager at site 1. Then, SLM forwards this lock request to all the sites having replica of data item A. So, site 1, site 2, and site 4 receive lock request for A from SLM. We need to remember that SLM can not make any decision about this transaction here. It just coordinates TC's and TM's of all the sites involved and manages locks. SLM does not have access to database or any particular data item. Rather, it uses

Date :

a dictionary (with mapping between available data items and sites where their corresponding replicas are available) to request lock for a particular data item to corresponding sites with replica. TM of the corresponding site basically decides on whether to give/grant lock for the data item. SLM, then, connects the Transaction coordinator at site4 (TC4) with the transaction manager (TM) to continue the transaction T41. Here, TM which responds and grants lock first involves in the later transaction process.

So, the steps are —

Steps

i. TC4 sends lock_S(A) request to SLM.
ii. SLM forwards lock_S(A) request to TM1, TM2, TM4.
iii. SLM forwards granted lock_S(A) from any one of those sites to TC4.

Ans.

answer – 15.2

1605023

let us consider the transaction T_{21} at site2. The transaction adds Tk.1000 to account B. There is single lock manager at site 1.

i. write the transaction with lock.

ii. write the steps to obtain the lock by T_{21}.

Solution

(i) Transaction

i. <START T_{21}>
ii. LOCK_x(B)
iii. READ(B)
iv. B:= B + 1000
v. WRITE(B)
vi. COMMIT(B)
vii. UNLOCK(B)
viii. <END T_{21}>

WRITE(B) must be performed on both replicas of B at site3 and site4.

(ii) steps

i. TC2 sends lock_x(B) request to SLM at site1.
ii. SLM forwards lock_x(B) request to TM3 and TM4.
iii. Both TM3 and TM4 grant lock_x(B) request.
(In Read operation, lock grant from any one of the TM's is enough. But in write operation, lock grant from all the TM's involved is a must for maintaining consistency in database.)

iv. SLM forwards granted lock_x(B) to TC2.

Ans.

answer – 26·1

16050023

A data item P is replicated in 12 sites. Transaction T_2 has Lock-x(P). Find the minimum number of messages required to

a. obtain this lock using majority protocol?

b. unlock using majority protocol?

Solution

(a) replica of data item P is available at 12 sites.

Hence,

messages for lock request $= \frac{12}{2} + 1 = 7$

messages for lock grant $= \frac{12}{2} + 1 = 7$

$\therefore$ minimum $(7+7) = 14$ messages are required to obtain Lock-x(P) using majority protocol.

(b) Here,

messages for unlock $= \frac{\text{total replica}}{2} + 1 = \frac{12}{2} + 1 = 7.$

$\therefore$ minimum 7 messages are required to unlock using majority protocol.

Ans.

answer - 26.2

1605023

Data items A and B are replicated in 15 sites. Transaction T_{11} has Lock-x(A) and Lock-S(B).

a. What is the minimum number of sites required to obtain these locks using biased protocol?

b. Find the number of messages to obtain lock and release lock in each case.

c. Compare biased protocol with majority protocol in terms of performance.

Solution

(a) minimum number of sites required

- Lock-x(A) ⟶ 15 sites.
- Lock-S(B) ⟶ 1 sites.

(b) number of messages to obtain and release lock

- lock-x(A) ⟶ 2×15 or 30 msg to obtain lock.
  ⟶ 15 msg to release lock.
- lock-S(B) ⟶ 2×1 or 2 msg to obtain lock.
  ⟶ 1 msg to release lock.

(c)

majority protocol	biased protocol
more robust as transactions with both shared and exclusive locks can continue when majority of sites are up	less robust as transaction with exclusive lock fails when any one of the sites fails
Same number of messages are required for both shared and exclusive locks which degrades overall performance	shared lock requires less number messages which eventually enhances overall performance
better performance in write/update-intensive transactions	better performance in read-intensive transactions

Ans.

answer – 26.3

1605023

The data item P is replicated in 8 sites $S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8$ and the weight of the sites are 2, 2, 2, 3, 3, 3, 4, 4 respectively.

a. Find Q_w and Q_r such that the protocol is
   i. biased   ii. majority

b. Find the replicas to be locked for Q_w and Q_r other than (a) for the following cases:
   i. Lock-S(P)   ii. Lock-X(P).

Here, the options are:

option-1 → Starting the sites with lowest weights in ascending order.

option-2 → Starting the sites with highest weights in descending order.

Solution

a. Here, $S = \sum_{i=1}^{8} w_i = 2+2+2+3+3+3+4+4 = 23.$

i. for biased protocol, lock-S(P) requires 1 replica lock. and lock-X(P) requires 8 replica lock.

So, $Q_r = 1$ will be enough for lock-S(P) and $Q_w = S = 23$ will be enough for lock-X(P).

Here, $Q_r + Q_w = 24 > 23$ and $2 \times Q_w = 46 > 23$.

ii. for majority protocol, lock-S(P) and lock-X(P) (both of them) require lock from at least $(\frac{8}{2}+1) = 5$ replicas. Also, we can set $Q_w = Q_r$ for the majority protocol case. Considering 5 replicas with lowest weights will help us set Q_r and Q_w in such a way that quorum consensus protocol starts to act as majority protocol.

Therefore, $Q_r = 2+2+2+3+3 = 12$ and $Q_w = Q_r = 12$ will be enough for lock-s(P) and lock-x(P) respectively. Here, $Q_r + Q_w = 24 > 23$ and $2 \times Q_w = 24 > 23$.

(b) Assuming, $Q_r = 11$ and $Q_w = 13$ where $Q_r + Q_w > S (23)$ and $2 \times Q_w > S$.

i. Lock-S(P)

option-1

weights $= w_1 + w_2 + w_3 + w_4 + w_5$
$= 2+2+2+3+3$
$= 12 > Q_r (11)$.

So, sites to acquire locks from are S_1, S_2, S_3, S_4, S_5.

option-2

weights $= w_8 + w_7 + w_6$
$= 4+4+3$
$= 11 = Q_r$.

So, sites to acquire locks from are S_6, S_7, S_8.

ii. Lock-x(P)

option-1

weights $= w_1 + w_2 + w_3 + w_4 + w_5 + w_6$
$= 2+2+2+3+3+3$
$= 15 > Q_w (13)$.

So, sites to acquire locks from are $S_1, S_2, S_3, S_4, S_5, S_6$.

option-2

weights $= w_8 + w_7 + w_6 + w_5$
$= 4+4+3+3$
$= 14 > Q_w$.

So, sites to acquire locks from are S_5, S_6, S_7, S_8.

Ans.

answer – 27·1 | 1605023

construct a wait-for graph for the following.

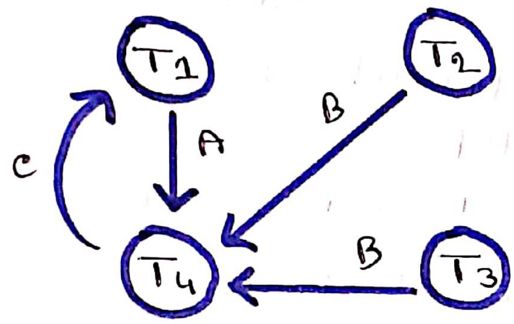
i. Transaction T_1 requests a lock for data A that is locked by T_4.

ii. Transaction T_2 requests a lock for data B that is locked by T_4.

iii. Transaction T_3 requests a lock for data B that is locked by T_4.

iv. Transaction T_4 requests a lock for data C that is locked by T_1.

Solution



Ans.

discussion – 27·1

why is site identifier for a transaction in LSB and local unique timestamp for that transaction in MSB of its global unique identifier? what would happen if it would be reverse?

Solution

We want to put more bias on the timestamp value when we are considering the priority for resolving deadlock. Putting transactions with nearby timestamp value close to one another ensures the aforementioned bias. We can do this by putting local timestamp in MSB and site id in LSB of global id.

If we put site id in MSB and local timestamp in LSB, then more bias will be imposed on the site id when resolving deadlock and we simply do not want that.

Ans.

answer – 17.2

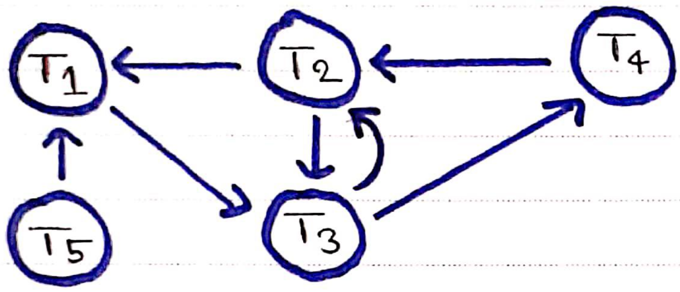
1605023

Construct the global wait-for graph from the given local wait-for graphs for sites S_1, S_2 & S_3 and find the deadlock status.

Date :

solution

global wait-for graph



Deadlock Status

There are, in total, 4 cycles in the global wait-for graph:

i. $T_1 \rightarrow T_3 \rightarrow T_2 \rightarrow T_1$

ii. $T_1 \rightarrow T_3 \rightarrow T_4 \rightarrow T_2 \rightarrow T_1$

iii. $T_2 \rightarrow T_3 \rightarrow T_2$

iv. $T_2 \rightarrow T_3 \rightarrow T_4 \rightarrow T_2$

These cycles indicate deadlock will arise during the execution of these transactions.

Ans.