Name : Manish kumar Jagnani

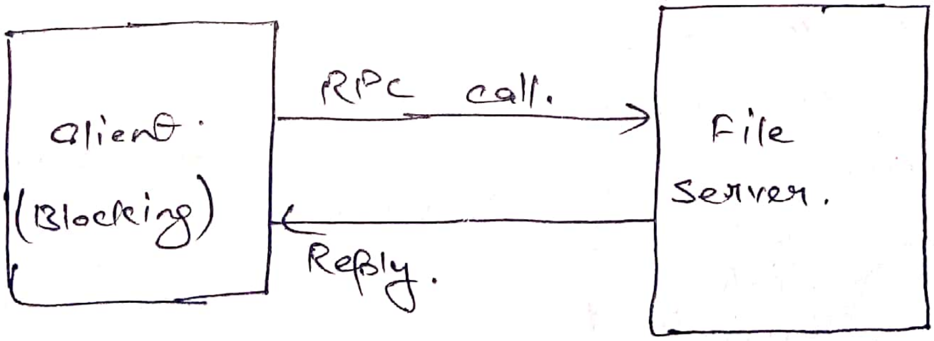
Roll : IIT2017503.

Q.2 What is CODA file system? How does communication perform is CODA file system?

Sol => CODA is a file system for a large-scale distributed computing environment composed of UNIX work-stations. It provides resilency to server and network failures through the use of two distinct but complementary mechanisms.

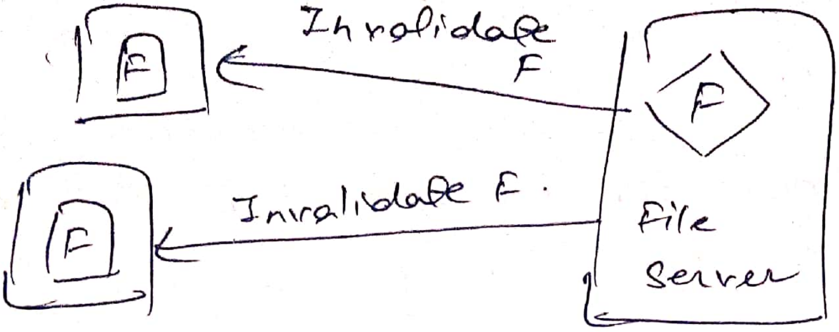
(a) Server replication : Stores copies of file at multiple servers.

(b) Disconnected operation : mode of execution in which a caching site temporarily assumes the role of a replication site.

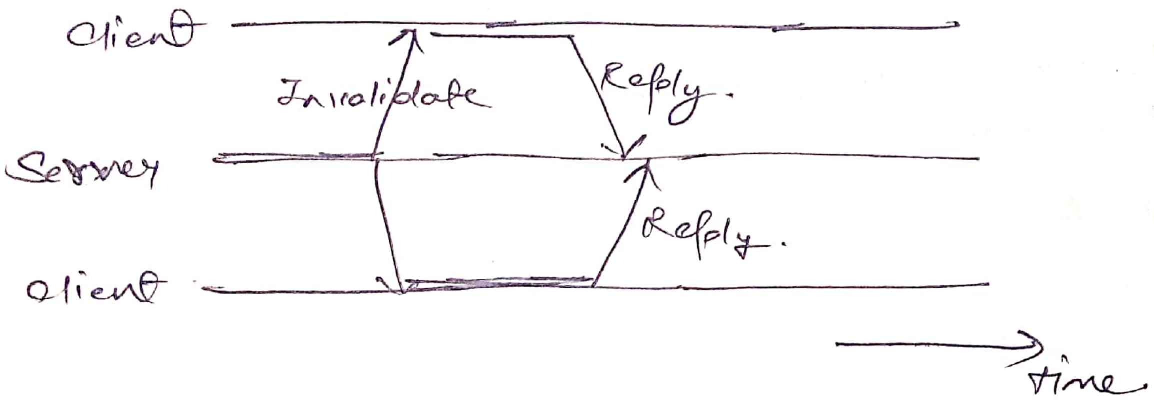


(1) The server keeps sending back messages to the client that it is still working on the problem

(2) If the server dies and the client notices it is not receiving any messages it reports back failure to the client application.

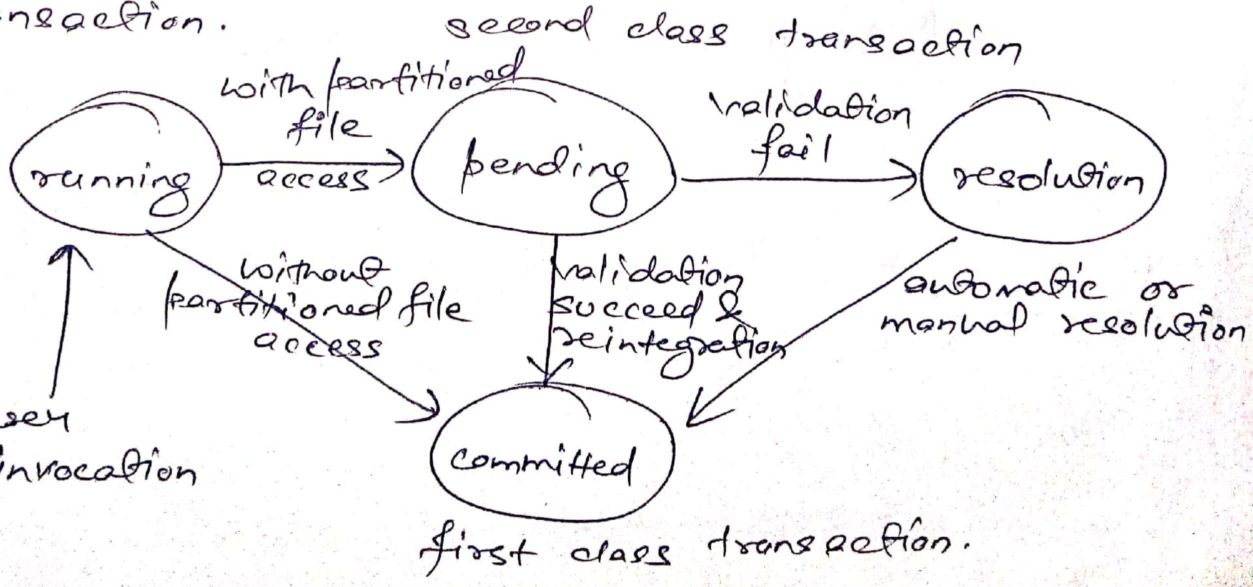


When a server notices updates in a file. It must informe the client which are caching a copy of it to invalidate that copy.



Q.3. Draw transition diagram for Isolation only transaction in ~~Cod~~ CODA. Disuss the difference between first class transaction & second class transaction.

Sol.


second class transaction
with partitioned file access
running
pending
validation fail
resolution
without partitioned file access
validation succeed & reintegration
automatic or manual resolution
user invocation
committed
first class transaction.

1. ~~T~~ A Transaction T is considered a first-class transaction if it does not have any partitioned file ~~acc~~ access, A second-class transaction have a partioned file-access.

2.) The result of a first-class transaction is immediately committed to FHs, a second-class transaction remains in The pending state untill connectivity is restored.

3.) Second-class transaction are guaronteed to be locally serializable ~~with all transaction that were previously resolved or committed at FH~~ among themselves. A-first class transaction is guaranteed to be serializable with all transaction that were previously resolved or committed at FH.

4.) upon reconnection, a second-class transaction is validated against one of two proposed serialization constraints. The first is global serializibility, if the local result of a pending transaction was written to FH, it would be serializable with all previously committed or resolved transaction. The second is global certifiability.

Q.1 A emergency patient dispatch query can be stated as follow: Find the right hospital or take the patient to the default hospital, then dispatch patient status to the emergency doctor for getting The correct treatment. Consider the Moflex transaction model, illustrate how the transaction fits into Moflex transaction structure.

Sol => In the system, an MH is the ambulance vehicle equipped with computers & the MH has to quickly find the proper emergency hospital for a patient. If it fails in cell 1, it tries cell 2. If it succeeds in either cell 1 or cell 2, the geographical information for the hospital is provided to the MH. If it fails in cell 1 & cell 2, the patient is transferred to the default center in cell 3. The activities are modelled as:

t1 : find the proper hospital.

t2 : transmit to the default emergency care center ~~in cell 3~~.

t3 : send the current emergency status for the proper care.

t4 : get the geographical information for the hospital.

t5 : get the patient record.

So $M = \{t1(c), t2(c), t3(c), t4(c), t5(c)\}$.

$S = \{t1 <_s t3, t2 <_s t3, t1 <_s t4\}$.

$F = \{t1 <_s t3\}$

$\Pi = \{L\}$ $L = \{t1, t4\}$.

$H = \{$restart$(t1)$, continue$(t2)$, ~~sp~~ continue$(t3)$, split_resume$(t4)$, continue$(t5)\}$.

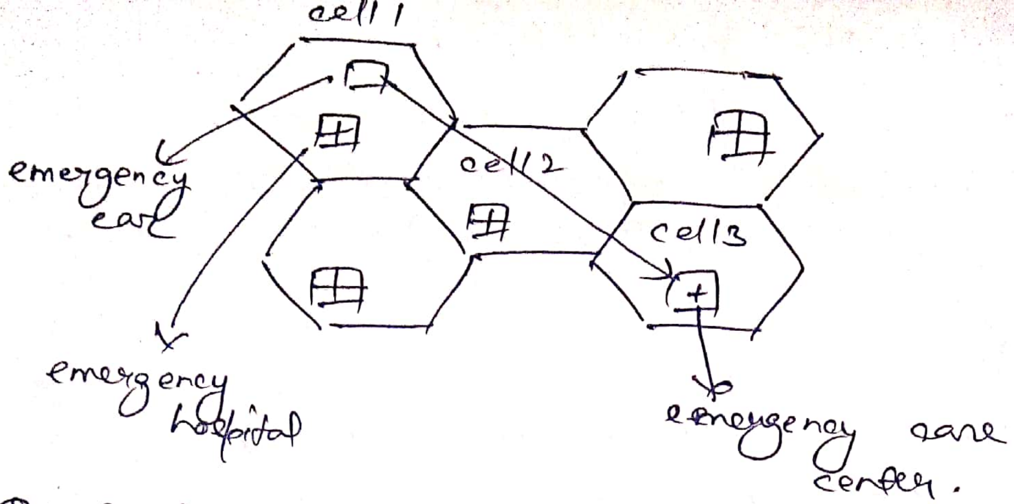
$J = \{$user$(t4)\}$.

$G = \{(s, -, s, s, s), (-, s, s, -, s)\}$

where s is successful execution of corresponding subtransaction

/ – means that the execution state of the subtransaction doesn't affect the decision

Scanned with CamScanner



Q.4 Consider the architecture of a given mobile database system. What types of scenario a transaction may encounter during its execution? Explain your own ideas in managing these situation successfully.

Sol. (i). MU does not move : A transaction arrives & completes its processing entirely at the MU. This is similar to conventional centralized data processing.

(ii). MU moves : A transaction arrives & entirely completes its execution at MU. Required data items are moved here from other nodes. This type of execution is called "local atomic".

(iii) Distributed processing & MU moves : A transaction originates at a MU & is fragmented. The subtransaction are distributed among the MU & a set of DBSs.

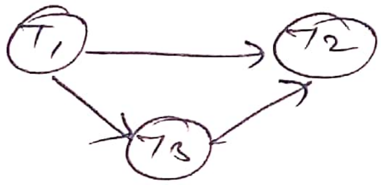
Ideas to manage these situation

(1) Static method : When a transaction originates at a MU, then the BS of this MU becomes the coordinator of the transaction & remains the coordinator. until the trans^n commits. The MU may continue to migrate from one cell to another while processing its subtransaction but the coordinator doesn't change. The MU moves from cell

e1 to c2 with subtransaction e1 leaving behind the coordinator at BS 1 which continue to manage the execution of e1 with the help of BS 2

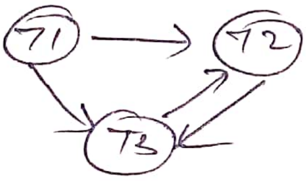
(ii) Dynamic method: In this method, the role of coordinator moves with MU. When MU moves to cell ~~a~~ c2, its base station BS2 becomes the coordinator. of the transaction being executed by the MU. Since a transaction is being processed by multiple DBs, they must know when a new coordinator is assigned to an existing ~~a~~ transaction.

Q.5 (i)



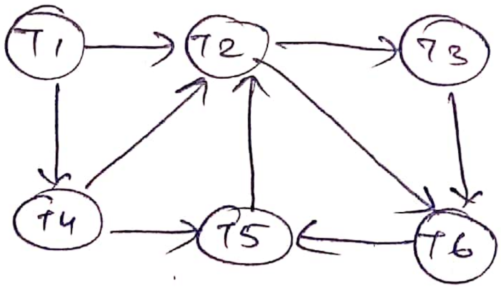
Since it doesn't have any cycle in it.
So it is deadlock free.

(ii)



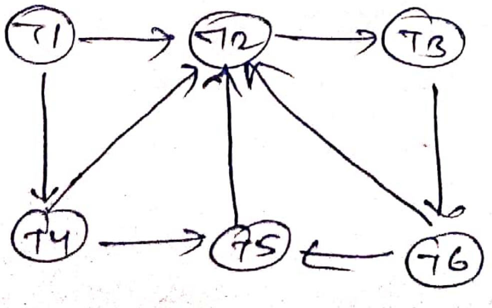
Since there is a cycle T1 → T2 → T3 → T2
so, it has deadlock.

(iii).



Since there is a cycle, T2 → T3 → T6 → T5 → T2
So, it has a deadlock.

(iv)



Since, there is a cycle, T2 → T3 → T6 → T2
So, it has a deadlock.

Q.6 Develop mobile transaction model & a way of executing them on a MDBS.

Sol => (i) A substransaction can share in ~~parallel~~ partial results with the parent transaction anytime & can commit independently

(ii) A substransaction can be forced to wait by other transactions & can be resumed after the other transection have executed.

(iii) Global transaction manager can be categorized into 2 layers. One layer can consist of Global Transaction Coordinators (GTC) in each Mobile Support station (MSS) & manage overall execution & migration of global transaction. The other layer can contain local site manager (LSM) & can be superwise the execution of necessary & unnecessary site transaction.

(iv) If the disconnection occurs from a catastrophic error, the transaction can be suspended. This way needless aborts will be minimized as there is no way to keep track of any ~~oo~~ ongoing transaction.