Backward Error Recovery
↳ operation-Based
↳ state-Based

Date / / Page No.: _____

Failure Recovery

Process that involves restoring an erroneous state to an error free state

Forward error Recovery:
↳ Remove errors in process/system state if errors can be completely assessed
↳ Continue process/system forward execution

Backward

Advantages:
1) simple to implement
2) can be used as general recovery mechanism

Disadvantages:
1) No guarantee that fault doesnot occur again
2) Some components can't be recovered.
3) perform penalty

Backward Error Recovery
Restore process/system to previous error-free state and restart from there.

forward

Advantages:
1) Less Overhead.

Disadvantages:
1) limited to use
2) cannot be used as general mechanism for error recovery

Data / /

Page No.

Application of Agreement protocols

- indivisible
- instantaneous
- uninterrupted

} Atomic Transaction

Atomic Commit

Phase 1: Every site executes their part of distributed transactions & broadcasts their decisions (commit / Abort) to all other sites.

Phase 2: Each site decides whether to commit or abort its part of distributed transactions based on what it has received from other sites in first phase.

Date __/__/____ Page No.: ____

Token Based Algo:

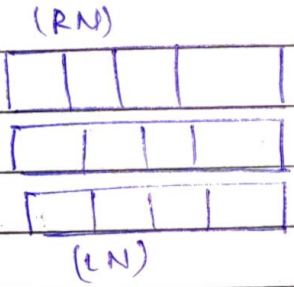
In token based algorithms, a unique token is shared among all sites. If the sites possess token then it is allowed to access the CS

Suzuki- Kasami Algorithm: (Broadcast)

Design Issues:

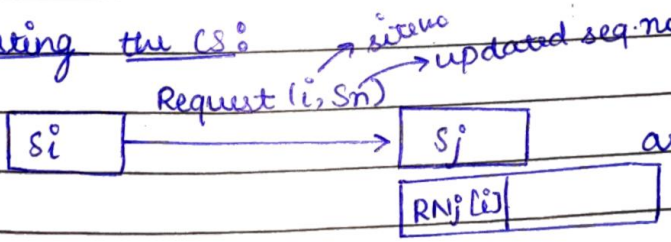
(i) Distinguish outdated Request Msg from Current Request Msg

(ii) Determine which site has outstanding req.



1) Requesting the CS:

(i)



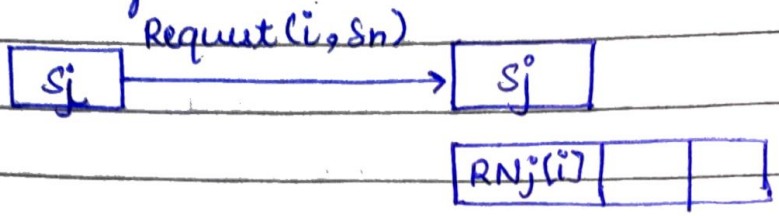
(ii) $RN_j[i]$ to $\max(RN_j[i], S_n)$.
S_j send idle token if $RN_j[i] = LN[i]+1$

2) Executing the CS:
Site S_i execute the CS when it received token.

3) Releasing the CS:
(i) $LN[i] = RN_i[i]$
(ii) updated ID of site not in token queue if $RN_j[j] = LN[j]+1$
(iii) Delete top site ID from queue

"In a gentle way, you can shake the world." –Mahatma Gandhi

1) Requesting the CS:

(i) 

(ii) RNj[i] to max(RNj[i], Sn)

Sj send idle token if

RNj[i] = LN[i] + 1

2) Executing the CS:

Site Si executes the CS when it received token

3) Releasing the CS:

(i) Set LN[i] = RNi[i]

(ii) Updated ID of site not in token only if RN[j] = LN[j] + 1

(iii) Pop queue.

Date __/__/____ Page No.: ____

Singhal's Heuristic Algorithm

$R = \{S_1 \dots S_n\}$

Data structure:

Site → SV[1..N], SN[1..N] (SV: state, SN: seq. no)

Token → TSV[1..N], TSN[1..N]

A site can be in following states:

R → req. the CS

E → execute the CS.

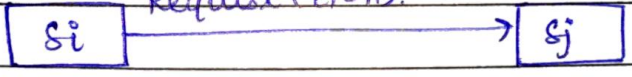
H → Hold the idle token

N → None

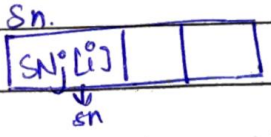
1) Requesting the CS:

(i) set $SV_i[i] = R$

(ii) Increment $SN_i[i] = SN_i[i] + 1$

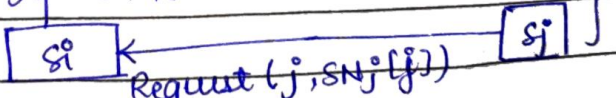


(iii) Discard if $SN_j[i] \geq sn$.



Start of site { j → N, set i → R }

{ j → R, if i ≠ R set i = R



}

{ j → E, set i → R }

{ j → H, set i → R }

Token ⇒ state → R

sequence NO → Sn.

Date ___/___/_____ Page No.: _____

2) Executing the CS:

Si execute CS and set SVi[i] = E

3) Releasing the CS:

Si → SVi[i] = N , TSV[i] = N

for all Sj

If SNi[j] > TSN[j]

then

TSV[j] = SVi[j]

TSN[j] = SNi[j]

else

SVi[j] = TSV[j]

SNi[j] = TSN[j]

If SVi[j] = N set SVi[i] = N

Token	RT	Sd.	Message (ll)	Message (hl)
Suzuki-Kasami	2T+E	T	N	N
Singhal-Heuristic	2T+E	T	N/2	N
Raymond	T(logN)+E	TlogN/2	log(N)	4

Date __/__/____ Page No.: ____

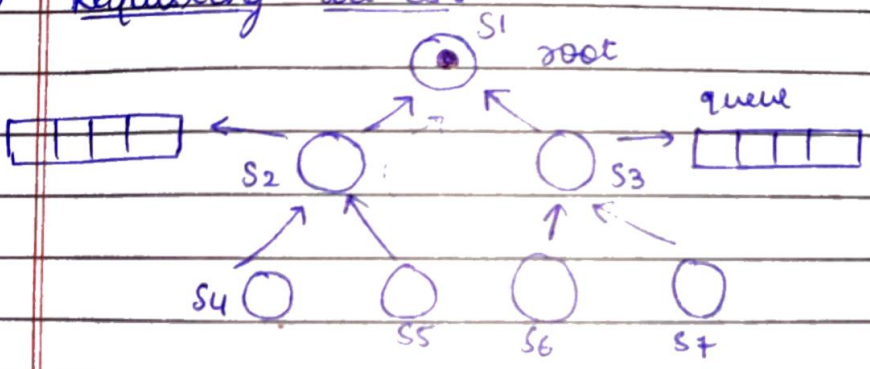
Logical variable holder

Raymond Tree Algorithm

Basic Idea:

1) The site are logically arranged as directed tree.
2) Every site has local variable holder that points to an immediate neighbour node.

1) Requesting the CS:



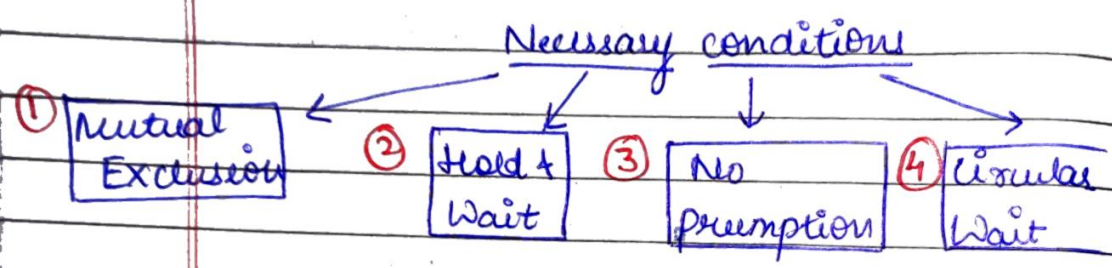
2) Executing the CS:
A site enter CS when it receives the token & deletes the top entry of its requesting queue

3) Releasing the CS:
If request queue is nonempty it delete top entry & send token to that site and set holder.

Date __/__/____ Page No.:

Deadlock

Defined as the permanent blocking of the process ie a set of process is waiting for an event held by some other process.



	Communication Deadlock	Resource Deadlock
1	Waiting is done for msgs	Waiting is done for resources
2	A process cannot proceed with the execution until it can communicate with one of the process for which it was waiting.	A process cannot proceed with the execution until it receives all the resources for which it was waiting.
3	A process can know the identity of those processes on the actions of which it depends.	The dependence of one transaction on actions of other transaction is not directly known.
4	Cant be prevented by safe state	can be prevented by safe state

Diagram

Date __/__/____ Page No.: ____

Centralized Deadlock Detection

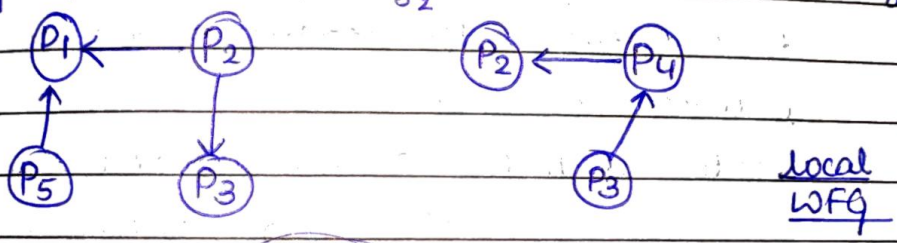
(i) Complete centralized Algo

$R = \{S_1, S_2 \text{ --- } S_n\}$

In this a designated site called control site maintain a WFG to check deadlock

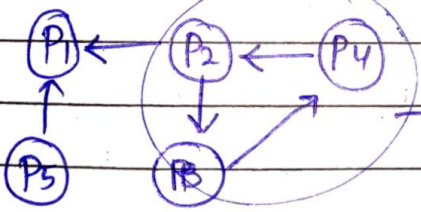
↳ Request Resource
↳ Release Resource
Info

eg. S_1 S_2



local WFG

Control site WFG S_3.



→ cycle ⇒ Deadlock

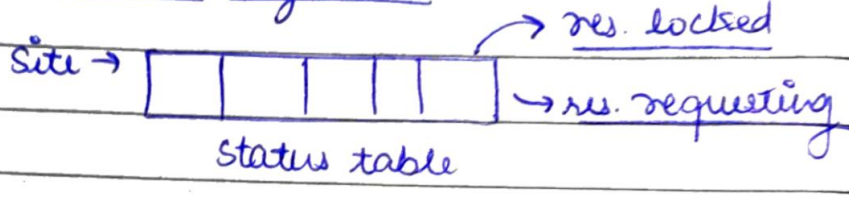
Disadvantages

↳ Dependent only on control site.

(ii) Ho - Ramamoorthy Algorithm

A) Two Phase B) One Phase

A) Two Phase Algorithm



1) every site maintain a status table that contains the status of all process initiated at that site
2) Status includes → all resources locked
→ all resources waited upon.
3) A designated site req. the status table from all sites & constructs a WFG from the info. received & searches it for cycle.
4) If no cycle, system is free from deadlock else, designated site again requests status table from all the sites & again constructs a WFG using only those transactions which are common to both reports.
5) If same cycle is detected, it is deadlocked

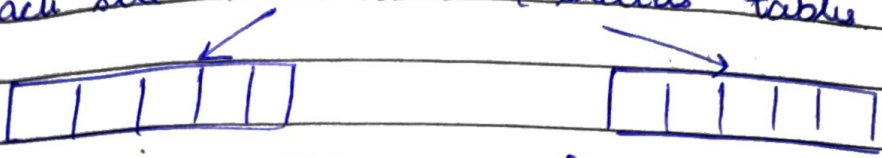
Limitations It may indeed report false deadlock.

Date __/__/____ Page No.: ____

B) One Phase Algorithm:

↳ only one status report from each site

↳ each site maintains 2 status tables



Resource status Table
Keeps track of the transactions that have locked by or waited for by all the transactions at that site

Process status Table
Keeps track of the resources locked by or waited for by all the transactions at that site.

↳ A designated site, requests both the tables from every site, constructs WFG using only those transactions for which entry in the resource table matches with process table & serches for the cycle.

↳ If no cycle is detected so no. deadlock

It does not detect false deadlocks b'coz it eliminate the inconsistency in state info by using only the info. that is common to both tables.

faster & less msgs, more storage → 2 tables.

Date ___/___/_____ Page No.:

Distributed Deadlock Detection

All the sites cooperate to detect a cycle in the state graph that likely to be distributed.

$R = \{S_1, S_2 \dots S_n\}$.

Obermark's Path-Pushing Algorithm

1) In this algo, info about the waitfor dependencies is propagated in the form of paths
2) In this, each site maintain its local WFG It includes Pex.

$P_i \rightarrow Pex$

$Pex \rightarrow P_j$

eg. S_i: $P_2 \rightarrow P_3 \rightarrow Pex$; S_j: $Pex \rightarrow P_4$, $P_3 \rightarrow P_4$

Algorithms:

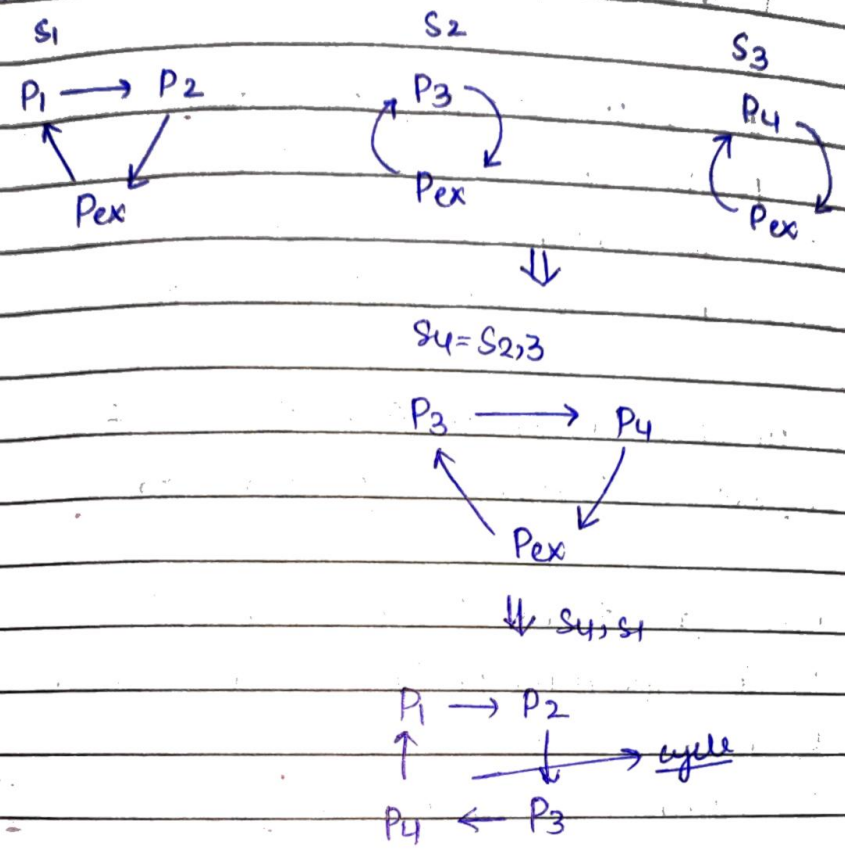
1) If local WFG contains cycle that doesn't involve Pex then system is in deadlock.
2) If Pex exists suppose

$Pex \rightarrow P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow Pex$.

3) S_i —msg update ex.→ S_j
4) S_j form WFG with new info to detect.

Date __/__/____ Page No.: ____

example



Centralized → completely centralized
→ Ho-Ramanoorthy

Distributed → Edge chasing
Path Pushing

Hierarchical → Manasc-Muntz
Ho-Ramanoorthy

Date __/__/__ Page No.:

Chandy Misra Haas's Algo

Edge chasing Algorithm

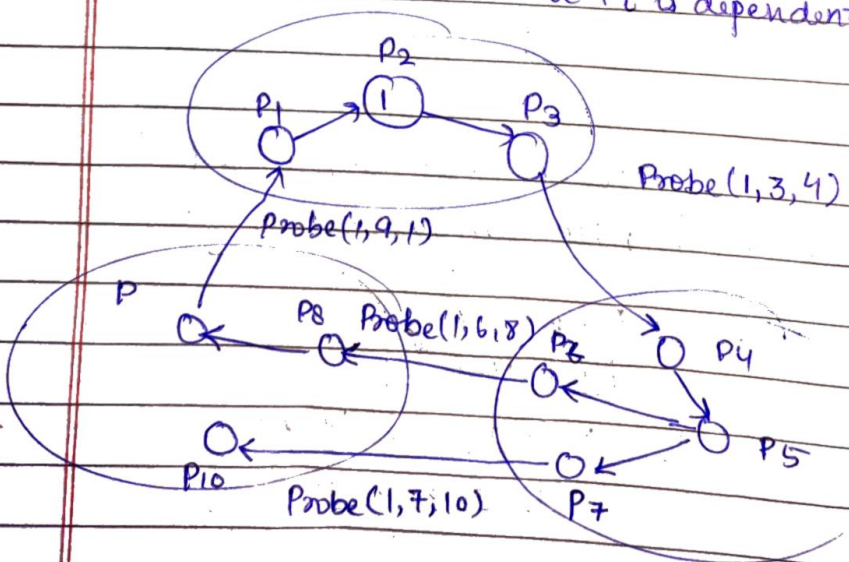
1) In this, we use a special msg called probe
A probe is a triplet (i, j, k) denoting that it belongs to a deadlock detection initiated for process P_i, P_j, P_k

2) Here we construct ~~GTWFG~~ GTWFG.
(Global Transition wait for Graph)

Data structure: Process (P_i, P_j, P_k)

↳ P_i is dependent on P_k if it exists a seq. of processes $\{P_i, P_{j1}, P_{j2}, P_{j3}, P_k\}$

↳ the system maintain a boolean array like dependent_j {if dependent_j(i) ie P_i is dependent on P_j}



Date __/__/____

Page No.: ____

Edge chasing Algorithm → phantom Deadlock Removal

3 steps

1) Initialization: Server initiates to detect deadlock
2) Detection: Detection consists of receiving probes and deciding whether deadlock has occured & forward probes.
3) Resolution: When a cycle is detected, a transaction in cycle is aborted to break deadlock.

Performance

1) $m(n-1)/2$ exchange of msg:
   $m \rightarrow$ no. of processes
   $n \rightarrow$ no. of sites
2) Delay $\rightarrow O(n)$

Date ___/___/_____ Page No.:

	Centralized control	Distributed control	Hierarchical control
1)	A control site has the resp. to detect Global WFG.	All site have the resp. to detect a GWFG.	Descendent site can detect GWFG.
2)	Have single point of failure	No single point of failure	No single point of failure.
3)	Easy to implement	Difficult to implement	Simple to implement
4)	Completely Centralized Algo, Ho-Ramamoorthy Algo.	Path Pushing Edge-chasing Algo.	Menasce Muntz Ho-Rama-moorthy Algo

Date __/__/____ Page No.: ____

Deadlock Handling Strategies

A) Prevention:

1) Never satisfy → Mutual exclusion
   → Hold & wait
   → No preemption
   ↓ circular wait

2) Methods 1) Collective Requests
   2) Ordered Requests
   3) Preemption

B) Avoidance:

A resource is assigned to a process if the state of global system is safe.

C) Detection:

Maintaining WFG and searching the cycles.

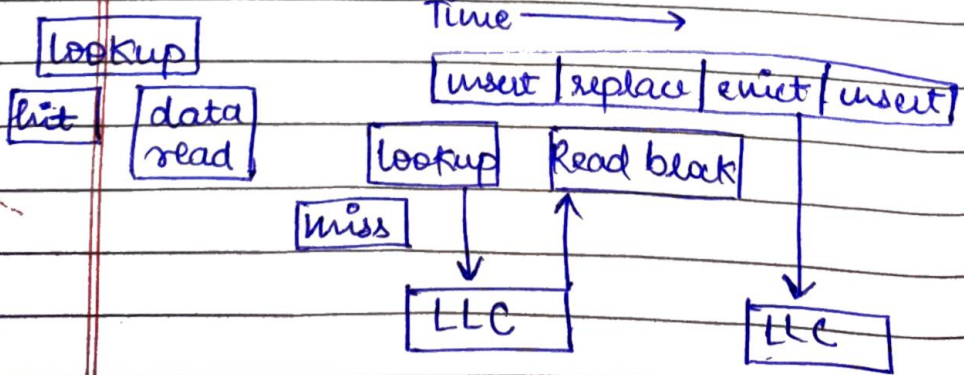
Date __/__/____ Page No.:

Cache

Data storage technique that provide the ability to access data or fill at higher speed.

Read Operation

1) Starts with lookup operation.
2) If cache hit, then cache return values to processor or higher level cache.
3) If cache miss, then cancel read opr & send request to lower level cache.
4) Lower level cache will perform the same seq. of accesses & return entire cache block.
5) Cache controller then extract the requested data from block & send it to processor.
6) Simultaneously, cache controller invokes the insert operation to insert the block into the cache.



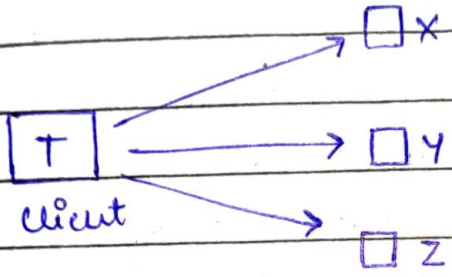
Distributed Transactions

Date __/__/____ Page No.: ____

Flat Transactions

↳ A flat transaction has a single initiating point and a single end point.

↳ for short activities.



Limitations:

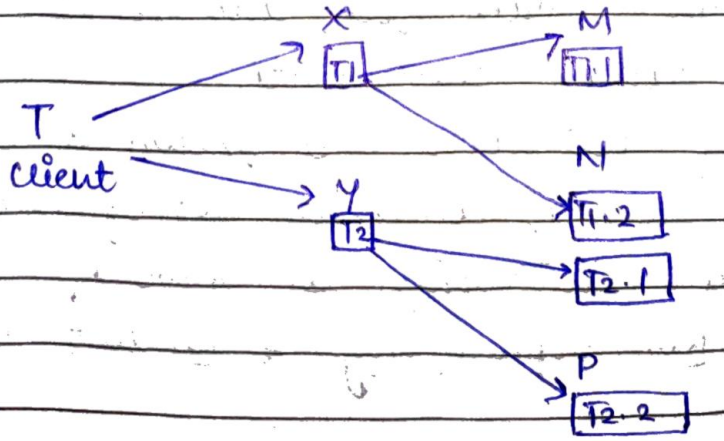
↳ All work is lost in the event of crash

↳ only one DBMS is used at a time

↳ No partial Rollback.

Nested Transactions:

↳ A transaction that includes other transactions within its initiating point & end point



Date __/__/____ Page No.: ____

Checkpointing:

↳ A mechanism that enables transactions to recover from inconsistent state using backward error recovery.

↳ In DS, all the sites save their local ~~sites~~ states which are known as local checkpointings, and the process of saving local state is called local checkpointing.

↳ All the local checkpoints, one from each site, collectively form a global checkpoint.

Strongly consistent set of checkpoints

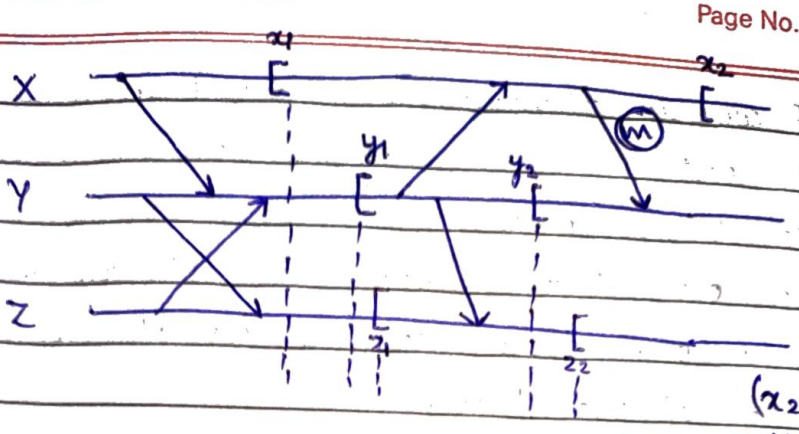
① Domino effect is caused by orphan msgs which themselves are due to rollbacks.

② To overcome domino effect, a set of local checkpoints are needed such that no info. flow takes place b/w any set of processes without saving checkpoints.

③ A set of checkpoints is known as Recovery line or strongly consistent set of checkpoints.

④ A Global checkpoint is SCSoC if there is no orphan msg & no lost msg

Date ___/___/_____ Page No.: _____



(x_1, y_1, z_1)
SCSOC

(x_2, y_2, z_2) not strongly consistent set of checkpoints

Method of taking Consistent set of cp:

① If every process takes a checkpoint after sending every msg, the set of most recent cp are always consistent.

② However it has high overhead, so we can resolve this, by taking cp after every K, K>1

Recovery in DDBS:

1) To enhance performance, availability, a DDBS is replicated whose copies are stored in diff sites.

2) The copies of DB at the failed site may miss some updates whereas sites which are not operational have inconsistent copies of data.

3) Proposed Approach

- Message spooler: Save all the updates directed towards failed sites
- Copier transactions: Read up-to-date copies at all operational sites & update the copies at recovering site. It runs concurrently with user transactions.

"In a gentle way, you can shake the world." —Mahatma Gandhi

Date ___/___/_____ Page No.: _____

4) Recovery algorithms should guarantee that-

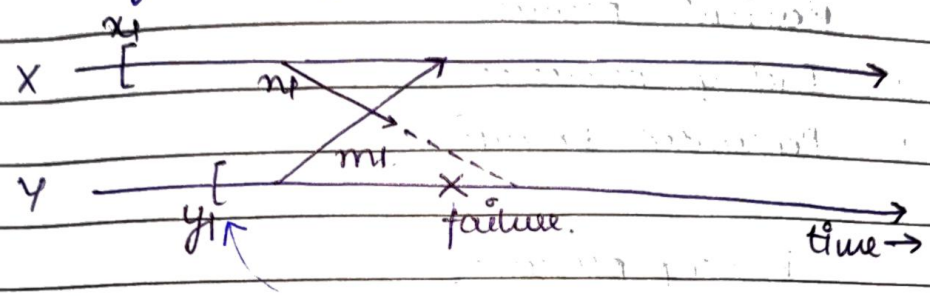
↳ The out of date replicas are not accessible to user transactions.

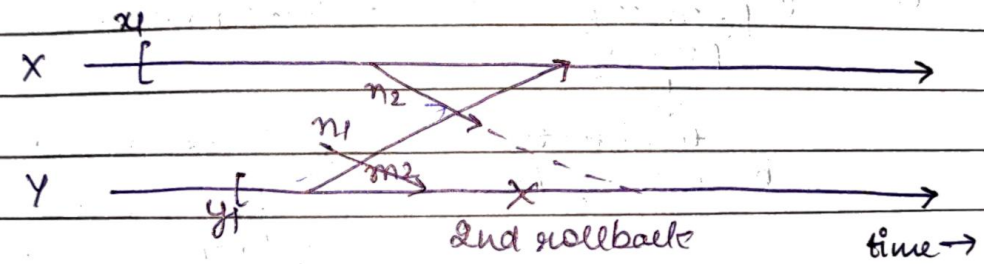
↳ Once outofdate replicas are uptodate by copies transactions, they are also updated along the other copies of user transactions

Date __/__/____ Page No.: _____

Livelocks:

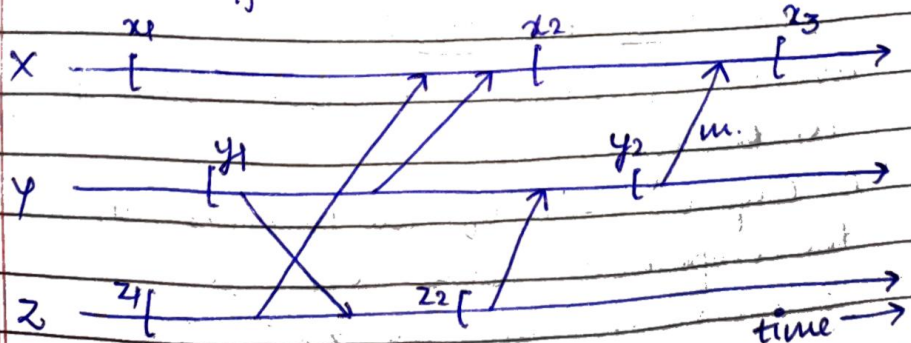
In rollback recovery, livelock is a situation in which single failure can cause an infinite no. of rollbacks preventing the system from making progress





Domino Effect:

When rolling back one process causes one or more other processes to rollback is known as Domino effect.



Failure Resilient Processes:

↳ A process is said to be resilient, if it masks failures and guarantees progress despite a certain no. of system failure.

↳ Approaches proposed to implement Resilient process:

a) Backup processes
b) Replicated Execution

Backup processes:

↳ ① primary process & ⑪ backup processes.

primary process terminates because of failure → backup process becomes active & takes over the job of primary process.

↳ To minimize computation that has to be redone by the backup process, the state of primary process is stored at appropriate intervals.

Replicated Execution:

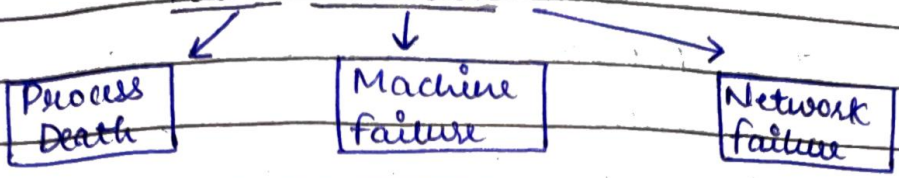
↳ several processes execute same program concurrently.

↳ As long as one of the processes survive failures, the computation/service continues.

↳ Increases reliability & Availability.

Date __/__/____ Page No.: ____

Fault Tolerance



Process Death:

↳ If process has reached a deadlocked state or had terminated abruptly, then all the resources allocated to the process must be recouped

↳ In client-server architecture, if client dies then server must be made aware about it so that all the resources can be released & vice-versa.

Machine failure:

↳ In this case, all processes running in the Machine will die.

Network failure:

↳ A communication link failure can partition a Network into subnet, making it impossible for a machine to communicate with another machine in a different subnet.

Date __/__/____ Page No.: ____

Commit Protocols

These protocols are used to ensure atomicity across sites

A transaction which executes at multiple sites must either be committed at all sites or aborted at all the sites.

2-Phase Commit Protocol

Assumptions:

- One of the cooperating process acts as a coordinator other is referred as cohorts.
- Stable storage is available at each site
- each site uses write-ahead protocol to achieve local fault recovery & rollback.
- At the beginning of transaction, the coordinator reads start transaction msg to every cohort.

PHASE 1:

- At coordinator
  a) coordinator —COMMITREQUEST→ cohorts
  b) wait for reply.
- At cohorts
  a) If transaction executing at cohort is successful, it writes UNDO & REDO log on stable storage & sends AGREED msg to coordinator else send ABORT msg to the coordinator.

Date __/__/____ Page No.: ____

PHASE 2:

↳ at coordinator:

a) If all cohorts agree & coordinator agree, then coordinator writes a COMMIT record into the log. & sends COMMIT msg to all the cohorts else ABORT msg.

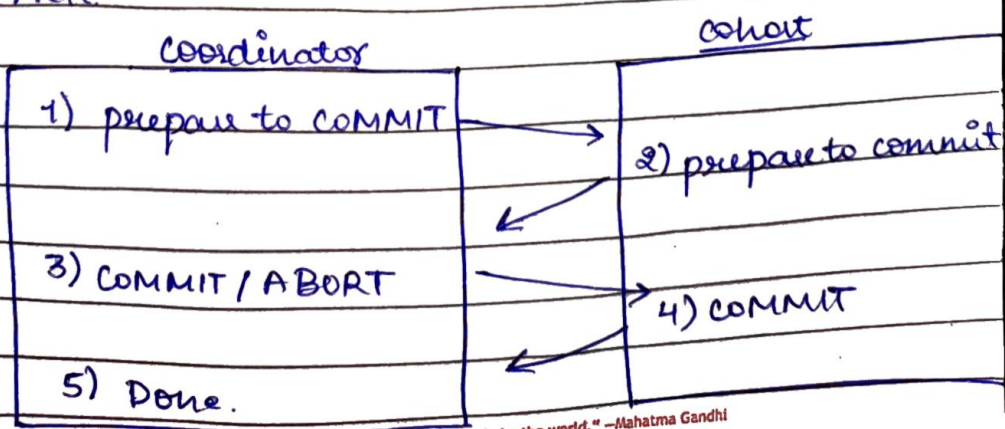
b) Waits for ACK from each cohort, If ACK is not received from any cohort within timeout, coordinator resends the COMMIT/ ABORT msg to that cohort.

c) If all ACK's are received then coordinator writes a COMPLETE record to the log.

↳ at cohorts:

a) On receiving COMMIT msg, a cohort releases all the resources & locks held by it for executing transactions & send an ACK.

b) On receiving ABORT msg, cohort undoes the transaction using UNDO log record, releases all the resources & locks held & sends an ACK.



Date __/__/____ Page No.: ____

Voting Algorithm

↳ More fault Tolerant

↳ Allow data access under ① Network partition
② Site failure
③ Msg Losses

↳ Every replica is assigned certain no. of votes
This info is stored in Stable Storage.

↳ R/W ops is permission when a certain no. of votes are collected by the req. process.

Assumptions

① each file access require that an appropriate lock ~~access~~ is obtained (R/W Lock)
② each Site has Lock Manager
③ each file has a version no. = no. of time it has been updated.
④ each replica has certain no. of votes.
⑤ vote allocation → stable storage.
⑥ R/W opr requires a quorum

Date __/__/____ Page No.: ____

Static Voting Algo

Requesting set; $R = \{S_1, S_2 \ldots S_n\}$

1) S_i send LOCK_REQ to its local lock manager.
2) If request granted, S_i sends VOTE_REQ to all sites in R.

3) S_j send LOCK_REQ, if granted then it return Version No. & No. of votes assigned to replica.

4) S_i decides whether it has quorum or not.
   - ↳ If S_i doesnot have quorum, issue RELEASE_LOCK
   - ↳ If S_i is successful, check if its copy of file is current
   - ↳ If request is read, read local copy
   - ↳ If request is write, update copy & Version No. and send RELEASE_LOCK request

Dynamic Voting Protocols:

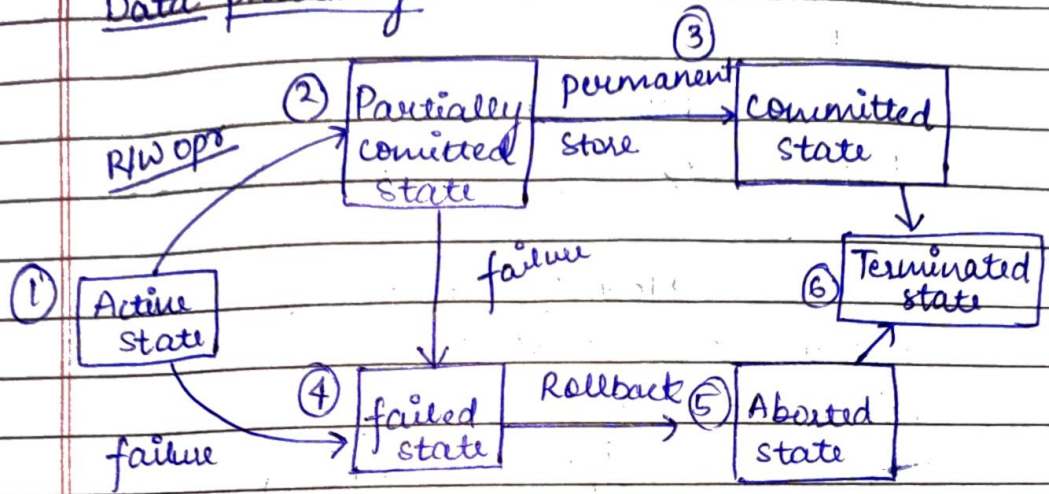
- ↳ change set of sites that can form majority
- ↳ change distribution of votes

Date __/__/____ Page No.: ____

Static Mote Protocol	Dynamic Mote protocol
1) Non-adaptive in nature	1) Adaptive.
2) can cause comm. failure	2) prevents comm. failure
3) may / may not ensure availability	3) ensure availability.
4) Selection time of replica doesn't depend on system state.	4) Selection time of replica depends on system state.

Date __/__/____ Page No.: ____

Transactions : A program including collection of DB opr, executed as a logical unit of Data processing



Properties of Transactions (ACID)

1) **Atomicity**:
   - ↳ either transaction executes completely or it doesn't occurs at all.
   - ↳ Responsibility of Transaction Control Manager

2) **Consistency**:
   - ↳ ensures integrity constraints are maintained, system remains consistent before & after the transaction.
   - ↳ Responsibility of DBMS & Application programmer

3) **Isolation**:
   - ↳ ensure multiple transactions can occur simultaneously without causing inconsistency, each transaction exe. is isolated
   - ↳ responsibity of Concurrency Control Manager

Date __/__/____ Page No.: ____

4) **Durability:**

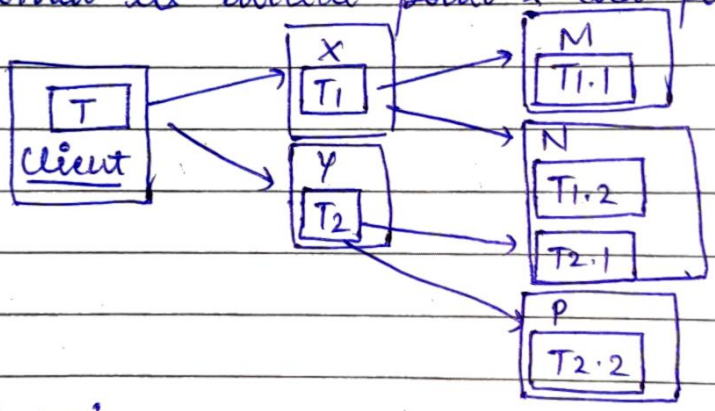
↳ ensures all the changes made by transaction after successful execution are written successfully in disk.

↳ Changes never lost in case of failure.

↳ Responsibility of Recovery Manager

Nested Transactions

↳ A transaction that includes other transaction within its initial point & end point



Rules:

(1) When a parent transaction commits then all the sub transactions that have provisionally commit can commit.

(2) When a parent aborts, all of its subtransactions are aborted.

(3) When a child completes, it makes an idependent decision either to commit psovisionally or abort.

(4) When a child abort, parent can decide whether to abort or not.

Date __/__/____ Page No.: ____

Locks : Variable associated with shared resource that ensures serializability, by providing access to a data item in mutually exclusive manner

Types of locks :

1) Binary locks
   ↳ locked (1)
   ↳ unlocked (0)

2) Shared / Exclusive lock

(i) Shared : If Ti locks X in shared mode then before Ti unlock X, no other transaction Tj can write into X, only Tj can read.

(ii) Exclusive : If Ti lock X in exclusive mode then before Ti unlocks X, no other transaction Tj can read or write into X.

Strict → 2PL → No shrinking phase
→ Release all the locks when whole transaction commit.

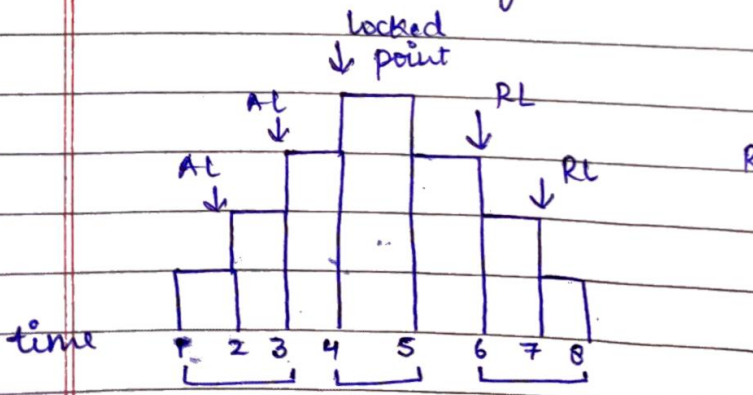
Date __/__/____ Page No.: ____

Two Phase locking Protocol

2PL is a method of concurrency control in DS that ensures serializability by applying lock to transaction data which blocks others trans. to access same data simultaneously.

1) Growing Phase:
- ↳ When transaction begins to execute, it requires permission of locks on DI it needs.
- ↳ In this phase, transaction may obtain any locks but maynot release any lock.

2) Shrinking phase:
- ↳ After executing transaction ops, it releases all the acquired locks on DI
- ↳ In this phase, transaction can only release locks but not obtain any locks



Date __/__/____ Page No.: _____

Timestamp Ordering Protocol

TS(Ti) Timestamp : unique identifier created by DS to identify a transaction

for two transaction, Ti & Tj

If TS(Ti) < TS(Tj)

⇒ Ti started before Tj

With every shared data item, 2 timestamps are associated.

(i) R_TS(X) → largest TS value that execute successful read operation Read(X)

(ii) W_TS(X) → largest Ts value that execute successful Write operation Write(X)

Conditions :

1) Ti issues Read(X) opr.
   - ↳ If W_TS(X) > TS(Ti) then opr is rejected
   - ↳ If W_TS(X) ≤ TS(Ti) then opr is executed.
   - ↳ timestamps of all data items updated.

2) Ti issues Write(X) opr.
   - ↳ If TS(Ti) < R_TS(X) then opr is rejected
   - ↳ If TS(Ti) < W_TS(X) then opr is rejected & Ti is rolled back.
   - ↳ else operation is executed.

Validation-Based Protocol (Optimistic Concurrency Control Technique)

In VBP, transaction is executed in 3 phases.

1) Read Phase:
↳ Transaction T is read & executed
↳ It is used to read the value of various data items and store them in temp variable. It can perform all write operations on temp variable without update of actual database.

2) Validation Phase:
In this phase, temp variable is validated against the actual data to see if it violates the serializability.

3) Write Phase:
If validation of transaction is validated, then the temporary results are written in db/system otherwise transaction is Rolled back.

Each Phase has the following Timestamps:
(i) Start(T_i): time when T_i starts its execution
(ii) Validation(T_i): time when T_i finishes its read phase and starts validation phase.
(iii) finish(T_i): time when T_i finishes its write phase.

Date __/__/____

Page No.: _____

Comparison of Methods of Concurrency Control

Locking	Timestamp	OCC
The locking schemes are used to restrict the availability of DI for other transactions	It works on checking of TS TS is the start time of transaction generated by logical clock.	All transactions are allowed to proceed but some are aborted when they attempt to commit

Date ___/___/_____ Page No.: _____

Concurrency Control in Distributed Transaction

DBS contains 3 Modules

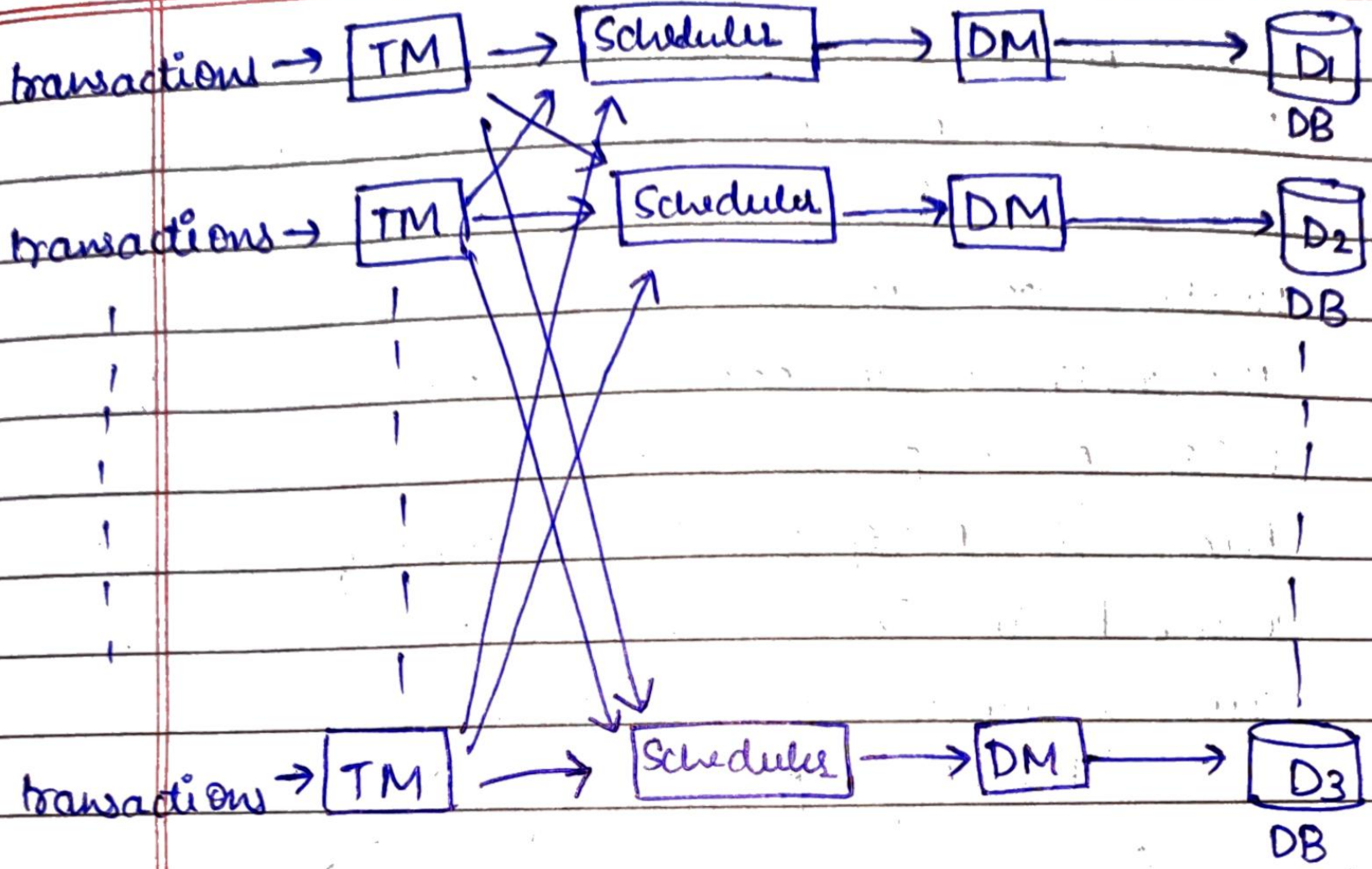
① Transaction Manager ② Data Manager ③ Scheduler

1) Transaction Manager:
- ↳ supervise the execution of transaction
- ↳ Interact with DM to carry out the eot.
- ↳ It assign a TS to a transaction or issue request to lock / unlock data obj. on behalf of user.
- ↳ Acts as interface b/w user & DBS.

2) Scheduler:
- ↳ used to enforce concurrency control.
- ↳ It grant / release locks on data objects as requested by a transaction.

3) Data Manager:
- ↳ Manages the DB.
- ↳ It carries out R/W request issued by TM on behalf of a transaction by operating them on DB.
- ↳ DM is interface between schedules & DB.
- ↳ Execution of transaction at TM results in execution of its actions at the DM.

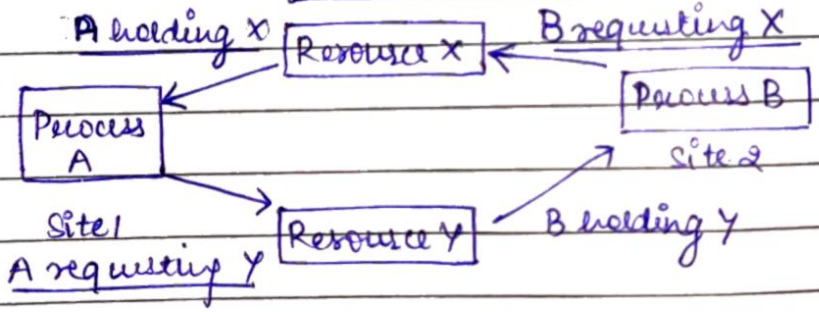
Date __/__/____ Page No.: _____


transactions
TM
Scheduler
DM
D1
DB
transactions
TM
Scheduler
DM
D2
DB
transactions
TM
Scheduler
DM
D3
DB

Distributed Deadlock

DS consists of a no. of sites connected by a network each site with globally unique identifies, makes resource req. to a controller
The controller at each site could maintain WFG on a process request.
Each site WFG could be cycle free & yet DS could be in Deadlock.
This is called Global Deadlock.

eg:



Transaction Recovery : REPLICATION

↳ Replication is the maintenance of copies of data multiple sites.

↳ Enhance performce, Higer Availability Fault Tolerance, Increase Reliability less Access time, Easy load Balancing

Problems : Data Inconsistency

Date __/__/____ Page No.: ____

Group Communication

↳ It occurs when source process sends a msg to a group of processes (Multicast Comm^n).

↳ Groups are useful for managing Replicated data and in other system where processes cooperate towards a common goal by receiving & processing the same set of multicast msgs.

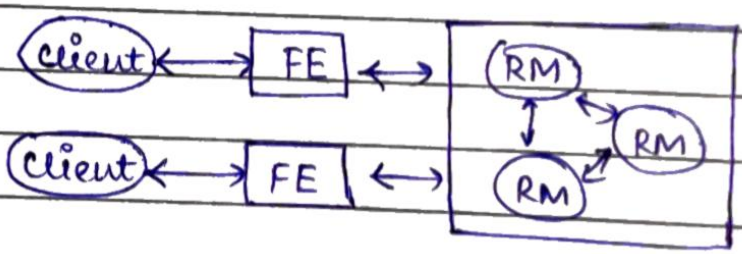
Date ___/___/_____ Page No.: _____

Highly Available service:

Replication technique is used to make service highly available.
i.e. client access the service with reasonable response time.

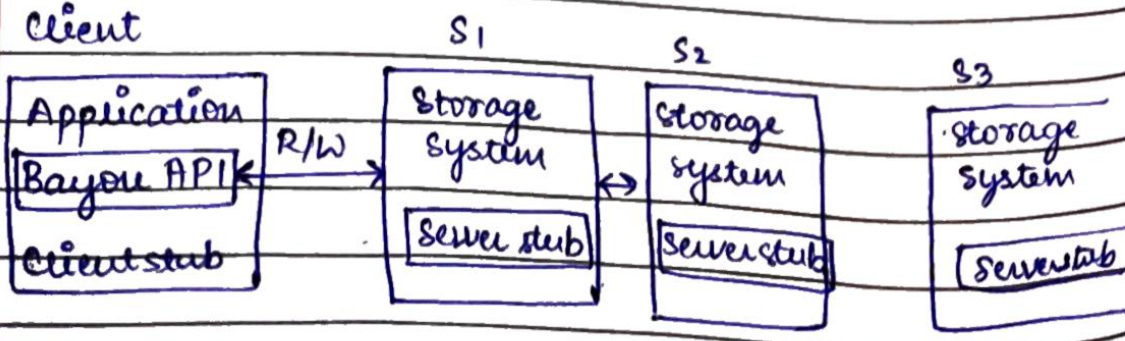
(i) Gossip Architecture:

↳ Client request service operation which are processed initially by front-end.



↳ Replica Managers update one another by exchanging gossip msg which contain the most recent update.

(ii) Bayou's Basic System Model:



↳ It is a replicated, weakly consistent storage designed for Mobile computing envt.

Date __/__/____ Page No.: ____

(iii) Coda file System:

↳ Purpose of CFS is enabling disconnected operations.

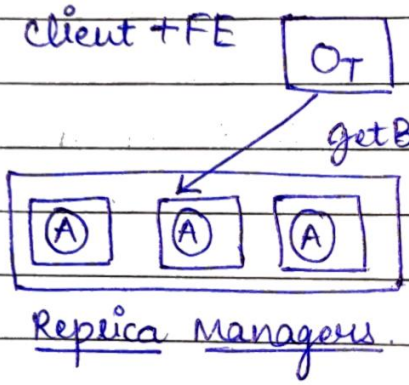
↳ Client communicate over high bandwidth network with server

↳ Coda use optimistic replical control for high availability

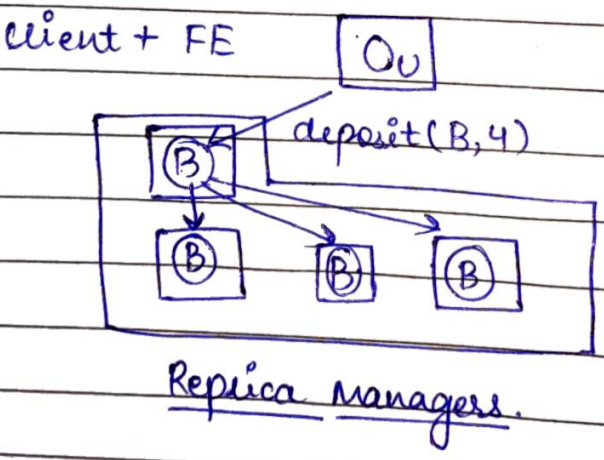
Date ___/___/_____ Page No.:

Transaction with Replicated Server

(i) Read operation :



(ii) Write Operation :



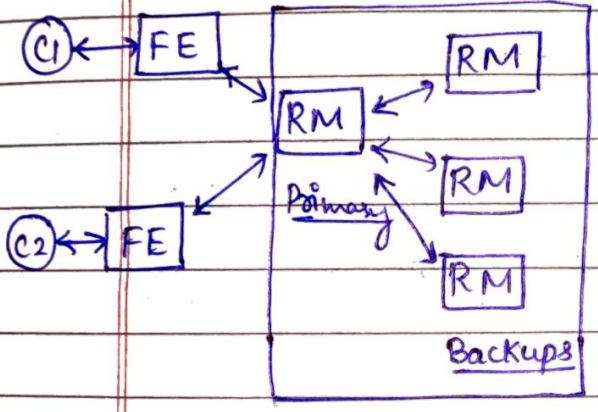
Date ___/___/_____ Page No.: _____

Fault Tolerant services

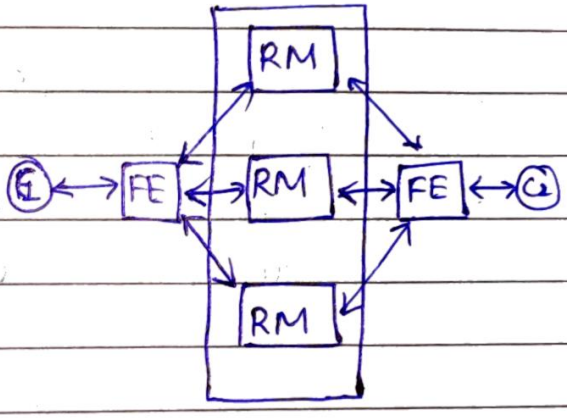
↳ FTS are obtained by using replication.

↳ By using Multiple Independent Server replica Managers each managing replicated data.

Passive Replication



Active Replication



RM acts as state Machines

Sequence of Events in Both Replications

unique identifier

1. Request
2. Coordination
3. Execution
4. Agreement
5. Response

Fault is anything that deviates from what we expect (when something unexpected happens in systems.

Errors are manifestation of faults in system and when propagate through system can lead to failure.

Types of faults

1. Transient occur once & goes away
2. Intermittent occurs, disappear occurs again
3. Permanent occurs once & continues until fixed.

Types of failures

1. Method failure
2. System failure
3. Disk failure
4. Media/ Network failure