Ajay Kumar Garg Engineering College, Ghaziabad

Department of CSE/IT

Model Solution- ODD Semester (2017-18)

Sessional Test -2

Course	:	B.Tech
Subject Code	:	NCS-701
Subject Name	:	Distributed System
Names of Faculty Teaching with Signature	:	

Dr. Vikas Goel
Mr. Dharmendra Kumar
Mr. Rahul Sharma
Ms. Pooja Arora

Section - A

A. Attempt all the parts

(1) List out some issues in DFS (Distributed File System)

Some Issues in Distributed file Systems are:

(i) Name & Name Resolution

(ii) Cache on Main memory or disk

(iii) Writing Policy

→ Write through

→ Delayed Writing

→ Delayed writing until file is closed.

(iv) Caching consistency

→ Server Initiated

→ Client Initiated

(v) Availability

(vi) Scalability

(vii) Semantics

(Q) State Byzantine agreement Problem.

Ans In Byzantine agreement problem one of the processor is randomly selected as source processor & broadcasts its value to all other processors. The agreement states that:

Agreement : All non faulty processors must agree on a common value.

Validity : If source processor is non-faulty then the agreed upon common value by all processors is initial value of source processor.

If source processor is faulty then all non faulty processors can agree on any common value.

(3) What are the deadlock handling strategies in distributed system?

Ans The three deadlock handling strategies are:

(i) Deadlock prevention : Either the processor acquires all the resources it needs or preempt any process.

(ii) Deadlock Avoidance : It can be done by checking if system is in global safe state or not.

(iii) Deadlock Detection : Wait for graph is constructed & detected for presence of any cycles.

(4) What do you mean by the memory coherence?

Ans ~~It~~ It refers to maintaing the consistency of shared data variable among various processes. Coherence refers to the dependency of one process to another during the access of shared data item.

Various coherence consistencies are:

(a) Processor consistency
(b) General consistency
(c) Weak consistency
(d) Release consistency

(5) Write down the performance metrics for distributed exclusion algorithms.

Ans

Algorithm	Response time	Synchronization Delay	Messages (Low load)	Messages (High load)
Non Token Based				
(i) Lamport's Algo	$2T+E$	T	$3(N-1)$	$3(N-1)$
(ii) Ricart-Agarwala	$2T+E$	T	$2(N-1)$	$2(N-1)$
(iii) Maekawa's Algo	$2T+E$	$2T$	$3\sqrt{N}$	$5\sqrt{N}$
Token Based				
(i) Suzuki-Kasami Algo	$2T+E$	T	N	N
(ii) Singhal Heuristic	$2T+E$	T	N	$N/2$
(iii) Raymond Tree Algo	$T(\log N)+E$	$T(\log N)/2$	$\log N$	4

Section - B

B. Attempt all the parts.

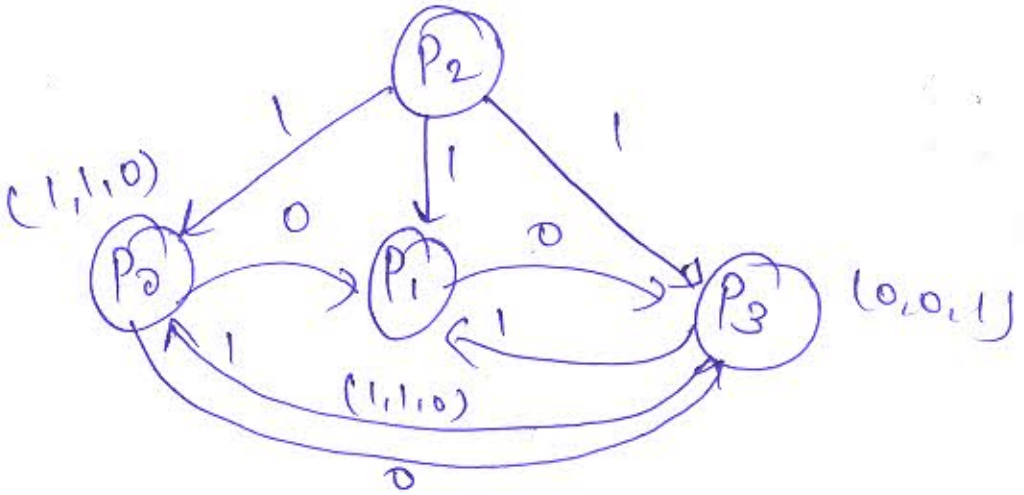
6 Show that Byzantine agreement cannot always be reached among four Processors if two processors are faulty.

<u>condition1</u>: All non faulty processors must agree on common value.

<u>condition2</u>: If source processor is non faulty then agreed upon value by all non faulty processors is initial value of source.

Now suppose we have four processors P_0, P_1, P_2, P_3. Out of these 4 suppose P_0 & P_1 is faulty

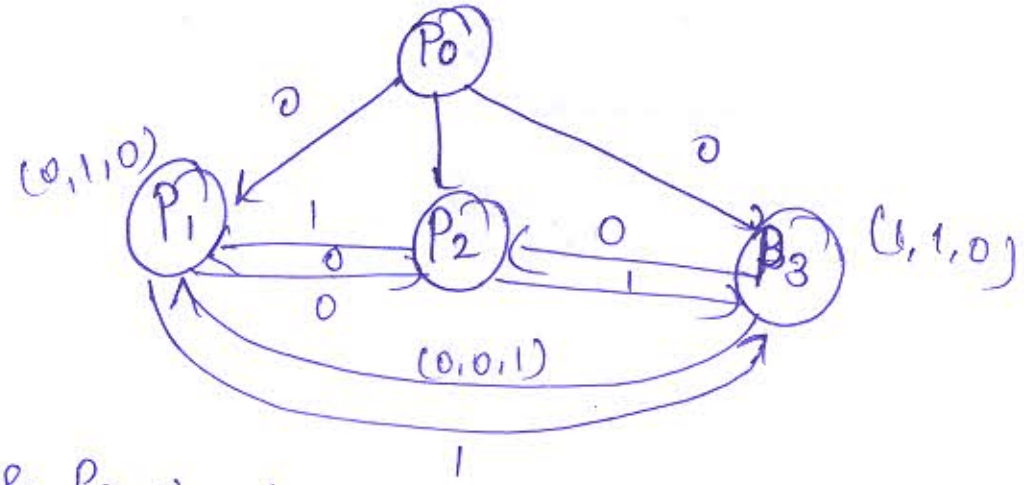
Case:1 → When source processor is non-faulty



Since P_2 is source processor, it has broadcasted 1 to all processors. Now P_0 & P_1 are faulty so they send conflicting values to other processor.

Since P_3 is only non faulty processor whose value is 0 (majority of (0,0,1)) and condition 2 is violated. Hence in this case byzantine agreement fails.

<u>Case 2:</u> When source processor is faulty



Since P_0 is faulty hence it has sent conflicting values to other processors. Now P_1 is also faulty so it has sent conflicting values to P_2 & P_3. Now the final values of P_2 & P_3 are:

Final value of P_2 = (0,0,1) = 0

" " " P_3 = (1,1,0) = 1

∵ Non faulty processors are agreed on different values.

condition 1 is violated.

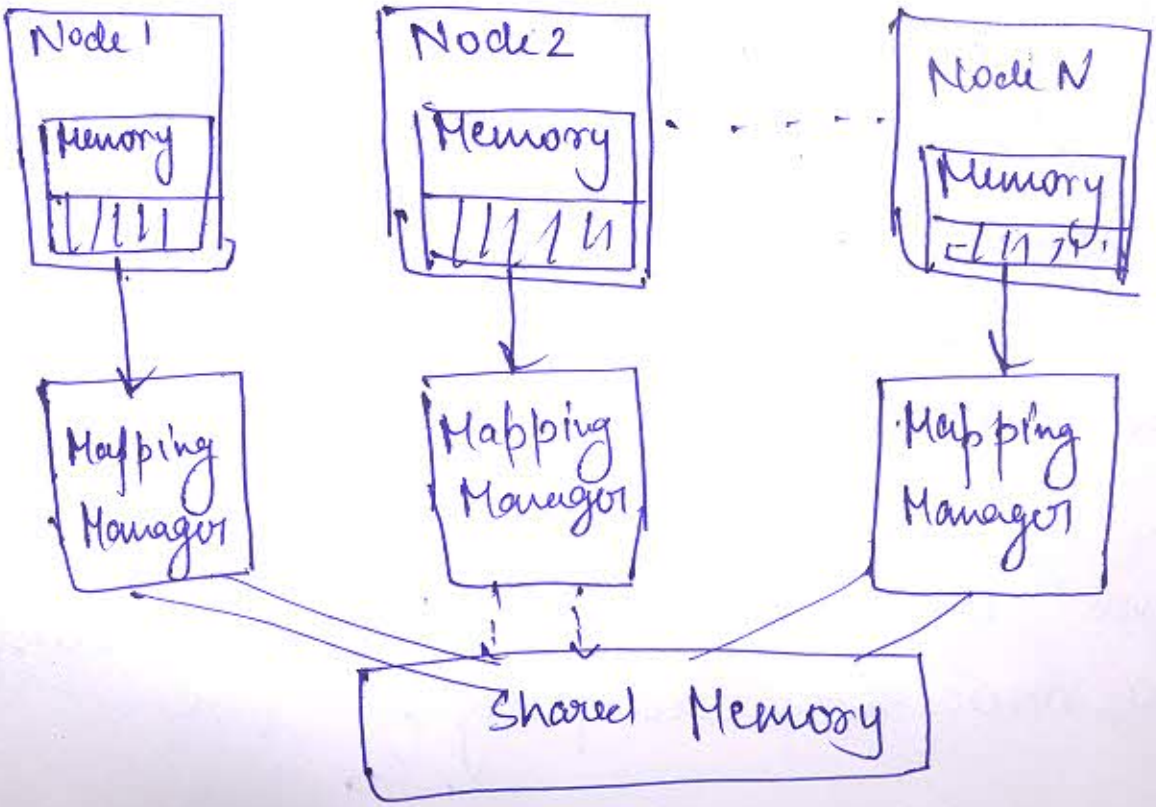
Hence we can say that Byzantine Agreement cannot be reached among four processors if two processors are faulty.

7 Discuss the architecture of distributed shared memory and its advantage.

Ans Architecture of Distributed Shared Memory

Distributed Shared Memory (DSM) is a resource management component of distributed operating systems that implement shared memory model without presence of any physically shared Memory.

Architecture of DSM: In DSM, data can be moved from main memory to secondary memory or between main memories at different nodes. Each node own a data object & when data object is moved to different location ownership changes.



There is a mapping manager in the system that maps the memory of nodes to the shared memory. In shared memory as there are no physical memory, virtual memory comes into existence.

Advantage of DSM

(1) It is easy to write & design parallel algorithms for DSM as compared to explicit message passing method.

(2) The complex structure can be passed by reference and hence DSM takes advantage of locality of reference & also makes it easy to develop algorithms for distributed application.

(3) In DSM, instead of specific piece the entire block is moved from one place to another that reduces delays & latencies.

(4) DSM is much cheaper than tightly coupled multiprocessor system.

(5) The programs written for multiprocessor system can run on DSM with little or no modification.

(6) DSM is fast as compared to multiprocessor system.

Q Discuss Obermarak's path pushing algorithm.

Ans In path pushing algorithm, the wait for dependencies are propagated in the form of paths. Obermark's Algo implements path pushing algorithm for distributed deadlock detection mechanism.

Obermarck's Algo: The algorithm is developed for Distributed Database System (DDBS). Here processes are referred as transactions. $T_1, T_2, \ldots T_n$. Each transaction consists of several subtransaction that are present at different site. At a time almost one subtransaction execute in the system. Execution passes sequencially from subtransaction to subtransaction. These subtransactions are distributed throughout the system that means for concurrent processing these subtransactions need to be synchronized to avoid deadlock.

Non local portion of the graph can be distinguished by the External (Ex) nodes.

Algorithm: The algorithm constitutes following steps:

(i) The sites wait for deadlock related information from all the other sites present in the system.

(ii) With the received information sites combine them with local transaction wait for graph (TFWG) & constructs global TWFG. If then detects all the cycles in the graph & breaks all those cycles that do not contain Ex nodes.

(iii) All the cycles that contain the Ex node are the potential member of the deadlock. The cycle of the form $Ex \rightarrow T_1 \rightarrow T_2 \rightarrow Ex$ is broken down by the site & sent in the form of string to other sites as $Ex\ T_1\ T_2\ Ex$

<u>Performance</u>: If there are n processors in the system then $n(n-1)$ exchanges are required to detect a deadlock. $O(n)$ is required to break cycle.

<u>Disadvantage</u> :- Disadvantage of Obermarck's Algo is that it detect false deadlocks called as phantom Deadlock.

(9) Differentiate between the following:

(i) Token and Non token based mutual exclusion algorithms.

(ii) Centralized, distributed and hierarchal deadlock detection.

Ans

(i)

Token Based Algo	Non-token Based Algo
(a) To execute the critical section the site must posses a token which is a PRIVLEGE message.	— Two or more sites exchange messages in order to decide who will get a chance to execute CS.
(b) The token based algorithm has better performance than non-taken based in case of number of messages per CS execution.	— Non-token based algorithm has worse performance than token based due to large no. of messages per CS execution.
(c) Response time of token based algorithm improves in case of Raymond tree algo.	— Response time of non-token based algo remains 2T + E throughout.
(d) In token based algorithm, a process can in its critical section as soon as it possesses token	— In this algorithm, a process can be in its critical section as soon as it does not send release message to sites.
(e) Synchronization delay is reduced in token Based Algo	— Synchronization delay is more in this case
eg: Suzuki-Kasami Algo, Singhal Heuristic Raymond Tree Algo	eg: Lamport's Algo, Ricart-Agarwala Algo.

(ii)

Centralized	Distributed	Hierachical
(a) In this, there is a central control site that checks for dead lock.	In this, all the sites equally participate in detecting deadlock	All the sites are arranged in hierarchial manner
(b) In this, all the request & release messages are given to control site & control site decides whom to give the resource.	In this, all the sites in the system checks for a global safe state in a mutual exclusive manner.	In this, the site can detect the deadlock only in its descendents in the hierarchy.
(c) It has the worst performance as central control site becomes a bottleneck.	It has better performance as the failure of one site doesnot affect others,	It combines the best feature of both centralised & distributed algorithms
(d) It is simplest to implement	It is moderate to implement	It is most complex to implement.
(e) It don't detect false deadlock	It can detect false deadlock.	It detect false deadlock.
eg: HO-Ramamoorthy two phase & one phase algorithm	eg: Path pushing, edge chasing Algo	eg: Miller-Munitch algorithm

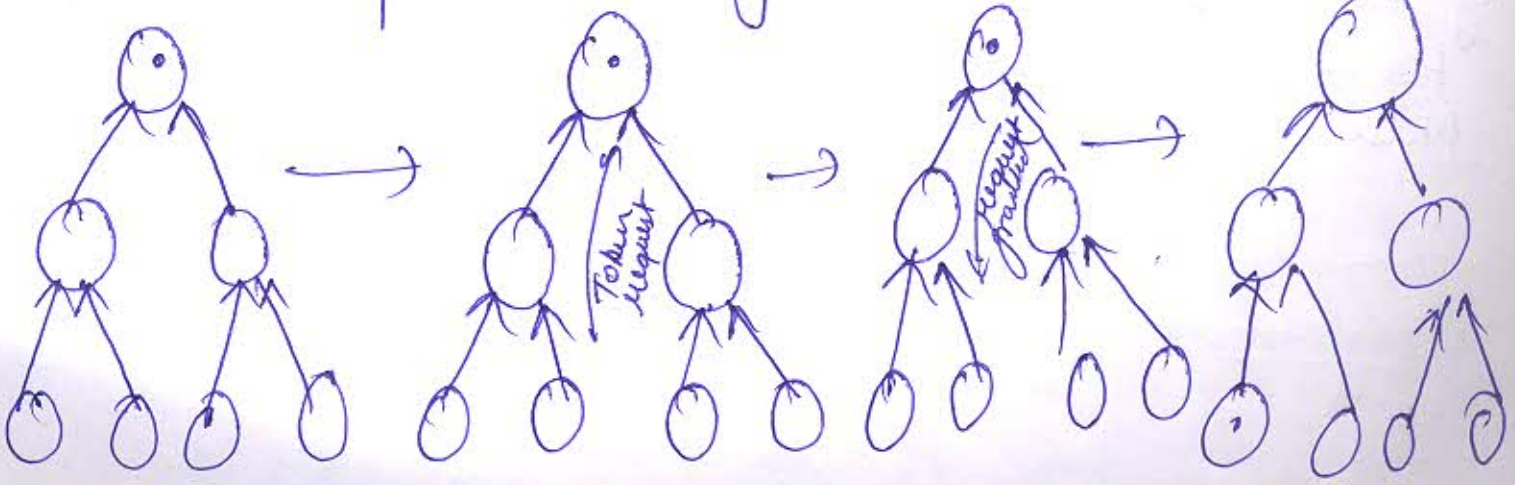
10 Write and explain a token based mutual exclusion algorithm. Describe its performance on important metrics.

Ans A token based algorithm is one in which each site who wishes to enter its CS must possess a token till the time it is executing its CS.

Various token based Algorithms are:

a) Suzuki Kasami Algo
b) Singhal Heuristic Algo
c) Raymond Tree Algo

Raymond Tree Algo:- In this algo all the sites are arranged in a tree manner and edges are directed towards node because node possesses the token. Each node has a holder variable that point to the immediate neighbour of that node. A request q is also maintained at every site which contains the request made by other sites to that site.



Algorithm:

Requesting the CS:

a) Whenever the sites wants to have token, it sends the request message to the directed path towards root provided that it does not contain token & its request-q is empty.

b) When a site receives a request message it forwards the request to the directed path towards node and puts the request in its request-q

c) When the root gets the request message it forwards the request to the requesting site & points holder variable to that site.

d) When the site gets the token, it deletes the top entry from its request q & sends the token to this site and points holder variable to this site.

Executing the C.S : When the site gets the token, it executes its critical section

Releasing the CS:

(a) When the request-q is non empty then the site deletes the top entry from the queue & forwards the token to that site.

(b) When the request-q is empty, it forwards the token to the site pointed by holder variable

Performance on Several Metrics:

(i) Response time: $T(\log N) + E$ where $T(\log N)$ is needed to access the node for tree & E is CS execution Time.

(ii) Synchronisation Delay: $T(\log N)/2$

(iii) Messages in low load: $(\log N)$ because the time complexity to access any node of a tree is $O(\log N)$

(iv) Messages in high load: In high load only four messages will be exchanged, 2 requests messages and 2 messages to transfer the token.

Section-C

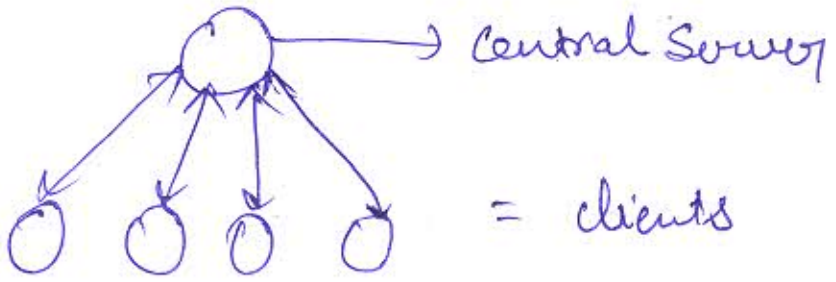
C. Attempt all the parts

11. Describe the various algorithms for implementing DSM (Distributed Shared Memory)

Ans Distributed Shared Memory (DSM) is the resource management component of distributed operating systems which implement shared memory model. Various ~~Sever~~ Algorithm to implement DSM are:-

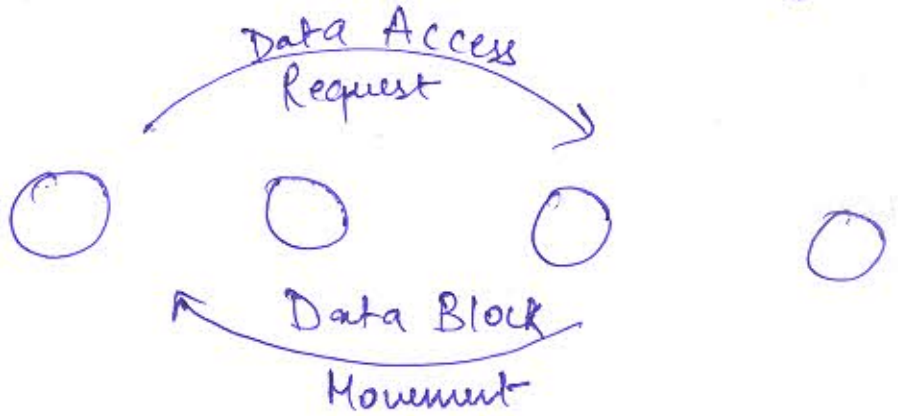
1) Central Server Algorithm
2) Migration Algorithm
3) Read - Replication Algorithm
4) Pull - Replication Algorithm

1) Central Server Algorithm :- In central server algorithm, there is a central server that accepts all read requests from the client & issues them the required data item. It maintains the shared data item. In case of write requests, it is the responsibility of central server to write or update the shared data & send acknowledgements to the clients. Timeout mechanism is used to resend the messages in case of failed transmissions. Duplicate requests can be discarded by the help of sequence no. Each request is assigned a sequence number. If two requests of same sequence no. arrives then one of them is discarded.



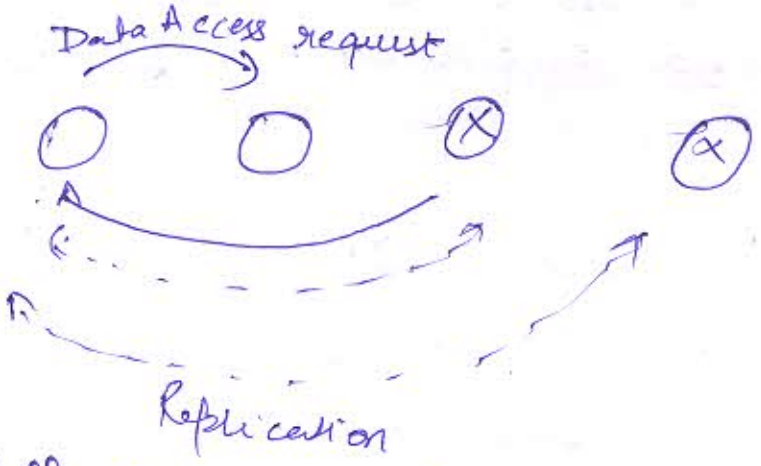
Central Server becomes a bottleneck in case of heavy traffic. Failure of central server will stop the entire system. Also congestion will be increased.

2) Migration Algorithm: In contrast to the central server algorithm where data access requests are forwarded to the data, data in migration algorithm is forwarded to the data access request. In this algo, the whole block containing the data item is moved from one location to another. Hence it takes the advantage of locality of reference where cache is exploited by moving the entire block of data. But the disadvantage is that thrashing occurs in case of large no. of requests. In case of thrashing, the movement of pages from block to block arrives more frequently then servicing the requests.



3) Read Replication Algorithm:- It is the extension of migration algorithm. There multiple read operations are allowed but a single write can be performed. In read replication, the performance of the system is improved due to multiple concurrent read operations.

But write operation is very expensive because in this replicas But write operation is very expensive because in this replicas of data blocks are created and write operation has to be performed on each data block that is a expensive task.



4) Full Replication Algorithm - It can be seen as an extension of read - replication algorithm. Here replicas of data blocks are formed and multiple read as well as multiple write operations can be performed. Hence the performance of system is enormously improved. But due to multiple write operations, the concurrent requests are to be made consistent otherwise system will move to an inconsistent state.

To remove this problem, gap-free sequencer is used. All the modification requests of several clients are given to sequencer. The sequencer assigns a sequence no. to each requests. These requests are executed in order of their sequence no. If there is a gap between sequence no.

no. of modification request & expected sequence no. then we say that one or more modification requests have been missed. Now the clients send the message for the retransmission of the missed requests.

12 Explain various issues that must be addressed in design and implementation of distributed file system.

Ans Distributed File System (DFS) is a resource management component of distributed operating system. The two main goals of DFS are

(i) Network Transparency
(ii) High Availability

Design & Implementation Issues in DFS: Various issues in DFS are:

1) Name & Name resolution
2) Cache on main memory or disk
3) Writing policy
4) Cache consistency
5) Availability
6) Scalability
7) Semantics

(2) Cache on main memory or disk :- Caching is done to reduce the delays in accessing the data but the actual ques is that where should caching be done: main memory or disk

a) Caching on main memory

(+) Diskless clients can take advantage of cache.

(+) Faster

(+) Single design for both at client or server.

(-) Large files cannot be cached.

(-) complex virtual memory management.

b) Caching on disk:

(+) Large files can be cached.

(+) Simple virtual memory management.

(-) Requires local disk space.

3) Writing Policy: When client performs write operation when should it be reflected at the server.

(i) Write through: As soon as client writes then immediately write operation should be performed at the server.

(+) Reliable

(-) Cannot take advantage of locality of reference

1) Name & Name Resolution: Various terminologies:

Name : Unique identifier given to data object

Name Resolution : Mapping a name to data object

Name Space : Collection of names with or without name resolution.

Various techniques for naming are:

(i) Concatenation of name of host to name of files on that host.

(+) Unique name

(+) Resolution is easy

(-) Conflicts with goal of Network transparency

(-) If files are moved to different location, then name has to be changed as well as applications that access it.

(ii) Mount global directories over local directions:

a) Requires that host of directory is known.

b) Files are referenced in location transparent manner (files doesn't reveal its physical location)

(iii) Have a single global directory:

It has the unique name for all the data objects But limited to single computing system

ii). Delayed Writing: The write operation performed at cache is not immediately written at server but it is delayed until multiple blocks have been written at client.

(+) Takes the advantages of locality of reference.

(−) Unreliable

(iii) Delayed writing Until file is closed:

Delayed writing means the write operation at client is not reflected at server till the time the file is opened at server.

In case of small page size, it behaves as above policy

In case of large page size, it is susceptible to loss of data.

(4) Cache Consistency: When multiple clients cache & modify the shared data then problem of cache consistency arises. It can be maintained in two ways:

(i) Server initiated Approach: Server informs the cache manager when the data in cache becomes stale. The cache manager at client side takes necessary step to whether bring new data or invalidate current data.

(ii) Client initiated approach: The cache manager at client side invalidate the data with the server.

5) Availability: What is the level of availability of data in the distributed system. In case of system failure or error, it is the duty of the DFS to make the data available to the client for access.

Replication is done to ensure availability.

6) Scalability: When the system is growing state, then what is the suitability of the system? If the client request increases then the system must not breakdown instead it must be scalable enough to handle several requests.

(7) Semantics: The expected semantics is that the data that is being read is the one that has been last modified. It should not be the case that stale value are being read even through data has been modified.