COMPUTER ENGINEERING DEPARTMENT

ASSIGNMENT NO. 2

Subject: Distributed Computing

COURSE: B.E Year: 2021-2022 Semester: VIII
DEPT: Computer Engineering
SUBJECT CODE: CSC802 DUE DATE: 25/03/2022

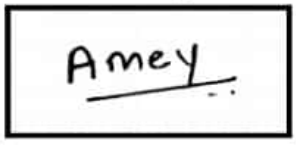
Roll No.: 50 Name: Amey Mahendra Thakur

Class: BE-Comps B Date of Submission: 25/03/2022

DC Assignment - 2

Sr. No.	Questions
1	Compare Task assignment approach, Load Balancing Approach and Load Sharing Approach.
2	Discuss Code Migration in detail.
3	Justify the need for a client consistency model in comparison with the data-centric consistency model and discuss any two client-centric consistency models.
4	Discuss any 4 data-centric consistency models.
5	Describe file caching schemes.
6	Discuss Google case study in detail.

Student Signature:



AMEY THAKUR B-50 Amey.

Q1 Compare Task assignment approach, Load balancing approach and load sharing approach.

Ans:

Task Assignment Approach

- In task assignment, each process is viewed as a collection of tasks.
- These tasks are scheduled to suitable processors to improve performance
- This is not widely used because it requires characteristics of all the processes to be known in advance
- This approach does not take into consideration the dynamically changing state of the system.
- In this approach, a process is considered to be composed of multiple tasks and goal is to find an optimal assignment policy for the tasks of an individual process.
- The typical assumptions are:
  1. Minimize IPC cost
  2. Efficient resource utilization
  3. Quick turnaround time
  4. A high degree of parallelism

AMEY THAKUR B-50 Amey

Load Balancing Approach

- It is a concept that aims to make a network more efficient.
- It distributes the traffic load evenly across a network with multiple-paths in order to get optimal resource utilization, maximize throughput & minimize resource time.
- The reason this term is less preferred than load sharing is because it is difficult to achieve perfect load balancing.
- Load balancing will split the traffic down the configured paths equally towards the destination.
- Because of the different ways, we can achieve load balancing but it can be difficult to achieve true load balancing across each of the paths.
- Eg. With two 768 Kbps links & 800 kbps traffic at any point, conceptually with load balancing each path should have 400 kbps worth of traffic.

AMEY THAKUR B-50 Amey

Load Sharing

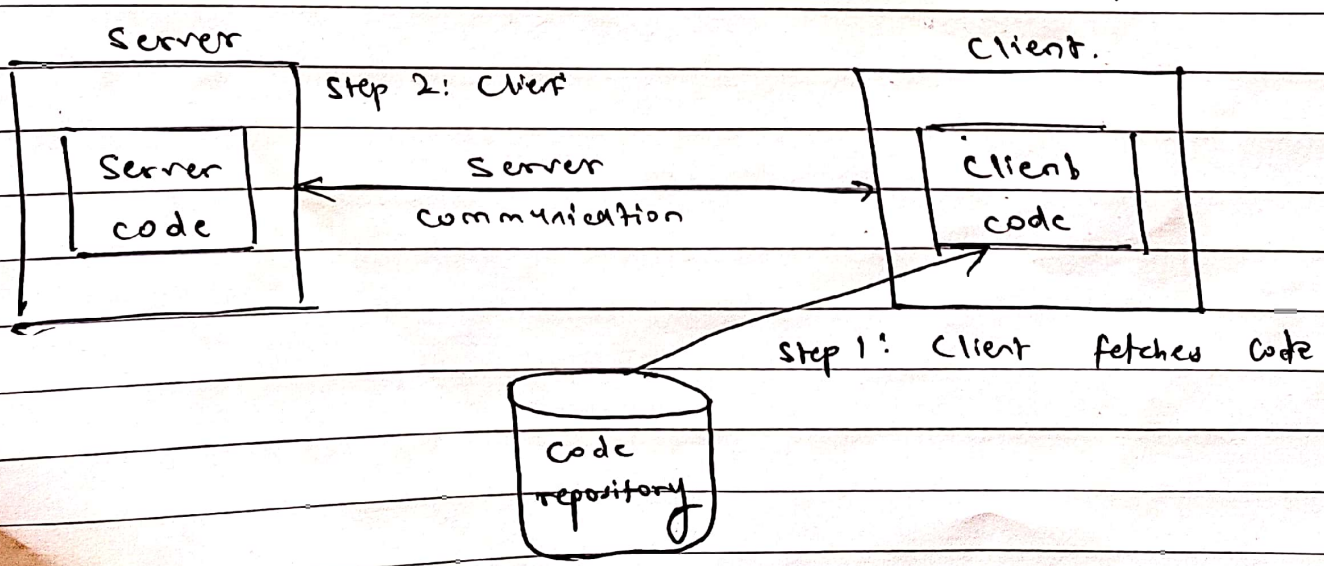
- It means one can split the traffic from a network to be transported by different paths.
- It is inherent to the forwarding process of a router to share the forwarding of traffic, if the routing table has multiple paths to a destination
- If equal paths, the forwarding process will decide the manner of forwarding & forward packets based on the load sharing algorithm used.
- If unequal paths, the traffic is distributed inversely proportionally to the cost of the routers, i.e paths with lower cost are assigned more traffic & vice-versa
- Eg - Lets say link one is 9 mbit/sec & other one is 3 mbit/sec, For every three packets we send through the 9 mbit link, we would want to send one packet down the 3 mbit link.
- The result is that the 9 mbit/sec link would send a higher proportion of traffic than the 3 mbit/s link.

AMEY THAKUR B-50 Amey

Q2. Discuss code migration in detail.

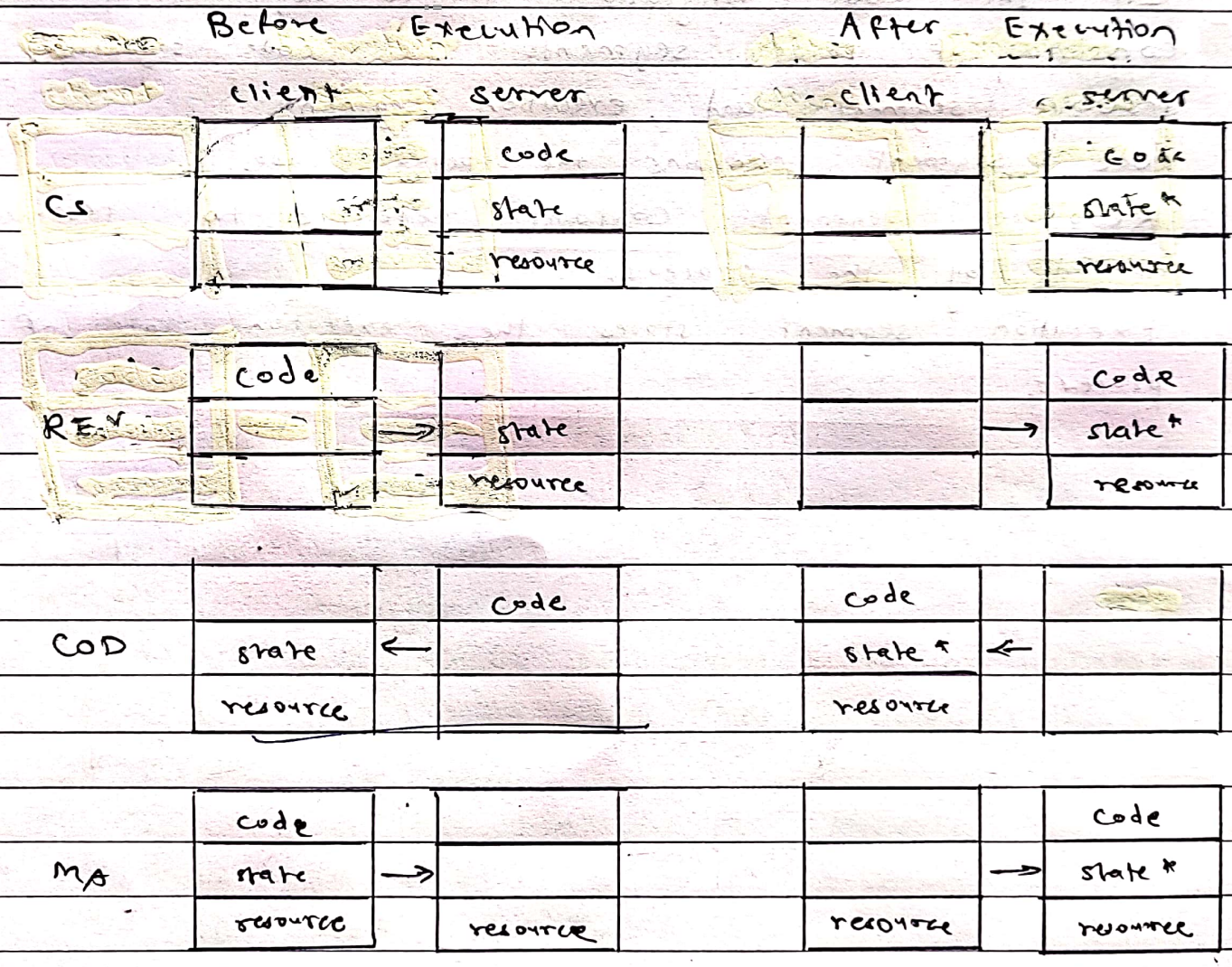
Ans!

- In this process, an entire process has to be moved from one machine to another.
- But this may be a crucial task, but it has a good overall performance.
- In some cases, a code needs to be migrated rather than the process.
- Code migration refers to the transfer of program from one node to another.
- Eg: Consider a client server system in which server has to manage a big database.
- The client application has to perform many database operation.
- In such situation, it is better to move part of the client application to the server & the result is sent across the network & thus code migration is a better option.
- Code migration is used to improve the overall performance of the system by exploiting parallelism.



AMEY THAKUR B-50 Amey

- As shown in figure, the server is responsible to provide the client's implementation, when the client binds to the server.
- Advantage of this approach is the client does not need to install all the required software, software can be moved in as & when necessary & discarded when it is not needed.



CS: client server	① The mobile agent moves from site to site
REV: Remote Evaluation	② COD is used to migrate part of server to client
CoD: Code on Demand	③ REV is used to perform database operations involving large amount of data
MA: Mobile Agents	

AMEY THARUR B-50 Amey

Code Migration Issues:

- Communication in distributed systems is concerned with exchanging data between processes.
- Code migration in the broad sense deals with moving programs between machines with the intention to have those programs to be executed at the target.
- In the code migration framework, a process consists of three segments i.e code segment, resource segment and execution segment.
- Code segment contains the actual code
- Resource segment contains references to resources needed by the process.
- Execution segment stores the execution state of a process.

AMEY THAKUR B-50 Amey

Q3. Justify the need of client consistency model in comparison with data centric consistency model and discuss any two client centric consistency model.

Ans:

Need of client centric consistency model

- It is one of the type of consistency model in distributed system
- System wide consistency is hard, so usually client centric consistency model is more preferable
- It is easier to deal with inconsistencies.
- Using client centric consistency model many inconsistencies can be hidden in cheap way
- It aims at providing a consistent view on a database.

client centric consistency models:

① Eventual consistency

- It is a weak consistency model.
- It lacks simultaneous updates.
- It defines that if update do not occur for long period of time, all replies will gradually become consistent.
- It is a lot inexpensive to implement.
- Requirements are:
  - ① Few read / write conflicts
  - ② No write / write conflicts
  - ③ Clients can accept temporary inconsistency
- Eg. WWW.

AMEY THAKUR B-50 Amey

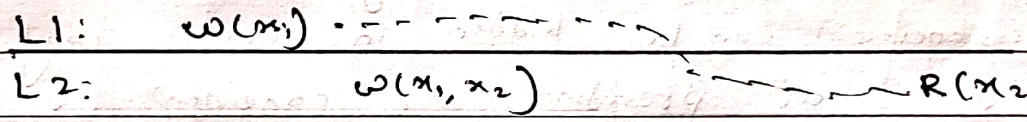
(2) Read Your Writes

- A data store is said to offer read your write consistency if following condition held.

"The effect of a write operation by a process P on a data item x at a location 1 will always be seen by a successive read operation by the same process."

- Example: Updating a web page & guaranteeing that the web browser shows the newest version instead of its cached copy

(a)



(b)

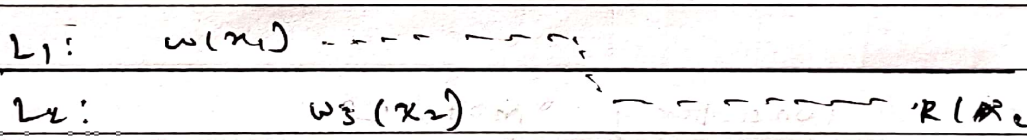


Figure a, demonstrate a data store that delivers read your writes consistency

Figure b, a data store that does not delivers read your writes consistency

AMEY THAKUR B-50 Amey

Q4. Discuss any four data centric consistency model

Ans:

① Strict consistency model

- Any read on a data item x returns a value corresponding to the result of the most recent write on x.
- This is the strongest form of memory coherence which has the most strictest consistency requirement

(a)
P1: W(x)a
P2: R(x)a

(b)
P1: W(x)a
P2: R(x)NIL R(x)a

- Behaviours of two processes operating on the same data item.
  (a) A strictly consistent store
  (b) A store that is not strictly consistent

AMEY THAKUR B-50 Amey

② Sequencial Consistency

- It is an important data centric consistency model which is slightly weaker consistency model then strict consistency.
- A data stored is said to be sequentially consistent if the result of any execution is same as if the (read & write) operations by all processes on the data store were executed in same sequential order & operation of each individual process should appear in this sequence in specified order.

P1 : W(x)a	P1: W(x) a
P2 : W(x) b	P2: W(x)b
P3 : R(x)b R(x) a	P3: R(x)b R(x) a
P4 : R(x) b R(x) a	P4: R(x)a R(x)b
(a) A sequentially consisted data store	(b) A data store that is not sequentially consistent.

AMEY THAKUR B-50 Amey.

③ Linearlizability
- It is weaker than strict consistency but stronger than sequential consistency
- A data store is said to be linearlizable when each operation is timestamped & result of any execution is the same as if the (read & write) operations by all processes on the data store were executed in some sequential order.
- The operations of each individual process appear in sequence order specified by its program
- If ts opr (x) < ts opr (y) then operation opr(x) should preceed opr(y) in this sequence

④ Casual Consistency
- It is weaker model than sequential consistency
- In this, all processes see only those memory reference operations in the correct order that are potentially casually related
- Memory reference operations which are not related may be seen by different processes in different order.
- A memory reference operation is said to be casually related to another memory reference operation if the first operation is influenced by the second operation.
- If a write (w_2) operation is casually related to another write (w_1) - the acceptable order is (w_1, w_2).

AMEY THAKUR B-50 Assign

Q5: Describe file caching schemes.

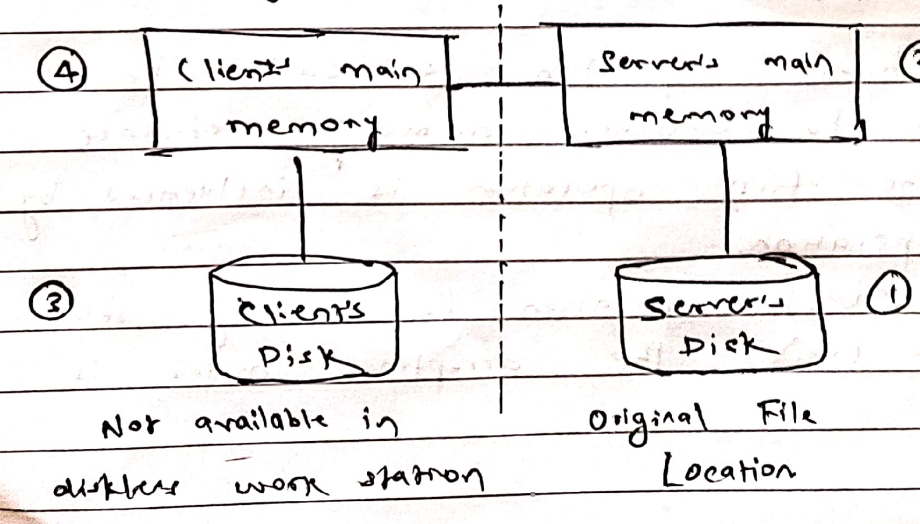
Ans:

File caching schemes

- In order to improve file I/O performance in centralized time sharing system, file caching has been implemented and executed in numerous DFS.
- Main goal of file caching scheme is to retain recently accessed data in main memory
- So that repeated access to the same information can be efficiently handled
- In implementing file caching scheme, one has to make several key decisions like granularity of cached data
- Three design issues are involved in file caching

① Cache location

- Cache location refers to the place where cached data is stored
- There exists three possible cache locations servers main memory, clients disk, clients main memory



① No caching
② Cache located at server's main disk
③ Cache located in client's main disk
④ Cache located in main memory

AMEY THAKUR B-50 Amey

Server's main memory
- In this, file must be first transferred to the servers' main memory & then across the network
- Hence the disk access cost is reduced.

Client's Disk: The cache located in the client's disk eliminates network access cost but requires disk access cost.

Client's main memory: The cache located in clients main memory eliminates both network access cost & disk access cost.

(2) Modification Propogation
- When cache is located on client's node, a file data may simultaneously be cached on multiple nodes
- It is possible for caches to become inconsistent when file data is changed by one of the clients & corresponding data cached at other nodes are not changed or discarded
- A variety of approaches are used to hadle these issues

Write through Scheme:
- When cache entry in modified new value is immediately sent to the server for updating the master copy of the file

AMEY THAKUR B-80 Amey

Delayed - Write Scheme

- In this, when cache entry is modified the new value is written only to the cache.
- The client just makes a note of the cache entry that has been updated.
- All the cache entries are collected together & sent to the server later on.
- It's approaches are as follows:

(a) Write on ejection from cache

- Modified data in the cache is sent to the server when the cache replacement policy has decided to eject it from the client's cache.

(b) Periodic write

- Caches are scanned at regular intervals.
- Cached data is modified right from last scan is sent to the server.

(c) Write on close

- Modifications made to a cached data by a client is sent to the server when the corresponding file is closed by the client.

AMEY THAKUR B-50 Amey

Client initiated Speech

- In this method, the client contacts the server to check if locally cached data is inconsistent with master copy
- File sharing semantics depends on the frequency of the validity check & can be used below approaches:

(a) Check before every access
- main pyrpose is defeated in this because the server has to be contacted on every access

(b) Periodic check
- A check is initialized after fixed interval of time

(c) Check on file open
- Client cache entry is validated only when client opens a corresponding file for use

AMEY THAKUR B-50 Amey

Server initiated approach

- A client informs the file server when it opens a file indicating whether the file is being opened for reading, writing or both.
- The file server keeps a record of which client has which file opened and in what mode.
- Thus, the server keeps a track of the file usage models being used by different clients and reacts in case of inconsistency.
- Inconsistencies occur when more clients try to open a file in conflicting modes.
- Eg. If a file is open for reading other clients may be able to open it to read a new file without any problem but not for writing.
- Similarly, a new client must not be allowed to open a file which is already open for writing.

AMEY THAKUR B-50 Amey

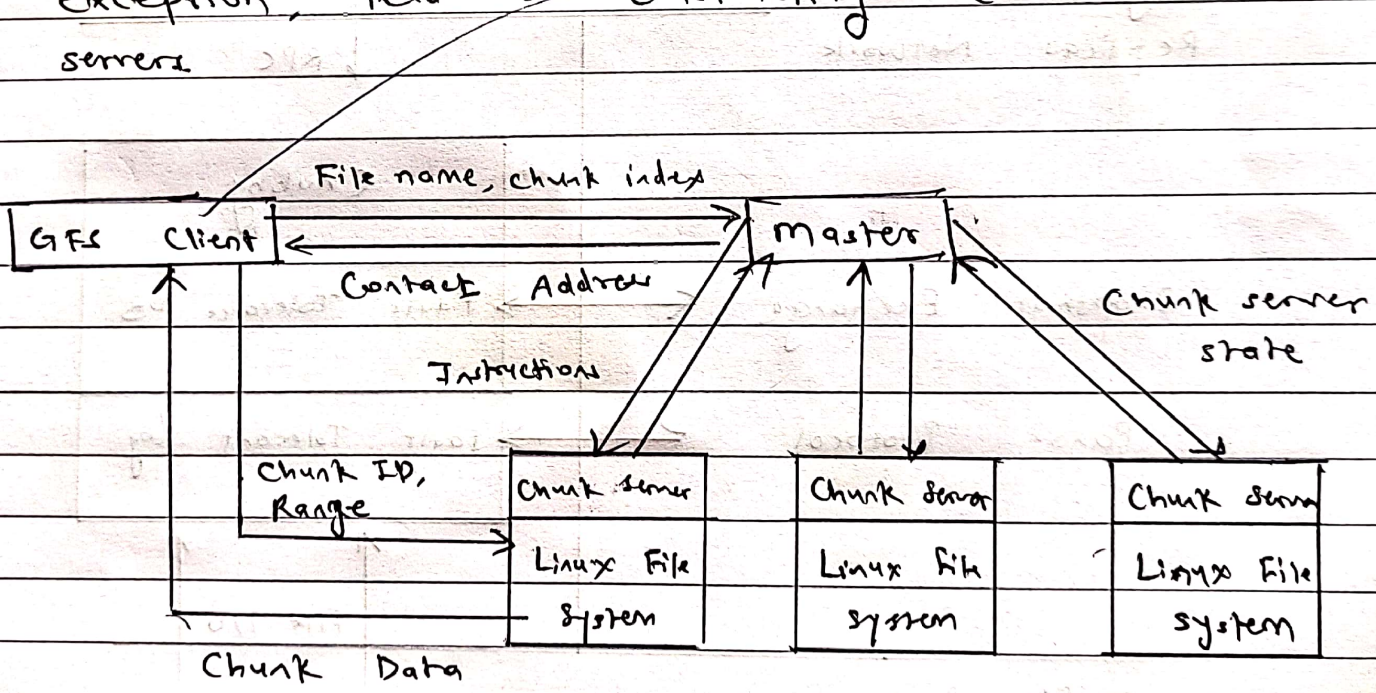
Q6. Discuss Google case study in detail

Ans:

Google case study on Google File system (GFS)

① GFS:

- Google has developed its own file system i.e. GFS
- Google files tend to be very large commonly ranging up to multiple gigabytes where each one contains lots of smaller objects
- Moreover, updates to files usually take place by appending data rather than occuring parts of a file
- These observations along with the fact that server failure are the norm rather than the exception, lead to constructing clusters of servers



- Each cluster has one master + multiple chunk servers.
- Files replicated on by default three chunk servers.
- System manages fixed size chunks 64 mb (very large)
- Over 200 clusters some with up to 5000 machines offering 5 Petabytes of data + 40 Gb/sec throughput.

AMEY THAKUR B-50 Amey

② Chubby Lock Service

- Google has opened / developed chubby, a lock service for loosely coupled distributed systems.
- It is designed for coarse-grained locking & also provides a limited but reliable distributed file system.
- It provides simple file system API.
- It is used for many purposes such as small scale filing system as a name-server to achieve distributed consensus / locking & use of lamports Paxos Algorithm.

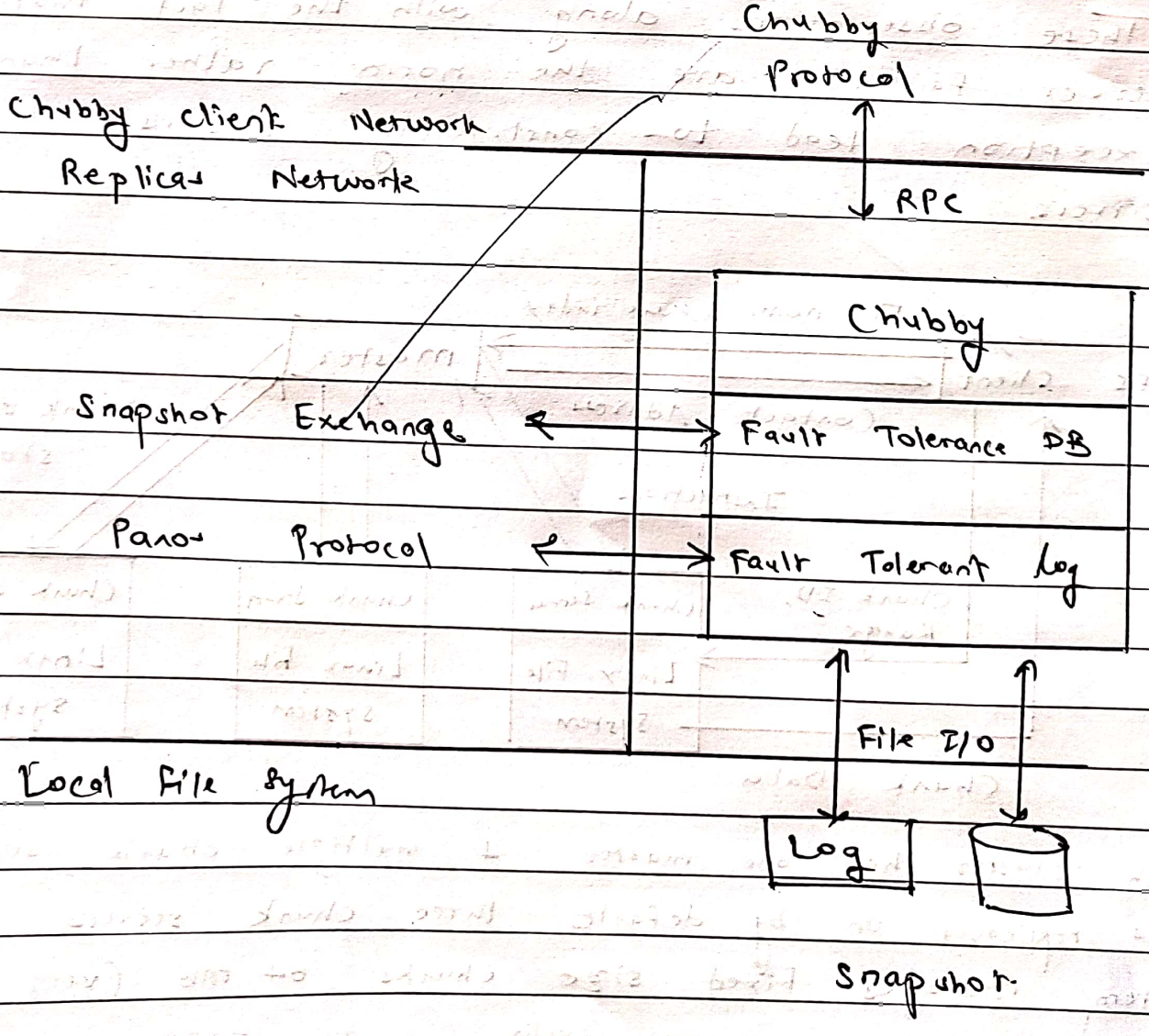


Figure: Chubby Lock Service

AMEY THAKUR B-50 Amey.

③ Big Table :

- It is a very large scale distributed table indexed by row, column & timestamp.
- It supports semi-structured & structured data on the top of GFS.
- It is used by over 60 google products including google earth.

④ map Reduce :

- It is a service offering a simple programming model for large scale computation over very large datasets.
- It hides parallelism, load balancing, optimization, failure / recovery techniques.

BS 25/3/22