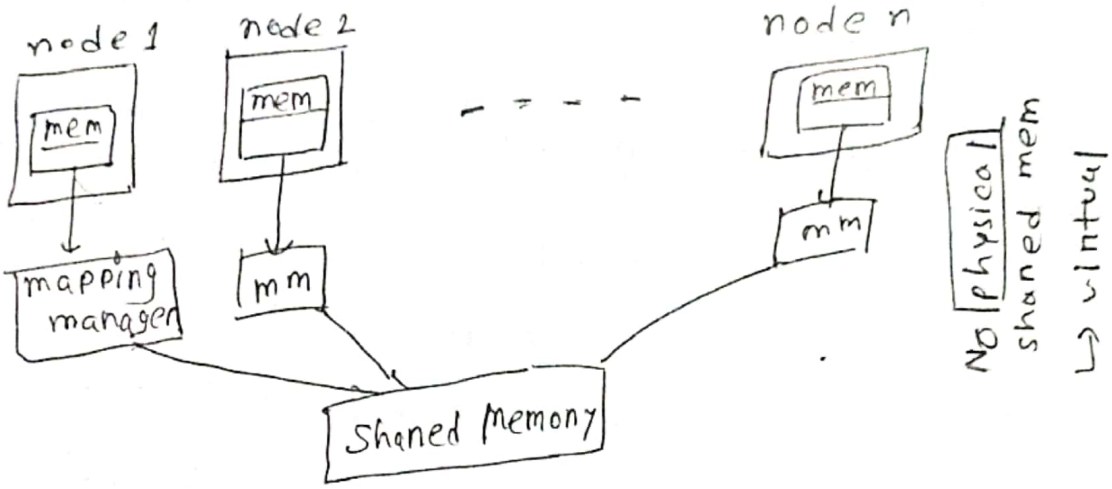
collection of many nodes/devices which are connected through some network and all have local memories

Distributed Shared Memory

→ provides a virtual address space

↓

shared among all nodes in DSM



What is mapping manager?

→ A layer of s/w, perhaps bundled with the OS or as a runtime library routine.

→ When a process accesses data in the shared address space, the mapping manager maps shared memory address to physical memory (local or remote)

Pros:

1) Easy Abstraction: same address space
   → easy data mignation
   → simplen than RPC

2) Easien pontability → common interface

3) Locality of data:
   → data fetched in large blocks
   → need in future

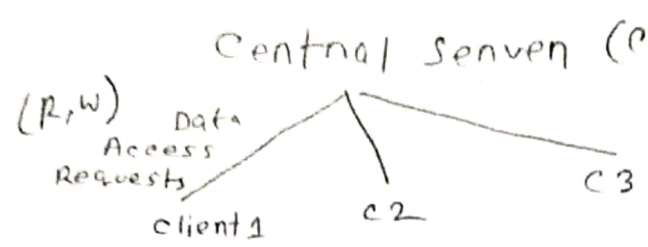
4) Langen memony space
5) Betten penfonmance
6) Flexible communication environment
7) Variables direct shane
8) Less cost

Cons: 1) ~~fast~~ slow Access
2) Not efficient
3) Process can't nun simultaneously
4) common bus → mone traffic
5) communication delay

Types of algo:

1) Central - senven
2) Data mignation
3) Read - neplication
4) Full neplication

1) Central Server Algorithm



→ simple to implement

→ CS maintains all the shared data < read / write

→ CONS: performance not good
→ not reliable

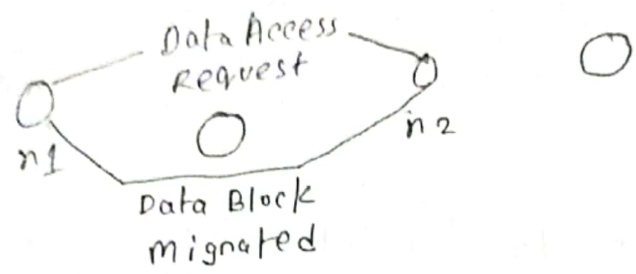
Soln: 1) Use a mapping function to distribute/ locate data

2) partition shared data between several servers.

* if timeout → resend request
* repeated failure → send it
* Detect duplicate write requests
→ with associated sequence number

2) Migration Algo:

→ migrate data elements on request



→ যে block এ data আছে, ঐ block এ request পাঠাইলে data migrate করে - for further access and processing

→ only 1 node can access data at a time

→ whole block is migrated to that node.

3) Read replication Algo:

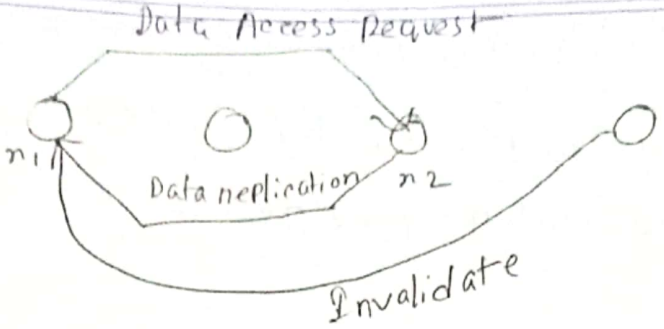
→ replicate data at multiple nodes for read access

→ write:

1) invalidate all copies of shared data at various nodes

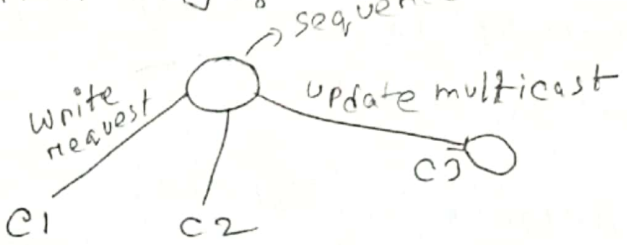
2) update with modified value

→ low read and high write cost



2 imple:
1) IVY
2) PLUS

4) Full-replication algo:



* memory cohenence:
→ most updated value is returned by read operation (expected)
→ control access

* sequential consistency:
→ The result of any execution of operations of all processors is the same as if they were executed in sequential order.

* General consistency:
All copies of mem loc contains same data — after write by CPU complete

* Processon consistency:
operations issued by processon are performed in the onden they're issued

* Weak consistency:
memony is consistent only often a synchronization operation.

* Release consistency:
→ further relaxation of weak consistency
→ acquine and release
(lock) (unlock)

* Coherence protocol?
→ why needed:
• how ensune –
→ all neplicas have same info
→ nodes not access state data (old)

Types of protocols: (2)
1) write-invalidate protocol:
→ invalidates data of all copies except 1 before write

→ can't access invalidated data

Ex: IVY, Clouds, Dash

Pros: - Locality of nefenence
- ~~goo~~ many updates
> good penfonmance in these

Cons: inefficient

2) Write - update protocol:

→ all copies data update

Cons: mone complex

* Cache cohenance protocol: 1) PLUS system
2) munin "