Normalization: process of organizing data in database.

used to remove or reduce redundancy from a relation.

col-row redundancy

Anomaly: 3 type

1) Insert 2) Update 3) Delete

- missing data
- data inconsistency
- inconnect data

Emp id (PK)	Name	Contact No	Dept	Dept mgmt

Soln:

table 1

Emp id (PK)	Name	contact	Deptid (FK)

Dept id (PK)	Name	mgmt

- solve 3 anomaly

1NF: an attribute of a relation can't hold multiple values.

① — No multi-valued attributes.

② — row not same

— attribute domain (type) → same (data type)

— every column → unique name

— order data doesn't matter.

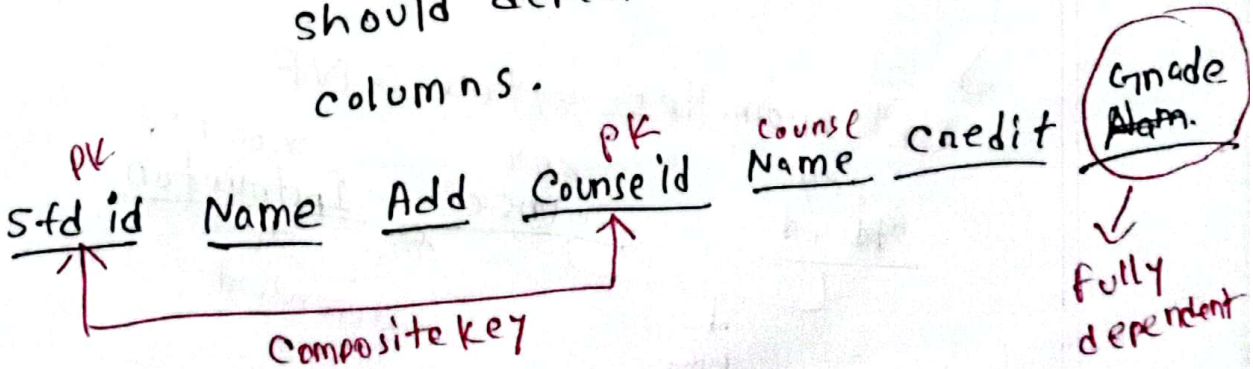
2NF:

1) Follow 1NF
2) Single column primary key can't depend (partially) on any subset of candidate key relation.

partial dependency should be removed.

in case of composite key,

— all columns (non-key column) should depend on each of key columns.



3 NF:

1) 2NF follow
2) Transitive dependency – not allowed

↓

যদি কোনো non-key col অন্য কোনো non-key col এর উপর dependent হয়ে transitive dependency

Counse Name	Teachen id	Iname	Credit
PK key col	non key	non	non

Soln:

PK C Name	Tid	Credit	Table 1

PK Tid	Tname	→ Table 2

3.5
BCNF → Boyce-Codd :

1) 3 NF follow
2) key col, non-key col এর উপর dependent হবে না.

solve ↓

still anomalies after 3 NF

key std id	key Counse	non-key Instructon

↓ PK

dependent

~~NF~~ 4NF:

1) 3.5 NF follow
2) no multi-valued dependency

course	Inst	Supervison
C	Fanhan	S1
	Hossan	S2
DBMS	Mawa	S3
	Suma	

5NF:

1) 4 NF follow
2) table can't be decomposed into any no. of smallen tables without loss of data.

6NF: not standandized

1NF

- no multivalued attri
- unique row

RoH	Course
1	C, C++
2	C, Java

2 NF

- 1NF
- no pantial dependency

A B C

~~com posite~~ Candidate ~~key~~

Pantial
- B → C ✗
- A → C ✗

3NF

- 2NF
- no transitive dependency

monkey monkey dependency ✗

~~key not depend on non-key~~

cName Tid Tname credit

3.5NF

- 3NF
- key not depend on non-key

Stdid Course Instn.

4 NF

- 3.5NF
- no. multivalue dependency

5NF

- 4NF
- lossless decomposition

Centralized DBS

(embedded sys.) single user

multi user (server sys.)

coarse grained parallelism (upto 10 processors)

Fine grained parallelism (large no. of CPU) - data center

---

server system

transaction server
RDBMS

data server
- high performance
- transaction processing

---

client → send request → server

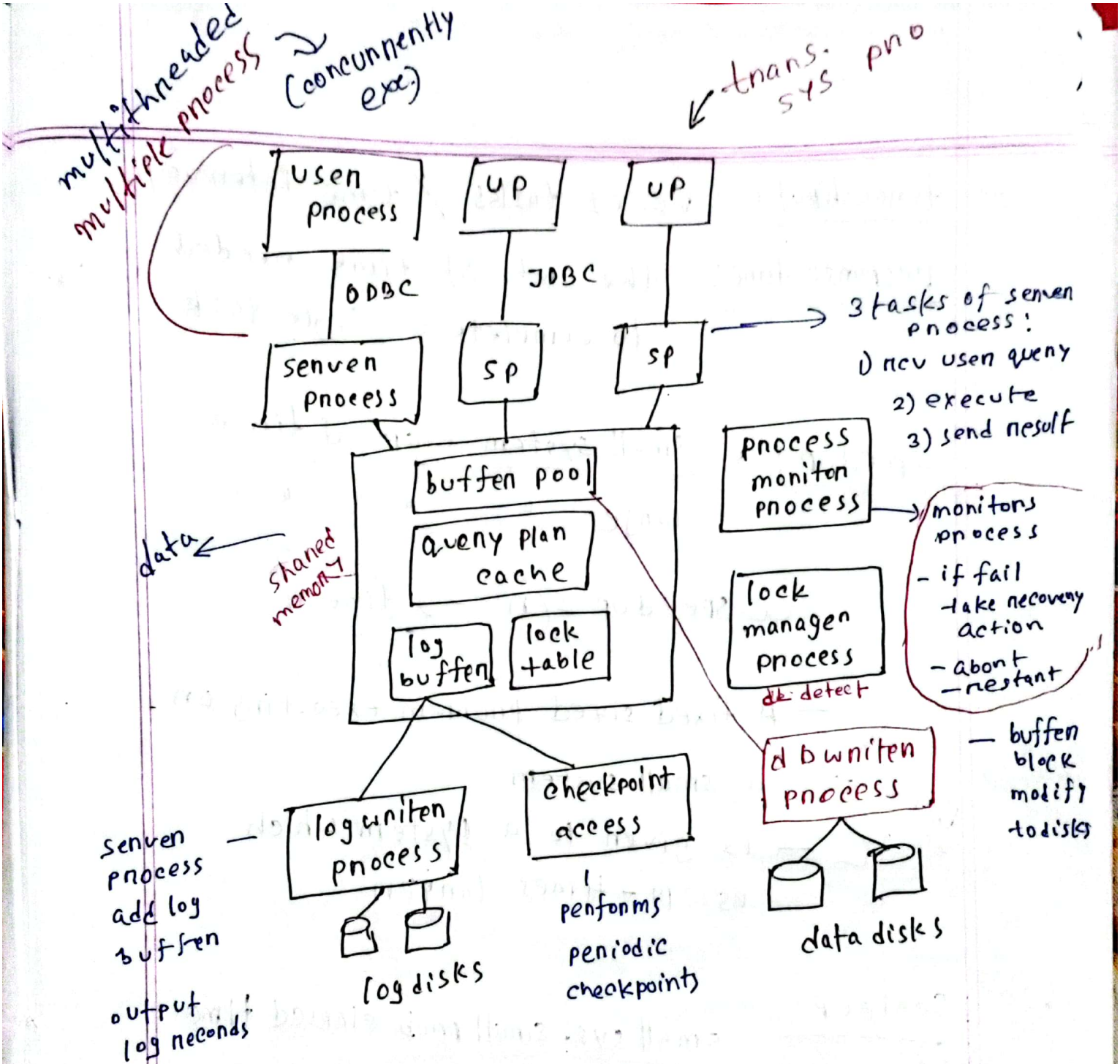
server → transaction execute → client

client ← send result ← server

transaction server → query server SQL

- requests : SQL
- communication mechanism:
  RPC
  - Remote Procedure Call
- APIs : ODBC
  JDBC

trans
- local
- global



A transaction processing system (TPS) is a component of a DBMS that manages and processes transactions. A transaction is a logical unit of work that consists of one or more database operations, such as reading, writing, or modifying data.

4 properties:
1) Atomicity
2) Consistency
3) Isolation
4) Durability

A parallel system refers to a database architecture that utilizes multiple processors or computing resources to process database operations concurrently. It aims to improve system performance by dividing a workload into smaller tasks that can be executed simultaneously on multiple processors or nodes.

throughput: no. of tasks / time interval

response time: the amt. of time needed to complete a single task

speedup: $\frac{\text{small system elapsed time}}{\text{large " " "}}$

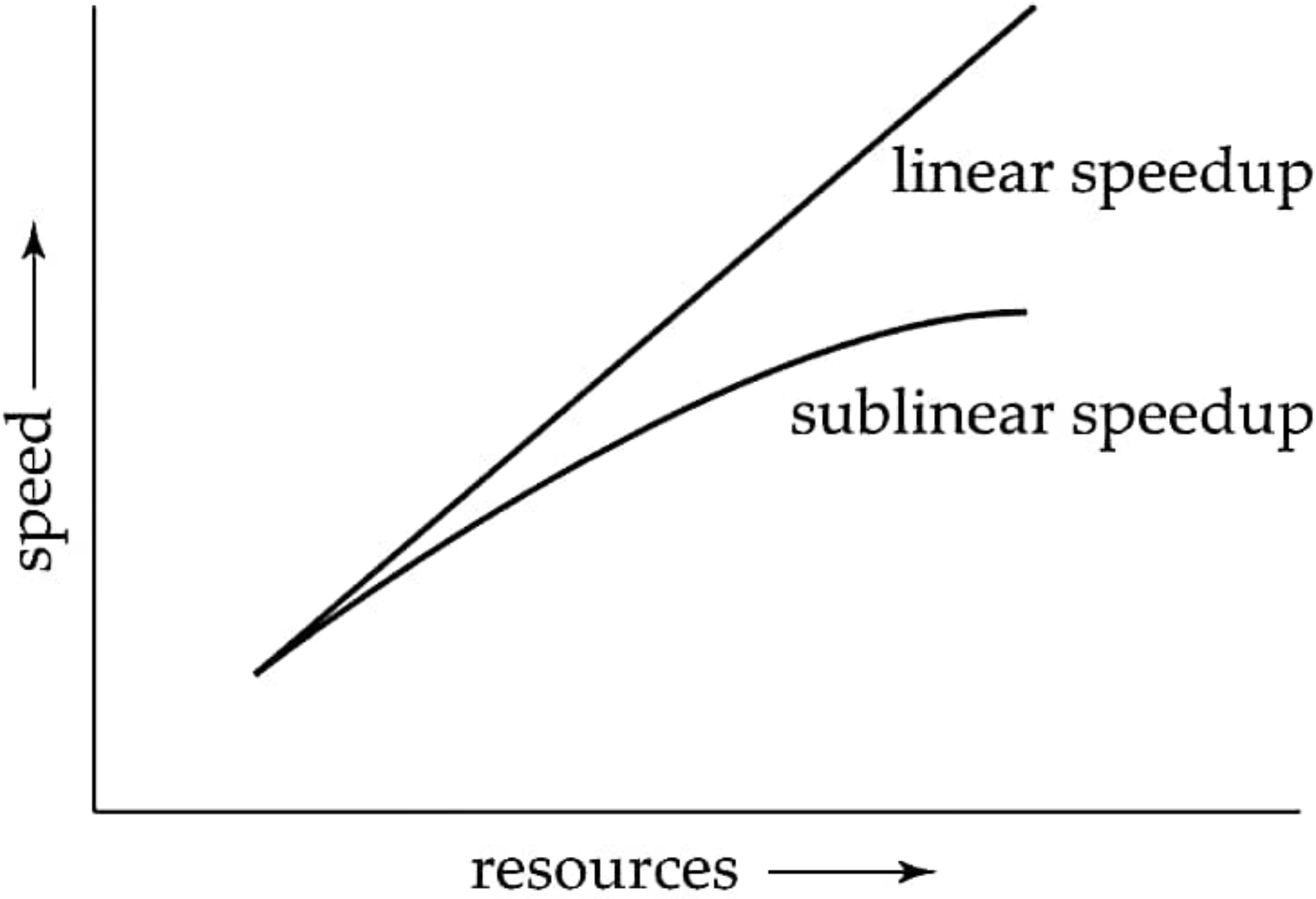
speedup = (n) → (linear)

Def:
- A fixed sized problem executing on a small system
- is given to a system which is N-times larger

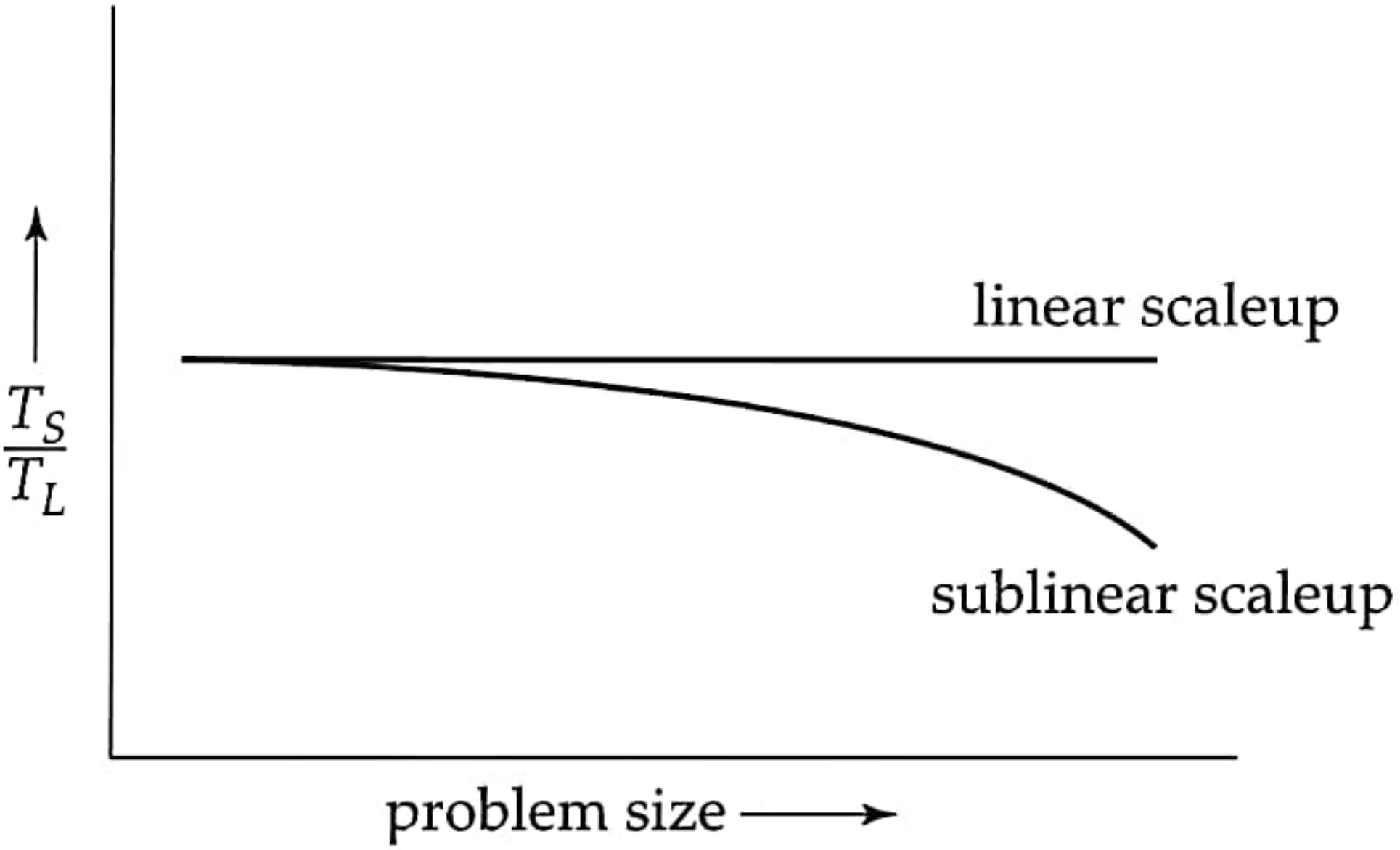
scaleup: $\frac{\text{small sys. small prob elapsed time}}{\text{big " big " " "}}$

scaleup = (1) → linear

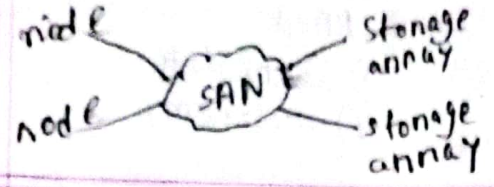
Speedup



Scaleup



modern shared Disk Archi



SAN - Storage Area Network

Atomic :

Test and Set

1) Mem_loc = 0
2) mem_loc = 1, ret old value
   a) 0 → process acquired mutex
   b) 1 → someone is holding mutex, try again
   c) release of mutex
      m = 0

CAS Compare and Swap

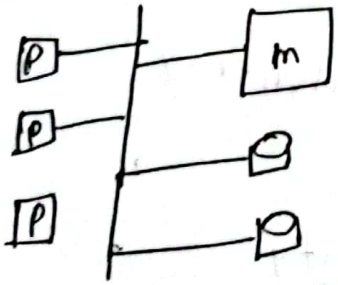
(m, V1, V2)

1) a) if m = V1, set m = V2 ret success
   b) else ret failure

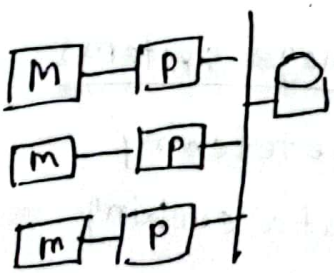
initially CAS(m, 0, 1)

2) CAS(m, 0, id)
   id - tid
      - pid
   t/p - has mutex

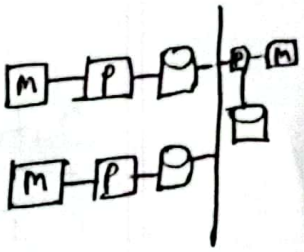
Parallel DB archi



shared memory (P and disc have common mem)
- bus connection

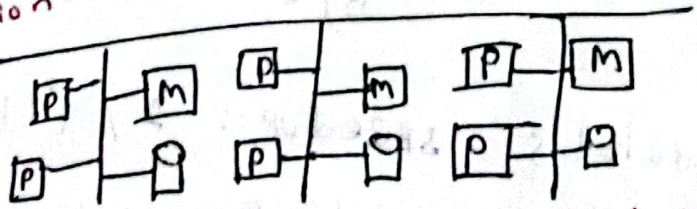


shared disk



shared nothing
- costly

top level - SN/SM
each node - sm/SN



Hierarchical (lots of shared mem)

- combines characteristics of rest 3

memory bannier

load ban.

store ban.

distri dB → PROS
- homo
- hetero

1) Share data
2) autonomy

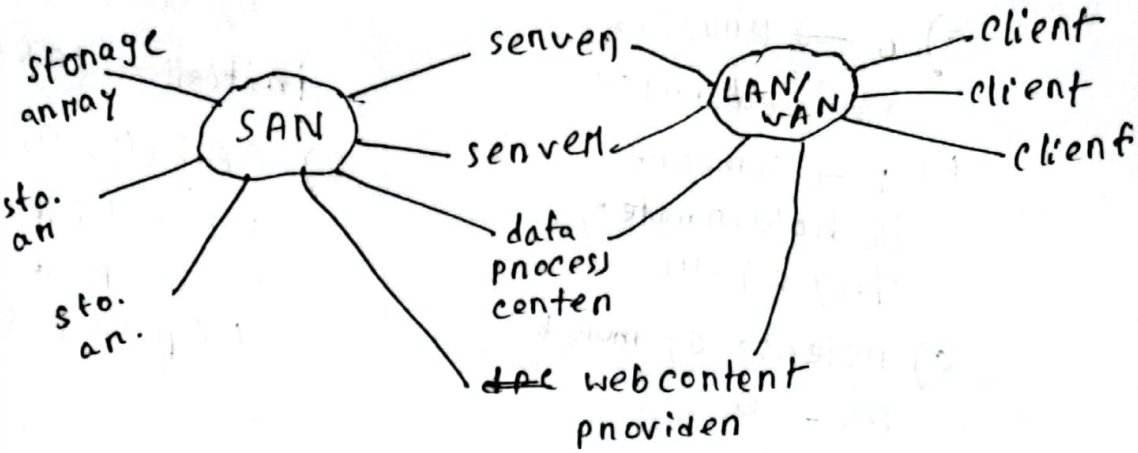
dB — ref — dB
dB

Shared Memory

NUMA - Non-uniform memory anchi

Shared Nothing

RDMA - Remote Direct memory Access



Data storage systems

1) Prefetching
2) Data caching → cache coherency
3) Lock "
4) Adaptive lock granularity
   a) " " escalation
   b) " " de-escalation

Amdahl's speedup : $1 / [(1-P) + (P/n)]$

Gustafson's scaleup : $1 / [n(1-P) + P]$