CSE 4125: Distributed Database Systems.
Chapter – 1: Part B

Distributed Databases: An overview

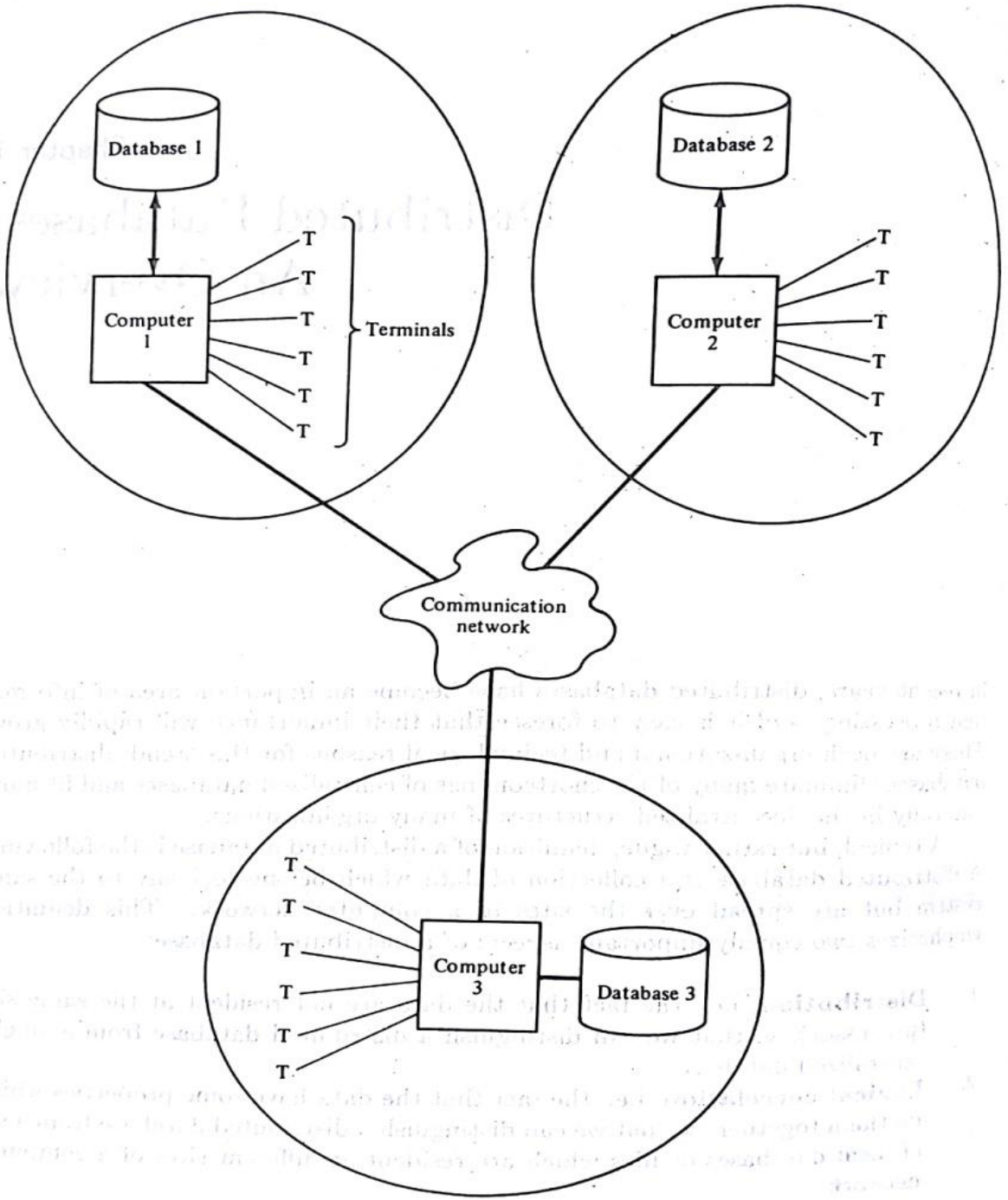
Outline

- DDB vs. traditional DB.
- Necessity of DDB.

DDB vs. Traditional DB

- **Centralized Control:**
  - **Traditional:** Database Admin (DBA).
  - **Distributed:** Hierarchical Responsibility (Global → Local DBA); Depends on Architecture.
- **Data Independence:**
  - **Traditional:** Organization of data is transparent to programmer (*conceptual schema*).
  - **Distributed:** Programs are written as if the databases are not distributed (*distributed transparency*).

ID	NAME


Database 1
Computer 1
T
T
T
T
T
T
Terminals
Database 2
Computer 2
T
T
T
T
T
T
Communication network
Computer 3
Database 3
T
T
T
T
T
T

❑ **Reduction of Redundancy:**

–**Traditional:** Redundancy is not desired and reduced for two 2 reasons:

1. Inconsistencies among several copies of the same logical data are avoided
2. Storage space is saved

–**Distributed:** Desired. Because:

1. Locality increases
2. Availability of the system increases

❑ **Efficient Access:**

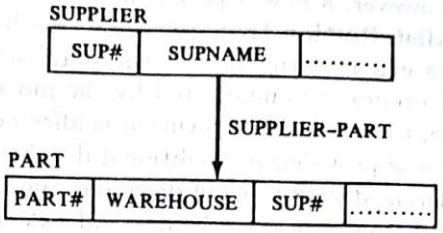
–**Traditional:** Complex physical structure.

•Navigate at record level.

–**Distributed:** Distributed access plan.

•Not navigate at record level.

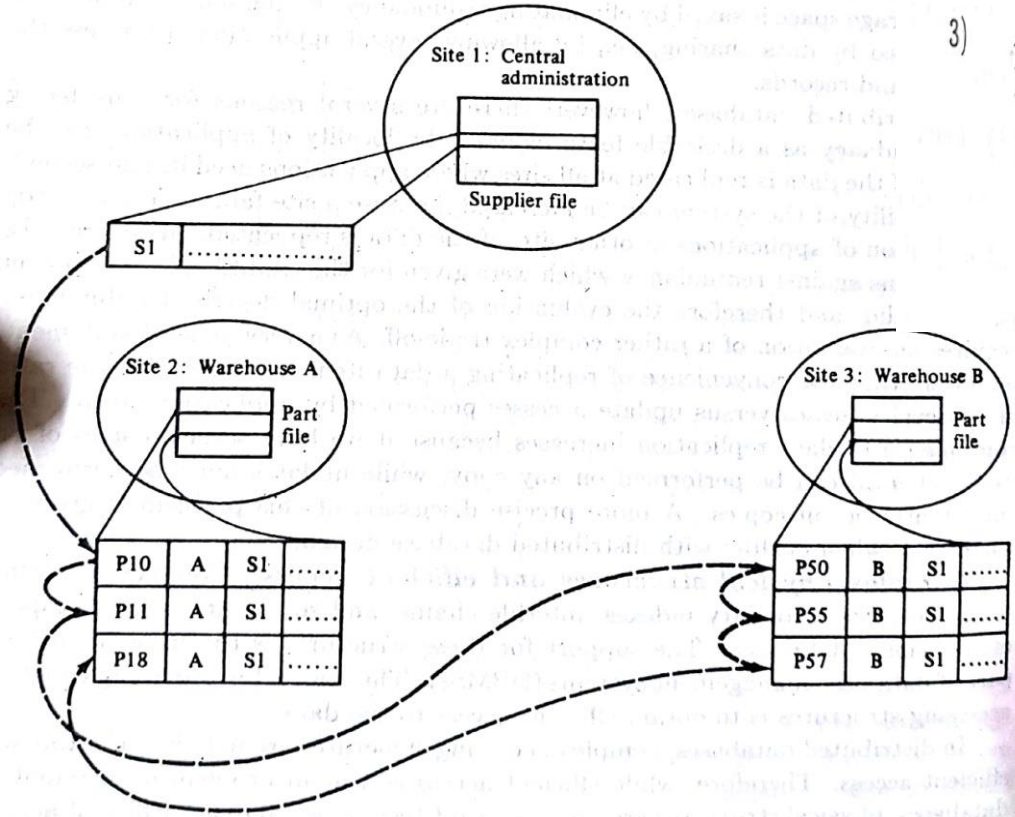
Distributed Access Plan



(*a*) A Codasyl database schema.

```
Find SUPPLIER record with SUP# = S1;
Repeat until "no more members in set"
    Find next PART record in SUPPLIER-PART set;
Output PART record;
```

(*b*) A Codasyl-DBMS-like program for finding parts supplied by supplier S1.



(*c*) Distribution of the SUPPLIER-PART set.

Figure 1.4 A distributed Codasyl-like database.

1) *At site 1*
   Send sites 2 and 3 the supplier number SN
2) *At sites 2 and 3*
   Execute in parallel, upon receipt of the supplier number, the following program:

   Find all PARTS *records having*
   SUP # = SN;
   Send result to site 1.

3) *At site 1*
   Merge results from sites 2 and 3;
   Output the result.

Figure 1.5 Example of access plan.

PART#	WAREHOUSE	SUP#
P10	A	S1
P11	A	S1
P18	A	S1
P50	B	S1
P55	B	S1
P57	B	S1

- **Integrity, recovery and concurrency control:**

–Common issue/ problem in both types.

–Solution: transaction management.

Transaction

- An atomic unit of execution.
- Sequence of operation.
- Either completely performed, or not performed at all.

Example: Fund transfer.

Integrity:

–Assuring one state to another.

Recovery:

–Preserving states while failure.

Concurrency:

–Synchronization.

- **Privacy and Security:**

–**Traditional:**

> DBA ensure the authorized access.

> More vulnerable than distributed, without specialized control procedures.

–**Distributed:**

>The owner of local data feel more protected.

> Security problems are intrinsic (natural) to distributed system in general

Necessity of DDB

- Organizational and economic reason.
  - ✓If the organization is –
    - Decentralized
  - ✓DDB fits more economically.

- Interconnection of existing DB.
  - If need to exchange data between different database.
  - If global application is necessary.

- Incremental growth.
  - If an organization grows by adding new autonomous units (new branches, warehouses etc.) then DDB is best fit for a smooth incremental growth.
  - Less expensive to implement.
- Reduced communication overhead.
  - One advantage of DDB is : local application does not engage communication network (example #1).
  - Workload is distributed.

- Performance consideration.
  - Parallel processing can be done in DDB.

- Reliability and availability.
  - Redundant data.
  - Graceful degradation.
  - Complete system crash is rare.