# Database Management Systems

## Distributed Databases

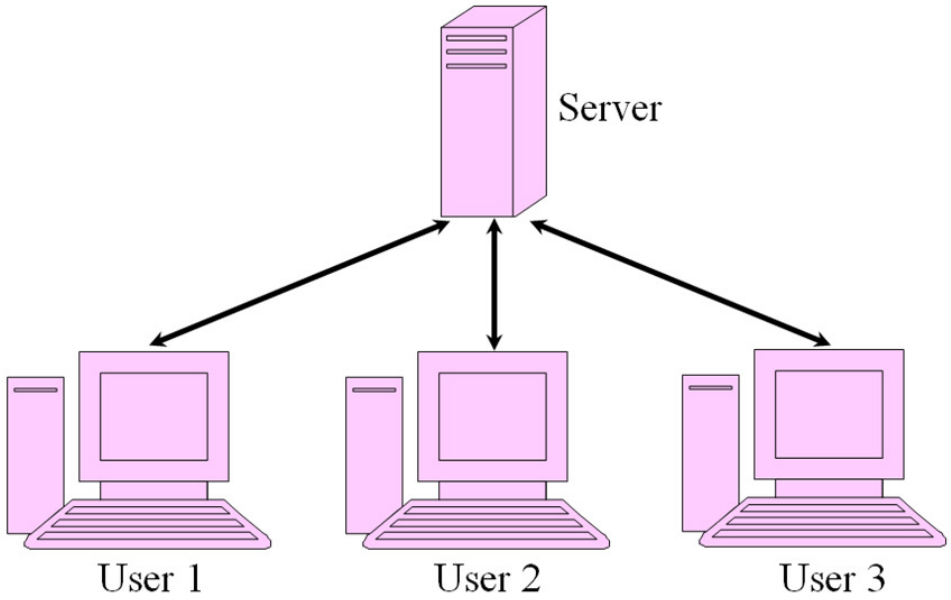
Malay Bhattacharyya

Assistant Professor

Machine Intelligence Unit
and
Centre for Artificial Intelligence and Machine Learning
Indian Statistical Institute, Kolkata

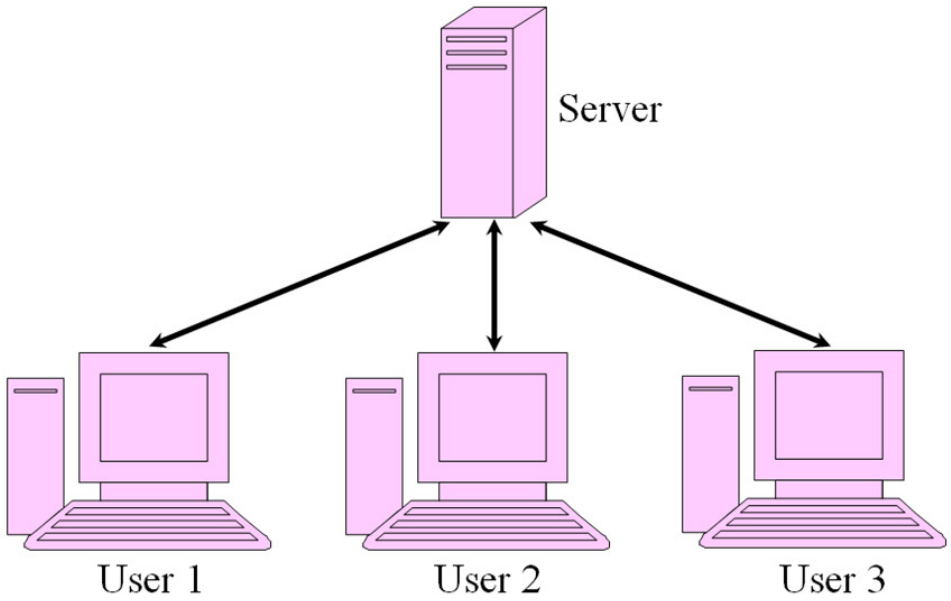
April, 2020

# Basics




**Centralized client-server architecture**

# Basics




**Centralized client-server architecture**

A distributed database system consists of loosely coupled sites that share no physical component. Database systems that run on each site are independent of each other. Transactions may access data at one or more sites.

# Homogeneous and heterogeneous databases

**In a homogeneous distributed database**

– all sites have identical software

– all are aware of each other and agree to cooperate in processing user requests

– each site surrenders part of its autonomy in terms of right to change schemas or software

– the entire system appears as a single system to the user

# Homogeneous and heterogeneous databases

**In a homogeneous distributed database**

– all sites have identical software

– all are aware of each other and agree to cooperate in processing user requests

– each site surrenders part of its autonomy in terms of right to change schemas or software

– the entire system appears as a single system to the user

**In a heterogeneous distributed database**

– different sites may use different schemas and software

– difference in schema is a major problem for query processing

– difference in software is a major problem for transaction processing

# Data distribution

**Data can be distributed in two ways:**

- Replication – The system maintains several identical replicas (copies) of the relation, and stores each replica at a different site. The alternative to replication is to store only one copy of a relation.
- Fragmentation – The system partitions the relation into several fragments, and stores each fragment at a different site.

# Data distribution

**Data can be distributed in two ways:**

- Replication – The system maintains several identical replicas (copies) of the relation, and stores each replica at a different site. The alternative to replication is to store only one copy of a relation.
- Fragmentation – The system partitions the relation into several fragments, and stores each fragment at a different site.

**Note:** The fragmentation can be lossless (original relation can be restored from the partitions) or lossy (original relation can not be restored from the partitions).

# Data distribution

| Replication | Fragmentation |
|---|---|
| Advantageous in terms of high availability | Might not be readily available |
| Advantageous in terms of time complexity but not space complexity | Maintains a balance between the time and space complexity |
| Disadvantageous in view of the redundancy and for updating | No redundancy or problem in updating |

# Data transparency

Data transparency denotes the degree to which a system user may remain unaware of the details of how and where the data items are stored in a distributed system.

# Data transparency

Data transparency denotes the degree to which a system user may remain unaware of the details of how and where the data items are stored in a distributed system.

It can be of the following types:

1. Replication transparency – Users are not required to know what data objects have been replicated, or where replicas have been placed.
2. Fragmentation transparency – Users do not have to be concerned with how a relation has been fragmented.
3. Location transparency – Users are not required to know the physical location of the data.

# Horizontal fragmentation

| Name | Age | Area |
|---|---|---|
| Malay | 37 | Crowdsourcing |
| Ansuman | 43 | High Performance Architectures |

| Name | Age | Area |
|---|---|---|
| Sasthi | 46 | Wireless Networks |
| Sourav | 39 | Theoretical Computer Science |

# Horizontal fragmentation

| Name | Age | Area |
|---|---|---|
| Malay | 37 | Crowdsourcing |
| Ansuman | 43 | High Performance Architectures |

| Name | Age | Area |
|---|---|---|
| Sasthi | 46 | Wireless Networks |
| Sourav | 39 | Theoretical Computer Science |

**Note:** Horizontal fragmentation is lossless when union of the fragments produces the original relation.

# Vertical fragmentation

| Name | Age |
|---|---|
| Malay | 37 |
| Ansuman | 43 |
| Sasthi | 46 |
| Sourav | 39 |

| Area |
|---|
| Crowdsourcing |
| High Performance Architectures |
| Wireless Networks |
| Theoretical Computer Science |

# Vertical fragmentation

| Name | Age |
|---|---|
| Malay | 37 |
| Ansuman | 43 |
| Sasthi | 46 |
| Sourav | 39 |

| Area |
|---|
| Crowdsourcing |
| High Performance Architectures |
| Wireless Networks |
| Theoretical Computer Science |

**Note:** Vertical fragmentation is lossless when natural join of the fragments produces the original relation.

# Advantages of horizontal and vertical fragmentation

**Horizontal:**

- It allows parallel processing on fragments of a relation.
- It allows a relation to be split so that tuples are located where they are most frequently accessed.

# Advantages of horizontal and vertical fragmentation

**Horizontal:**

- It allows parallel processing on fragments of a relation.
- It allows a relation to be split so that tuples are located where they are most frequently accessed.

**Vertical:**

- It allows tuples to be split so that each part of the tuple is stored where it is most frequently accessed.
- Here tuple-id attribute allows efficient joining of vertical fragments.
- It allows parallel processing on a relation.

# Advantages of horizontal and vertical fragmentation

**Horizontal:**

- It allows parallel processing on fragments of a relation.
- It allows a relation to be split so that tuples are located where they are most frequently accessed.

**Vertical:**

- It allows tuples to be split so that each part of the tuple is stored where it is most frequently accessed.
- Here tuple-id attribute allows efficient joining of vertical fragments.
- It allows parallel processing on a relation.

Vertical and horizontal fragmentation can be mixed (hybrid fragmentation) and the advantage is that fragments may be successively fragmented to an arbitrary depth.

# Hybrid fragmentation

A hybrid fragment neither include all the tuples for an attribute (likewise vertical fragmentation) nor all the attributes for a tuple (likewise horizontal fragmentation).

| Name | Age |
|---|---|
| Malay | 37 |
| Ansuman | 43 |
| Sourav | 39 |

# Distributed transaction management

- Transaction may access data at several sites.

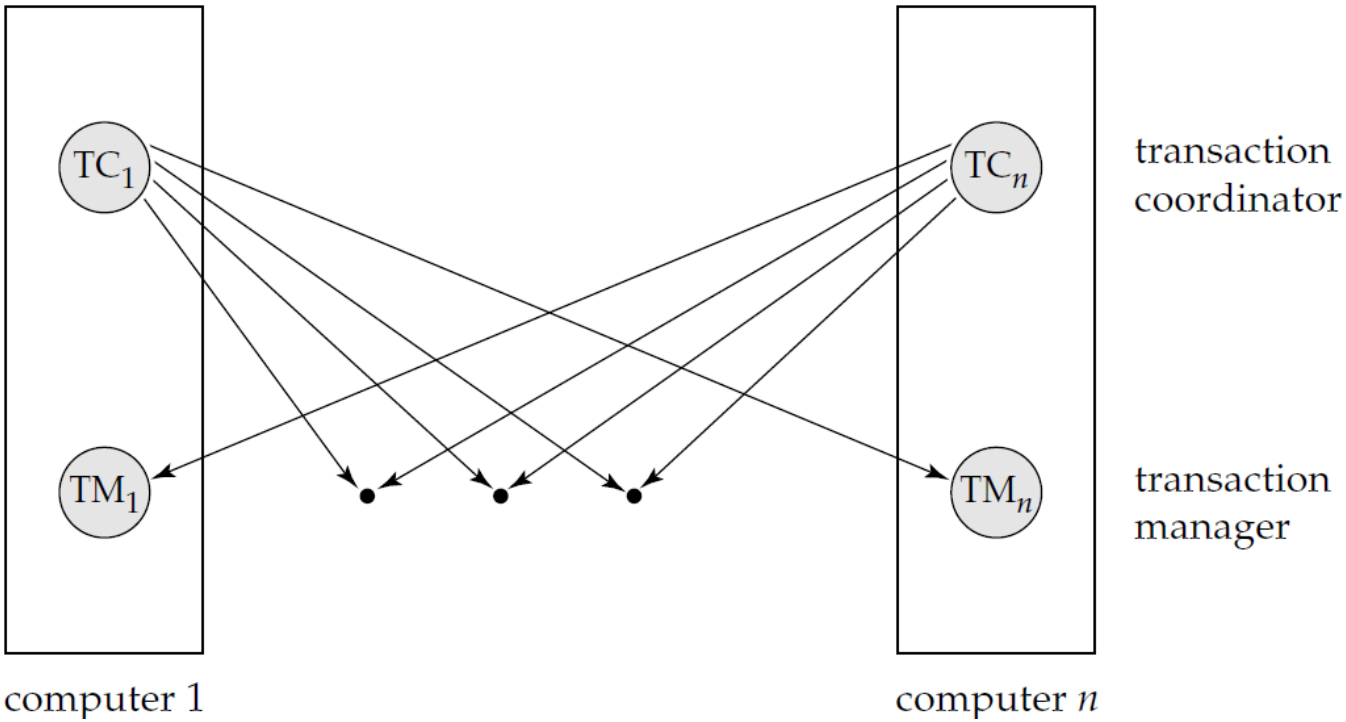
# Distributed transaction management

- Transaction may access data at several sites.
- Each site has a local *transaction manager* responsible for:
  1. Maintaining a log for recovery purposes
  2. Participating in coordinating the concurrent execution of the transactions executing at that site.

# Distributed transaction management

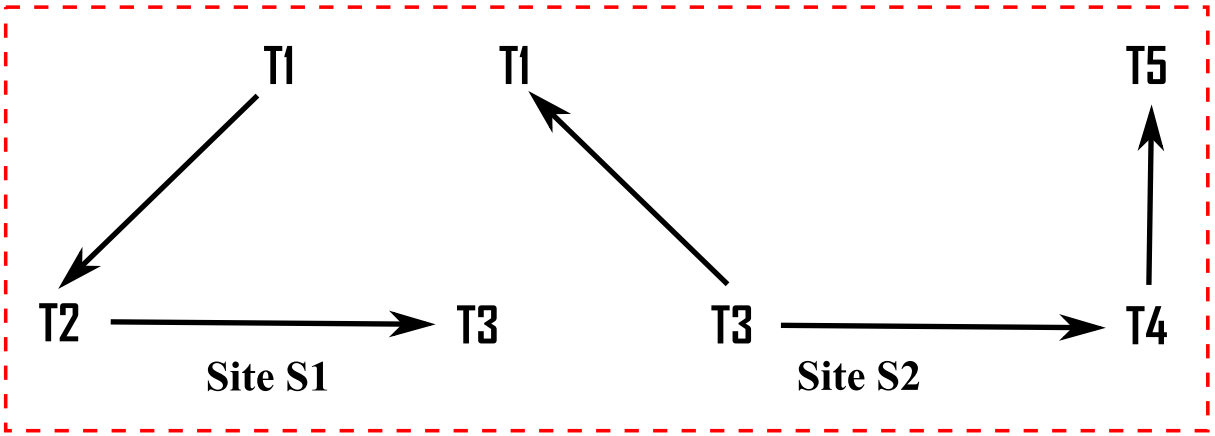
- Transaction may access data at several sites.
- Each site has a local *transaction manager* responsible for:
  1. Maintaining a log for recovery purposes
  2. Participating in coordinating the concurrent execution of the transactions executing at that site.
- Each site has a *transaction coordinator*, which is responsible for:
  1. Starting the execution of transactions that originate at the site.
  2. Distributing subtransactions at appropriate sites for execution.
  3. Coordinating the termination of each transaction that originates at the site, which may result in the transaction being committed at all sites or aborted at all sites.

# Distributed transaction management

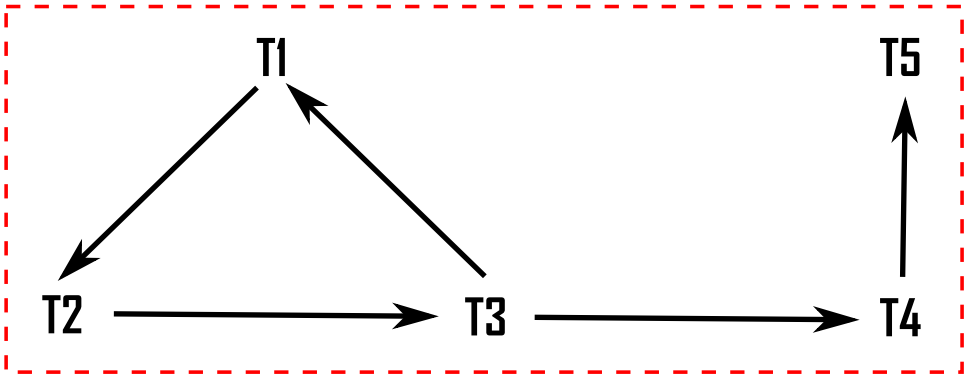



**Distributed system architecture for transaction management**

# Distributed deadlock handling




**Local view**




**Global view**

**Local and global wait-for graphs**