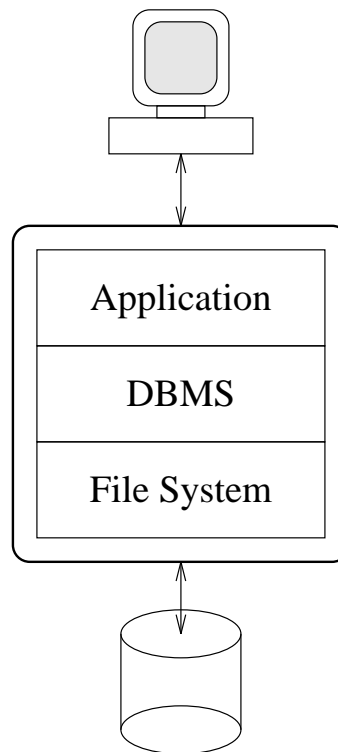


## **VII. Database System Architecture**

### Lecture Topics

- Monolithic systems
- Client/Server systems
- Parallel database servers
- Multidatabase systems

## Monolithic System



Each component presents a well-defined interface to the component above.

## Component Functions

- **Applications**

- User interaction: input of queries and data, display of results
- Application-specific tasks

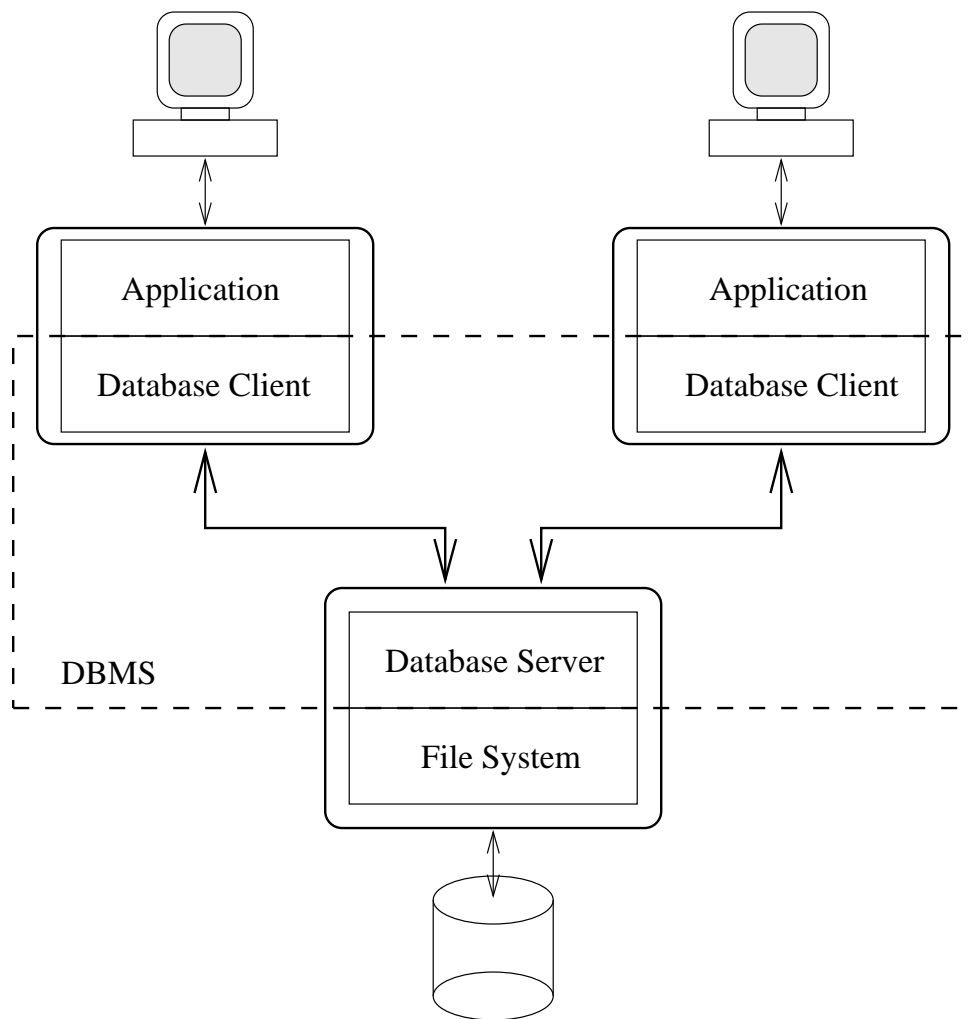
- **The DBMS**

- Query optimization: selection of one of many possible procedures for executing a query
- Query processing: execution of the selected query
- Buffer management: allocation and control of memory
- Transaction management: concurrency control, rollback, and failure recovery
- Security and integrity management: access control and consistency checking

- **The File System**

- Storage and retrieval of unstructured data

### Client/Server System



## Client/Server System (cont.)

- **DBMS Client:** packs application requests into messages, sends messages to server, waits for and unpacks the response
- **DBMS Server:** all database system functions, including query processing and optimization, transaction management, security and integrity management, buffer management

---

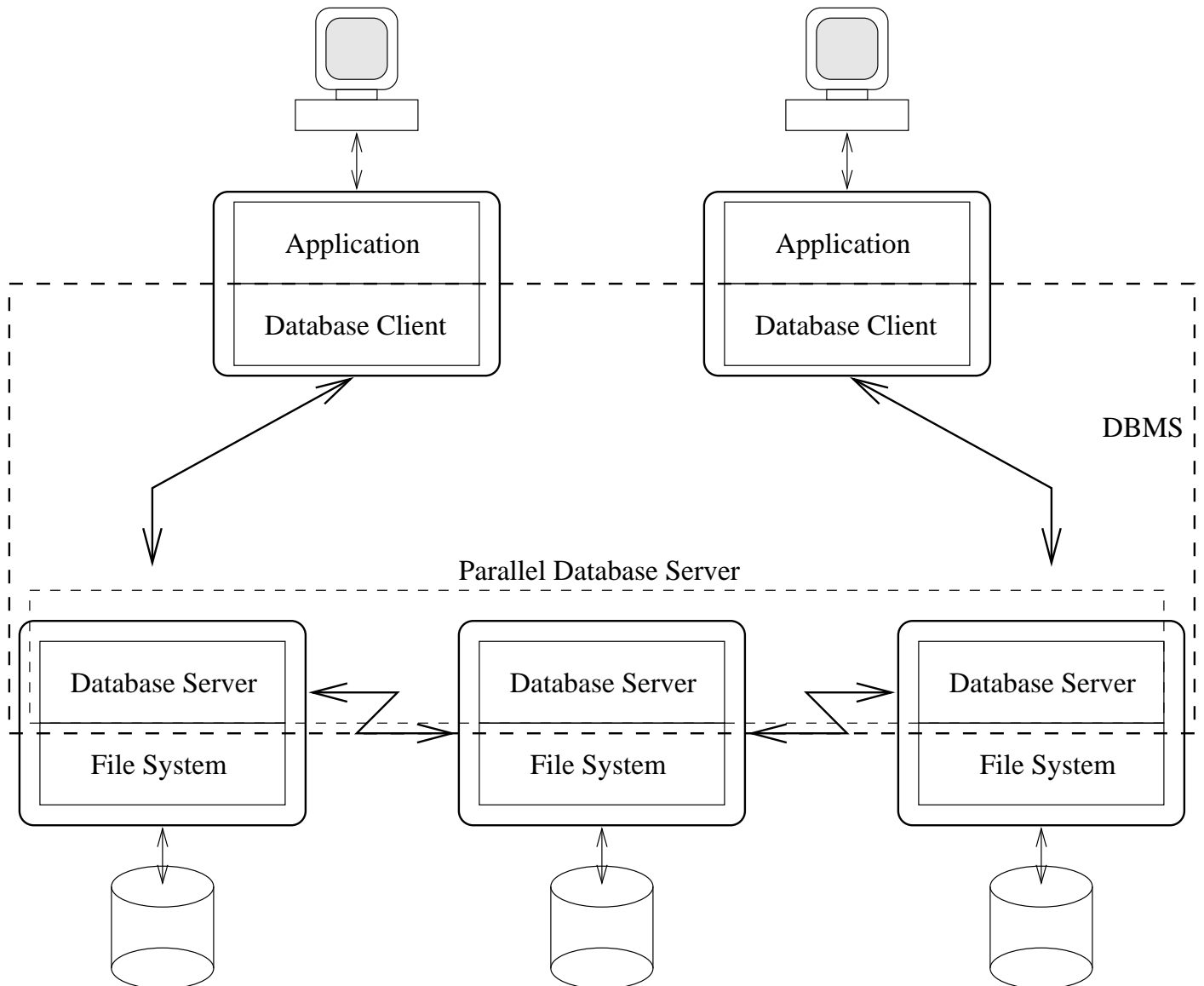
---

Client/server separation allows user interaction and database management to be performed by different processors

---

---

### Parallel/Distributed Database Server



## Parallel/Distributed Database Server (cont.)

- Data is distributed across the **sites**
- Relations may be fragmented
- Relations (or fragments of relations) maybe replicated at several sites

---

---

Component servers are intended to operate as a team, not independently.

---

---

- A single, common schema
- Distribution of data is transparent
- Distribution of computation is transparent
- Replication is transparent
- Fragmentation is transparent

## Parallel vs. Distributed

- **Parallel Database Server**

- servers in physical proximity to each other
- fast, high-bandwidth communication between servers, usually via a LAN or shared memory
- queries often processed cooperatively by all servers

- **Distributed Database Server**

- servers may be widely separated
- server-to-server communication may be slower, possibly even via a WAN
- queries often processed by a single server

## Parallel/Distributed: Why?

- Reliability and Availability: if one server fails, another can take its place
- Faster Query Processing: several servers can cooperate to process a query

## Horizontal Fragmentation

**Complete relation:**

Vno	Vname	City	Vbal
1	Sears	Toronto	200.00
2	Kmart	Ottawa	671.05
3	Eatons	Toronto	301.00
4	The Bay	Ottawa	162.99

**Horizontally fragmented relation (two sites):**

### Site 1 (Ottawa site)

Vno	Vname	City	Vbal
2	Kmart	Ottawa	671.05
4	The Bay	Ottawa	162.99

### Site 2 (Toronto site)

Vno	Vname	City	Vbal
1	Sears	Toronto	200.00
3	Eatons	Toronto	301.00

## Vertical Fragmentation

**Complete relation:**

Vno	Vname	City	Vbal
1	Sears	Toronto	200.00
2	Kmart	Ottawa	671.05
3	Eatons	Toronto	301.00
4	The Bay	Ottawa	162.99

**Vertically fragmented relation (two sites):**

### Site 1

Vno	Vname	City
1	Sears	Toronto
2	Kmart	Ottawa
3	Eatons	Toronto
4	The Bay	Ottawa

### Site 2

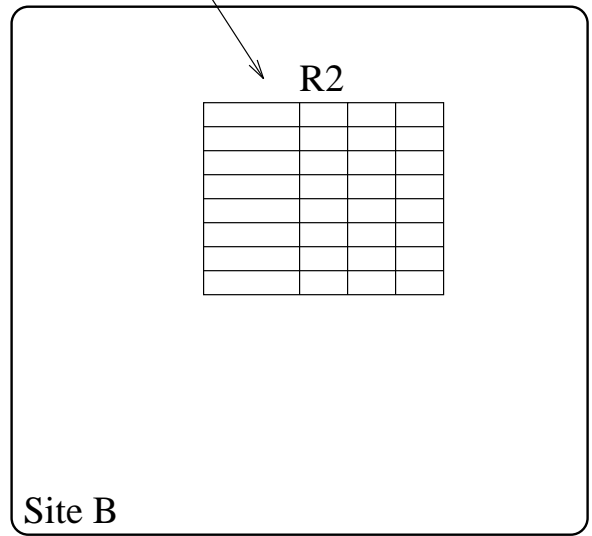
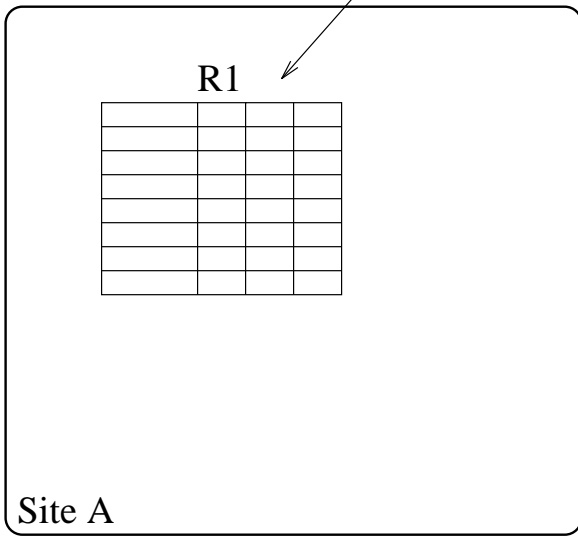
Vno	Vbal
1	200.00
2	671.05
3	301.00
4	162.99

### Data Distribution

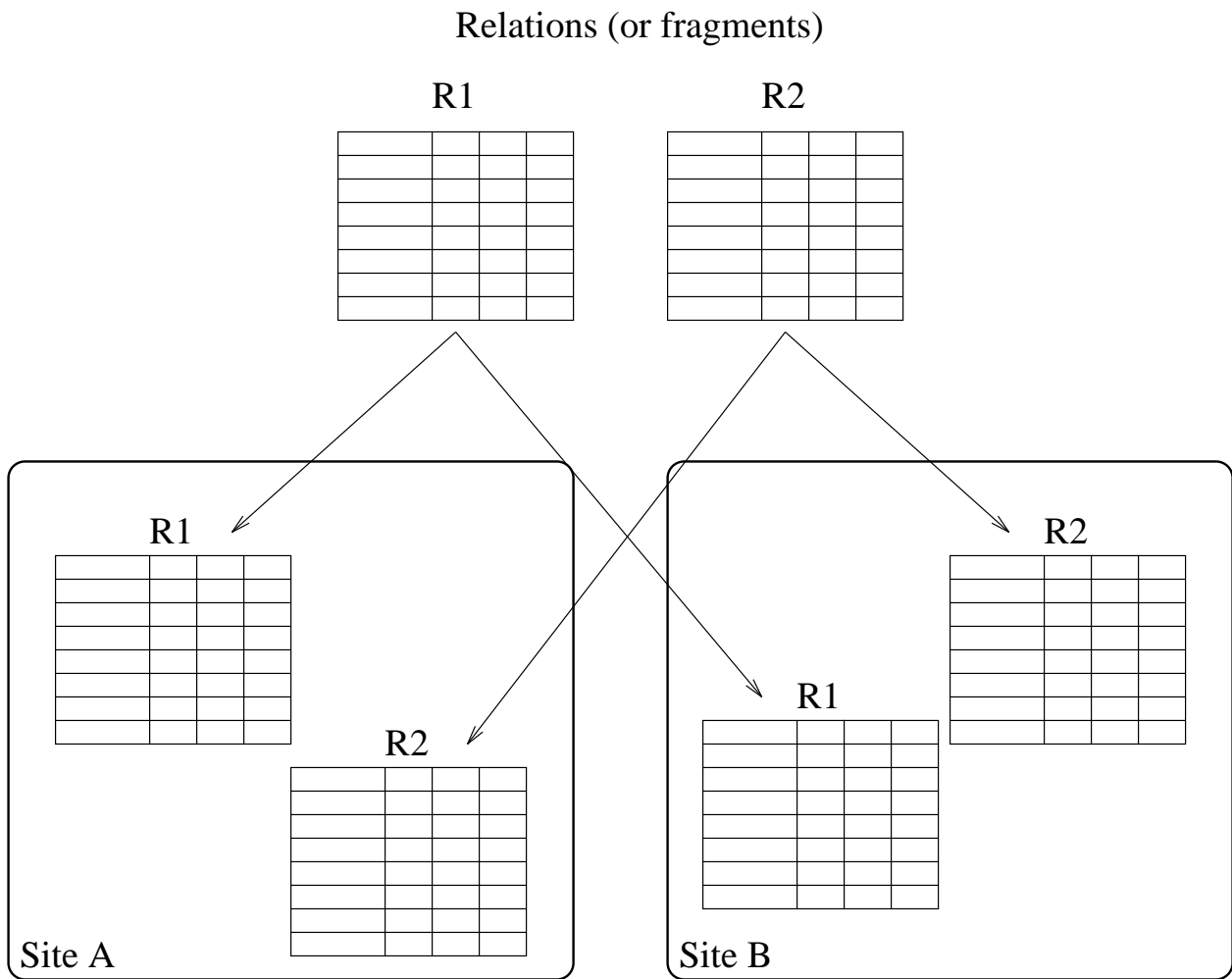
Relations (or fragments)

R1


R2

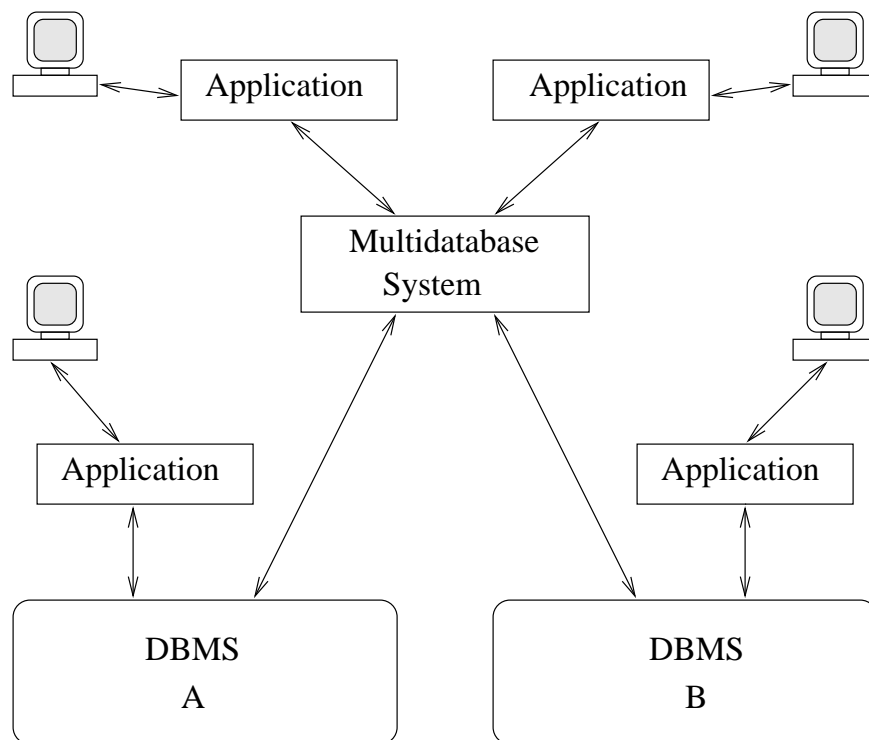



# Data Replication



## Multidatabase System

- Application perceives a single database system
- Database systems are autonomous



### Multidatabase System w/ Gateway

