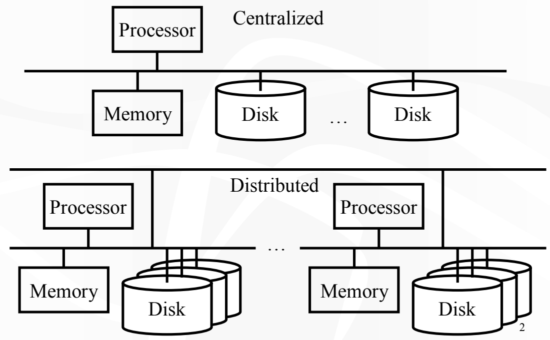
Distributed Databases

CPS 216
Advanced Database Systems

Centralized versus distributed DBMS



Parallel versus distributed DBMS

- Parallel DBMS
  - Fast interconnect
  - Homogeneous hardware/software
  - Total control over components
- Distributed DBMS
  - Geographically distributed
    - Disconnected operations possible
  - Heterogeneous hardware/software
    - Performance, data formats, data processing capabilities
  - Autonomy of individual sites

Distributed DBMS issues

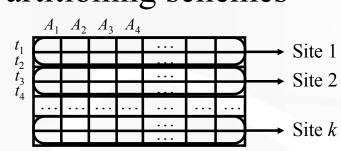
- Database management with multiple sites that are possibly autonomous and heterogeneous
  - Data organization
  - Query processing and optimization
  - Concurrency control and recovery

Data organization

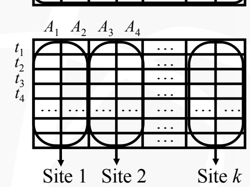
- Top-down approach
  - Have a database
  - How to partition and/or replicate it across sites
- Bottom-up approach
  - Have existing databases at different sites
  - How to integrate them together and deal with heterogeneity and autonomy
- Focus for today
  - Data partitioning using a top-down approach

Partitioning schemes

- Horizontal



- Vertical



- Or hybrid

Horizontal partitioning schemes

- Round-robin partitioning
- Hash partitioning
- Range partitioning
- Predicate-based partitioning
- Derived horizontal partitioning

7

Properties of a correct partitioning

$R \rightarrow \{ R_1, R_2, \ldots, R_k \}$

- Completeness and reconstructability

  $R = R_1 \cup R_2 \cup \ldots \cup R_k$

- Disjointness

  $R_i \cap R_j = \emptyset$ for any $i \neq j$

8

Round-robin partitioning

R	R_0	R_1	R_2
t_1	t_1		
t_2		t_2	
t_3			t_3
t_4	t_4		
…			

- Evenly distributes data
- Good for full relation scans
- Not good for range queries

9

Hash partitioning

R		R_0	R_1	R_2
t_1	hash(k_1) = 2			t_1
t_2	hash(k_2) = 0	t_2		
t_3	hash(k_3) = 0	t_3		
t_4	hash(k_4) = 1		t_4	
…				

- Evenly distributes data (assuming a good hash function)
- Good for point queries and equijoins on the partitioning attribute
- Not good for range queries

10

Range partitioning

R	partitioning vector: <4, 7>	R_0	R_1	R_2
t_1	$k_1 = 5$		t_1	
t_2	$k_2 = 8$			t_2
t_3	$k_3 = 2$	t_3		
t_4	$k_4 = 3$	t_4		
…				

- Good for range queries on the partitioning attribute
- The choice of partitioning vector is important
  - Bad vector may result in both data skew and execution skew

11

Predicate-based partitioning

- Fragmentation
  - Decide how to divide a relation horizontally into fragments using a set of predicates
- Allocation
  - Decide which fragments go to which site

12

Predicate-based fragmentation

- Given a relation R and a set of simple predicates $P = \{ p_1, p_2, \ldots, p_n \}$
- Generate minterm predicates
  - $M = \{ m \mid m = \wedge_{(1 \le k \le n)} p_k^* \}$, where p_k^* is either p_k or $\neg p_k$
  - Simplify minterms in M and eliminate useless ones
- For each m in M, generate a fragment $\sigma_m R$

13

Example

- Say queries use simple predicates: $A < 10$, $A > 5$, $D =$ 'CS', $D =$ 'EE'
- Generate, simplify, and eliminate minterms
  ~~$A < 10 \wedge A > 5 \wedge D =$ 'CS' $\wedge D =$ 'EE'~~ eliminated
  ~~$A < 10 \wedge A \le 5 \wedge D =$ 'CS' $\wedge D \ne$ 'EE'~~ $A \le 5 \wedge D =$ 'CS'
  …
- Final set of fragments

$\sigma_{5 < A < 10 \wedge D = \text{'CS'}} R$ $\quad \sigma_{5 < A < 10 \wedge D = \text{'EE'}} R$
$\sigma_{A \le 5 \wedge D = \text{'CS'}} R$ $\quad \sigma_{A \le 5 \wedge D = \text{'EE'}} R$
$\sigma_{A \ge 10 \wedge D = \text{'CS'}} R$ $\quad \sigma_{A \ge 10 \wedge D = \text{'EE'}} R$

14

Choice of simple predicates

- Completeness
  - There is an equal probability of access by every application to any two tuples in the same minterm fragment
    - If p is used in fragmentation, then $\sigma_p R$ either accesses all tuples in a fragment or none in a fragment
- Minimality
  - If a predicate causes a fragment f to be further fragmented into f_i and f_j, there should at least one application that accesses f_i and f_j differently

» Use all relevant predicates in frequent queries!

15

Allocation of fragments

- Tough optimization problem
  - Do we replicate fragments?
  - Where we place each copy of each fragment?
- Metrics: minimize query response time; maximize throughput; minimize network traffic; …
- Constraints: available storage, bandwidth, processing power; response time requirement; …
- Issues: origin of queries; selectivity of fragments; query processing strategies; consistency enforcement; …

16