

Other Topics

December 12, 2008

Outline

1 Parallel Databases

2 Distributed Databases

Introduction

- Motivation:
 - cheap processing power
 - increasing database size
 - genetic information
 - multimedia data

Introduction

- Motivation:
 - cheap processing power
 - increasing database size
 - genetic information
 - multimedia data
- Large-scale parallel DBs
 - store large volume of data in parallel
 - process time-consuming decision support queries in parallel
 - high throughput

Parallelism in Databases

- DB are ripe for parallelism

Parallelism in Databases

- DB are ripe for parallelism
- Data can be partitioned across multiple disks for parallel I/O
- Each relational operation (sort, join, ...) in parallel for single query (intra-query parallelism)
 - data partition $\Rightarrow$ multiple processors work on independent partitions

Parallelism in Databases

- DB are ripe for parallelism
- Data can be partitioned across multiple disks for parallel I/O
- Each relational operation (sort, join, . . .) in parallel for single query (intra-query parallelism)
 - data partition $\Rightarrow$ multiple processors work on independent partitions
- Multiple queries can run in parallel (inter-query parallelism)
 - need concurrency control
- Query still in SQL

I/O Parallelism

- Reduce relation-retrieval time by partitioning relations on disks
- Usually for large relations: horizontal partitioning
 - divide tuples of a relation: one tuple per disk

Partitioning Techniques

- **Round robin:**

Partitioning Techniques

- **Round robin:**
 - send i^{th} tuple to disk $i \bmod n$

Partitioning Techniques

- **Round robin:**

- send i^{th} tuple to disk $i \bmod n$

- **Hash partitioning:**

- hash function, h , range: $0 \dots n - 1$
- choose attributes $a_1, \dots, a_m$
- For tuple i : $h(a_1, \dots, a_m) = k$;
 - $\Rightarrow$ send i to disk k

Parallel System Design Issues

- Parallel loading of data from external sources to handle large volumes of incoming data

Parallel System Design Issues

- Parallel loading of data from external sources to handle large volumes of incoming data
- Resilience to failure of some processors or disks.
 - Probability of some disk or processor failing is higher in a parallel system.
 - Operation (perhaps with degraded performance) should be possible during failure.
 - Redundancy achieved by storing extra copy of every data item at another processor.

Outline

1 Parallel Databases

2 Distributed Databases

Introduction

- DBMS consists of loosely coupled sites
- Individual site runs (fairly) independent of others
- A transaction can access data from any site

Distributed Data Storage

- **Replication:**

- copies of data in several sites: fast retrieval, fault tolerance

- **Fragmentation:**

- Relation is partitioned at several sites (like parallel DB)

Distributed Data Storage

- **Replication:**
 - copies of data in several sites: fast retrieval, fault tolerance
- **Fragmentation:**
 - Relation is partitioned at several sites (like parallel DB)
- *Transparent* to user:
 - distribution (network)
 - replication
 - fragmentation
- Can combine fragmentation and replication

Advantages of Distributed Databases

- Cheaper to build multiple small sites
- Improve performance:
 - just like parallel DB
 - also, with replication, load-balanced response
- Increases modularity

Issues with Distributed Databases

- Locality of reference
 - How near is the queried data?

Issues with Distributed Databases

- Locality of reference
 - How near is the queried data?
- Reliability and availability
 - Does the strategy improve fault tolerance and access?

Issues with Distributed Databases

- Locality of reference
 - How near is the queried data?
- Reliability and availability
 - Does the strategy improve fault tolerance and access?
- Performance
 - Are there bottlenecks or under-utilisation of resources?

Issues with Distributed Databases

- Locality of reference
 - How near is the queried data?
- Reliability and availability
 - Does the strategy improve fault tolerance and access?
- Performance
 - Are there bottlenecks or under-utilisation of resources?
- Storage
 - What is the effect on size of data storage?

Issues with Distributed Databases

- Locality of reference
 - How near is the queried data?
- Reliability and availability
 - Does the strategy improve fault tolerance and access?
- Performance
 - Are there bottlenecks or under-utilisation of resources?
- Storage
 - What is the effect on size of data storage?
- Communication costs
 - How much network traffic will result from the strategy?