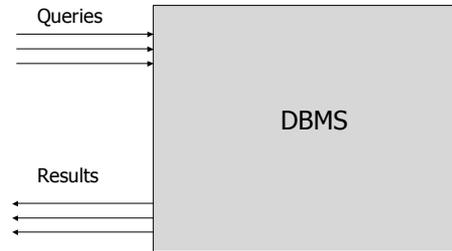


CS422 Principles of Database Systems

Introduction to Database Performance Tuning

Chengyu Sun
California State University, Los Angeles

DBMS – A User’s Perspective

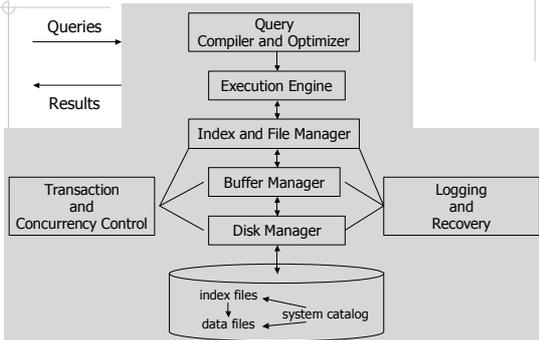


Database Tuning

Tuning rests on a foundation of informed common sense. This makes it both easy and hard.

Dennis Shasha and Philippe Bonnet
Database Tuning: Principles, Experiments and Troubleshooting Techniques

DBMS – Under the Hood

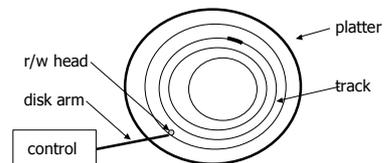


Hardware Tuning

◆ Disk I/O is the bottleneck for most DB applications

- Faster CPU doesn't help much
- The more memory, the better
 - ◆ Aim for 90+% buffer hit rate, or
 - ◆ Add memory until buffer hit rate level off
- Disk
 - ◆ Disk characteristics
 - ◆ RAID

Disk Characteristics ...



- ◆ Seek time
- ◆ Rotational delay
- ◆ Read/write time

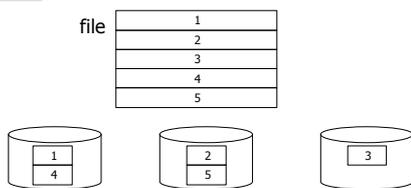
... Disk Characteristics

- ◆ Disk accesses are slow
- ◆ Random accesses are even slower
- ◆ Solutions:
 - Buffering
 - Access by disk page
 - ◆ 1k, 4k, or 8k block
 - ◆ Clustering
 - Multiple disks
 - ◆ put tables, indexes, and logs on different disks

RAID

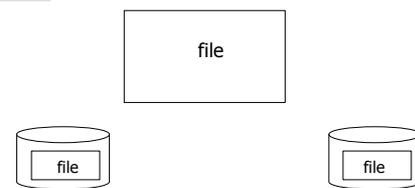
- ◆ Redundant Array of Independent Disks
- ◆ RAID levels
 - 0 – striping
 - 1 – mirroring
 - 5 – rotated parity mirroring
 - 0+1 – mirrored striping

RAID 0



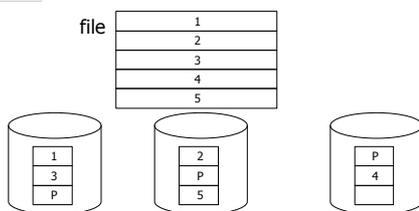
- ◆ Benefits large sequential read
- ◆ Problem??

RAID 1



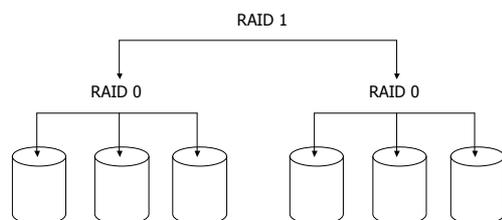
- ◆ Redundancy
- ◆ Performance improvement??

RAID 5



- ◆ Benefits of both RAID0 and RAID1 with some redundancy and better storage utilization

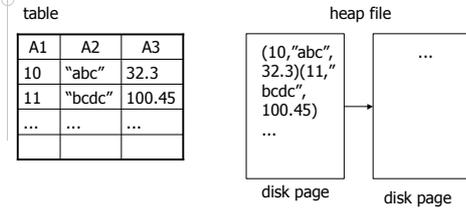
RAID 0+1



RAID in DBMS

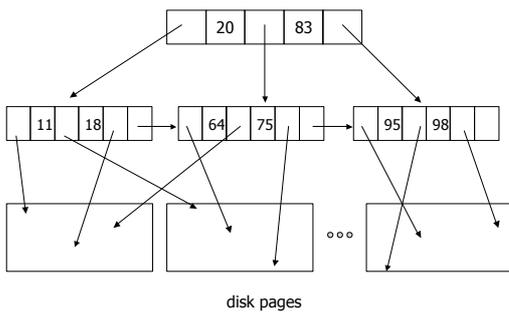
- ◆ Log files – RAID 1
- ◆ Temporary files – RAID 0
- ◆ Data and index files – RAID 5 if read traffic dominates

Tables – Heap File

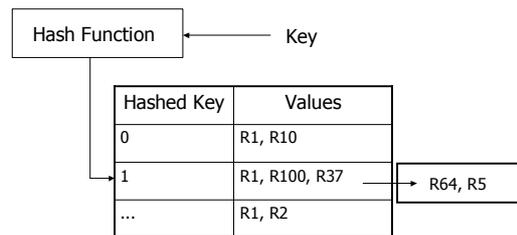


- ◆ by tuple
- ◆ unordered – low update cost

Indexes – B+-tree (dense)



Indexes – Hash Index



Clustered vs. Non-clustered

- ◆ Clustered (primary) index
 - Tuples are clustered by the indexed attribute(s)
- ◆ Non-clustered (secondary) index

Composite Index

- ◆ Composite index – an index which involves multiple attributes

Index Choices

- ◆ Index or not?
 - update, small table, search attributes
- ◆ Hashing or B-tree?
 - query types
- ◆ Clustered or non-clustered?
- ◆ Composite index?

Vertical Partitioning and Denormalization

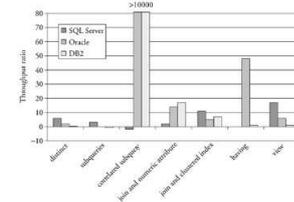
- ◆ Vertical partitioning
 - (AccountID, Address, Balance)
 - (AccountID, Address) and (AccountID, Balance)
- ◆ Denormalization
 - (SupplierID, SupplierRegion) and (SupplierID, ProductID, Quantity)
 - (SupplierID, ProductID, Quantity, SupplierRegion)

Query Tuning

- ◆ DISTINCT, ORDER BY, HAVING
- ◆ Subqueries
- ◆ Use of temporary tables
- ◆ Joins

Query Tuning Experiments

SQL Server 2000
Oracle 8i
IBM UDB V7.1



Database Tuning: Principles, Experiments and Troubleshooting Techniques, Shasha and Bonnet, 2003

Performance Tuning Tools

- ◆ Query plan tools
 - Are the query optimizer doing its job?
- ◆ Trace and profiling tools
 - More detailed information about query costs.
- ◆ Analysis tools
 - Keep the table stats up-to-date
- ◆ Tuning helpers
- ◆ Maintenance tools

Some Performance Tuning Tools in PostgreSQL

- ◆ explain
- ◆ cluster
- ◆ vacuum
- ◆ analyze

Summary

- ◆ Knowledge of DBMS and computer systems in general
- ◆ Performance tuning heuristics
- ◆ Performance tuning tools