LUT University

LAND OF THE CURIOUS

TABLE OF CONTENTS

- Performance management
- System performance
- Database performance

CT60A7650 – DATABASE SYSTEMS MANAGEMENT

PERFORMANCE MANAGEMENT

Lecture

Jiri Musto, D.Sc.

INTRO TO PERFORMANCE MANAGEMENT

- Five factors influencing database performance
  - Workload: transactions, queries, analyses, system commands, etc.
  - Throughput: Capability of the system (I/O, CPU, efficiency)
  - Resources: Hardware & software resources
  - Optimization: How optimized everything is
  - Contention: Conflicts when using resources during workload
- Performance management is often reactive, but proactive is better

DEFINING PERFORMANCE

- Database performance can be defined:
  - “Optimization of resource use to increase throughput and minimize contention, enabling the largest possible workload to be processed
- Performance should not be managed in a vacuum
  - There are many parts affecting the performance
  - Sometimes connecting applications that are not your responsibility affect performance
- Remember to place limits to the scope

BASIC DATABASE PERFORMANCE ROADMAP

- Manageability—the ability to create and maintain an effective environment that delivers service to users
- Recoverability—the ability to reestablish service in the event of an error or component failure
- Reliability—the ability to deliver service at specified levels for a stated period
- Serviceability—the ability to determine the existence of problems, diagnose their cause(s), and repair the problems.
- 80/20 rule
  - The Pareto Principle
  - 80 % of results come from 20 % effort
  - 20 % of your applications cause 80 % of your problems

TUNING THE DATABASE: FINDING THE PROBLEM

- Application
  - SQL, host language code
- Database
  - Indices, database and index organization
- System / subsystem
  - Locking, parameters, address spaces, etc.
- Environment
  - Network, disk, operating system, etc.
- Only tune one thing at a time

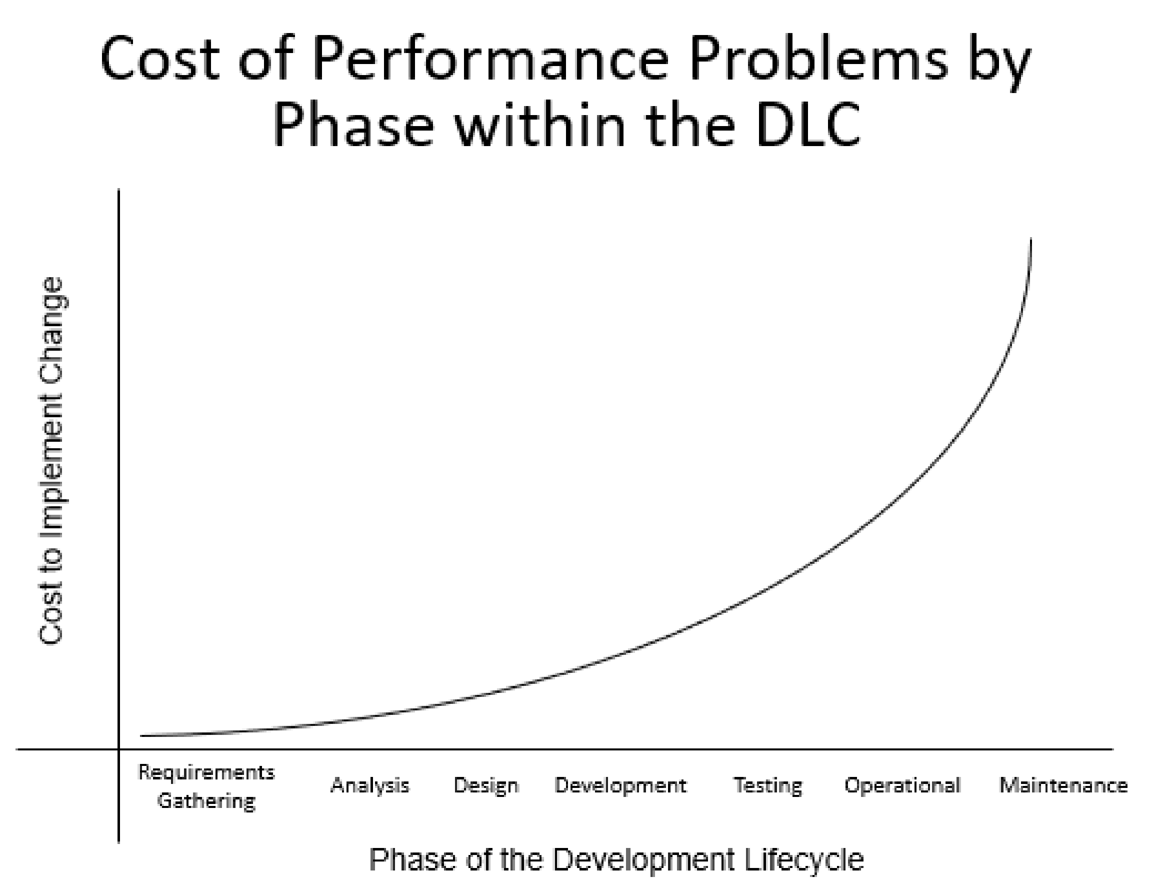
MANAGEMENT INCLUDES

- Monitoring
  - Identifying and finding problems
- Analysing
  - How to fix the problem
- Fixing
  - Resolve the problem and optimize the environment
- Usually reactive response
  - Proactive approach requires skills to identify rising problems before they emerge
- Identify problems early in development!

PROACTIVE VS REACTIVE

Proactive	Reactive
Forethought	Fire fighting
Planning	Problem exits that needs to be corrected
Actions taken before problems occur	Unplanned problems
Automated	Can never be completely eliminated
Minimize reactive actions	

COST TO IMPLEMENT CHANGES



MONITOR AND ANALYZE HISTORICAL TRENDS

- Know what is **normal** so you identify what is **abnormal**
- Resource usage trends and performance statistics over time
- Monitor key performance statistics
- When is database
  - Most active
  - Least active
  - Slower than usual
  - Etc.

SERVICE-LEVEL MANAGEMENT

- “99.95 % uptime from 9:00 to 22:00 on weekdays
- “Average response time is less than 2 sec with 500 or fewer users”
- Service-level management is a measure of operational behaviour
  - “Disciplined, proactive methodology used to ensure that adequate levels of service are delivered to all users”

PERFORMANCE TUNING

- System tuning
  - Database instance or subsystem, system parameters, configurations
- Database tuning
  - Database structure, data organization
- Application tuning
  - Application code and SQL

DBMS PERFORMANCE MANAGEMENT

- Basics:
  - Don’t over tune
  - Remain focused
  - Communicate clearly
  - Accept reality
- Performance management tools
  - Monitors, estimation, capacity planning
  - SQL analysis, system analysis
  - Reorganization, caching, compression, sorting

CT60A7650 – DATABASE SYSTEMS MANAGEMENT

SYSTEM PERFORMANCE

Lecture

Jiri Musto, D.Sc.

DBMS INSTALLATION AND CONFIGURATION ISSUES

- Different DBMS have different requirements
- DBMS offer system parameters for configuring the databas environment
- Different methods for configuration
  - Execute system procedures
  - Edit files
  - Issue commands at DBMS prompt
  - Assemble parameter specifications within DBMS
- Change parameters or leave default values
  - Defaults are bad in the long run
- Parameters can be changeable dynamically or non-dynamically

DISK STORAGE, I/O AND MEMORY

- I/O operations are the largest bottleneck
- Data can be stored on a disk or cloud
  - Disk is a mechanical device that has limitations
- Faster disks can improve performance
  - SSD vs. HDD
  - Performance comes with a price (literal)
- Memory is consumed by all operations and the lack of memory can be another bottleneck
  - Memory is also used for caches

AREAS OF MEMORY CONSUMPTION

- User connections
  - Each concurrent user requires memory to maintain
- Devices
  - Similar to users, devices using database require memory
- Open databases
  - It is possible to have multiple database open and running on one DBMS, each requiring memory
- Open objects
  - Open database objects (tables, indices, etc.) require memory to maintain
- Locks
  - Each lock held requires memory. The amount of concurrent locks can be limited
- Caches

HOW MUCH MEMORY IS ENOUGH?

- No one correct answer
- Need a balance between cost of memory and return of investment
- You can calculate what you need based on size and usage of database objects
  - And add some extra room

CACHES TO SOLVE I/O PROBLEMS

- There are multiple caches / buffers that can reduce I/O cost
- Data cache
- Procedure cache
- Sort cache
- Internal structure cache
- Database log cache

MONITOR AND TUNE DATA CACHE

- Efficiency of the data cache relies on proper sizing
  - Too large wastes memory, too small requires swapping and frequent writes
- DBMS may offer multiple buffer pools that can be configured
- Or they offer a single data cache per database
- The primary duty of cache is to avoid physical disk I/O operations
- Read efficiency shows the percentage of times data is found in the cache
  - Aim for 80 % or better

$$Read\ efficiency = \frac{(\#database\ IO\ requests) - (\#physical\ IOs)}{(\#database\ IO\ requests)}$$

ADDITIONAL MEMORY TUNING

- Open database objects
  - How many allowed simultaneously
  - Configure settings
  - Better understanding as time goes by
- Database logs
  - Log modifications done to database
  - Avoid disabling database logs when using valuable data, even if they cost memory
  - Use proper checkpoint intervals

DATABASE LOG CONFIGURATION

- Buffers
  - Input and output
  - Logging can be configured to be asynchronous
- Log offloading
  - Archive old logs and switch to new ones
- Log file definitions
  - Dual logs
- Not all operations may be logged
  - For example create index and bulk copy
- Logging can be turned off
  - Proceed with caution

LOCKING AND CONTENTION

- Lock suspensions
  - Process requires a lock that is held by another
  - Suspended until lock is available
- Timeouts
  - Process is terminated because of suspension
- Deadlocks
  - Two or more processes hold locks others need
  - None can proceed
  - Deadlock detection cycle – time interval between checking deadlocks

SYSTEM CATALOG

- Specifies the framework of the database
  - Group of tables and views that have vital details concerning the database
- Location and setup of the system catalog will impact the performance
  - Where to store, how much space to allocate
  - Decisions made during installation
- Should not be directly manipulated unless absolutely necessary
  - A mistake can cause the whole database to be inoperable

OTHER CONFIGURATION OPTIONS

- Nested trigger calls
  - How many nested triggers are allowed
- Security options
  - Users, authorization, rights, external security
- Identity values
  - Configure identity values rather than using default options
- Distributed database
  - Options to connect multiple databases

CT60A7650 – DATABASE SYSTEMS MANAGEMENT

DATABASE PERFORMANCE

Lecture

Jiri Musto, D.Sc.

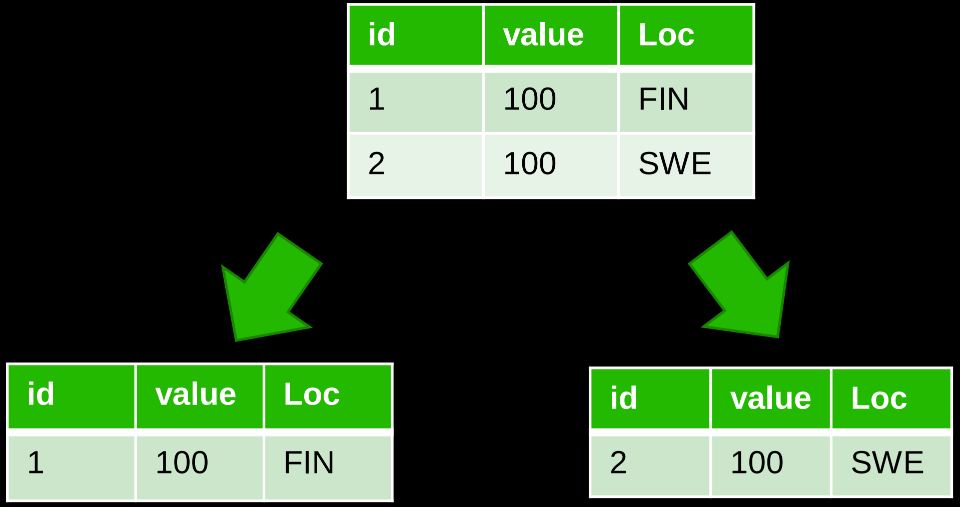
OPTIMIZING DATABASES

- You can optimize database by
  - Making an efficient database structure and
  - Defining better database objects
  - Creating efficient SQL queries
  - Defining indices
- SQL tweaking and system tuning do not resolve poorly designed database

LUT University

PARTITIONING

- Distribute data from one table into multiple sub tables
  - Divide based on ranges or values
- Most used in distributed database systems
- Useful if
  - Different users require different data
  - Different access rights
  - Queries for specific data
- Partitions can be partitioned as well



id	value	Loc
1	100	FIN
2	100	SWE

id	value	Loc
1	100	FIN

id	value	Loc
2	100	SWE

PARALLELISM

- Multiple tasks to access database in parallel
- Partitioning helps
- Parallel request may use multiple simultaneous reads for a single SQL statement
- Can reduce the time used for queries
- Some types of parallelism
  - Single query divided into multiple request
  - Spread work across multiple databases

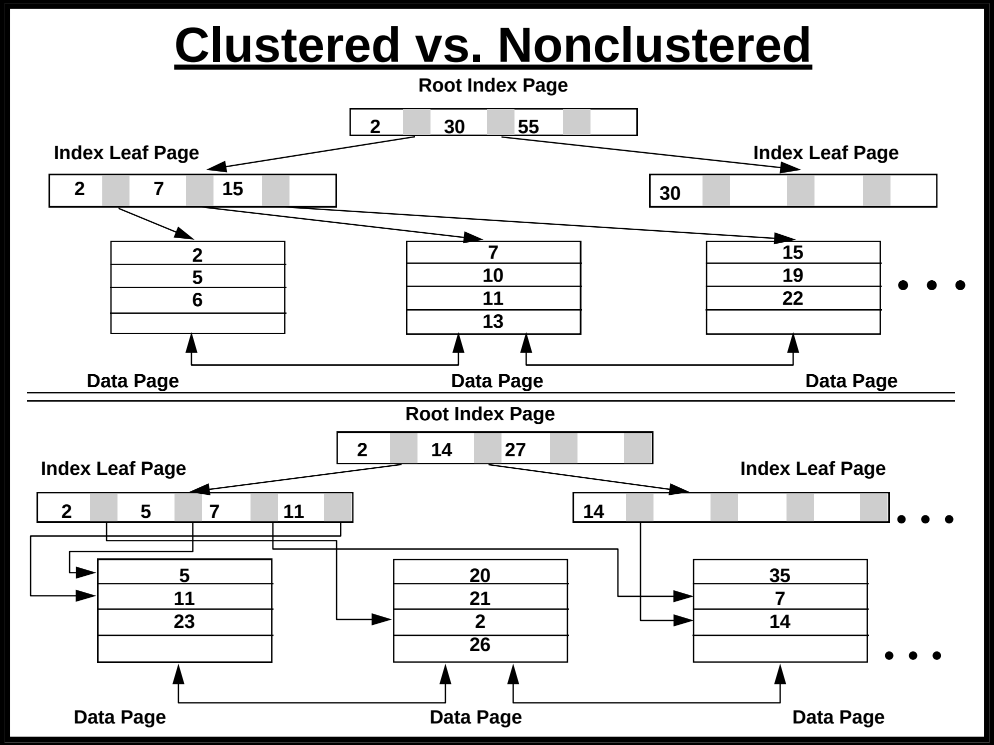
LUT University

USING INDEX

- Single greatest / easiest performance tuning technique
- Indices are between database performance and application performance
- Useful in most databases
  - Provides a sorting mechanism
  - Locate non-primary key values faster
- Understand what types of queries are used
  - Create “correct” indices
  - Can drastically reduce query time
  - Design indices so that they support multiple queries
- Avoid indexing variable length columns or when column has a small amount of possible values

CLUSTERING

- Clustered table stores rows physically in order using specified column(s)
- Usually enforced by DBMS when using clustered index
- If DBMS supports clustering, define clustering for each table created, except for small tables



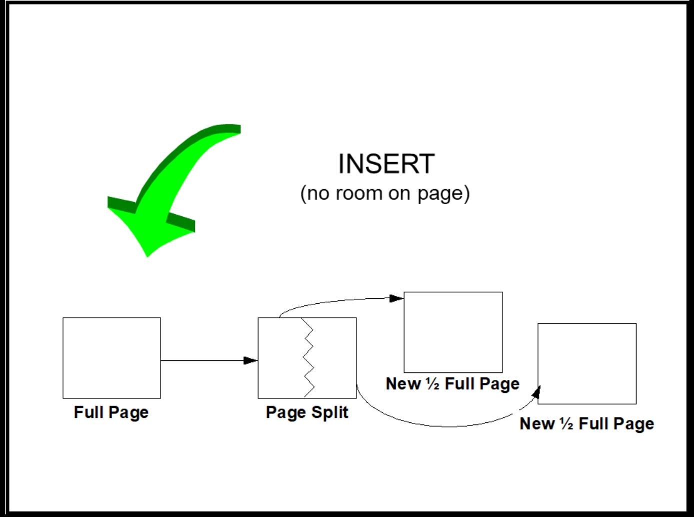
Clustered vs. Nonclustered

PAGE SIZE AND DISK ALLOCATION

- Rows are stored physically in a table space page or block
- DBMS can limit the size of the page per user configuration
- Determine the best page size based on
  - Row size
  - Number of rows per page
  - Free space requirements
- DBMS may require disk allocation for databases
  - DBMS specific commands for initialization
  - Ensure that enough space is free

FREE SPACE AND PAGE SPLITTING

- DBMS can specify how much “free space” to leave for new data
- Reduces the need for reorganization
- If new data is inserted and there is no “free space”, new page is created to store the data
  - This is called page splitting
- Page splitting creates a new empty page and puts half of the existing data to the new page.



PLACING LOGS AND DISTRIBUTED DATA

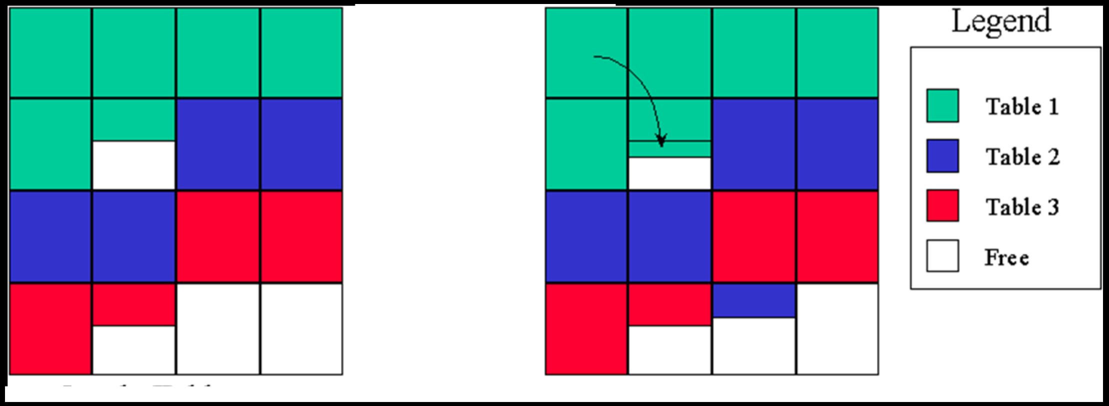
- Database logs can be placed on separate disks from the actual data
  - Minimizes dual writes on the same disk
  - Enables backing up transaction logs independently

- Place distributed data for effective access
  - Close to the user
  - Lappeenranta data should be in Lappeenranta, Lahti data in Lahti

FINAL OPTION: DATABASE REORGANIZATION

- Automate reorganization
- Traditional reorganization requires database to be offline
- Base the reorganization on thresholds
- Plan and schedule reorganization to resolve disorganization issues

- Causes of disorganization
  - Unclustured data
  - Fragmentation
  - Row migration
  - File extents



LUT
University