CPSC 404: Advanced Relational Database Systems

Instructor: Dr. Ed Knorr
Department of Computer Science
University of British Columbia

September-December 2017

Learning Goals for this Unit

- List some job responsibilities of various DB personnel.
- Gain an appreciation for, and explain, the scope and complexity of DB-related jobs in an organization.
- Explain the benefits of logical and physical data independence brought about by the relational model.
- Identify some database challenges in providing 24x7 operations.
- Explain why DBMSs are so hard to configure "properly" for an organization.
- Justify the use of self-managing (sometimes called "autonomic" or self-tuning) database systems.
- Give examples of quantifiable components that determine the "all in" cost of delivering IT or database services.

Introduction

- Course Outline
  - Handout provided; please read it carefully
  - I'll keep it updated on Connect. I'll make an announcement in class, or I'll post a note when any significant change has been made.
- Carl Wieman Science Education Initiative (CWSEI)
  - Part of the Faculty of Science's mandate to improve learning
    - Best practices from evidence-based research
  - www.cwsei.ubc.ca has some great resources; you may wish to explore the site, and especially the Resources page and its links
  - In this course, there will be an emphasis on *deliberate practice* and problem solving
    - We're encouraging students to develop "expert-like thinking" regarding a task.
    - We'll use clickers, peer instruction, pre-reading, pre-class exercises, worked examples, more frequent testing, etc.

Introduction (cont.)

- CWSEI (cont.)
  - Learning Goals—very important!
    - Less of a focus on memorizing facts, and more of an emphasis on: applying concepts and problem solving
    - Bloom's Taxonomy (with a % estimate of typical post-secondary science test questions, shown below)
      - However, CS tends to have more questions in levels 3-6 than other sciences.

Level #	% Questions	Level/Category	Some Verbs Used for Questions
6 (highest)	< 2%	Evaluation	Evaluate, Justify, Argue, Judge, …
5	< 2%	Synthesis	Create, Generate, Propose, …
4	2-5%	Analysis	Analyze, Determine, …
3	12-15%	Application	Apply, Use, Choose, …
2	20%	Comprehension	Explain, Show, Extrapolate, …
1 (lowest)	60%	Knowledge	Describe, List, Name, Identify, …

Course Objectives: We'll Study …

- The big picture:
  - Consider the scope, complexity, and integration of an organization's data assets, including its data management and analysis strategies.
- The relationship among bytes, pages, disks, buffer pools, data tables, indexes, metadata, etc.
- Indexing strategies
- SQL query evaluation and optimization
  - Performance issues
- Transaction Processing and Concurrency Control
  - Schedules, Serializability, Deadlock, Locking Protocols, Isolation Levels
- Crash Recovery and Application Recovery
  - Logging
  - Backups (Image Copies) and Recoveries

Historically, Three Major Types of DBMSs

- From oldest to more recent …
- Network DBMSs
  - e.g., IDMS from Computer Associates (CA)
  - Has lots of pointers, similar in spirit to the WWW
  - More difficult to program, less flexible
- Hierarchical DBMSs
  - e.g., IBM's IMS (Information Management System)
  - Many applications have natural hierarchies of data.
  - Many production systems throughout the world (e.g., banking, government – Government of Ontario is still a huge user), but nowhere near as many as for relational systems
- Relational DBMSs
  - IBM's DB2, Oracle, Microsoft's SQL Server, MySQL, etc.
  - Heavily used throughout the world. Bruce Lindsay, IBM: "Relational databases form the bedrock of Western civilization" (biased quote?).
  - The focus of this course

Newer Technologies

- OODBMSs
  - Object-Oriented DBMSs haven't really caught on, at least not the way some people had predicted
  - Very small market share
- NoSQL
  - Not Only SQL (or in some circles: No SQL)
  - An "eventually consistent" capability rather than the much stronger ACID properties: atomicity, consistency, isolation, and durability
  - Lack of a schema
  - Examples:
    - Cassandra
    - CouchDB
    - Hbase
    - MongoDB
  - Not the focus of this course
- But, RDMBSs are still a huge workhorse, and represent a lot of the data in the dark Web.

Some DB-Related Jobs

- DB-related staff in a large IT shop (for a single company):
  - Systems / Technical Support
    - Installation, configuration, customization, problem determination, interaction with vendor, patches, monitoring, performance & tuning, availability, etc.
  - Database Administration (e.g., DBA = Database Administrator or Database Analyst)
    - Requirements analysis, data modeling (logical DB design), physical DB design, data dictionary, standards, documentation, capacity planning, interact with users & programmers, analyze SQL (recommend changes, add indexes, identify bottlenecks), support business cases, interact with management, etc.
    - Middleware and DBMSs
  - Database Support (often combined with Database Administration)
    - Day-to-day support of production jobs including backups & recoveries, binds & rebinds, authorizations; set up DBs in test & production; do performance & tuning; interact with above groups; etc.
    - Lots of 24x7 action; nightly callouts are not uncommon

Some DB-Related Jobs (cont.)

- Programming
  - Historically, each "area" had its own programming group (e.g., for an oil & gas company: payroll, human resources, accounting, land, exploration, production, ...)
  - DBA interacts with each group regularly
- Space Management
  - Disk drives, tapes, archivals, capacity planning, chargeback

❖ In conclusion, there are many DB-related jobs

- Some large IT consulting shops use people in each of the above categories to service multiple clients, although some customers are so big that multiple consultants are often devoted to—and are permanently stationed at—a single client
- In a small shop, a single employee is often responsible for many of the above tasks.
- Many organizations are interested in cost-recovery for IT operations, and often need a business case to justify new IT expenditures.
  - They're interested in "all in" costs.

Some DB-Related Jobs (cont.)

❖ Companies often save money by outsourcing their IT departments (or major parts of it).

- e.g., BC Hydro → Westech Information Systems and Western Integrated Technologies → BC Hydro → Accenture and others
- e.g., Does each corporation really need to develop, maintain, and support its own proprietary payroll, HR, and tax programs?
  - e.g., BC Hydro moves to PeopleSoft HR software → later Oracle buys PeopleSoft
- Software and hardware change frequently; networks are frequently upgraded; mobile access and security are "moving targets"
- The all-in cost of an employee is approximately double his/her salary.
- Training costs
  - $3000-5000 for a 5-day DB tuning course (or DB conference) in Toronto, *plus* one-week's loss of an employee
- There are trade-offs (e.g., costs, security, privacy, stability, knowledge of the business area, time to respond).
- Ask yourself: Do you want to be a specialist or a generalist?

Data Independence

❖ Joe Hellerstein's Inequality:

- $d\,\text{application}/dt \ll d\,\text{environment}/dt$
- What are the implications for applications that use databases?

❖ From CPSC 304, recall some major benefits of a relational DBMS:

- Sharing, Redundancy, Integrity, Concurrency, Backup, Recovery
- Physical independence (disks and other hardware)
- Logical independence (schema views, no pointer manipulation, a declarative query language)

❖ CPSC 304 dealt with usage of a DBMS; in CPSC 404, we look under the hood (i.e., at the internals)

Availability & Performance

❖ Here are some desirable characteristics of a DBMS to permit high-levels of availability (e.g., 24x7) and performance:

- Perform a schema change *without taking the database offline.*
  - Note that there could be complex changes that affect queries, updates, application programs, optimizations, backup/recovery, etc.
  - Creating a "schema evolution tool" is a very hard problem.
    - Important area of research
- Online backup (i.e., while the data is still "live")
- Online reorganization (of a table and its indexes)
- Add/drop indexes on the fly.
- Optimize the buffer pool to help minimize the number of disk I/Os.
- Be proactive about poorly performing queries or resource hogs.

Self-Managing DBMSs

- Skilled DBAs are hard to find, esp. in small shops where a single person cannot be a jack of all trades.
  - Performance and tuning are important DBMS tasks that require substantial know-how.
  - DBAs tend to work longer hours than most IT people, including some evenings and weekends.
- Many DBMSs have hundreds of tuning parameters, such as:
  - Checkpoint frequency, log size, sizes and types of buffer pools, rollback wait time, how often to check for deadlocks, max # of connections, max # of concurrent users, lock escalation levels, sizes of work files, # of concurrent users, security issues, etc.
  - Hard: What *defaults* should be provided to DBMS customers?
- So … DBMS vendors are putting more effort into self-managing database systems (e.g., IBM's "Autonomic Computing" initiative)

Self-Managing DBMSs (cont.)

- Example 1: When should tables be backed up?
  - Common Practice:
  - Preferably:
- Example 2: When should tables be reorganized, and indexes be rebuilt?
  - Common Practice:
  - Preferably:
- A big part of this course is about performance.
  - What is "good performance"?
  - How do we design for good performance?
  - How can we fix bad performance?

Self-Managing DBMSs (cont.)

- Example: REORG
  - Reorganization of a Table(s) and Index(es)
  - Traditionally, a reorg involved taking the database offline (making it unavailable to users)
  - Better:
    - Make a duplicate of the data and reorganize the duplicate
    - Keep the original online for read and write access
    - When the reorganization of the copy is done, use the database log to apply the latest changes that were made to the production data, to the new copy.
      - Iterate this update process
    - Switch the two datasets
- Lots of vendor or third-party tools to help a DBA

Keeping Systems Up

- [Mullins, 2013]
  - A major vendor reports that 70% of database downtime is due to human error, such as DBA mistakes.
  - Planned outages represent up to 70% of downtime.
  - Up to half of the unplanned outages are due to problems encountered during the planned downtime.
- To help keep systems running round the clock, use automated DBA tools as much as possible.
- Be sure to have well-trained and experienced staff.
- Use clusters of machines and storage area networks (SANs)
  - Can run standby systems
    - Redundant hardware with copies of DB log changes being propagated to other systems, ready to take over, if needed
  - Can provide fast hardware failover support
  - Can take a machine offline for service while the DBMS continues to run