Data Processing Systems

COMP60029

Rahul George (rg922)

Written & Maintained Autumn 2024

Imperial College London

October 20, 2024

Disclaimer!

Content may be wrong, inaccurate, or otherwise lacking. Don't treat these as a sole resource for studying. All credit goes to the lecturers teaching the module, this was constructed using their resources.

Contents

1 Introduction

1.1 Online Lecture

The lecturer, Holger, may not like coffee. He is also a stereotypical German in that he likes efficiency, which is the main goal for the course - we should never sacrifice efficiency when working with databases.

Formally, this module looks into DBMS (database management systems). In context, a database is any structured collection of data points - this of course includes relational tables, but could also be a set, vector, stack of cards, etc. The "management" part refers to the parts of an application that works with data - this may consist of the operations people may want to apply. Note crucially that data management is distinct from data processing, where the former controls the entire lifecycle of data, from it's creation to it's deletion. The latter are a strict super-sets of data management systems, and support a part (or entirety) of the lifecycle.

Data-intensive applications are those which acquires, store, or gather significant amounts of information. Any such application will require data management. In practice, this can be a difficult task. However, classes of applications will share certain characteristics, allowing us to build systems for large problem spaces. Some of the most common patterns include:

a) **Online Transaction Processing (OLTP):** Lots of small updates to a persistent database. Focus is on throughput, and ACID is key.

b) **Reporting:** Running several data analysis tasks, given fixed time budgets. Focus is resource efficiency. Queries are known in advance.

c) **Online Analytical Processing (OLAP):** Running a single data analysis task. Focus is on latency. Queries are ad-hoc.

d) **Hybrid Transactional/Analytical Processing:** Small updates interwoven with larger analytics.

Now, we formally define the data management system. It is a generic system, made from several specific components, that provides some number of the following functionalities: storage; data ingestion; concurrency; data analysis; standardized programming model; user-defined functions; access control; self-optimizing. In a typical application stack, we will have a user interface written with HTML, followed by the core application logic written is some standard programming language, and lastly, a database management system in C or C++.

Some of the non-functional requirements for a DBMS include:

a) **Efficiency:** They should not be slower than hand-written applications.

b) **Resilience:** A system should be able to recover from problems like power outages, hardware faults, and software crashes.

c) **Robustness:** Queries should have predictable performance.

d) **Scalability:** Efficient use of available resources, and increased resources should improve performance.

e) **Concurrency:** Multiple simultaneous clients can be served simultaneously and transparently.