高级数据库系统

胡卉芪
华东师范大学
数据科学与工程学院
hqhu@dase.ecnu.edu.cn

课程简介

- 先修课：数据库概论
  —如何使用数据库系统？
- 本课程：高级数据库系统
  —数据库理论的拓展。
  —数据库系统的一些内部实现方法

本课程的内容

- 数据库系统的设计思想
  - 为什么长这样？
  - 数据库系统的实现
- 主要内容
  - 存储、查询引擎、事务管理、分布式一致性协议

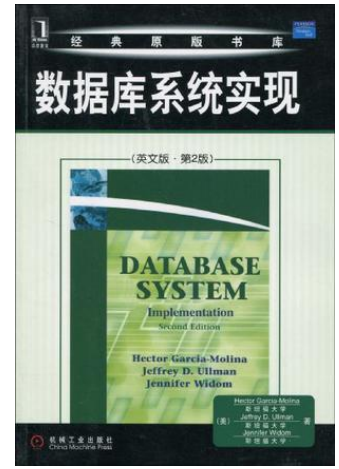
课程地址

- 课程地址（课件&Lab）
  - https://github.com/dase314/db_impl_course/blob/main/README.md

参考书籍？

- 没有特别好的参考
- 参考书
  - 《数据库系统实现》-- (美) Hector Garcia-Molina, et al.

部分内容讲解有些繁琐，有些当下数据库技术缺乏



课程教材（推荐的）

- CMU 15445（网络公开课）
  - https://15445.courses.cs.cmu.edu/spring2023/
- 《Designing Data-Intensive Applications》
  - 数据系统设计的核心要素

我们先做一些回顾

什么是数据库系统？

一种系统软件

数据库系统

- 功能
  - 数据存储
    - 可靠性？、容量
  - 数据访问
    - 表达能力、准确性？

数据库系统

关系数据库系统

ORACLE®

Larry Ellison

IBM DB2

PostgreSQL

SQLite

MySQL®

关系数据库

- 功能
  - SQL表达能力很强
- 性能
  - 满足大部分应用的需求
- 易用性
  - 局限性明显？

时代变化 → 技术更新

数据库的发展历史–简要

- 传统数据库–>NoSQL–>NewSQL

DB	RDBMS	OLTP	OLAP	NoSQL	Cloud&NewSQL
1960s	1970s	1980s	1990s	2000s	2010s
• 64: concept • 65: network • 68:hierarchical	• 70: relation • 74: system R • 74: ingres • 76: ER • 79: oracle	• 83: DB2 • 85: OO • 86: PG • 88: SQLServer	• 93:OLAP • 95:MySQL	• 00:SQLite • 04:CStore • 07:Neo4j • 08: Cassandra • 09:Mango • 09: Redis •	• 10:Hive • 13:F1 • 14: Spark SQL • 14: Aurora • 14: snowflake • 15: cosmos

No-SQL

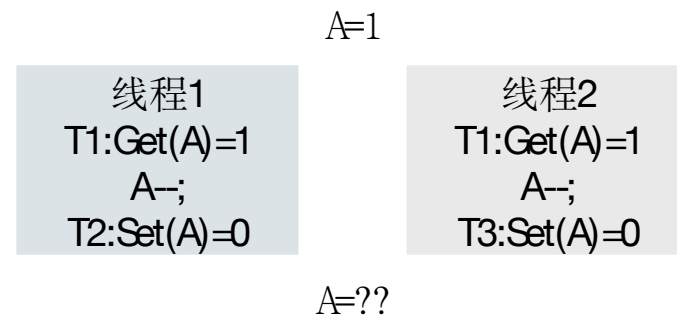

APACHE
HBASE
Cassandra
riak
CouchDB
relax
mongoDB
HYPERTABLE INC
Neo4j
redis

OldSQL vs. NoSQL vs.NewSQL

	OldSQL	NoSQL	NewSQL
Data model	Relational	---	Relational
Interface	SQL	Variance	SQL
Consistency/Concurrency control	Strong	Weak	Strong
Fault tolerance	Strong	Fine	Strong
Performance	Poor	Good	Good
Scalability	Poor	Good	Fine

ACID和SQL的重要性

- ACID的重要性
  - 保证强一致性
- SQL的优势
  - 标准化、通俗易懂、简单易学
  - 没有powerful的查询语言，应用开发不容易



对数据库的需求变化

- 数据量和负载激增
  - 扩展性变得更重要
  - 性价比变得更重要
- 云计算的普及
  - 易用性变得更重要

数据库系统的演进

Google Cloud Platform

如何评价数据库系统性能？

- 性能
  - 响应时间
  - 吞吐率

系统的要素

- 功能
- 性能
- 易用性
  - 访问语言是否容易掌握
  - 管理是否方便

功能与性能折中能否给出一些例子?

更具体一点，看看需要关注什么

1.性能

- 通常有三种性能指标
  - 时延（响应时间）->SQL(查询)的执行时间
  - 吞吐量->事务/简单查询的吞吐率
  - 可扩展性-> 增加计算资源(节点，CPU)后系统性能的增长趋势
- 还有一些特殊的指标
  - 比如？

从系统内部看什么会影响性能？

- CPU/IO/Network Efficiency
  - e.g. code path，cache locality
- Scalability
  - e.g. Blocking，critical path

2.容错

- 系统如何应对节点问题？
  - 节点宕机、重启
  - 网络分区、延迟
- 数据库系统中有何体现？
  - 日志管理
  - 一致性副本

3.数据一致性

- 注意数据一致性在不同场景中有不同意义，常用的三种
  - 读写操作一致性(多核CPU, KVS系统)
  - 读写集合操作(事务 ACID)
  - 副本间数据一致性

4.实现

- 比如：
  - 日志的实现(e.g. WAL)
  - 并发控制的实现(e.g. OCC, 2PL, TO, MVCC)

4实现

- 一个典型的数据库系统架构图
  - 存储
  - 查询
  - 事务
  - 副本

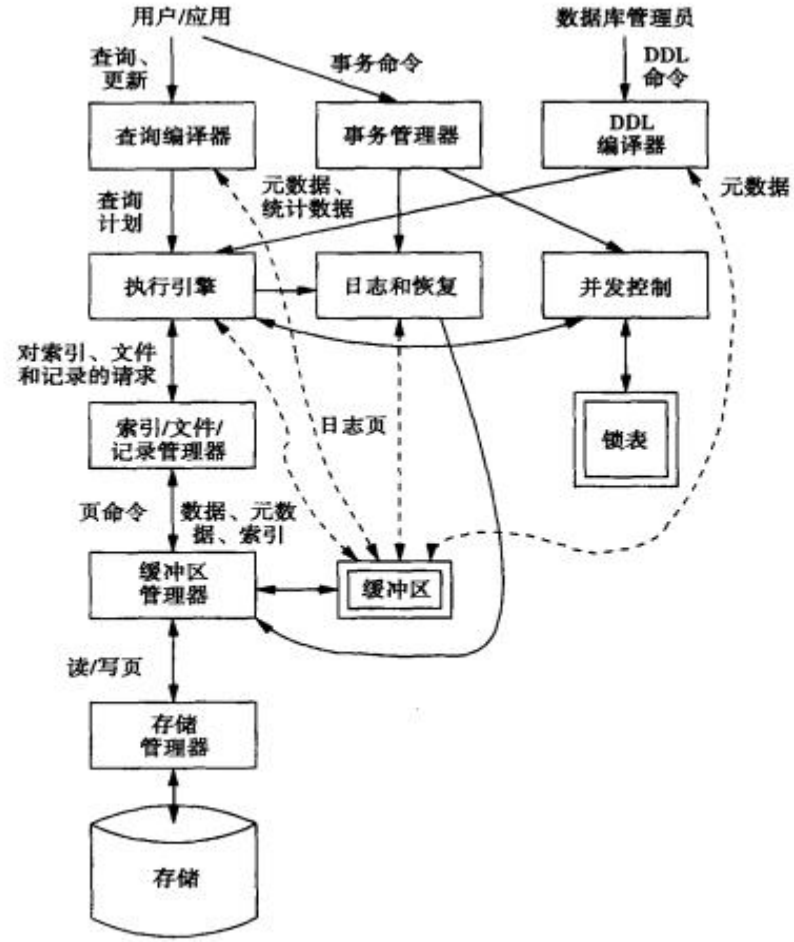


图 1-1 数据库管理系统成分

课程成绩

- A Survey Paper（40%=32%+8%）
- A Project（30%=10%+10%+10%）
- 上机（20%=10%+10%）
- 其他（10%）

Project

- Survey参考的Readling Lists
  - 基础知识
    - https://github.com/pingcap/awesome-database-learning
  - 数据库的研究方向
    - SIGMOD，VLDB，ICDE会议
- 推荐
  - 向量索引
  - 确定性事务执行

https://dase314.notion.site/dede2cf8573544b98c0ccbf380023052?pvs=25