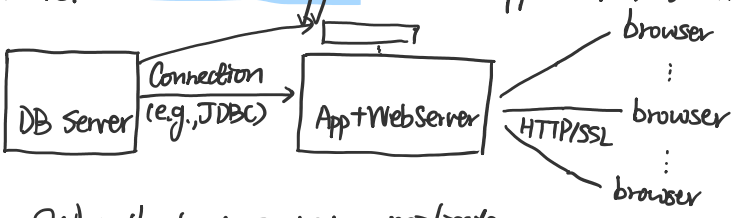
NoSQL

Monday, April 22, 2019 1:58 PM

- Two classes of DB apps
  - OLTP (Online Transaction Processing)
    - Consistency is critical: transactions
      - same item have the same values.
      - weaker physical data independence
    - 0 or 1 join
    - Many updates (e.g. insert order, update payment)
  - OLAP (Online Analytical Processing)
    - a.k.a. Decision Support
    - Many joins, group by
    - No updates.

- DBMS Architectures
  - Serverless
    - Easy Consistency
    - one datafile, user, DBMS app.
  - 2-tier: Client-Server
    (JDBC: Java Database Connectivity)
    - One server running DBMS, RDBMS, Clients connect via JDBC or ODBC protocal.
    - Need to issue consistency.

- 3-tier: Web-based apps. client ↔ app-server ↔ db-server



  - When there're more users, replicate App+Web Server.

    for consistency, we don't replicate DB server.

- NoSQL

  - Give up OLAP, consistency, functionality.
  - simple data model, restricted updates
  - Two basics approach
    - Scale up through partitioning - "sharding"
      replication
      - Consistency is much harder to enforce
  - Scale through Replication

    - Spread queries across these replicas
    - Increase throughput & lower latency

      Increase fault-tolerance.
    - easy for read but expensive to write.
  - Scale through Partitioning

    - Increase throughput
    - Easy to write but expensive to read

- easy to write but expensive to read
- Queries stay across the machines