Database Systems:

Module 12, Lecture 3 – The Relational Problem

Instructor: Alan Paradise

The Relational Problem

LESSON OBJECTIVES

- To be able to describe the challenges that the explosion of "Big Data" brings to traditional **relational database** technology
- Describe various database technology solutions that organizations can implement to handle the explosion of "Big Data"

The Relational Problem

Situation:

- Big Data is exploding all around us, all over the world
- The explosion of Big Data is growing at an increasing rate
- Organizations struggle to cope with Big Data

Problem:

40-year-old Relational Database technology cannot easily handle the volume, velocity and variety of Big Data.

The Relational Problem

The Relational Problem:

The Relational Database is 40+ year-old technology

- Demands STRUCTURE: Tables, Rows, Columns, Keys, Indexes
- Demands ACID Transaction Compliance
  - Keep my data consistent across transactions
  - Consistency VS Fast Performance and Throughput
- Prior RDBMS Software Choices
  - Investment in code
  - Investment in training staff

Traditional Database Server Architecture

- Memory, CPU, Storage
- The cost of scaling “UP" versus Scaling "OUT"

The Relational Problem

Questions:

My organization relies on operational software that sits on top of relational databases.

- How can I expand the capacity of my apps and their databases to handle our Big Data explosion?
- If I shift away from relational DBMS will I have to rewrite all my application code?
- If I shift away from relational DBMS will I have to retrain or replace all my software developers and support staff?
- Can I change my current systems (without replacing my databases) to expand capacity and run our queries faster?

The Relational Problem

One approach:

- Keep my existing Relational DBMS systems and staff
- SCALE my systems to handle more data and run faster

I can SCALE my database servers

- UP: Adding CPU, Memory, Storage ("vertical")
- OUT: Add server nodes to a cluster ("horizontal")

The Relational Problem

Scaling Up may be very costly

- Requires scalable, expandable database servers ($$$)
- Requires purchasing more memory ($$$)
- Requires purchasing more CPU capacity ($$$)
- Requires purchasing more disk storage ($$)

The Relational Problem

Scaling Out is a better solution

- Purchase cheap, commodity server hardware

- Connect multiple database servers together to form a "cluster"
  - Multiple database servers running a single database
  - Working together as a single server

The Relational Problem

Google pioneered the concepts of scaling out clustered database servers

We can learn about clustering from Google's experience

The Relational Problem

Be sure to watch the Google Container Data Center video. (2009, getting started with clustering)

https://www.youtube.com/watch?v=zRwPSFpLX8I

Challenges:

- Heat
- Maintenance
- UPS – Uninterruptible Power Supply

The Relational Problem

Be sure to also watch the Google Data Center tours (9 years later)

https://www.youtube.com/watch?v=zDAYZU4A3w0

https://www.youtube.com/watch?v=XZmGGAbHqa0

https://www.youtube.com/watch?v=avP5d16wEp0

The Relational Problem

In summary

- Google mastered horizontal scaling through clustering
- Cheap, commodity server hardware with directly attached disk storage
- Google figured out how to handle
  - server maintenance
  - heat
  - expandability

The Relational Problem

Next Topic: Scaling Relational Systems