Database Systems:

Module 12, Lecture 4 – Scaling and Clustering

Instructor: Alan Paradise

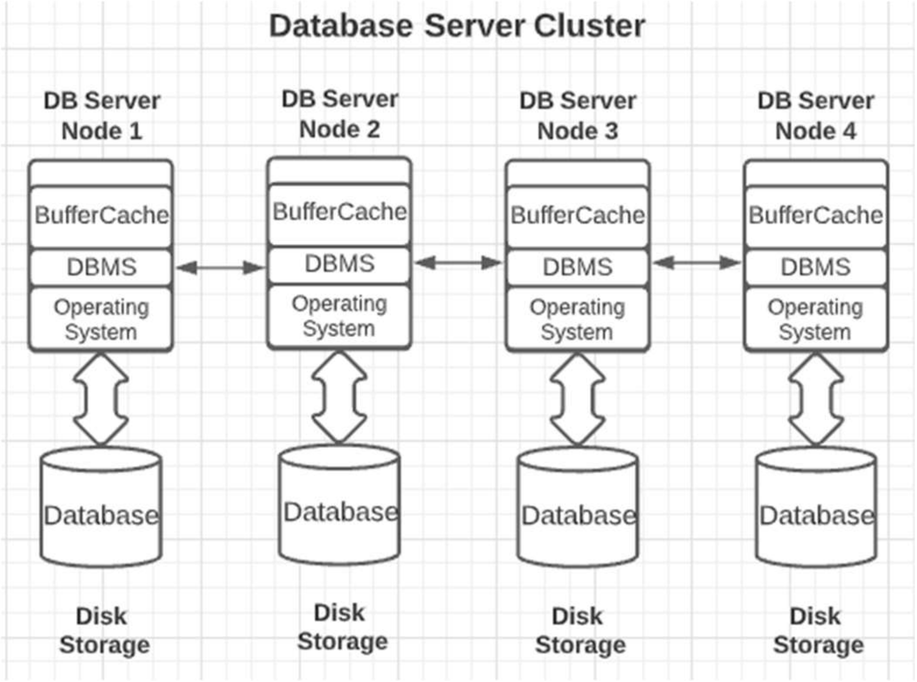
Solving the Relational Problem

LESSON OBJECTIVES

- To be able to describe how scaling and clustering offer a solution to the relational problem
- Describe various approaches to replication, sharding, and clustering

Solving the Relational Problem

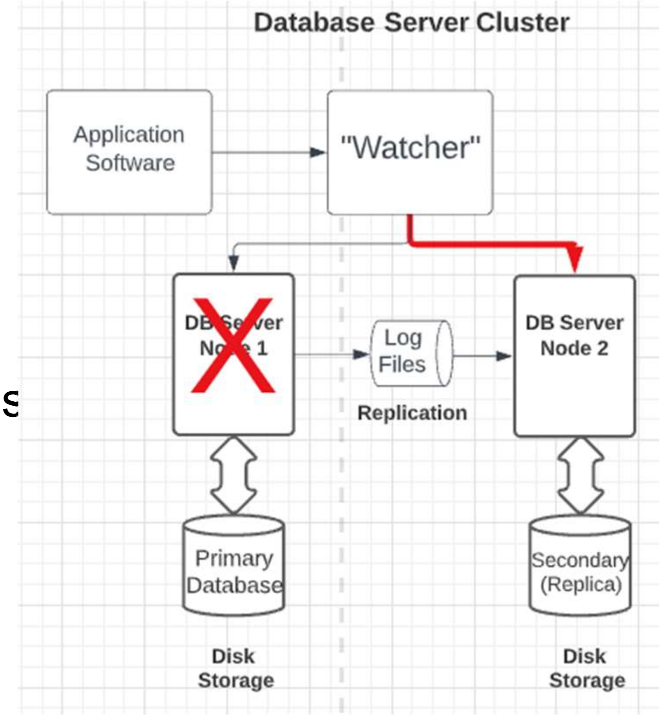
Multiple server nodes in a cluster work together as one



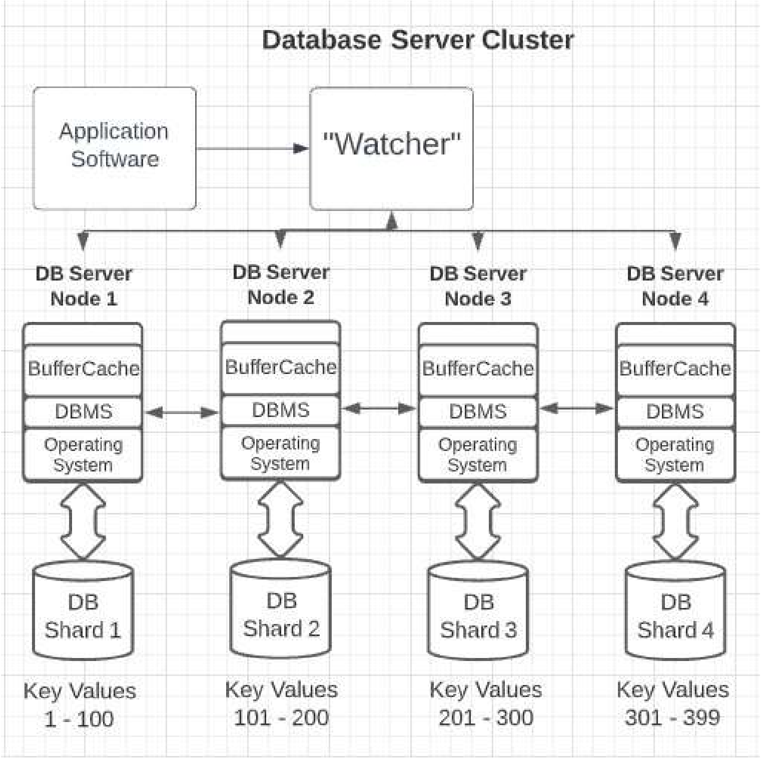
Solving the Relational Problem

The role of the "Watcher" in Replication

- Sometimes called a "coordinator" node
- A component of the replication engine
- A special server node that manages the cluster
- Directs traffic from application software
- Detects if a node goes down, and initiates a FAIL OVER to a secondary node
- Provides HA (High Availability)
- Eliminates any SPOF (Single Point Of Failure)



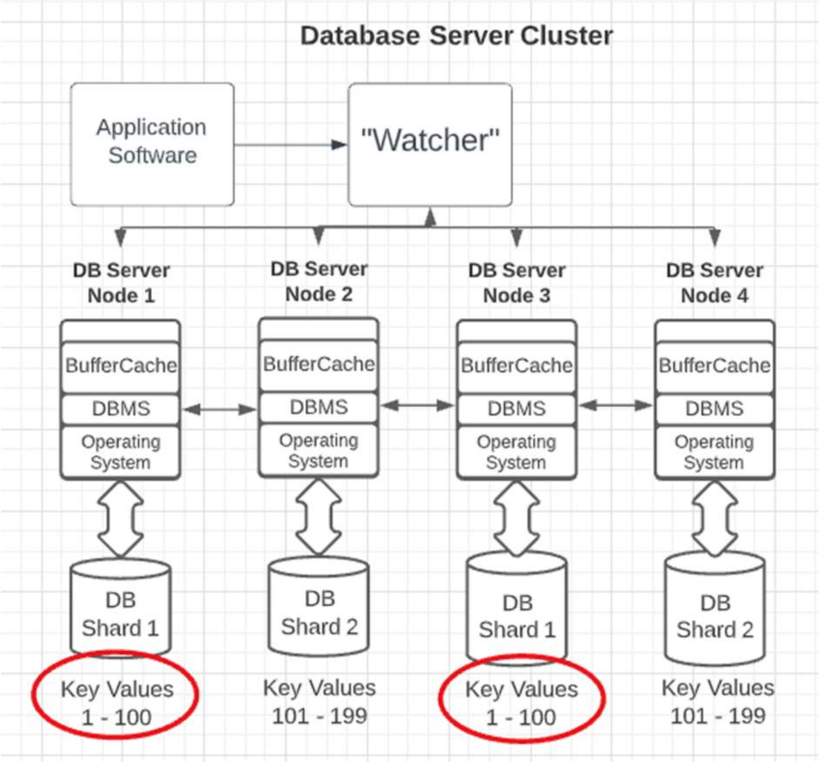
Solving the Relational Problem



Sharding

- Distribute data partitions across nodes
- Based on key range values
- The watcher directs query traffic based upon key values
- Allows for parallelization

Solving the Relational Problem



Sharding + Replication

- Data is distributed via sharding

AND

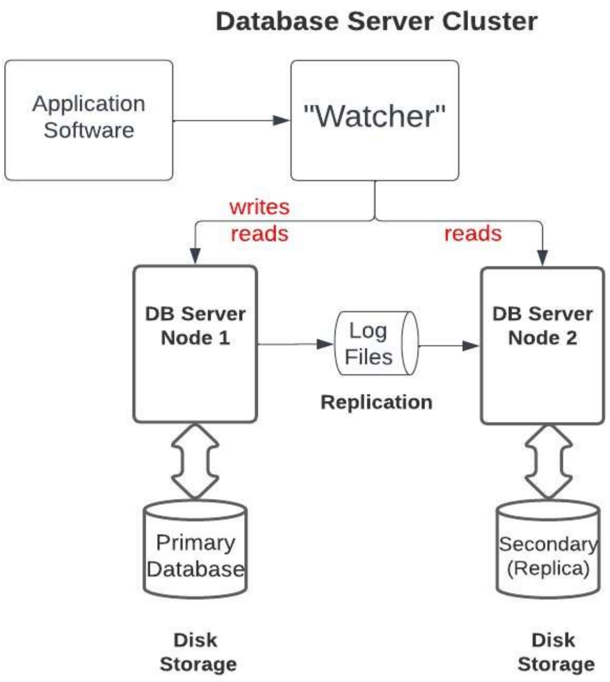
- Data is stored redundantly via replication
- Further supports parallelization and HA
- Eliminates SPOF

Solving the Relational Problem

Two modes of replication

- Primary-to-Secondary (also called "Master-Slave")
- Peer-to-Peer

Solving the Relational Problem



Primary-to-Secondary Replication

Solving the Relational Problem

Primary-to-Secondary Replication

- All UPDATES must go to Primary node
- READ activity MAY be able to run against either node**

Advantages

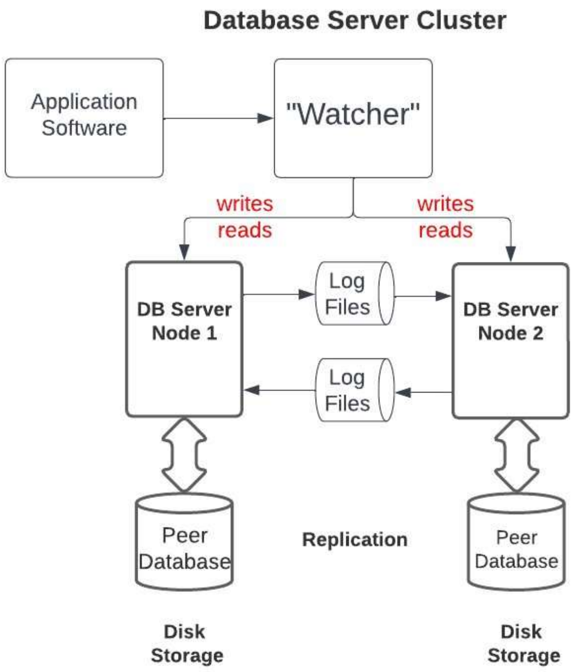
- Good READ Scalability – just add more secondary nodes
- Guarantees UPDATE isolation

Disadvantages

- Constrained by the capacity of the Primary node

** Depends on your DBMS and its replication engine...

Solving the Relational Problem



Peer-to-Peer Replication

Solving the Relational Problem

Peer-to-Peer Replication

- UPDATES and READs can go to any node

Advantages

- Good Scalability – just add more nodes
- Provides robust HA in case of node failure

Disadvantages

- Update propagation is very complex
- Difficult to guarantee update isolation

Solving the Relational Problem

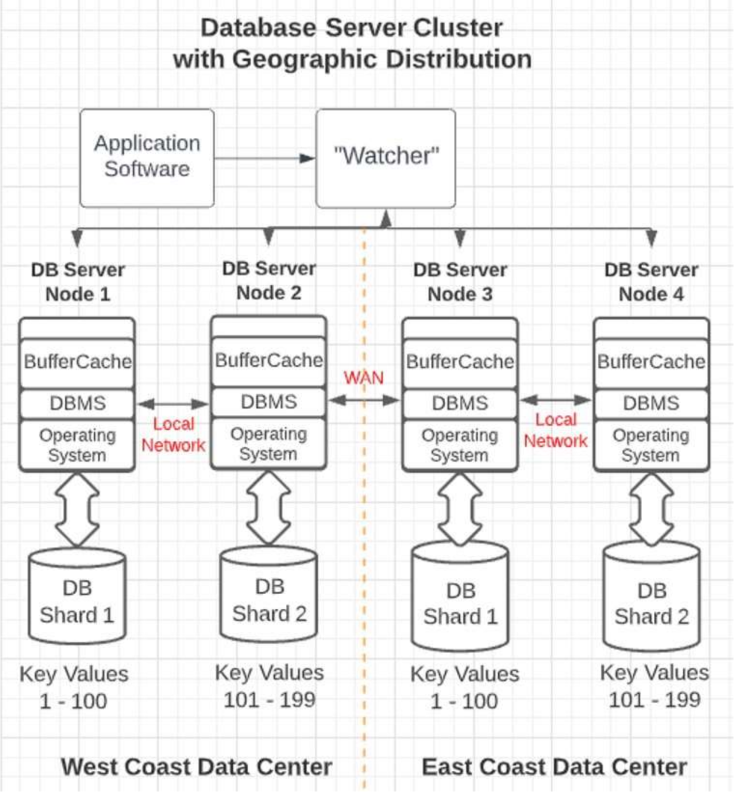
What if replicas are geographically distributed?

- There can be latency (delay) as updates are propagated across a network
- This could cause READ anomalies

Trade-off

- Data Consistency versus Processing Speed

Solving the Relational Problem



Geographically Distributed Nodes

- Update on Node 1
- Isolation Issues: READ against Node 3 before the update on Node 1 is propagated
- What if the WAN is down?

Solving the Relational Problem

Geographically Distributed Nodes

Trade-offs:

- Do I require absolute Data Consistency?
- Do I want to delay remote READs while waiting for update propagation?
- Or, do I seek the fastest possible execution at the cost of perfect consistency?

Solving the Relational Problem

Decision Point:

Enhance my RDBMS architecture to handle Big Data

OR

Abandon Relational DBMS and adopt NoSQL solutions

Solving the Relational Problem

How do we decide?

- Is new, BIG data critical to our organization?
- How much new data is there and how rapidly is it growing?
- Is the data structured or unstructured?
- How fast (or slowly) are our queries running?
- What does it cost to retrain staff?
- What do we save by NOT needing to rewrite all our application code?

Solving the Relational Problem

Pros & Cons

Option 1: Keep Using Relational Database Systems

  - Leverage existing Staff, with some retraining
  - Leverage existing Application Software/Code

- Expand by scaling out (horizontally)
  - Use the Cloud where possible
  - Requires new database server hardware

- Utilize Clustering, Replication, Sharding
- Leverage Parallelization (queries run in parallel on different nodes)
- Relax ACID compliance where possible for faster throughput

Solving the Relational Problem

Pros & Cons

Option 2: Adopt a new NoSQL solution

- Handles unstructured data
- Requires that we retrain or replace staff
- Requires that we rewrite and test existing Application Software/Code
- Requires new database server hardware

- NoSQL Takes advantage of horizontal scaling (clustering)
- NoSQL Utilizes Replication, Sharding, Parallelization
- NoSQL Relaxes ACID compliance
- NoSQL opts for speed over consistency

Solving the Relational Problem

Next Topic: NoSQL Solutions