



Distributed Systems

Krishna Desai
Abdul Mueez

PES1PG20CS013
PES1PG20CS001



Introduction

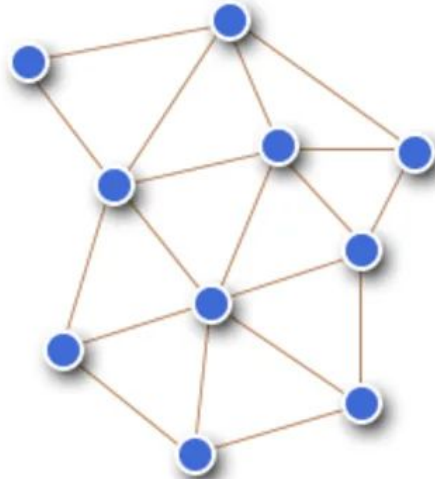
Distributed Systems is the collection of independent computers that appears to its user as one computer.

Characteristics of Distributed Systems

- No global clock : Programs coordinate action by exchanging messages
- Concurrency : Concurrent execution of programs , share resources
- Independent Failures : If one system fails , others may not know

Three majors of Distributed System

1. Storage
2. Computation
3. Coordination



Storage

Problem

With increase in user requests, processing from a single database gets complicated.

Solution

Scaling out database by implementing horizontal scaling

Ways to do Horizontal Scaling

- Adding **read replicas** to handle Read-Heavy workloads.
- **Sharding** your database into multiple servers to improve read-write performance

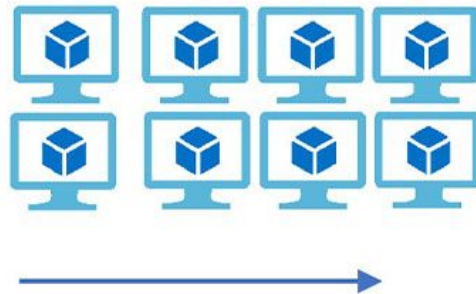
Vertical Scaling

(Increase size of instance (RAM , CPU etc.))

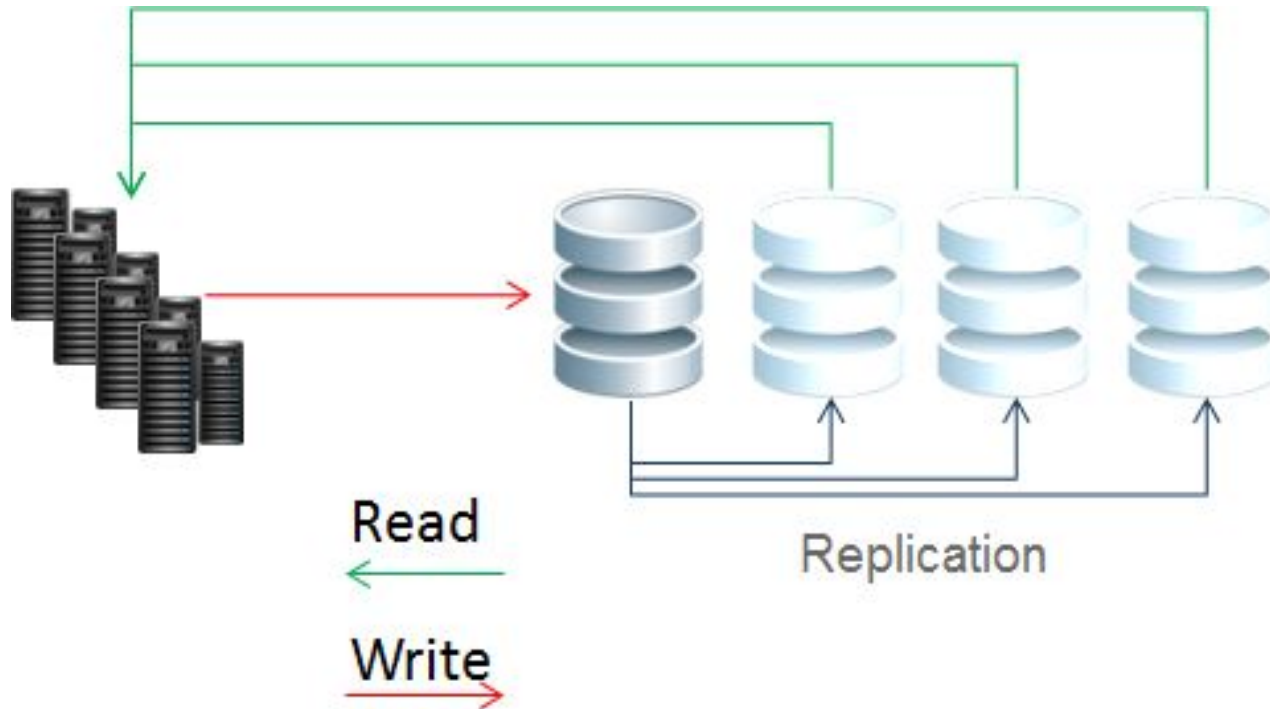


Horizontal Scaling

(Add more instances)



Database Read Replicas



Database Read Replicas



PROS:

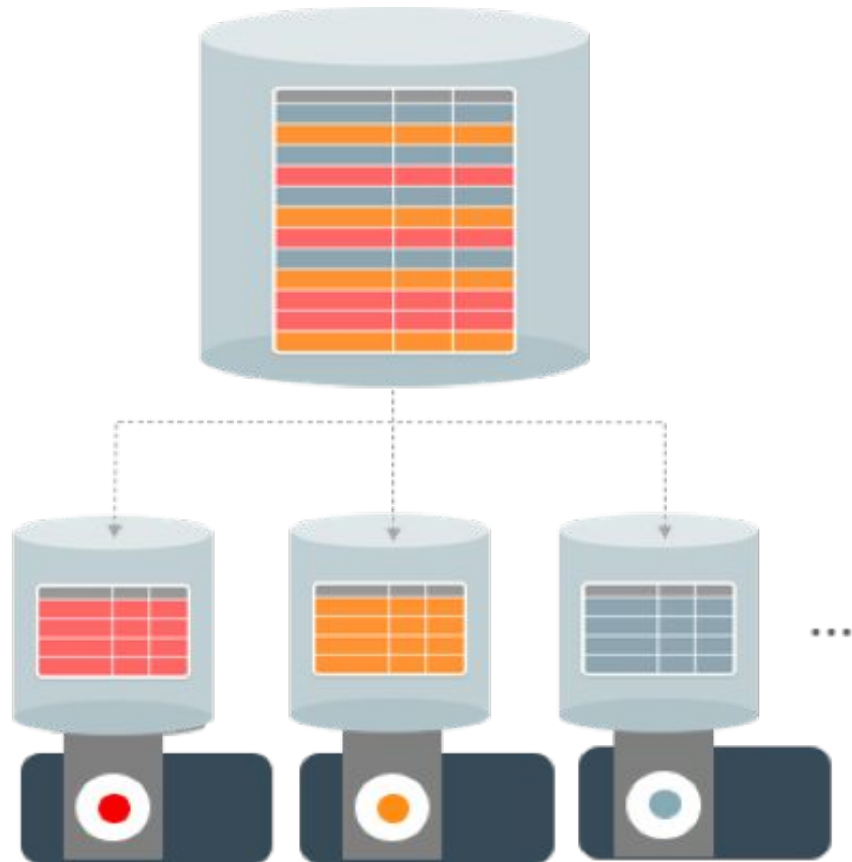
- Any update to the primary DB instance is automatically replicated to the associated read replicas.
- If one of read replica goes down, traffic routes to rest of the available read replicas.
- If primary DB instance goes down, one of the read replicas is promoted as the new primary DB

CONS:

- There is a slight replication lag between the primary database and the replicas
- Data write is performed only on the primary database

Database Sharding

Ex: HBase, MongoDB,
HDFS, Cassandra



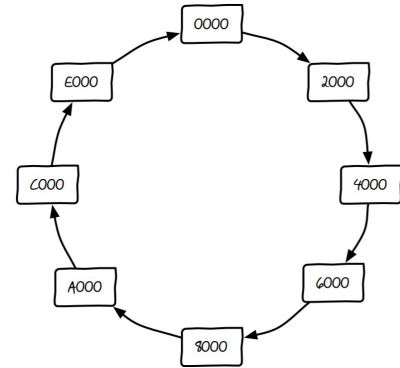
Database Sharding

- Sharding is also referred as horizontal partitioning
- Increases cluster storage capacity, speeds up processing, higher availability
- Data is split into a number of nodes based on a shard key
- Scale out write transaction as well as read transaction from database

Consistent Hashing

Shard based on a hash strategy

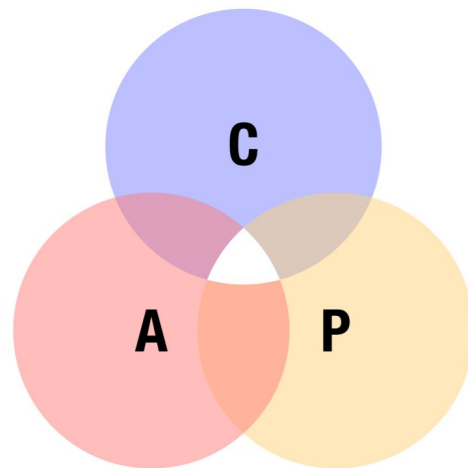
Hash match algorithm for hashing on an entity field



Problem : One of the node fails during write may cause inconsistency

CAP Theorem

- **Consistency:** same data at the same time
- **Availability:** every request gets a result
- **Partition tolerance:** cluster must continue to work despite any number of communication breakdowns between nodes in the system



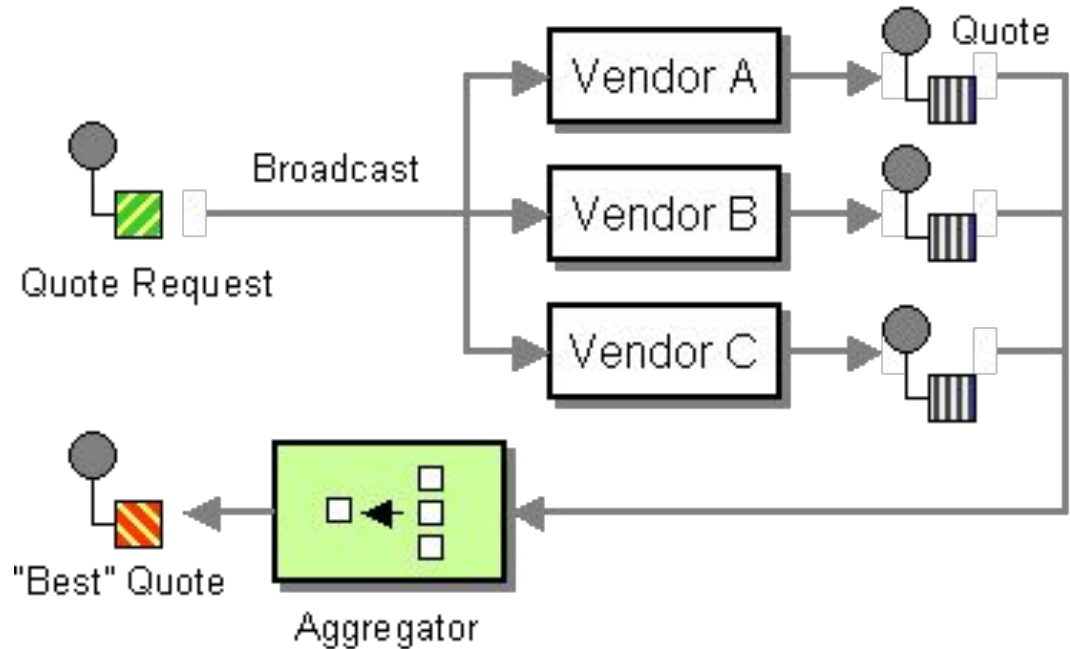
$$R + W > N$$

R : number of Reads
W : number of Writes
N : number of Replicas

Distributed Computation

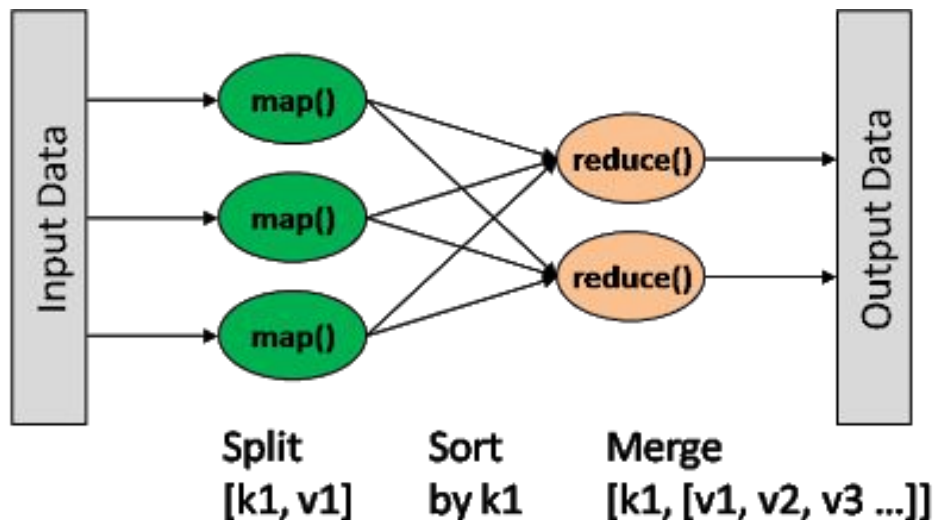
Scatter/Gather

Broadcasts a message to multiple recipients and re-aggregates the response back into a single message.



MapReduce

- It is a way to organize processing by taking advantage of clusters
- It gained prominence Google's MapReduce Framework
- It was then implemented in Hadoop Framework



Hadoop



- An Open-Source software for distributed storage of large dataset on commodity hardware
- Provides a programming model/Framework for processing large dataset in parallel

Key Points

- MapReduce API
- Job Tracker, Task Tracker
- Distributed File system (HDFS)
- Enormous Ecosystem

Spark



- A big data analytics cluster-computing framework written in Scala
- Provides in-memory analytics based on RDD
- Highly compatible with Hadoop Storage API

Key Points

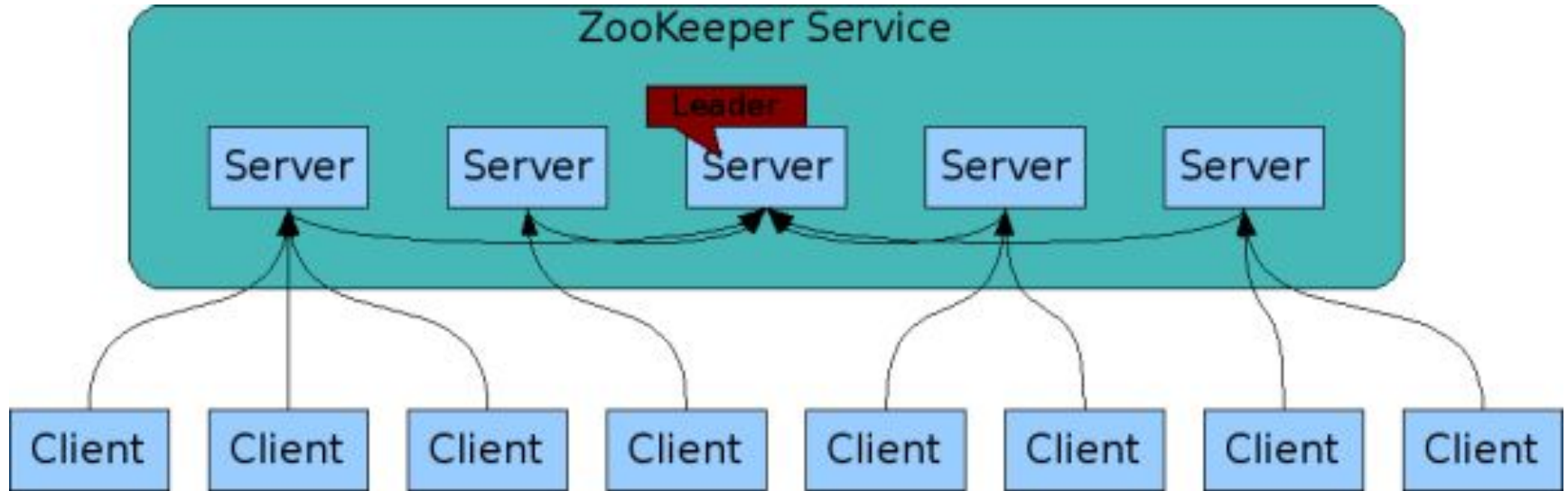
- RDD: Parallel collection with partitions
- Results in a DAG(Directed Acyclic Graph) of operators
- DAG is compiled into stages
- Each stage is executed as a series of Tasks
(one task for each partition)

Distributed Coordination



- The coordination part of a distributed system handles the communication and cooperation between processes.
- **ZooKeeper** is a distributed, open-source coordination service for distributed applications.
- Coordination services are especially prone to errors such as race conditions and deadlock.
- The motivation behind ZooKeeper is to relieve distributed applications the responsibility of implementing coordination services from scratch.

Distributed Coordination





Conclusion

In the words of Leslie B. Lamport, an American computer scientist, best known for his seminal work in distributed systems,

“A distributed system is one in which the failure of a computer you didn’t even know existed can render your own computer unusable.”



Thank you!