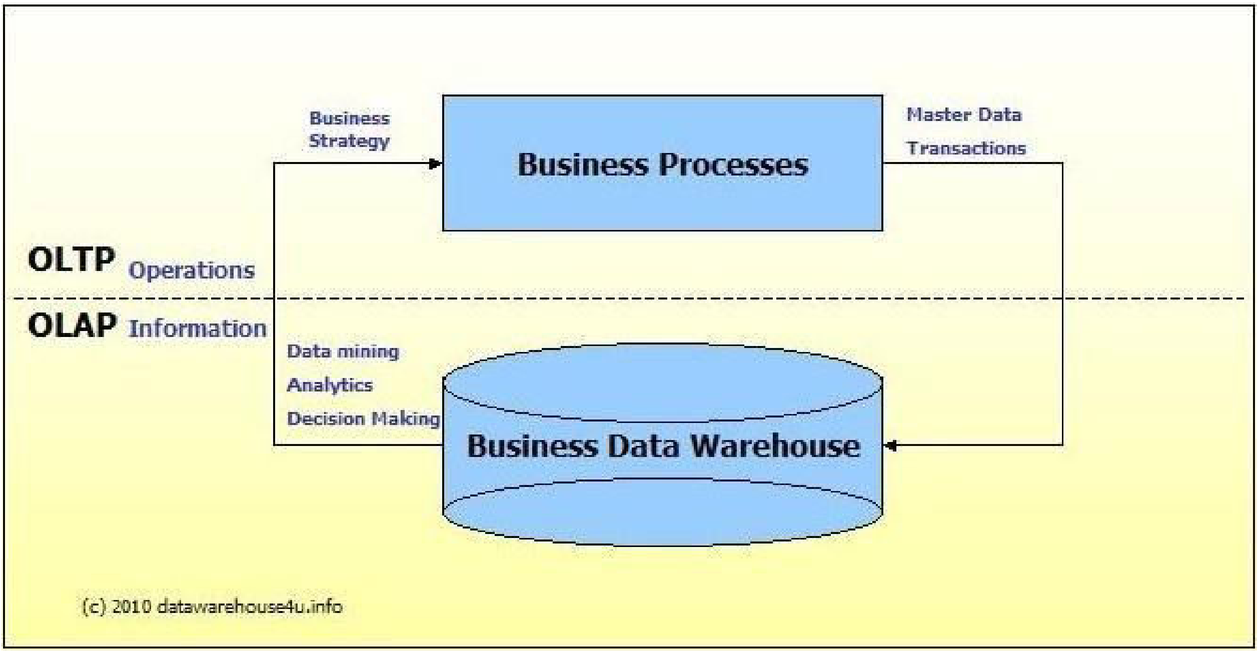
NoSQL databases and MongoDB



ACID properties :

1. Atomicity : Transaction is a logical unit of work. An entire transaction should be completed, with all its data operations or the entire transaction must not be performed,i.e no data operation.
2. Consistent : This means that the integrity constraints must be maintained before and after a transaction leading to correctness of database. Example : Total amount in 2 accounts with transactions must be equal.
3. Isolated : All transaction data operations are independent of each other and occur concurrently without incosistency in database state.
4. Durability : The effects of a transaction must be permanent sand non-volatile. All data updates and modifications to database are written to disk and they persist even after system failure.

Distributed systems :

Multiple computers and software components connected through a network to share resources and achieve a common goal.

Advantages :

1. High computing power and speed
2. Reliability : Even if 1 computer crashes, rest can keep on working
3. Scalability : More machines can be added
4. Resource sharing
5. Flexible

Disadvantages :

1. Troubleshooting
2. Less software support
3. Networking
4. Security

NoSQL :

- No relational database management system.
- No fixed operations, avoid joins and scales horizontally

- NoSQL evolved to manage and process huge amounts of data

RDBS vs NoSQL

Property	RDBMS	NoSQL
Full form	Relational Database Management System	Not only SQL/non SQL/not relational DB
Data	Structured and organized data	Unstructured and unpredictable data
Data storage	Data and relationships in tabular formats	Structured, unstructured, semi structured data forms for storage
Schema	Fixed/static predefined schema	Dynamic schema
Hierarchal data storage	Not suited	Suited
Complex queries	Suited	Not suited
Scalability	Vertical (Increase RAM, CPU, SSD)	Horizontal Sharding/add more servers
Theorem and properties	ACID properties	CAP theorem and BASE propert
Consistency	Tight	Eventual
	Thousand transactions per second	Million transactions per second

CAP theorem :
This theorem states that in a distributed system with replication and network shared data systems all 3 – consistency, availability, partition tolerance, cannot be guaranteed. Only 2 of them can be guaranteed.
1. Consistency : All nodes have access to the same copy of the data item (the most recent write) for various transactions. Same view of data for all clients. (Diff from ACID C)
2. Availability : System is always available to take requests,no downtime. Each read/write request will either be performed successfully or an error message saying that the operation hasn’t been completed is given. “Every” node gives a response in a reasonable amount of time.
3. Partition Tolerance : It means that a system can continue operating even if a network fault arises leading to 2 or more partitions, where each node can only communicate in its partition.

BASE properties : (gives up on consistency, is AP)
1. Basically available : System is guranteed to be available even in times of failures.
2. Soft state : The state of a system may change over time even without any application input due to the eventual consistency model.
3. Eventual Consistency : System will be consistent after some time, given that the system does not receive input in that period of time.

NoSQL features :
1. Horizontal linear scaling
2. Dynamic data models – schema less and flexible
3. Built in replication and failover abilities with fast fault recovery
4. Simple API

5. High throughput – million transaction per second – simultaneous users

Disadvantages of NoSQL :
1. No complex querying or joins
2. No standardization

NoSQL data models :
1. Key-value stores
2. Document stores
3. Wide column stores
4. Graph stores

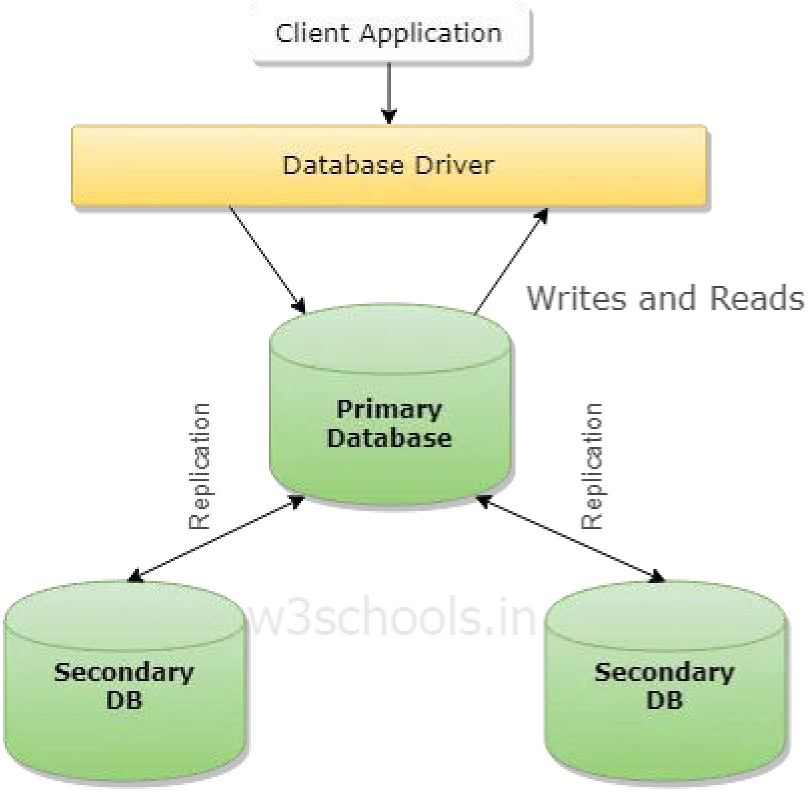
1. Key-value stores : Contain key value pairs and is schema less. Example : Amazon SimpleDB,riak
2. Document stores : Data is stored in documents in Javascript like JSON and schema less. Popular for web frameworks. Example : MongoDB, CouchDB
3. Wide column/Extensible record stores: Semi structured (not schema free) Group of columns(column families) must be specified. Master column names, row may have different column names. Examples : Cassandra
4. Graph databases : Focus on relationship between entites rather than entities. Social graphs. Data is stored as node objects. Examples : Sones, OrientDB

MongoDB :
1. Open source
2. Document store NoSQL data model
3. High performance data persistence (support embedded data models and indexing)
4. Automatic scaling : Horizontal scaling, sharding, replica sets with eventual consistency reads
5. High Availability : Replication facility called “Replica Sets” provides “automatic failover”

SQL	MongoDB
Database	Database
Table	Collection
Row	Document
Column	Field
Join	Embedded document and linking
Primary key is any unique column or column gp	Primary key automatically set

Replication in MongoDB
Objective :
1. Synchronization of the same data on multiple servers to ensure availability of data across multiple servers. Thus enables protection during system failure, disaster management and data backup.
2. Load balancing : Users can be connected to multiple servers for equal distribution of load.
3. Data kept safe with redundant backup.
4. Minimize downtime for maintainence

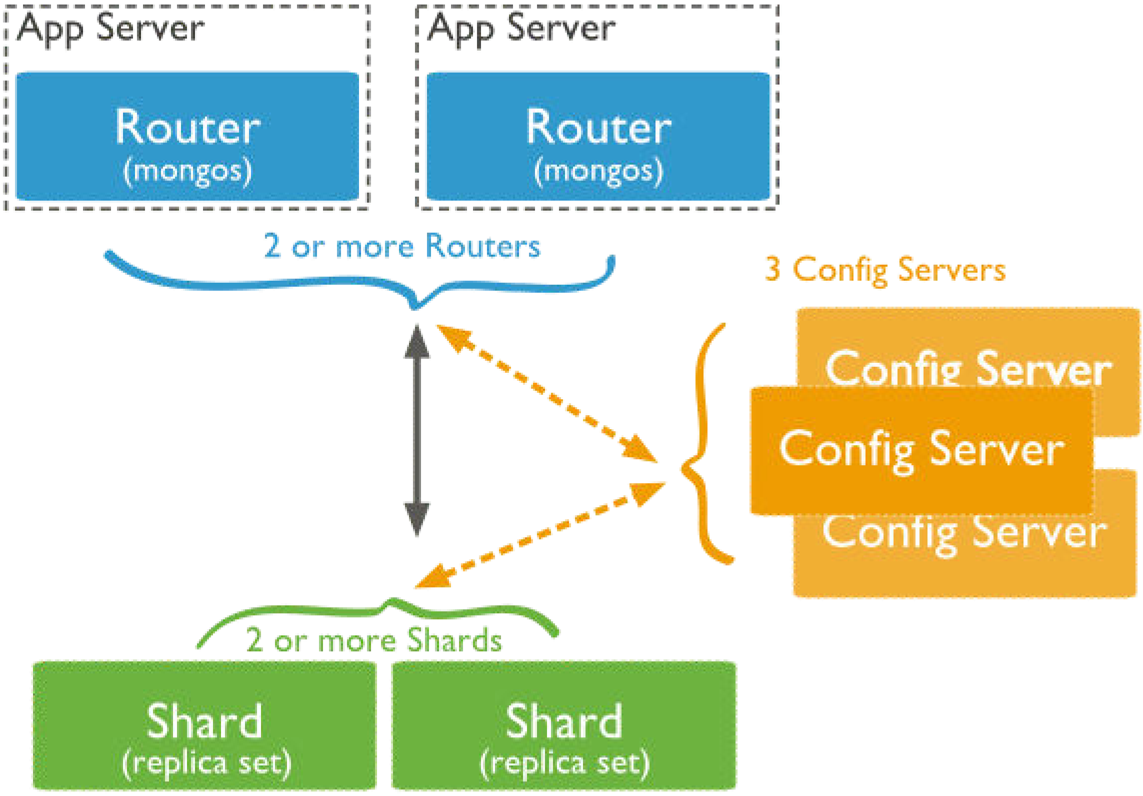
Disadvantage :
1. More space and processing required.
Working:
MongoDB uses “replica sets” to achieve replication. Replica sets are a collection of MongoDB instances which aim to host the identical data.
1. To perform this, at least 3 nodes are required.
2. One node acts like the primary node, the rest like the secondary nodes.
3. All transactions take place on the primary and data replicates in the secondary.
4. In case of automatic failure or maintainence of the primary node, another secondary node is elected to be the primary node. The failed node, after data recovery, acts like a secondary node now.



The Replication in MongoDB

Sharding :
Process in which large datasets are split into smaller datasets which are stored on multiple mongodb instances. This is done because querying on large datasets leads to high CPU utilization. Horizontal scaling

In Mongo, sharding breaks collections into smaller collections called shards.



1. Shard : small data subset for high data availability and consistency
2. Config server : Stores metadata regarding what subset from the database is stored in what shard
3. Query Router : interface to client application responisble for navigating to correct shard