NoSQL and MongoDB

An Introduction

NoSQL: Features

- NoSQL stands for “Not Only **SQL**”
- Next Generation Database
  - Non Relational
  - **Distributed**
  - **Horizontally Scalable**
  - Can be deployed on **Commodity Hardware**
  - Can handle huge amount of **Semi-Structured and Unstructured** data

NoSQL: Web Scale Database

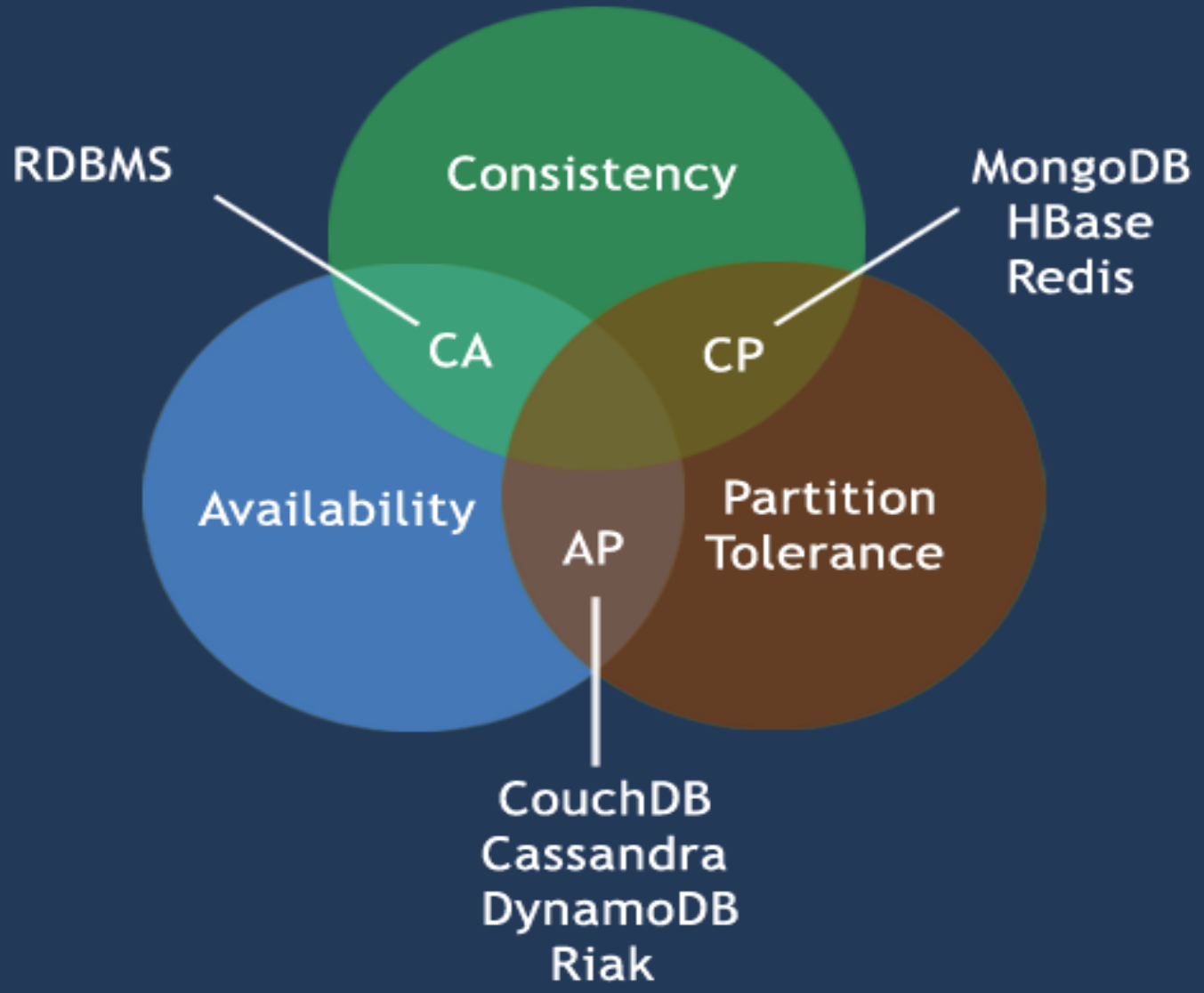
- Modern Web Scale Database
  - Simple **API**
  - **Schema-free** (flexible schema)
  - Easy **replication** support, Automatic **Failovers**
  - Follows Eric Brewer's **CAP theorem**
  - **BASE** Compliant, NOT **ACID**

NoSQL: CAP Theorem

- Applicable on Distributed Database Systems.
- Also known as Brewer's Theorem – Named on Scientist Eric Brewer
  - **Consistency -** Every read would get the most recent write. Commits are atomic across the distributed system.
  - **Availability -** Every node (if not failed) always executes queries.
  - **Partition Tolerance -** Even if the connections between nodes are down, the other two (A & C) promises, are kept

"No Distributed Database System can met all three requirements (CAP) simultaneously" **(maximum two can be achieved simultaneously)**

CAP Theorem


RDBMS
Consistency
MongoDB
HBase
Redis
CA
CP
Availability
AP
Partition
Tolerance
CouchDB
Cassandra
DynamoDB
Riak

NoSQL: BASE Model/Properties

- BASE Compliant
  - **B**asically **A**vailable
  - **S**oft state
  - **E**ventually consistent
- **Basically Available** indicates that system should be available all the time, though few nodes may be down.
- **Soft state** indicates that the state of the system may change over time, even without input. This is because of the eventual consistency model.
- **Eventual consistency** indicates that the system will become consistent over time, given that the system doesn't receive input during that time.

NoSQL Databases: Types

- Key-Value Store
  - Simplest NoSQL databases
  - Every single item in the database is stored as an attribute name (key) together with its value
  - e.g. BerkeleyDB, Oracle's NoSQL Database
- [Wide] Column Store
  - Optimized for queries over large datasets
  - Store columns of data together instead of rows.
  - e.g. HBase, Apache Cassandra, Cloudera

NoSQL Databases: Types

- Document Store
  - Pair each key with a data structure known as a document
  - Documents can contain many different key-value pairs or key-array pairs, or even nested documents
  - e.g. **MongoDB**, Apache **CouchDB**
- Graph Databases
  - Used to store information about networks, such as social connections.
  - e.g. **Neo4J**, HyperGraphDB

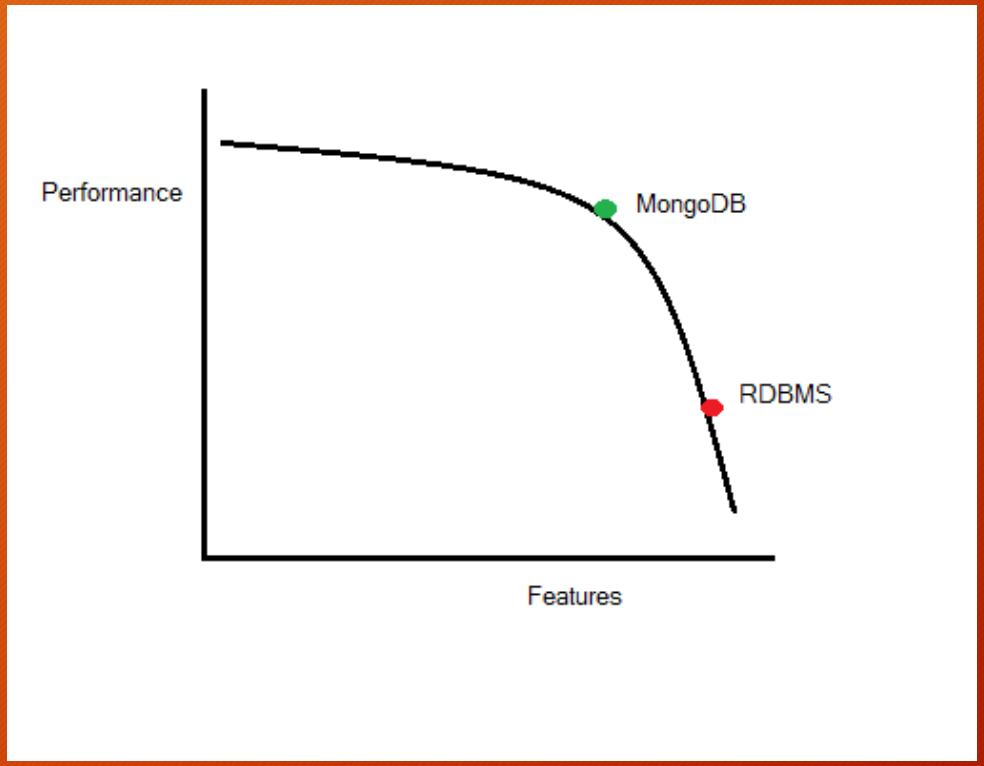
MongoDB

- Open Source
- Free Software Foundation's GNU AGPL 3.0 License
- Commercial License also available from MongoDB Inc.
- Available for Windows/Linux/Mac/Solaris etc.
- Also available on Cloud as a Service through MongoDB Atlas
- Supports Big Data and Map-Reduce
- Uses Document-Oriented Model

MongoDB

- Highly Scalable
- Distributed Horizontal Scaling (Scale-Out)
- High Performance
- High Availability through Replica Sets
- Advanced GUI, Monitoring and Backup Service
  - MongoDB Compass - GUI
  - MongoDB Monitoring Service

Performance Vs. Features



MongoDB: Features

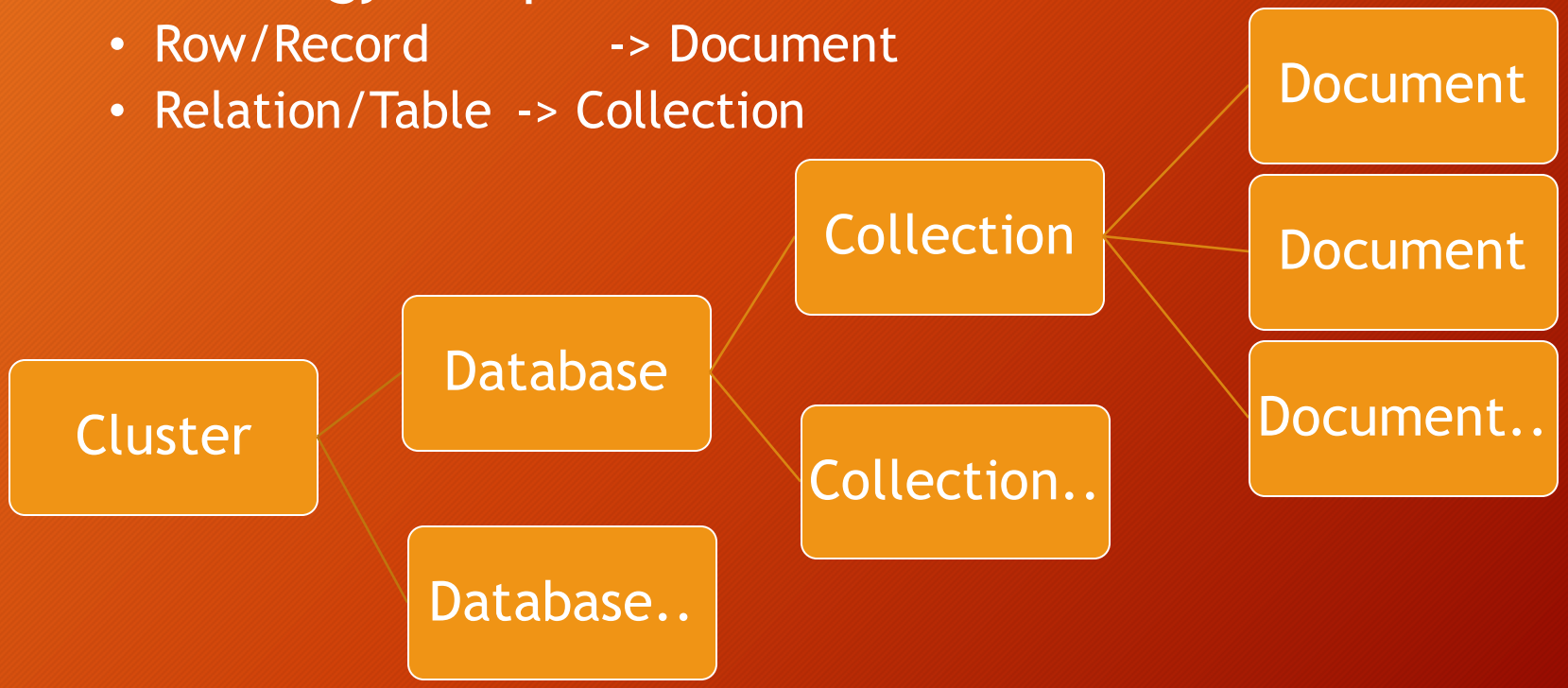
- Provides **Consistency** & **Partition Tolerance** as per CAP Theorem. Some data may not be available, but the available data is Consistent & Accurate.
- Handling of complex/unstructured data possible
- Fast application development
- Does NOT support Complex Transaction
- Does NOT support Joins
- Supports Document Size upto 16 MB

MongoDB: Document Oriented

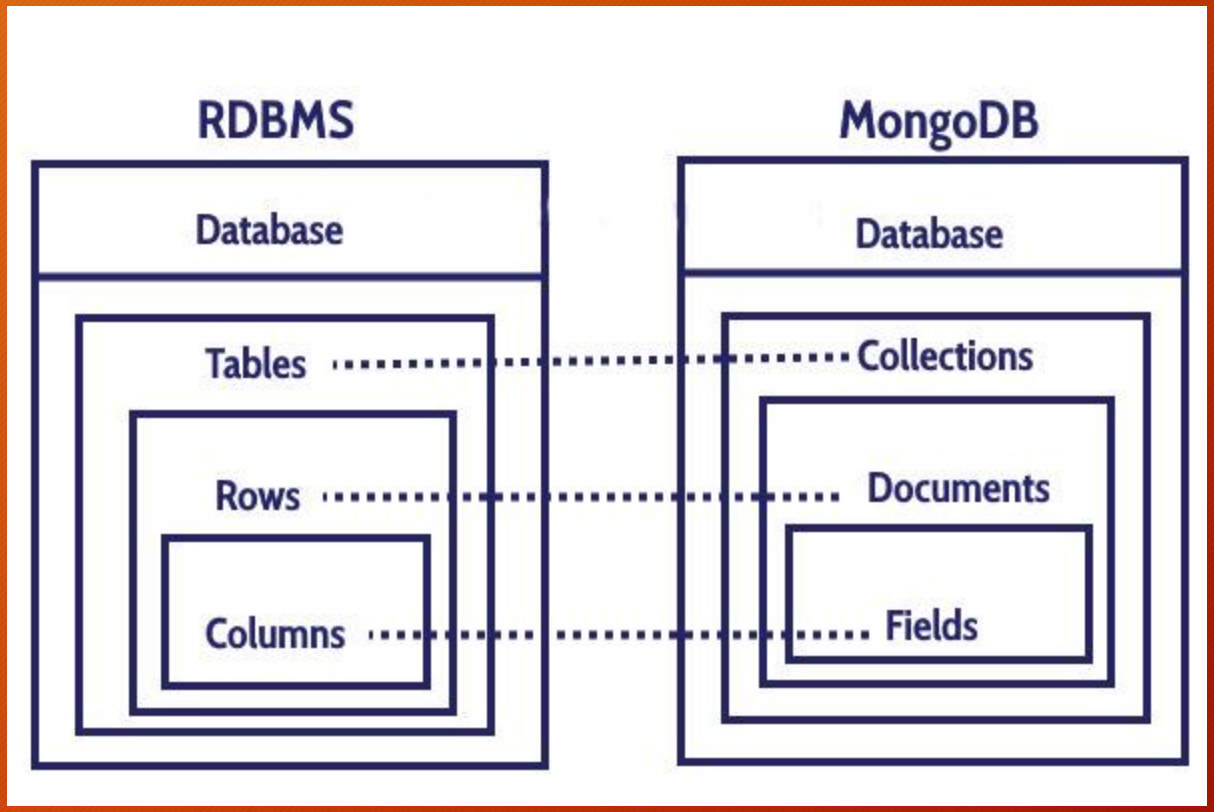
- Stores data in Key-Value pair, grouped in documents
- Documents are of two types –
  - JSON : Java Script Object Notation
  - BSON : Binary JSON

MongoDB Object Hierarchy

- Terminology compared to RDBMS
  - Row/Record -> Document
  - Relation/Table -> Collection



RDBMS vs MongoDB: Terminology



JSON Sample Document

```
JSON  {
        id : “A1”
        X  : 3
        Y  : “abc”
        Z  :  {1,2}
        E :
}
```

JSON

- Supports Nested Documents
- Support Arrays
- No Relations – No Joins

Starting & Connecting with MongoDB

- mongod -> binary to start Mongo Process/Demon
- mongo -> binary to start Mongo Shell
- To Start Mongo Process -

  > **mongod**
- Mongo db runs on Port : 27017 by default
- To connect to Mongo Shell -

  > mongo

  OR

  > mongo localhost/demo