NoSQL Databases

Agenda

- Why NoSQL databases and challenges with RDBMS
- Different RDBMS cluster approaches and methods to scale it.
- NoSQL, its Features and Types.
- Advantages and disadvantages of NoSQL databases.
- CAP Theorem
- Key values databases, document databases, columnar database and graph databases

Why NoSQL?

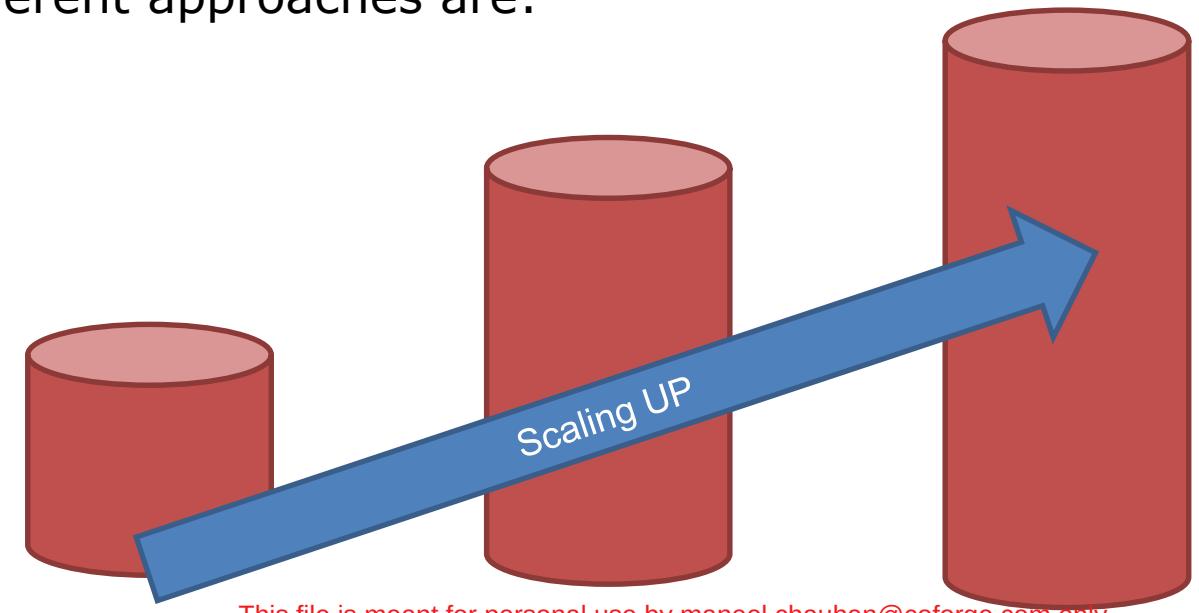
- Relational databases are mainstay of business.
- Spikes were caused by web-based applications.
- Growth of social media sites (Facebook, Twitter) with large data requirements.
- Example of such data : Personal user information, geo location data, social graphs , user generated contents , machine-logging data, sensor generated data etc.
- The rise of cloud-based solutions like Amazon S3 (a simple storage solution).

Challenges with RDBMS

- Hooking RDBMS to web-based application becomes trouble.
- RDBMSs are now being fronted with memcache or other caching mechanisms are integrated within the application(i.e.. Ehcache).
- As datasets grew, the simple memcache/MySQL model (for lower-cost startups) started to become problematic.
- Hence, developers look forward to improve existing applications and develop new applications which can meet the needs of Big Data.

Issue with Scaling Up

- Scaling up when the dataset is just too large.
- The RDBMS was not designed to be distributed.
- Examined solutions for multi-node databases.
- A horizontal scaling process is known as scaling out.
- Among the different approaches are:
- Master-slave
- Sharding



Different RDBMS Cluster Approach

Master-Slave

- Reads are performed against replicated slave databases. All writes are written to the master.
- There is a chance that critical reads are incorrect because writes may not have been propagated.
- Data sets with large amounts of data can pose problems as the master must duplicate the data to slaves.

Partition or sharding

- It scales well both for reading and writing.
- Partition-aware applications are not transparent.
- Having relationships/joins across partitions is no longer possible.
- Referential integrity is lost across shards.

Some other methods to scale RDBMS

- Replication with multiple masters.
- Only INSERT, exclude UPDATES/DELETES.
- No JOINs, thus reducing query time.
- De-normalizing data is involved with this process.
- Using in-memory databases.

NoSQL - Definition

Stands for Not Only SQL

- In 1998, Carl Strozzi introduced the term NOSQL as a name for his file-based database.
- During an event to discuss open source distributed database systems, Eric Evans reintroduced it.
- Evans said that the purpose of seeking alternatives is to deal with a problem that relational databases aren't well-suited to.

Features of NoSQL

- Non-relational
- Do not need schema
- This system replicates data across multiple nodes (fault-tolerant & identical) and can be partitioned:
- Replaceable down nodes.
- There are no single points of failure.
- Scalable horizontally
- It is open-source
- Have a massive write performance
- Gives fast key-value access

Benefits of NoSQL databases

- **High Scalability** : Ability to execute more and more queries and store more and more data without having any upper limit.
- **High Availability :** Ability to run Read/Write queries even when some servers are down.

Disadvantage of NoSQL databases

- Relations aren't fully supported

  Join, group, and order operations are not allowed (except within partitions).

  Referential integrity is not constrained across partitions.

- SQL is not a declarative query language, so more programming is required.

- Reduced guarantees due to relaxed ACID (see CAP theorem).

- There is no easy integration with other applications that support SQL.

Who are users of NoSQL

- All the e-commerce companies such as Flipkart, Amazon, Walmart, etc. use.
- NoSQL database for storing huge volume of data and large amount of request from user.
- All the Cab aggregator companies such as OLA and UBER.
- The mobile app companies like Kobo and Playtika.
- Consumer appliances companies such as LG, Samsung etc use NoSQL for IOT use cases.
- NOSQL has been used by some of the mobile gaming companies like, electronic arts, zynga and tencent for Social Gaming use cases.

Tradeoff in Cluster Databases

ACID

A DBMS should support ACID transactions, which include:
Atomicity: either all the steps are completed or none of them.
Consistency: Only valid data is written.
Isolation: Each operation is isolated separately.
Durability: Once committed, it remains that way.

CAP

Consistency: all clustered data is identical.
Availability: Reads and writes are always accepted by the cluster.
Partition Tolerance: Even if some machines cannot communicate with other machines due to network failures, partition tolerance guarantees properties are maintained.

CAP Theorem

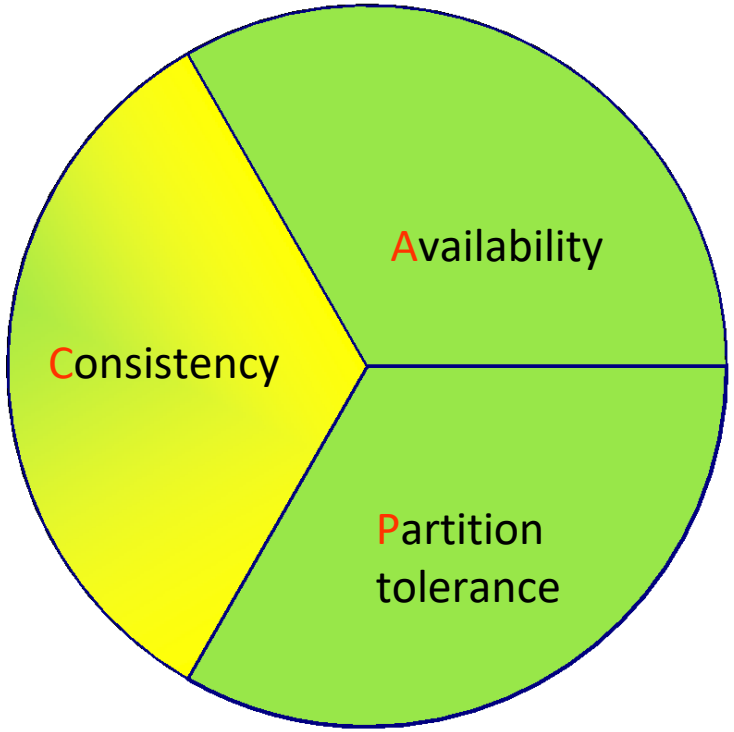
Brewer’s CAP Theorem:

- The combination of all three properties is "impossible" for any system that shares data.
- A shared-data system can have no more than two of these three characteristics.
- At some point, very large systems will "partition":

  C or A are the options left (traditional DBMS prefer C over A and P).

  Most of the time, you'd choose A over C (except in specific cases such as order processing).

CAP Theorem

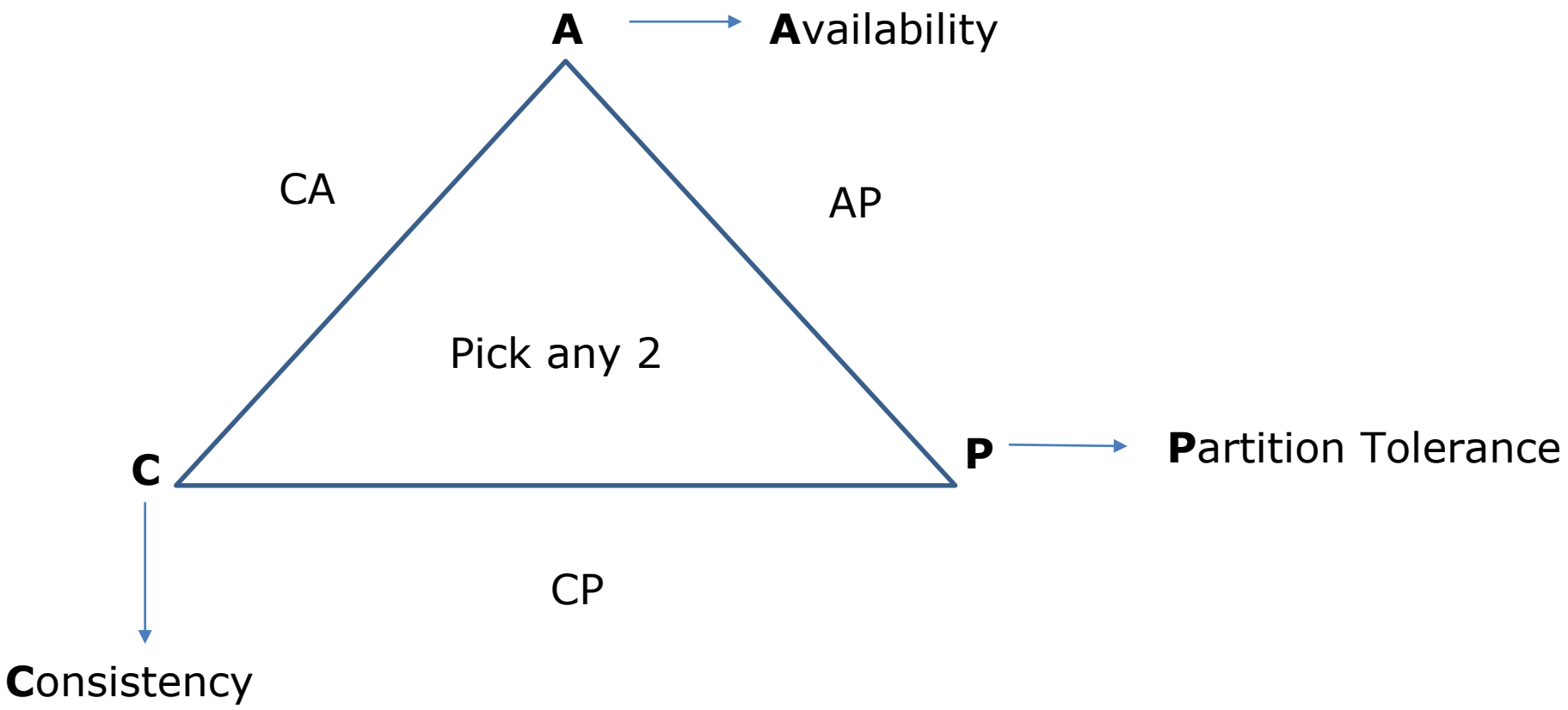


Consistency

There are two types of consistency that all clients have:

- Strong consistency – ACID (**A**tomicity, **C**onsistency, **I**solation, **D**urability)
- Weak consistency – BASE (**B**asically **A**vailable **S**oft-state **E**ventual consistency)

CAP (Visual Guide to NoSQL Systems)



Types of NoSQL databases

1. **Key-value**

   Example: DynamoDB, Voldermort, Scalaris

2. **Document-based**

   Example: MongoDB, CouchDB

3. **Column-based**

   Example: BigTable, Cassandra, HBase

4. **Graph-based**

   Example: Neo4J, InfoGrid

- Most NOSQL storage systems have "No-schema" as one of their characteristics.
- Data types should be "flexible".

Key Value Databases

- Key-value stores are the simplest NoSQL databases.
- Every single item in the database is stored as an attribute name (or "key"), together with its value.
- Examples of key-value stores are Riak and Voldemort.
- Some key-value stores, such as Redis, allow each value to have a type, such as "integer", which adds functionality.
- Key-value NoSQL databases are ideal for database for lookup queries with extremely quick and optimized retrieval.

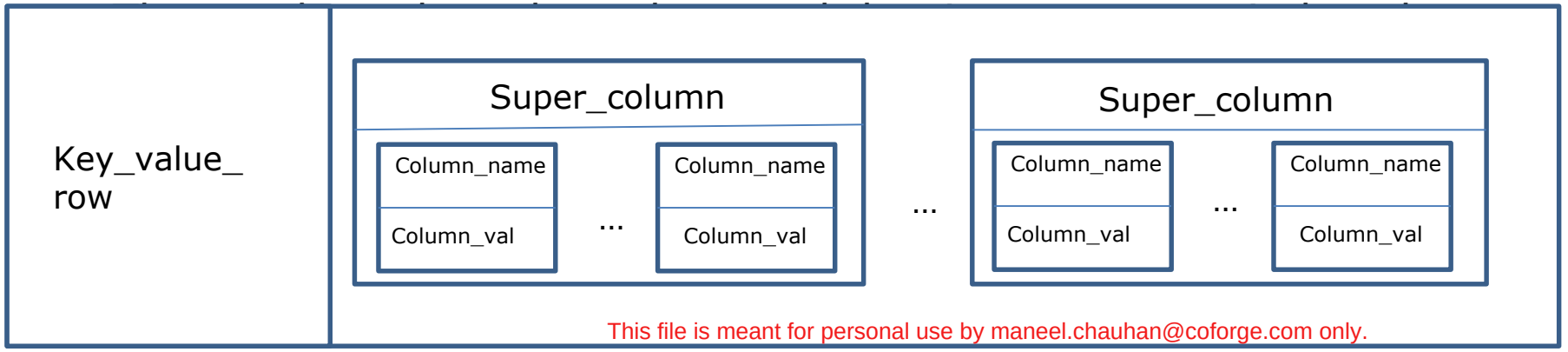
Document Databases

- {Name:“Mridul", Address:“FlatNo 16,Asnara Aparment ,BLR,India”,Grandchildren: {Ashwini: “10", Brinda: “12", "Mugdha: “9", "Kristen: “3", "Owaisi: “5", Rachna: “5“} Phones: [“91-987-7865”, “91-765-5432”] }

- Example: MongoDb, CouchBase, etc.

Columnar Databases

- In Google's BigTable paper, they are compared to column oriented relational databases (which store data in column order).
- RDBMS-like tables, but handle semi-structured data.
- **Data model:**
  - Column families are grouped in this collection.
  - Column family ☐ (key, value) where value ☐ set of **related** columns (standard, super)



Columnar Databases

- A column family can have a variable number of columns.
- A column family is sorted "physically".
- Most cells have null values, very sparse.

Comparison: Querying multiple tables with an RDBMS versus a NOSQL schema.

RDBMS: It must gather data from several places on disk and glue it together.

Column-based NOSQL: If a query only uses the columns in a certain column family, multiple rows can be retrieved with a single read operation à data locality (all columns in a column family are stored together on disk).

Example : HBase, Cassandra, HyperTable etc

Graph Databases

- Modeling the structure of data (interconnectivity).
- Scales of data complexity.
- Based on mathematical Graph Theory (G=E,V).
- **Data model:**

The graph may contain nodes (including IDs) and edges (including labels and roles).

Compared single-step, path expressions, and full recursion.

Example:

Neo4j, FlockDB, Pregel, InfoGrid …

Thank You