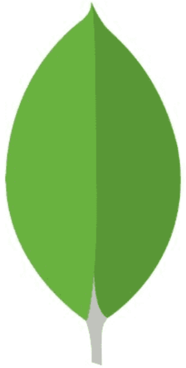
~~SQL~~ → NoSQL

↓

Anti



mongo

Cassandra

neo4j

redis

NoSQL Databases

Trainer: Mr. Nilesh Ghule

RDBMS vs NoSQL

	RDBMS	NoSQL
Types	All types support SQL standard	Multiple types exists, such as document stores, key value stores, column databases, etc
History	Developed in 1970	Developed in 2000s
Examples	SQL Server, Oracle, MySQL	MongoDB, HBase, Cassandra, Redis, Neo4J
Data Storage Model	Data is stored in rows and columns in a table, where each column is of a specific type	The data model depends on the database type. It could be Key-value pairs, documents etc
Schemas	Fixed structure and schema	Dynamic schema. Structures can be accommodated
Scalability	Scale up approach is used	Scale out approach is used
Transactions	Supports ACID and transactions	Supports partitioning and availability
Consistency	Strong consistency	Dependent on the product [Eventual Consistency]
Support	High level of enterprise support	Open source model
Maturity	Have been around for a long time	Some of them are mature; others are evolving

NoSQL database

- NoSQL databases are non-relational.
- There is no standardization/rules of how NoSQL database to be designed.
- All available NoSQL databases can be broadly categorized as follows:
  - Key-value databases
  - Column-oriented databases
  - Graph databases
  - Document oriented databases

Key-value database

- Based on Amazon’s Dynamo database.
- For handling huge data of any type.
- Keys are unique and values can be of any type i.e. JSON, BLOB, etc.
- Implemented as big distributed hash-table for fast searching.
- Example: redis, dynamodb, riak, ...

Column-oriented databases

- Values of columns are stored contiguously.
- Better performance while accessing few columns and aggregations.
- Good for data-warehousing, business intelligence, CRM, ...
- Examples: hbase, cassandra, bigtable, …

Graph databases

- Graph is collection of vertices and edges (lines connecting vertices).
- Vertices keep data, while edges represent relationships.
- Each node knows its adjacent nodes. Very good performance, when want to access all relations of an entity (irrespective of size of data).
- Examples: Neo4J, Titan, …

Document oriented databases

- Document contains data as key-value pair as JSON or XML.
- Document schema is flexible & are added in collection for processing.
- RDBMS tables → Collections
- RDBMS rows → Documents
- RDBMS columns → Key-value pairs in document
- Examples: MongoDb, CouchDb, …

MongoDb Databases

Trainer: Mr. Nilesh Ghule

Mongo Db

- Developed by 10gen in 2007
- Publicly available in 2009
- Open-source database which is controlled by 10gen

github.com → most of developers.

- Document oriented database → stores JSON documents
- Stores data in binary JSON. (BSON)
- Design Philosophy
  - MongoDB wasn't designed in a lab and is instead built from the experiences of building large scale, high availability, and robust systems.

Install MongoDB

- Install MongoDB by downloading community edition
  - (https://www.mongodb.com/download-center/community)
- Linux and Mac Users
  - Extract the downloaded file somewhere in the disk.
  - Set the environment path to use the tools without going to the bin directory in the ~/.bash_profile or ~/.basrc file.
- Ubuntu (20.04) Mongo installation
  - terminal> wget -qO - https://www.mongodb.org/static/pgp/server-4.4.asc | sudo apt-key add -
  - terminal> echo "deb [arch=amd64,arm64] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/4.4 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.4.list
  - terminal> sudo apt-get update
  - terminal> sudo apt-get install -y mongodb-org
- Windows Users
  - Install the MongoDB by following all the steps in the installation wizard
  - Set the the environment path to include the <path>/bin

JSON

- Java Script Object Notation
- Hierarchical way of organizing data
- Defined as part of the JS language by JavaScript creator Douglas Crockford (2000).
- JavaScript objects are associative containers, wherein a string key is mapped to a value
- JSON shows up in many different cases.
  - APIs
  - Configuration files
  - Log messages
  - Database storage
- JSON is not ideal for usage inside of a database.
  - JSON is a text-based format, and text parsing is very slow
  - JSON’s readable format is far from space-efficient, another database concern
  - JSON only supports a limited number of basic data types
- Mongo stores JSON data into Binary form.

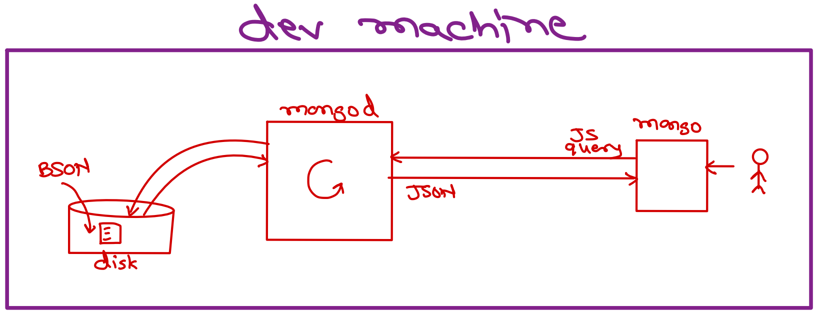
BSON

- BSON simply stands for “Binary JSON”
- Binary structure encodes type and length information, which allows it to be parsed much more quickly
- It has been extended to add some optional non-JSON-native data types
- It allows for comparisons and calculations to happen directly on data
- MongoDB stores data in BSON format both internally, and over the network
- Anything you can represent in JSON can be natively stored in MongoDB

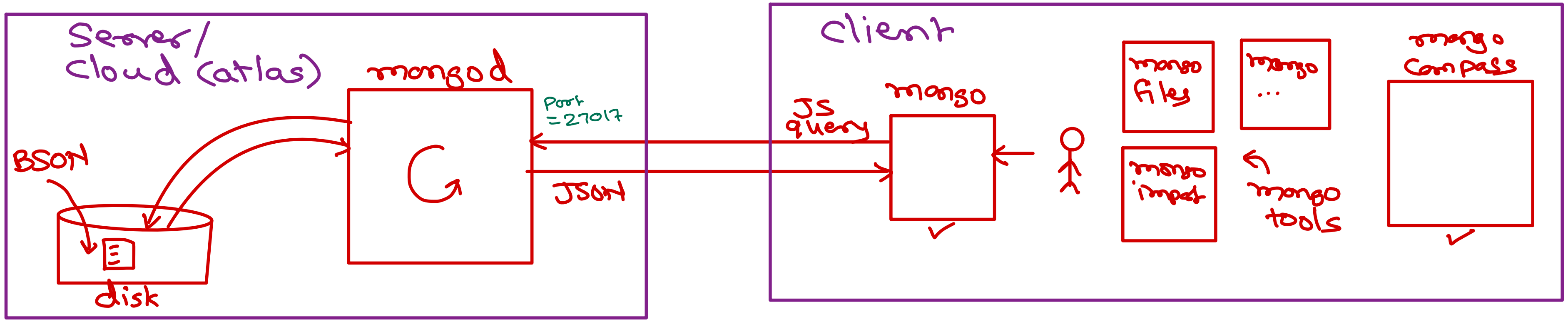
	JSON	BSON
Encoding	UTF-8 String	Binary
Data Support	• String • Boolean • Number • Array	• String • Boolean • Number • Integer • Float • Long • Decimal • Array • Date • Raw Binary
	Human and Machine	Machine Only

Mongo Server and Client

- MongoDb server (mongod) is developed in C, C++ and JS.
- MongoDb data is accessed via multiple client tools
  - mongo : client shell (JS).
  - mongofiles : stores larger files in GridFS.
  - mongoimport / mongoexport : tools for data import / export.
  - mongodump / mongorestore : tools for backup / restore.



- MongoDb data can be accessed in application through client drivers available for all major programming languages e.g. Java, Python, Ruby, PHP, Perl, …
- Mongo shell is follows JS syntax and allow to execute JS scripts.



MongoDb: Data Types

data	bson	values
null	10	
boolean	8	true, false
number	1 / 16 / 18	123, 456.78, NumberInt(“24”), NumberLong(“28”)
string	2	“….”
date	9	new Date(), ISODate(“yyyy-mm-ddThh:mm:ss”)
array	4	[…, …, …, …]
object	3	{ … }

MongoDB Terminology

- Database
  - Like database/schema in RDBMS.
  - mongo> show databases;
  - mongo> use dbname;
- Collection
  - Like table in RDBMS.
  - No fixed structure or schema.
  - mongo> db.createCollection("colname");
- Document
  - Like row in RDBMS.
  - Inserted in JSON format.
  - Each record can have different fields.
- Field
  - Like column in RDBMS.
  - A name-value pair in a document.

Mongo - INSERT

- show databases;
- use database;
- db.contacts.insert({name: "nilesh", mobile: "9527331338"});
- db.contacts.insertMany([

```
        {name: "nilesh", mobile: "9527331338"},
        {name: "nitin", mobile: "9881208115"}
]);
```

- Maximum document size is 16 MB.
- For each object unique id is generated by client (if _id not provided).
  - 12 byte unique id :: [counter(3) | pid(2) | machine(3) | timestamp(4)]

Mongo – QUERY

- db.contacts.find(); → returns cursor on which following ops allowed:
  - hasNext(), next(), skip(n), limit(n), count(), toArray(), forEach(fn), pretty()
- Shell restrict to fetch 20 records at once. Press "it" for more records.
- db.contacts.find({ name: "nilesh" });
- db.contacts.find({ name: "nilesh" }, { _id:0, name:1 });
- Relational operators: $eq, $ne, $gt, $lt, $gte, $lte, $in, $nin
- Logical operators: $and, $or, $nor, $not
- Element operators: $exists, $type
- Evaluation operators: $regex, $where, $mod
- Array operators: $size, $elemMatch, $all, $slice

SUNBEAM

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>