Why mongo Db?

Open source document oriented database

High performance database highly available.

It is horizontally scalable. We can add a server to increase its capacity instead of vertically scalable which is buying a bigger server.

Where to use?

1. High demanding application (real-time application)
2. Diverse ,mixed set of data
3. Massive concurrency
4. Application which are globally deployed in multiple sites
5. no down time is tolerated
6. Able to grow with user needs should not be stuck with data models should be able to evolve as per the market need.
7. High uncertainty in sizing (not sure how much the system will grow) needs fast scaling.
8. Where ever we need seamless and consistence experience
9. High performance while read, write and update happens at the same time.
10. For Real-time analytics mongo Db is the best where Hadoop is good doing analytics on existing data (batch analysis).
11. Straight forward replication( we can create replicas of mongo db in very less time)
12. Scaling on demand (we can add or remove mongo servers on demand)
13. Location based deployment ( singe cluster can expand over entire globe)
14. Geo spatial queries (location based queries like nearest restaurant , within a circle)
15. High Availability and auto failover(if there is any failure of any datacenter then it will be automatically failover to one of the replicas so there will be no down time )
16. Mongo db can be deployed not only commodity hardware but also virtual hardware like aws.
17. Text indexing
18. Compression
19. Full table scan can disrupt if lot of operations going on.

What MongoDB is not for?

Mongo is not a good choice for:

1. Billing system or general ledger system. Example Oracle RAC
2. Search engine Example Elastic search, SOLR

OLTP and OLAP

OLTP (On-line Transaction Processing) is characterized by a large number of short on-line transactions (INSERT, UPDATE, and DELETE). The main emphasis for OLTP systems is put on very fast query processing, maintaining data integrity in multi-access environments and an effectiveness measured by number of transactions per second. In OLTP database there is detailed and current data, and schema used to store transactional databases is the entity model (usually 3rdNormalForm).

Example: RDBMS, NoSQL DB (Mongo Db)

- OLAP (On-line Analytical Processing) is characterized by relatively low volume of transactions. Queries are often very complex and involve aggregations. For OLAP systems a response time is an effectiveness measure. OLAP applications are widely used by Data Mining techniques. In OLAP database there is aggregated, historical data, stored in multi-dimensional schemas (usually star schema).

Example: Hadoop

Why mongo Db came in to picture?

1. Usage of mobile devices increased drastically in recent years
2. Social media usage in business application increased a lot.
3. Unstructured data increasing at the rate of 2x as compared to structured data.
4. Classic database (RDBMS) does not know how to scale gracefully in a distributed fashion on commodity hardware.

Bson -> binary json in mongo db

Collection similar to tables

Document similar to record

Commands

db 🡪 shows current database

show dbs or show databsases 🡪 show databases

use dbname 🡪 creates and switches the database

> db

test

> show db

2018-10-29T16:04:27.168+0530 E QUERY [js] Error: don't know how to show [db] :

shellHelper.show@src/mongo/shell/utils.js:1055:11

shellHelper@src/mongo/shell/utils.js:766:15

@(shellhelp2):1:1

> show databases

admin 0.000GB

config 0.000GB

local 0.000GB

test 0.000GB

> switch local

2018-10-29T16:05:04.609+0530 E QUERY [js] SyntaxError: missing ( before switch expression @(shell):1:7

> db

test

> use local

switched to db local

> db

local

> use test

switched to db test

> show databases

admin 0.000GB

config 0.000GB

local 0.000GB

test 0.000GB

> show dbs

admin 0.000GB

config 0.000GB

local 0.000GB

test 0.000GB

> db.testcollection

test.testcollection

> db.testcollection.insert({"name":"Max"})

WriteResult({ "nInserted" : 1 })

> show dbs

admin 0.000GB

config 0.000GB

local 0.000GB

test 0.000GB

> use test

switched to db test

> db.testcollection

test.testcollection

> show dbs

admin 0.000GB

config 0.000GB

local 0.000GB

test 0.000GB

test2 0.000GB

> db.dropDatabase()

{ "dropped" : "test", "ok" : 1 }

**Create collection and insert data**

db.mycollection.insert([{"firstName":"Niranjan","lastName":"Panigrahi"},{"firstName":"Amit","lastName":"Saha"}])

**Get all data from collection**

db.getCollection('testcollection').find({})

show collections

**Create collection**

db.createCollection("mycollection2")

**Drop collection**

db.mycollection2.drop()

**Creates Database**

use school

db.getCollection('mycollection').find().pretty()

db.getCollection('mycollection').findOne()

db.mycollection.findOne()

mongoimport --db dbName --collection collectionName --file fileName.json

mongoimport --db dbName --collection collectionName --file fileName.json –jsonArray

C:\project\softwares\mongo

**Import documents**

C:\project\softwares\mongo>mongoimport --db test --collection students --file C:\students.json

C:\project\softwares\mongo>mongorestore --drop -d test -c tweets C:\tweets.bson