

Mongodb



Mongodb Abstract

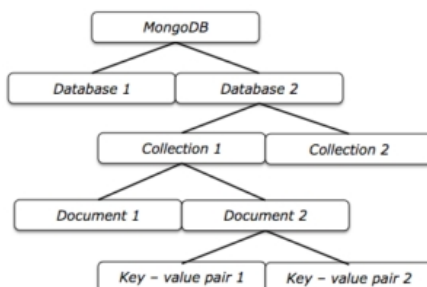
Mongo DB is scalable, open source, high performance, document oriented database. MongoDB is an open-source document database that provides high performance, high availability, and automatic scaling. MongoDB is available under General Public license for free and it is also available under Commercial license from the manufacture. It is an open source product, developed and supported by a company named 10gen. In simple words you can say that - Mongo DB is a document oriented database. Now it is used by company of all sizes, across all industry.



Mongodb Video

What is Mongodb?]

MongoDB is a cross-platform and open-source document-oriented database, MongoDB is built for scalability, high availability and performance from a single server deployment to large and complex multi-site infrastructures. This makes data integration for certain types of applications faster and easier. As a NoSQL database, MongoDB shuns the relational database's table-based structure to adapt JSON-like documents that have dynamic schemas which it calls BSON. MongoDB is a cross-platform and open-source document-oriented database, a kind of NoSQL database.



The following table shows the relationship of RDBMS terminology with MongoDB.

RDBMS

MongoDB

Database

Database

Table

Collection

Tuple/Row

Document

column

Field

Table Join

Embedded Documents

Primary Key

Primary Key (Default key `_id` provided by mongodb itself)

How MongoDB works ?]

MongoDB memory maps the database files. It allows the OS to control this and allocate the maximum amount of RAM to the memory mapping. As MongoDB updates and reads from the DB it is reading and writing to RAM. All indexes on the documents in the database are held in RAM also. The files in RAM are flushed to disk every 60 seconds. To prevent data loss in the event of power failure, the default is to run with journaling switched on. The journal file is flushed to disk every 100ms and if there is power loss is used to bring the database back to a consistent state. An important design decision with mongo is on the amount of RAM. You need to figure out your working set size - i.e if you are going to be reading and writing to only the most recent 10% of your data in the database then this 10% is your working set and should be held in memory for maximum performance. So if your working set is 10GB you are going to need 10GB for max performance - otherwise your queries/updates will run slower as pages of memory are paged from disk into memory. Other important aspects of MongoDB are replication for backups and sharding for scaling. There are a lot of great online resources for learning. MongoDB is free and opensource.

Mongoddb Advantages

- schema-less. If you have a flexible schema, this is ideal for a document store like MongoDB. This is difficult to implement

in a performant manner in RDBMS

- ease of scale-out. Scale reads by using replica sets. Scale writes by using sharding (auto balancing). Just fire up another machine and away you go. Adding more machines = adding more RAM over which to distribute your working set.
- Depends on which RDBMS of course, but MongoDB is free and can run on Linux, ideal for running on cheaper commodity kit.
- you can choose what level of consistency you want depending on the value of the data (e.g. faster performance = fire and forget inserts to MongoDB, slower performance = wait til insert has been replicated to multiple nodes before returning)
- It supports replica sets; in other words, a failover mechanism is automatically handled. If the primary server goes down, the secondary server becomes the primary automatically, without any human intervention.
- It supports the common authentication mechanisms, such as LDAP, AD, and certificates. Users can connect to MongoDB over SSL and the data can be encrypted.
- MongoDB can be a cost effective solution because improves flexibility and reduces cost on hardware and storage.

Mongoddb Disadvantages

- Data size in MongoDB is typically higher due to e.g. each document has field names stored it
- less flexibility with querying (e.g. no JOINS)
- no support for transactions - certain atomic operations are supported, at a single document level
- at the moment Map/Reduce (e.g. to do aggregations/data analysis) is OK, but not blisteringly fast. So if that's required, something like Hadoop may need to be added into the mix
- less up to date information available/fast evolving product