

HOME » RESOURCES » BLOG

# MongoDB: The Latest Features-Replication

Get in touch with a  
TriCore Solutions specialist

START HERE

SIGN UP FOR TRICORE'S  
BLOG

RESOURCES

- Customer Snapshots
- Whitepapers
- TriCore Videos
- Podcasts
- Blog
- Business Intelligence Knowledge Zone
- Oracle R12 Knowledge Zone
- TriCloud Knowledge Zone
- WHY WORK WITH A MSP

CATEGORIES

- Oracle (58)
- Oracle Tips (32)
- TriCore (17)
- Business intelligence (14)
- Oracle EBS (14)
- CLOUD (11)
- Oracle 12c (10)
- SAP (10)
- SQL Server (10)
- Databases (9)
- MySQL (9)
- Oracle R12 (9)

Blog | Apr 26, 2016

## MongoDB: The Latest Features-Replication

In MongoDB, Replication serves for high availability and redundancy of the database by keeping the same copy of data across several replica sets.

Introduction:

Let's pick up from where I left last. We will discuss Replication in detail. In MongoDB, Replication serves for high availability and redundancy of the database by keeping the same copy of data across several replica sets. As of version 3.2, we can have upto 50 members in a replica set while only 12 were possible in 3.0. However, the number of voting members is still limited to 7.

In version 3.2, a **new protocolVersion: 1** is introduced and for legacy reasons the **old protocolVersion: 0** is available. The new replication protocol has been designed in reference with the [Raft Consensus Algorithm](#) and several existing capabilities of MongoDB replica sets.

MongoDB\_replication.png

**A replica set is a group of mongod** processes that maintain the same data set. In a replica set, there is always one Primary (there may be more than one in some situations which it tries its best to avoid) which accepts the writes and replicates those to other secondary data bearing nodes. Data bearing nodes in a replica set actually contain the data and may or may not participate in an election. Voting members are those which participate in an election.

1. {votes: n} where n is generally 0 or 1.

Each voting member has one vote. It is recommended not to change this value unless non-voting members are required.

Instead, "priority" parameter should be changed in case lower or higher priority of a node is required.

2. {priority: n} n can be a whole number between 0 and 1000.

This parameter decides the eligibility of being a primary. A higher priority value of a member raises its probability of becoming a primary.

A lower priority member can also become primary for shorter durations but it is always highly likely that a member with higher priority will eventually be elected as primary. A member with priority 0 can never be primary and hence can only serve read queries if required.

3. Arbiter

Click here to chat

Career (7)  
 Business Analytics (6)  
 Compliance (6)  
 SharePoint (6)  
 Big Data (4)  
 MongoDB (4)  
 Tableau (4)  
 AWS (3)  
 PeopleSoft (3)  
 Events (2)  
 Linux (2)  
 Divestitures (1)  
 ITIL (1)  
 Windows Exchange (1)

---

Arbiter is a non-data bearing node which is used in a replica set having even number of members to make it odd numbered in order to ease the election process. Arbiter does not hold any data, so, it does not serve any queries, but participates in elections. An arbiter has its parameter `arbiterOnly` set to `true` (`arbiterOnly: true`) and its `votes` parameter is always 1.

#### 4. Hidden Members : {hidden: true}

Hidden members have a priority 0 and are invisible to the clients. These do not serve queries and cannot become primary. However, hidden members keep a copy of the primary's dataset and can be used for purposes such as reporting and backups. These members may or may not participate in an election depending on the "votes" configuration.

#### 5. Write Concern & Journaling

The term "Write Concern" in MongoDB is the level of acknowledgement for the write operations. This value specifies "after writing on how many replica set members will a write operation provide an acknowledgement back to the client".

A write operation should be considered permanent only when it has been replicated to a majority of replica set members.

Journaling allows for durability in the event of the failure of a mongod process by using write ahead logging to on-disk journal files. It can be enabled or disabled depending on requirements.

```
{w: <value>, j: <boolean>, wtimeout: <number> }
```

Value for 'w' specifies that the write should propagate at minimum to this many number of replica set members before providing an acknowledgement to the client. The possible values are number(0,1 or more), majority (e.g. {w: "majority"} ), or tag sets (to specify that the write operations have propagated to the members with specified tag)

For `protocolVersion`: 1

1. {w: "majority"} implies {j: "true"}
2. Regardless of the j option, secondary members are bound to write to on-disk journals before providing an acknowledgement for a write operation

#### 6. MongoDB Journal & Op-Log: Analogy for Oracle DBA

This section is particularly based on my personal views to carve out an analogy while moving from Relational to NoSQL world. In Oracle, Redo Logs record the change vectors in order to facilitate instance recovery in case of instance crashes.

Also, if you remember Oracle Logical Standby databases where Redo stream from Redo logs is changed to SQL statements and applied on the standby database. Same is the case with MongoDB except that we have two separate logs here to help in almost the same functions and may be more.

One is MongoDB Op-log which is a **capped collection** and stores record of all operations that modify the data stored in the database. These operations are then polled and applied by the secondary replica set members.

Another is Journal which allows for write-ahead logging of the operations in the form of change descriptions which allows for the writes to be durable on a single mongod instance in case of instance crash.

#### Conclusion:

[Click here to chat](#)

Hope you find this blog informative and understand MongoDB and associated environment better. You can read my previous blogs by [clicking here](#).

To read Part I about the latest features, go here: [MongoDB: The Latest Features](#).

For any questions on the topic click below:

**Ask the Expert**

GET IN TOUCH

T: 1-888-239-7775  
E: [info@tricoresolutions.com](mailto:info@tricoresolutions.com)

SERVICES

- Managed Applications
- Managed Infrastructure
- Consulting Services
- IT Staffing
- Managed Services BI Calculator

FOLLOW US