For example, if you have six collections in a database using the MMAPv1 storage engine and an operation takes a collection-level write lock, the other five collections are still available for read and write operations. An exclusive database lock makes all six collections unavailable for the duration of the operation holding the lock.

11.4.3 How do I see the status of locks on my `mongod` instances?

For reporting on lock utilization information on locks, use any of the following methods:

- `db.serverStatus()`,
- `db.currentOp()`,
- `mongotop`,
- `mongostat`, and/or
- the MongoDB Cloud Manager[18] or Ops Manager, an on-premise solution available in MongoDB Enterprise Advanced[19]

Specifically, the `locks` document in the `output of serverStatus`, or the `locks` field in the `current operation reporting` provides insight into the type of locks and amount of lock contention in your `mongod` instance.

To terminate an operation, use `db.killOp()`.

11.4.4 Does a read or write operation ever yield the lock?

In some situations, read and write operations can yield their locks.

Long running read and write operations, such as queries, updates, and deletes, yield under many conditions. MongoDB operations can also yield locks between individual document modifications in write operations that affect multiple documents like `update()` with the `multi` parameter.

MongoDB's *mmapv1* (page 91) storage engine uses heuristics based on its access pattern to predict whether data is likely in physical memory before performing a read. If MongoDB *predicts* that the data is not in physical memory, an operation will yield its lock while MongoDB loads the data into memory. Once data is available in memory, the operation will reacquire the lock to complete the operation.

For storage engines supporting document level *concurrency control*, yielding is not necessary when accessing storage, as the *intent locks* held at the global, database and collection level do not block other readers and writers.

Changed in version 2.6: MongoDB does not yield locks when scanning an index even if it predicts that the index is not in memory.

11.4.5 Which operations lock the database?

Changed in version 2.2.

The following table lists common database operations and the types of locks they use.

[18] https://cloud.mongodb.com/?jmp=docs

[19] https://www.mongodb.com/products/mongodb-enterprise-advanced?jmp=docs