creation options and their behavior.

Related

Some options that you can specify to `createIndex()` options control the *properties of the index* (page 503), which are *not* index creation options. For example, the *unique* (page 504) option affects the behavior of the index after creation.

For a detailed description of MongoDB's index types, see *Index Types* (page 486) and *Index Properties* (page 503) for related documentation.

Background Construction

By default, creating an index blocks all other operations on a database. When building an index on a collection, the database that holds the collection is unavailable for read or write operations until the index build completes. Any operation that requires a read or write lock on all databases (e.g. **listDatabases**) will wait for the foreground index build to complete.

For potentially long running index building operations, consider the `background` operation so that the MongoDB database remains available during the index building operation. For example, to create an index in the background of the `zipcode` field of the `people` collection, issue the following:

```
db.people.createIndex( { zipcode: 1}, {background: true} )
```

By default, `background` is `false` for building MongoDB indexes.

You can combine the background option with other options, as in the following:

```
db.people.createIndex( { zipcode: 1}, {background: true, sparse: true } )
```

Behavior

As of MongoDB version 2.4, a `mongod` instance can build more than one index in the background concurrently.

Changed in version 2.4: Before 2.4, a `mongod` instance could only build one background index per database at a time.

Changed in version 2.2: Before 2.2, a single `mongod` instance could only build one index at a time.

Background indexing operations run in the background so that other database operations can run while creating the index. However, the `mongo` shell session or connection where you are creating the index *will* block until the index build is complete. To continue issuing commands to the database, open another connection or `mongo` instance.

Queries will not use partially-built indexes: the index will only be usable once the index build is complete.

Note: If MongoDB is building an index in the background, you cannot perform other administrative operations involving that collection, including running `repairDatabase`, dropping the collection (i.e. `db.collection.drop()`), and running `compact`. These operations will return an error during background index builds.

Performance

The background index operation uses an incremental approach that is slower than the normal "foreground" index builds. If the index is larger than the available RAM, then the incremental process can take *much* longer than the foreground build.