Chapter 3: Slow Queries

Lab: Building an Index in the Foreground in Production

The answer is less than one second *after* the index build ended.

You can find your log files with:

```
ps -ef | grep mongod
```

🗐 COPY

Look for the item after the `--logPath` flag. You should see something like the following:

🗐 COPY

```
2017-03-07T04:59:43.902+0000 I INDEX     [conn3498] build index done.
scanned 2888268 total records. 14 secs
2017-03-07T04:59:43.914+0000 I COMMAND  [conn3498] command m312.$cmd
command: createIndexes { createIndexes: "employees", indexes: [ {
name: "last_name_1_first_name_1", key: { last_name: 1, first_name: 1
} } ], writeConcern: {} } numYields:0 reslen:113 locks:{ Global: { a
cquireCount: { r: 2, w: 2 } }, Database: { acquireCount: { w: 1, W: 1
} }, Collection: { acquireCount: { w: 1 } }, Metadata: {
acquireCount: { w: 1 } }, oplog: { acquireCount: { w: 1 } } }
protocol:op_query 14368ms
2017-03-07T04:59:43.915+0000 I COMMAND  [conn3502] command
m312.employees command: find { find: "employees", filter: { _id:
ObjectId('58be04b01e94b8092bee7a7d') } } planSummary: IDHACK
keysExamined:1 docsExamined:1 cursorExhausted:1 numYields:0
nreturned:1 reslen:482 lock s:{ Global: { acquireCount: { r: 2 } },
Database: { acquireCount: { r: 1 }, acquireWaitCount: { r: 1 },
timeAcquiringMicros: { r: 14313556 } }, Collection: { acquireCount: {
r: 1 } } } protocol:op_query 14325ms
2017-03-07T04:59:43.915+0000 I WRITE    [conn3504] update
m312.employees query: { _id: ObjectId('58be04b01e94b8092bee7a7d') }
```

```
planSummary: IDHACK update: { $inc: { count: 1 } } keysExamined:1
docsExamined:1 nMatched:1 nModified:1 numYields:0 locks:{ Global: {
acquireCount : { r: 2, w: 2 } }, Database: { acquireCount: { w: 2 },
acquireWaitCount: { w: 1 }, timeAcquiringMicros: { w: 14263073 } },
Collection: { acquireCount: { w: 1 } }, Metadata: { acquireCount: {
w: 1 } }, oplog: { acquireCount: { w: 1 } } } 14275ms
2017-03-07T04:59:43.915+0000 I COMMAND  [conn3506] command
m312.employees command: find { find: "employees", filter: { _id:
ObjectId('58be04b01e94b8092bee7a7d') } } planSummary: IDHACK
keysExamined:1 docsExamined:1 cursorExhausted:1 numYields:0
nreturned:1 reslen:482 lock s:{ Global: { acquireCount: { r: 2 } },
Database: { acquireCount: { r: 1 }, acquireWaitCount: { r: 1 },
timeAcquiringMicros: { r: 14208478 } }, Collection: { acquireCount: {
r: 1 } } } protocol:op_query 14220ms
2017-03-07T04:59:43.915+0000 I COMMAND  [conn3504] command m312.$cmd
command: update { update: "employees", ordered: true, updates: [ { q:
{ _id: ObjectId('58be04b01e94b8092bee7a7d') }, u: { $inc: { count: 1
} }, multi: false, upsert: false } ] } numYields:0 reslen:119 lo cks:
{ Global: { acquireCount: { r: 2, w: 2 } }, Database: { acquireCount:
{ w: 2 }, acquireWaitCount: { w: 1 }, timeAcquiringMicros: { w:
14263073 } }, Collection: { acquireCount: { w: 1 } }, Metadata: {
acquireCount: { w: 1 } }, oplog: { acquireCount: { w: 1 } } } protoc
ol:op_query 14275ms
```

You can see it very clearly if you only look at the **INDEX** operations and the **WRITE** operations:

```
                                                          COPY

logPath=$(ps -ef | grep mongo | grep rs1 | awk '{print $14}')
mlogfilter $logPath --component INDEX
```

Look for the log files immediately before and after the index build. There should be none at all in the seconds before, and they should start hitting the logs very quickly after the build is done.

The `.find()` queries will tell the same story. Any other operations you see in your logs will be from other operations, such as a `mongoimport`.

Proceed to next section