MongoDB cool administration tips

Gabriel Ciciliani - Percona Live Europe 2018

Pythian®
love your data

#1 `db.currentOps()` flat output

db.currentOps() flat output

```
mongo> db.currentOp(<filters>).inprog.forEach(
    function(d){
        var q = {};
            print('ConnectionId:'+d.connectionId
            +' Operation:'+d.op
            +' Namespace:'+d.ns
            +' Client:'+d.client
            +' Seconds_running:'+d.secs_running
            +' NumYields:'+d.numYields
            +' Query:'+JSON.stringify(d.command))
            }
    )
```

Filters: {"ns":/^customers\./,"op":"insert","secs_running":{$gt:60}}

db.currentOps() flat output

```
ConnectionId:88 Operation:command Namespace:sbtest.$cmd Client:172.19.0.1:59782 Seconds_running:53 NumYields:0
Query:{"createIndexes":"sbtest3","indexes":[{"name":"k_1","ns":"sbtest.sbtest3","background":false,"key":{"k":1}}],"$db":"sbtest"}

ConnectionId:86 Operation:command Namespace:sbtest.$cmd Client:172.19.0.1:59774 Seconds_running:53 NumYields:0
Query:{"createIndexes":"sbtest1","indexes":[{"name":"k_1","ns":"sbtest.sbtest1","background":false,"key":{"k":1}}],"$db":"sbtest"}

ConnectionId:87 Operation:command Namespace:sbtest.$cmd Client:172.19.0.1:59778 Seconds_running:53 NumYields:0
Query:{"createIndexes":"sbtest2","indexes":[{"name":"k_1","ns":"sbtest.sbtest2","background":false,"key":{"k":1}}],"$db":"sbtest"}
```

#2 Massive operations

Massive document update / removal

If the condition retrieves too many documents or there is no index for the condition key:

- Determine lowest and highest `_id` for the condition specified
- Split the above range in fixed size chunks, by storing the chunk's boundaries in a list.
- Iterate through the chunks and execute the update operation for each one

Massive document removal

If the condition key **is** indexed

- use `limit()`, specify a `chunksize` and a `delay`. Iterate until no documents matching the condition are found

Database drop

- `db.dropDatabase()` requires a global write lock, blocking other operations until it has completed.
- `db.collection.drop()` only requires a database level lock
- Drop all collections first, then drop the database

#3 Slow queries

Plan Cache Filters

```
db.runCommand(
    {
        planCacheSetFilter: <collection>,
        query: <query>,
        sort: <sort>,
        projection: <projection>,
        indexes: [ <index1>, <index2>, ...]
    }
)
```

* ***query + sort + projection = query shape***

Unused Indexes

- `db.orders.aggregate( [ { $indexStats: { } } ] )`

```
...
"host" : "myhost:30008",
"accesses" : {
    "ops" : NumberLong(4),
    "since" : ISODate("2018-07-25T10:12:59.106Z")
}
...
```

- Needs to be executed for each collection => script!

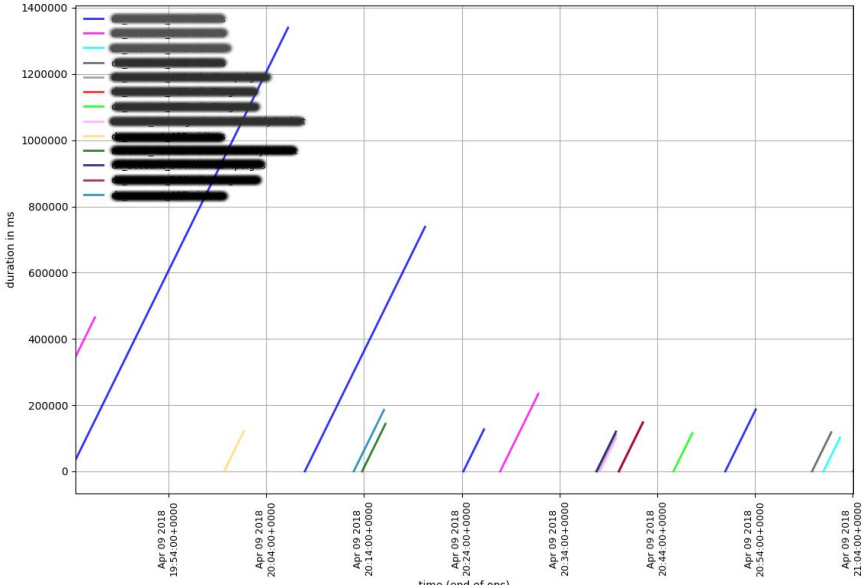
mtools

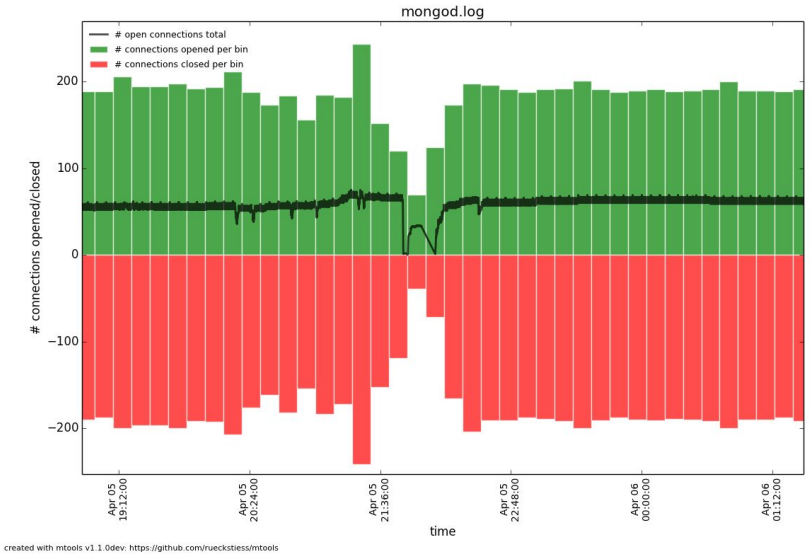
Written in python. For when the profiler is disabled.

- `mlogfilter`
  - `--operation OP`
  - `--pattern P`
  - `--slow MS`
  - `--scan`
  - `--from FROM --to TO`

- `mplotqueries`

mtools





#4 SSL troubleshooting

SSL troubleshooting

```
## Validate certificate - PK correspondence

$ openssl rsa -in mongo.key -noout -modulus |openssl md5
(stdin)= 45921d92f81d32ce48f02bcd904bdfb3
$ openssl x509 -in mongo.crt -noout -modulus |openssl md5
(stdin)= 45921d92f81d32ce48f02bcd904bdfb3


## Verifying alternate DNS names within the certificate
$ openssl x509 -in mongo.crt -text |grep DNS


## Verifying certificate against the CA certificates

$ openssl verify -CAfile /etc/ssl/mongo-rootAndChain.crt /etc/ssl/server.pem
/etc/ssl/server.pem: OK
```

SSL troubleshooting

```
## Configuration
ssl:
    mode:[ allowSSL | preferSSL | requireSSL | ]
    allowConnectionsWithoutCertificates: true
```

#5 Replicasets and Sharding

Sharded clusters maintenance

- Dealing with Jumbo chunks
  - Divide the chunk using `sh.splitAt()`
  - Manually clear the flag updating `config.chunks`

- Remember that `explain()` can be used to verify chunk migration operations

- Update mongoS cache after a manual chunks migration, clearing a jumbo chunk flag, removing a shard or issuing `movePrimary`
  - `db.adminCommand({ flushRouterConfig: 1 } )`

- Query `config.mongos` to obtain a list of mongoS nodes that ever connected to the cluster. Useful when migrating config servers

Preventing cross-region traffic

- Hide a replicaset member to make it invisible to applications
- Use tags to prioritize traffic on one region over the other

```
>>> from pymongo.read_preferences import Secondary
>>> db = client.get_database(
...     'test', read_preference=Secondary([{'dc': 'ny'}, {'dc': 'sf'}]))
```

Example scripts

Example script to process updates in chunks
https://github.com/gabocic/mongodb/blob/master/chunksfinder.js

Example to drop a database by dropping all collections first
https://github.com/gabocic/mongodb/blob/master/dropDatabasesLowImpact.js

Simple collection pruning example
https://github.com/gabocic/mongodb/blob/master/simpleCollectionPruning.js

Full collection pruning example including emergency halt and logging
https://github.com/gabocic/mongodb/blob/master/pruneDropCollections.js

Flat currentOps()
https://github.com/gabocic/mongodb/blob/master/flatcurrentops.js

Unused indexes
https://github.com/gabocic/mongodb/blob/master/unusedidx.js

Thanks!

Gabriel Ciciliani
ciciliani@pythian.com
www.linkedin.com/in/gabrielciciliani
@gabocic