Introduction: MongoDB with PHP

PHP UK Conference - London, UK - Feb 25th, 2012
Derick Rethans - derick@10gen.com - twitter: @derickr
http://joind.in/4976

About Me

Derick Rethans

- Dutchman living in London
- PHP mongoDB driver maintainer for 10gen (the company behind mongoDB)
- Author of Xdebug
- Author of the mcrypt, input_filter, dbus, translit and date/time extensions

About Me

Derick Rethans

- Dutchman living in London
- PHP mongoDB driver maintainer for 10gen (the company behind mongoDB)
- Author of Xdebug
- Author of the mcrypt, input_filter, dbus, translit and date/time extensions

What is mongoDB?

- mongoDB is a document storage and retrieval engine
- It requires almost no configuration to set-up a high available and high performant cluster of database servers
- Each document is stored into a collection, which is stored into a database, which is stored in a database server.

Getting started with mongoDB

Installation by downloading from :

```
wget http://fastdl.mongodb.org/linux/mongodb-linux-x86_64-2.0.2.tgz
tar xvzf mongodb-linux-x86_64-2.0.2.tgz
cd mongodb-linux-x86_64-2.0.2/bin
mkdir ../../data
./mongod --dbpath ../../data --logpath /tmp/mongod.log --fork
tail -f /tmp/mongod.log
```

Installation through apt-get:

```
$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv 7F0CEB10
$ sudo echo "deb http://downloads-distro.mongodb.org/repo/debian-sysvinit dist 10gen" >> /etc/apt/sources.list
# sudo apt-get update
$ sudo apt-get install mongodb-10gen
```

Tweak config in /etc/mongodb.conf if you must. mongo (the shell) is a useful application to try out things with.

The mongoDB PHP extension

- Maintained and supported by 10gen (Kristina and me)
- Can be installed with pecl: pecl install mongo
- Add extension=mongo.so to php.ini

Terminology

- JSON Document: the data (row)
- Collection: contains documents (table, view)
- Index
- Embedded Document (~join)

Connecting to mongoDB

No tables or collections have to do be explicitly created:

```
<?php
$m = new Mongo();
$database = $m->demo;
$collection = $database->testCollection;
?>
```

Different connection strings:

- new Mongo("mongodb://localhost");
- new Mongo("mongodb://localhost:29000");
- new Mongo("mongodb://mongo.example.com");
- new Mongo("mongodb://mdb1.example.com,mdb2.example.com", ['replicaSet' => 'testSet']);

Documents

- Stored as BSON (Binary JSON)
- Can have embedded documents
- Have a unique ID (the _id field)
- Are schemaless

Simple document:

```
{
   "_id" : ObjectId("4cb4ab6d7addf98506010001"),
   "handle" : "derickr",
   "name" : "Derick Rethans"
}
```

Document with embedded documents:

```
{
  "_id" : "derickr",
  "name" : "Derick Rethans",
  "talks" : [
    { "title" : "Profiling PHP Applications",
      "url" : "http://derickrethans.nl/talks/profiling-phptour.pdf",
    },
    { "title" : "Xdebug",
      "url" : "http://derickrethans.nl/talks/xdebug-phpbcn11.pdf",
    }
  ]
}
```

Inserting a document

```
{
  "_id" : "derickr",
  "name" : "Derick Rethans",
  "articles" : [
    {
      "title" : "Profiling PHP Applications",
      "url" : "http://derickrethans.nl/talks/profiling-phptour.pdf",
    },
    {
      "title" : "Xdebug",
      "url" : "http://derickrethans.nl/talks/xdebug-phpbcn11.pdf",
    }
  ]
}
```

```
<?php
$document = [
  "_id" => "derickr",
  "name" => "Derick Rethans",
  "articles" => [
    [
      "title" => "Profiling PHP Applications",
      "url" => "http://derickrethans.nl/talks/profiling-phptour.pdf",
    ],
    [
      "title" => "Xdebug",
      "url" => "http://derickrethans.nl/talks/xdebug-phpbcn11.pdf",
    ]
  ]
];

$m = new Mongo();
var_dump( $m->demo->articles->insert( $document ) );
?>
```

mongoDB supports many types

- null
- boolean
- integer (both 32-bit and 64-bit, MongoInt32, MongoInt64)
- double
- string (UTF-8)
- array
- associative array
- MongoRegex
- MongoId
- MongoDate
- MongoCode
- MongoBinData

IDs

- Every document needs a unique ID

```
<?php
$m = new Mongo();
$c = $m->demo->articles;

$c->insert( [ 'name' => 'Derick', '_id' => 'derickr' ] );

$d = array( 'name' => 'Derick' );
$c->insert( $d );
var_dump( $d );
?>
```

Checking for errors

So far, we have not checked for whether inserts worked.

```
<?php
$m = new Mongo;
$c = $m->demo->articles;

$c->insert( array( '_id' => 'derickr' ) );
$c->insert( array( '_id' => 'derickr' ) );
?>
```

"Safe" inserts:

```
<?php
$m = new Mongo;
$c = $m->demo->articles;

try {
    $c->insert(
        array( '_id' => 'derickr' ), // document
        array( 'safe' => true )      // options
    );
} catch ( Exception $e ) {
    echo $e->getMessage(), "\n";
}
?>
```

Checking for errors (2)

Safeness levels:

- Confirm change recorded in memory: array('safe' => true);
- Confirm committed to disk: array('fsync' => true);
- Confirm replication: array('safe' => true, 'w' => 2);

```
<?php
$m = new Mongo();
$c = $m->demo->articles;

$c->insert(
    array( '_id' => 'mongodb' ),       // document
    array( 'safe' => true, 'w' => 2 ) // options
);
?>
```

Querying and looping

Finding multiple documents is done with the find(), and returns an iterator in the form of a MongoCursor object.

```
<?php
$m = new Mongo();
$c = $m->demo->articles;

$cursor = $c->find( [ 'title' => [ '$exists' => true ] ] );

foreach ( $cursor as $r )
{
    echo $r['title'], ': ', join( ', ', $r['tags'] ), "\n";
}
?>
```

Cursor methods

```
<?php
$m = new Mongo();
$c = $m->demo->articles;

$cursor = $c->find();
$cursor->sort( array( '_id' => 1 ) )
       ->limit( 3 )
       ->skip( 3 );

// Only now does the query get send
foreach ( $cursor as $r )
{
    echo $r['_id'], "\n";
}
?>
```

Advanced querying operators

Besides testing for exact matches, mongoDB also has operators.
Operators start with a '$', so make sure to use single quotes!
Compare:

- $lt, $lte, $gt, $gte (<, <=, >, >=)
- $ne (not equal)
- $exists (field exists)
- $mod (modulo)

Logical:

- $or (one needs to match)
- $and (all need to match)
- $nor (not or)
- $not

Array:

- $in (value needs to be one of the elements of the given array)
- $nin (not in)
- $all (value needs to match all elements of the given array)
- $size (exact size of array)
- $elemMatch (element in an array matches the specified match expression)

Others:

- $type (check for type number, see)

Querying: $gte

```
<?php
$m = new Mongo;
$c = $m->demo->circus;

$c->insert( array(
  '_id' => 'circ1',
  'name' => 'Diaspora',
  'performers' => 43,
) );

$c->insert( array(
  '_id' => 'circ2',
  'name' => 'Sensible',
  'performers' => 27,
) );
?>
```

Find circuses with 40 or more performers.

```
<?php
$m = new Mongo;
$c = $m->demo->circus;

$r = $c->findOne(
  // query
  array( 'performers' => array( '$gte' => 40 ) ),
  // projection
  array( 'name' => true, 'performers' => true )
);
var_dump( $r );
?>
```

Updating documents

```
<?php
$m = new Mongo;
$c = $m->demo->elephpants;
$c->remove();

$c->insert( array( '_id' => 'e42', 'name' => 'Kamubpo' ) );
var_dump( $c->findOne( array( '_id' => 'e42' ) ) );

$c->update( array( '_id' => 'e42' ), array( 'name' => 'Bo Tat' ) );
var_dump( $c->findOne( array( '_id' => 'e42' ) ) );

$c->update( array( 'name' => 'Bo Tat' ), array( 'age' => '17' ) );
var_dump( $c->findOne( array( '_id' => 'e42' ) ) );
?>
```

update() replaces the document matching criteria entirely with a new object.

Modifying documents

```
<?php
$m = new Mongo;
$c = $m->demo->elephpants;
$c->remove();

$c->insert( [
  '_id' => 'e43',
  'name' => 'Dumbo'
] );

$c->update(
  [ 'name' => 'Dumbo' ], // criteria
  [ '$set' => array [ 'age' => '17' ] ] // modifiers
);

// document is now:
[
  '_id' => 'e43',
  'name' => 'Dumbo',
  'age' => '17',
]
?>
```

Updating documents

Update only updates the first document it finds by default.

```
<?php
$m = new Mongo;
$c = $m->demo->elephpants;
$c->drop();

$c->insert( array( '_id' => 'e42', 'name' => 'Kamubpo', 'age' => 17 ) );
$c->insert( array( '_id' => 'e43', 'name' => 'Denali', 'age' => 17 ) );

$c->update( array( 'age' => 17 ), array( '$inc' => array( 'age' => 1 ) ) );

var_dump( iterator_to_array( $c->find() ) );
?>
```

Updating documents

Update only updates the first document it finds by default.
You can set an option to get all matching documents to be updated

```
<?php
$m = new Mongo;
$c = $m->demo->elephpants;
$c->drop();

$c->insert( [ '_id' => 'e42', 'name' => 'Kamubpo', 'age' => 17 ] );
$c->insert( [ '_id' => 'e43', 'name' => 'Denali',  'age' => 17 ] );

$c->update(
    [ 'age' => 17 ],                // criteria
    [ '$inc' => [ 'age' => 1 ] ],   // update spec
    [ 'multiple' => true ]          // options: multiple
);
?>
```

Upserting documents

upsert: if the record(s) do not exist, insert one.

```
<?php
$m = new Mongo;
$c = $m->demo->elephpants; $c->drop();

function birthDay( $c, $name )
{
    $c->update(
        array( 'name' => $name ),                  // criteria
        array( '$inc' => array( 'age' => 1 ) ),    // update spec
        array( 'upsert' => true )                  // options
    );
    echo $c->findOne( array( 'name' => 'Santon' ) )['age'], "\n";
}

birthDay( $c, 'Santon' );
birthDay( $c, 'Santon' );
?>
```

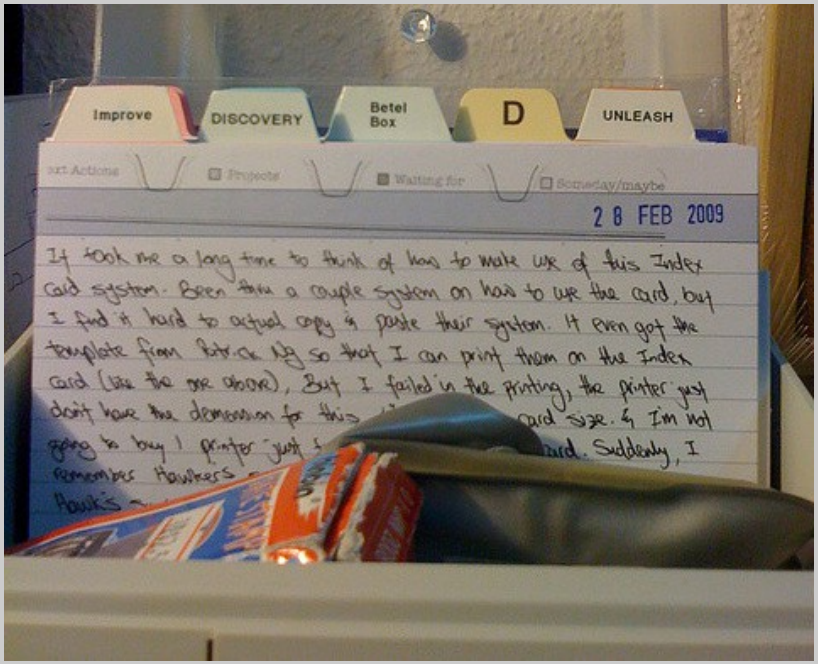
Document update modifiers: Single value manipulation

- $set (sets a field to a new value)
- $unset (removes a field)
- $inc (increments the value in a field)

```
<?php
$m = new Mongo;
$c = $m->demo->circus;
$c->remove();

$c->insert( array( '_id' => 'circ3', 'name' => 'Humberto', 'performers' => 12 ) );
$c->update( array( 'name' => 'Humberto' ), array( '$inc' => array( 'performers' => 4 ) ) );
var_dump( $c->findOne( array( 'name' => 'Humberto' ) ) );
?>
```

Indexes

- Just like a relational database, mongoDB also benefits from indexes.
- Every collection has (automatically) an index on _id.
- Indexes can be on single or multiple fields.
- MongoCursor->explain().



Indexes

```
<?php ini_set('xdebug.var_display_max_depth', 1);
$m = new Mongo;
$c = $m->demo->elephpants;
$c->drop();

$c->insert( [ '_id' => 'ele1', 'name' => 'Jumbo' ] );
$c->insert( [ '_id' => 'ele2', 'name' => 'Tantor' ] );
$c->insert( [ '_id' => 'ele3', 'name' => 'Stampy' ] );

var_dump( $c->find( [ 'name' => 'Jumbo' ] )->explain() );
?>
```

Indexes

```
<?php ini_set('xdebug.var_display_max_depth', 1);
$m = new Mongo;
$c = $m->demo->elephpants;
$c->drop();

$c->ensureIndex( [ 'name' => 1 ] );

$c->insert( [ '_id' => 'ele1', 'name' => 'Jumbo' ] );
$c->insert( [ '_id' => 'ele2', 'name' => 'Tantor' ] );
$c->insert( [ '_id' => 'ele3', 'name' => 'Stampy' ] );

var_dump( $c->find( [ 'name' => 'Jumbo' ] )->explain() );
?>
```

Geospatial Indexes

Helps you with finding POIs (pubs!) in a 2D space

```
<?php
$m = new Mongo; $c = $m->demo->pubs; $c->drop();

$c->ensureIndex( array( 'l' => '2d' ) );
$c->insert([ 'name' => 'Betsy Smith',   'l' => [ -0.193, 51.537 ] ]);
$c->insert([ 'name' => 'London Tavern', 'l' => [ -0.202, 51.545 ] ]);

$closest = $m->demo->command( [
    'geoNear' => 'pubs',
    'near' => [ -0.198, 51.538 ],
    'spherical' => true,
] );

foreach ( $closest['results'] as $res ) {
  printf( "%s: %.2f km\n", $res['obj']['name'], $res['dis'] * 6378 );
}
?>
```

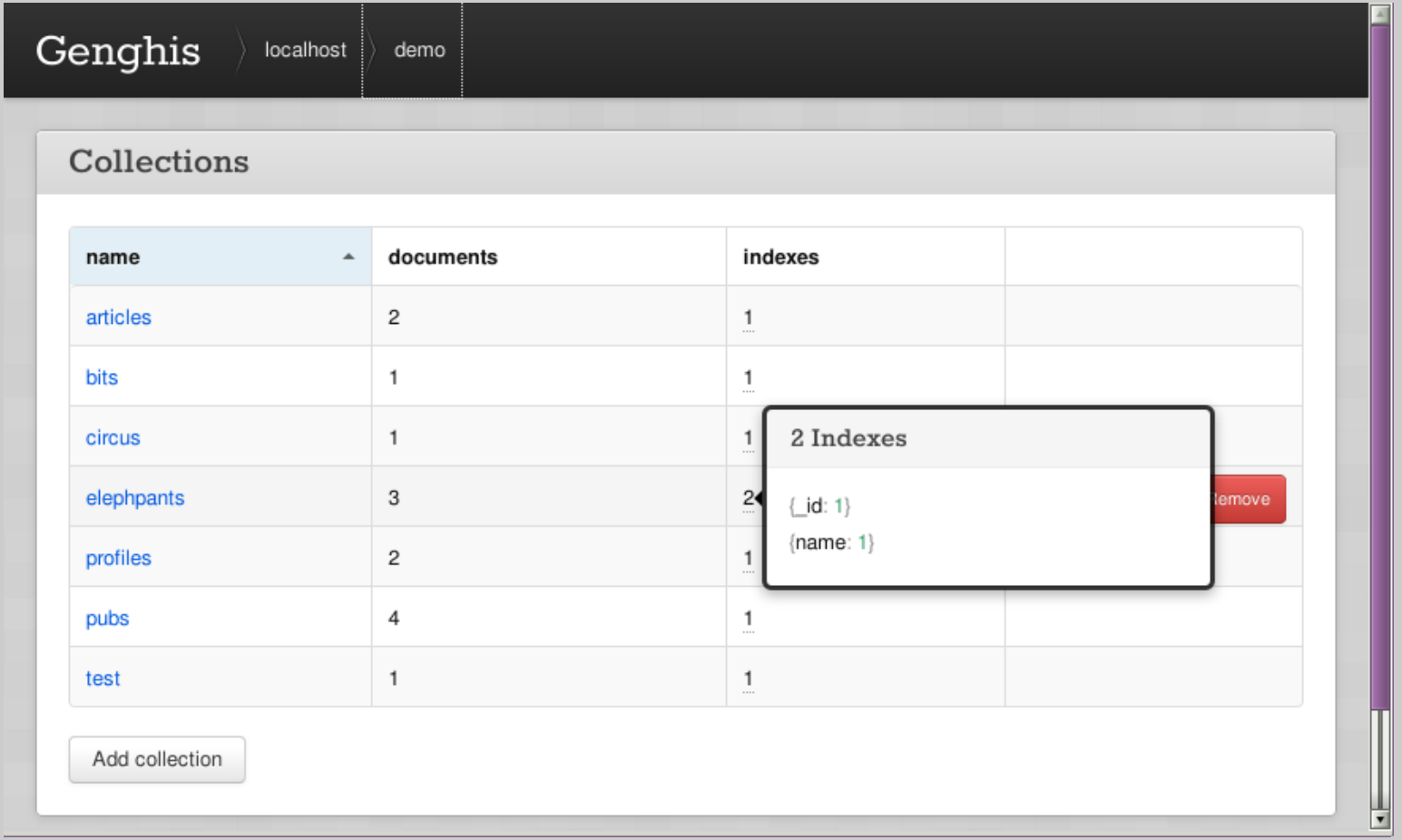
Distinct

```
<?php
$m = new Mongo; $c = $m->demo->pubs; $c->drop();

$c->insert( [ 'name' => 'Betsy Smith',   'city' => 'London' ] );
$c->insert( [ 'name' => 'London Tavern', 'city' => 'London' ] );
$c->insert( [ 'name' => 'Lammars',       'city' => 'Manchester' ] );
$c->insert( [ 'name' => 'Weatherspoons', 'city' => 'Coventry' ] );

$r = $m->demo->command( [
    'distinct' => 'pubs',
    'key' => 'city',
    'query' => [ 'name' => [ '$ne' => 'Weatherspoons' ] ]
] );
var_dump( $r['values'] );
?>
```

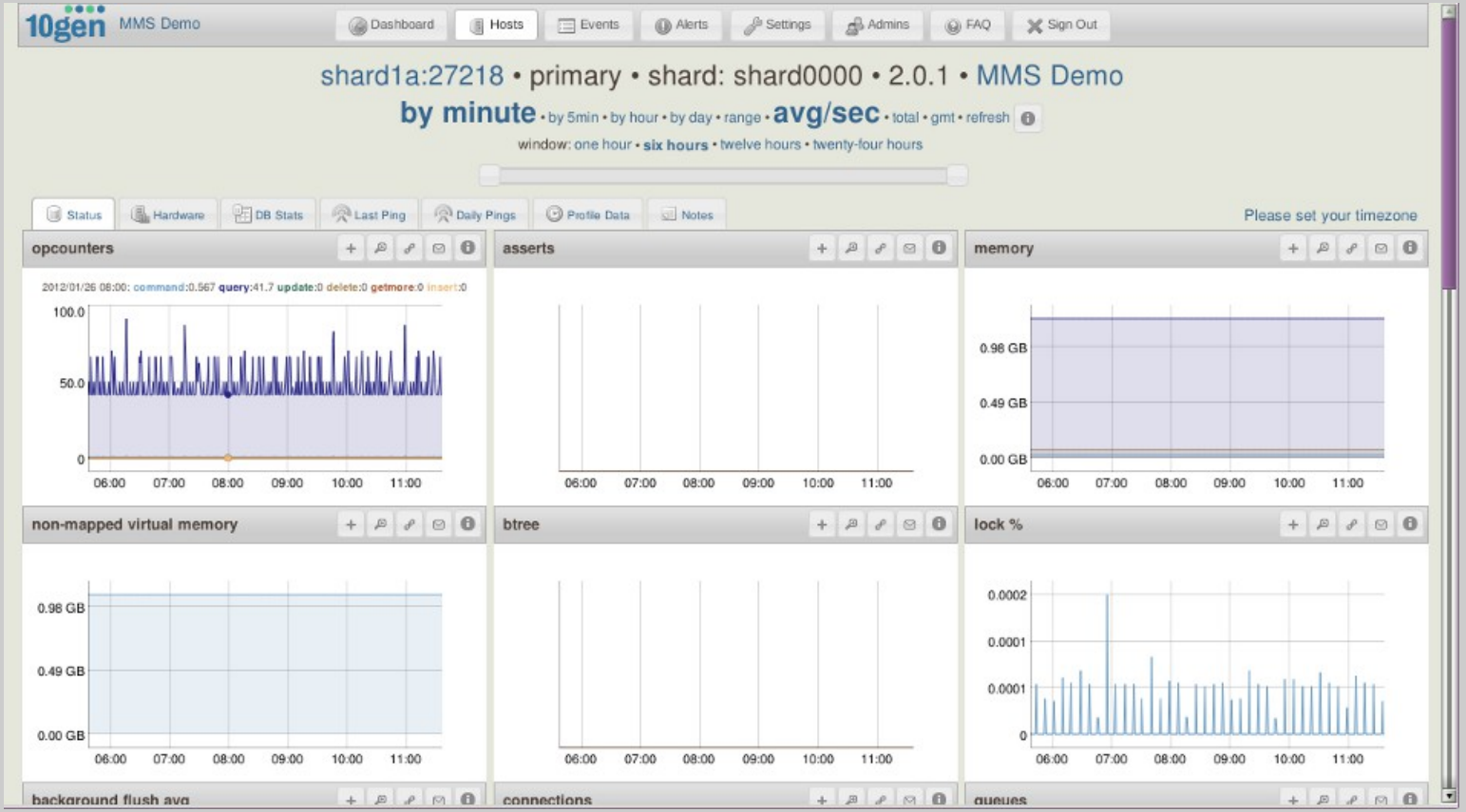
Tools and monitoring

- Ghengis: single file mongoDB management app (a la phpMyAdmin)
- RockMongo: mongoDB management app (a la phpMyAdmin)
- MMS: hosted application for instrumentation

Ghengis

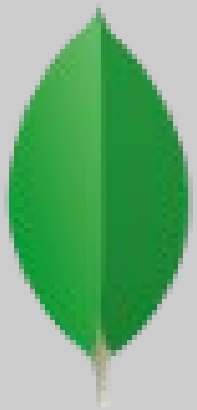

Genghis
localhost
demo
Collections
name
documents
indexes
articles
2
1
bits
1
1
circus
1
1
elephpants
3
2
profiles
2
1
pubs
4
1
test
1
1
2 Indexes
{_id: 1}
{name: 1}
Add collection

MMS


10gen MMS Demo
Dashboard
Hosts
Events
Alerts
Settings
Admins
FAQ
Sign Out
shard1a:27218 • primary • shard: shard0000 • 2.0.1 • MMS Demo
by minute • by 5min • by hour • by day • range • avg/sec • total • gmt • refresh
window: one hour • six hours • twelve hours • twenty-four hours
Status
Hardware
DB Stats
Last Ping
Daily Pings
Profile Data
Notes
Please set your timezone
opcounters
asserts
memory
non-mapped virtual memory
btree
lock %
background flush avg
connections
queues

Resources

- These slides: http://derickrethans.nl/talks.html
- Contact me: Derick Rethans: @derickr, derick@10gen.com
- Feedback: http://joind.in/4976



mongoDB