Welcome!

MongoDB introduction

London Web - London, UK - Mar 22nd, 2012

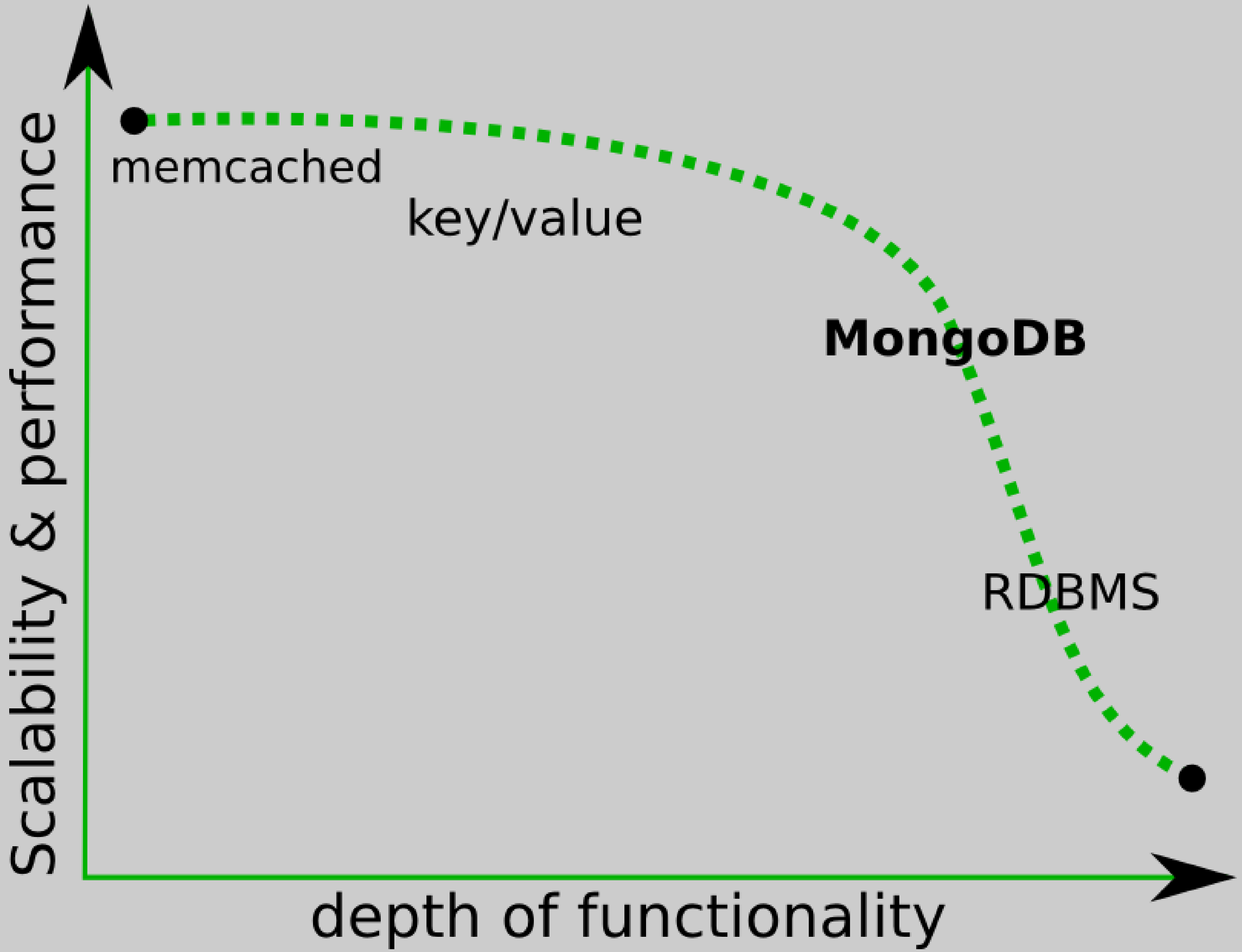
Derick Rethans - derick@10gen.com - twitter: @derickr

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

Database landscape



NoSQL

MEMCACHED
redis

Key/value

Cassandra
HBASE
riak
amazon web services
Amazon DynamoDB

Column

Neo4j
the graph database
sones

Graph

CouchDB
relax
mongoDB
terrastore

Document

Terminology

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

Documents

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

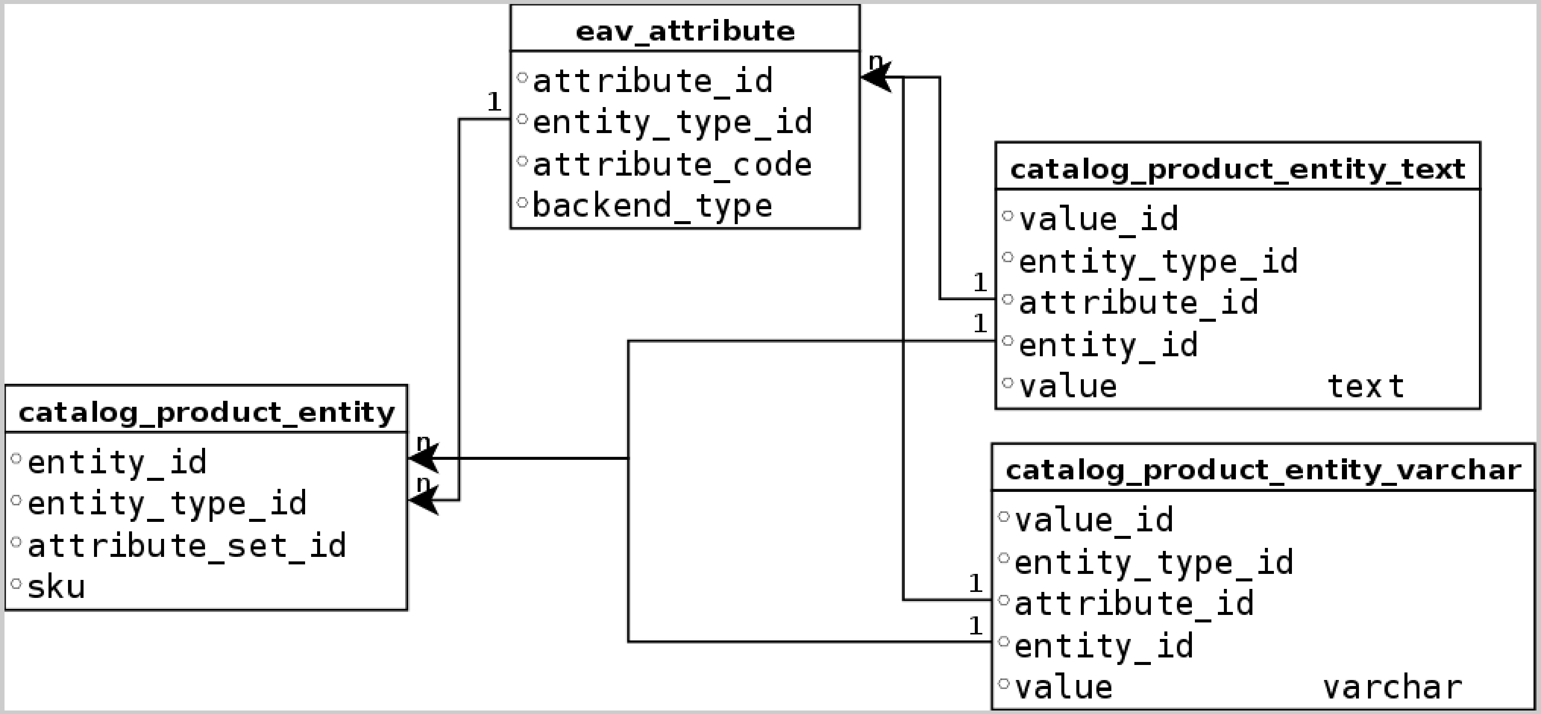
Simple document:

```
{
   "_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",
    }
  ]
}
```

EAV: Entity Attribute Value



```
SELECT cpe.entity_id, value AS name
FROM catalog_product_entity cpe

INNER JOIN eav_attribute ea
    ON cpe.entity_type_id = ea.entity_type_id

INNER JOIN catalog_product_entity_varchar cpev
    ON ea.attribute_id = cpev.attribute_id AND
      cpe.entity_id = cpev.entity_id

WHERE ea.attribute_code = 'name'
```

EAV: Entity Attribute Value

```
SELECT entity_id, attribute_code, value
FROM catalog_product_entity_text cpev
JOIN eav_attribute ea ON cpev.attribute_id = ea.attribute_id;

| entity_id | attribute_code         | value               |
+-----------+------------------------+---------------------+
|         1 | description            | Cute elephpant      |
|         1 | short_description      | It&#39;s cute       |
|         1 | meta_keyword           | NULL                |

SELECT entity_id, attribute_code, value
FROM catalog_product_entity_int cpev
JOIN eav_attribute ea ON cpev.attribute_id = ea.attribute_id;

| entity_id | attribute_code          | value |
+-----------+-------------------------+-------+
|         1 | status                  |     1 |
|         1 | visibility              |     4 |
|         1 | tax_class_id            |     2 |
```

In MongoDB

```
{
    '_id': 1,
    'name' : 'Elephpant',
    'url_key': 'elephpant',
    'description': 'Cute elephpant',
    'short_description': "It's cute",
    'status': 1,
    'visibility': 4,
    'tax_class_id': 2,
}
```

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']);

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

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 = array(
  "_id" => "derickr",
  "name" => "Derick Rethans",
  "articles" => array(
    array(
      "title" => "Profiling PHP Applications",
      "url" => "http://derickrethans.nl/talks/profiling-phptour.pdf",
    ),
    array(
      "title" => "Xdebug",
      "url" => "http://derickrethans.nl/talks/xdebug-phpbcn11.pdf",
    )
  )
);

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

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";
}
?>
```

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

Updating documents

```php
<?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.
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' );
?>
```

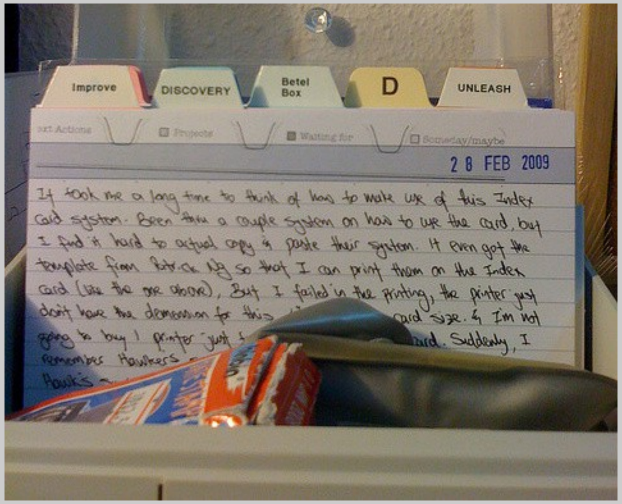
Document update modifiers: Single value manipulation

- $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( [ '_id' => 'circ3', 'name' => 'Humberto', 'performers' => 12 ] );
$c->update(
    [ 'name' => 'Humberto' ],             // query
    [ '$inc' => [ 'performers' => 4 ] ]  // update spec
);
var_dump( $c->findOne( [ '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().



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

 $c->insert( array( '_id' => 'ele1', 'name' => 'Jumbo' ) );
 $c->insert( array( '_id' => 'ele2', 'name' => 'Tantor' ) );

 var_dump( $c->find( [ '_id' => 'ele1' ] )->explain() );
 ?>
```

Indexes

```php
<?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

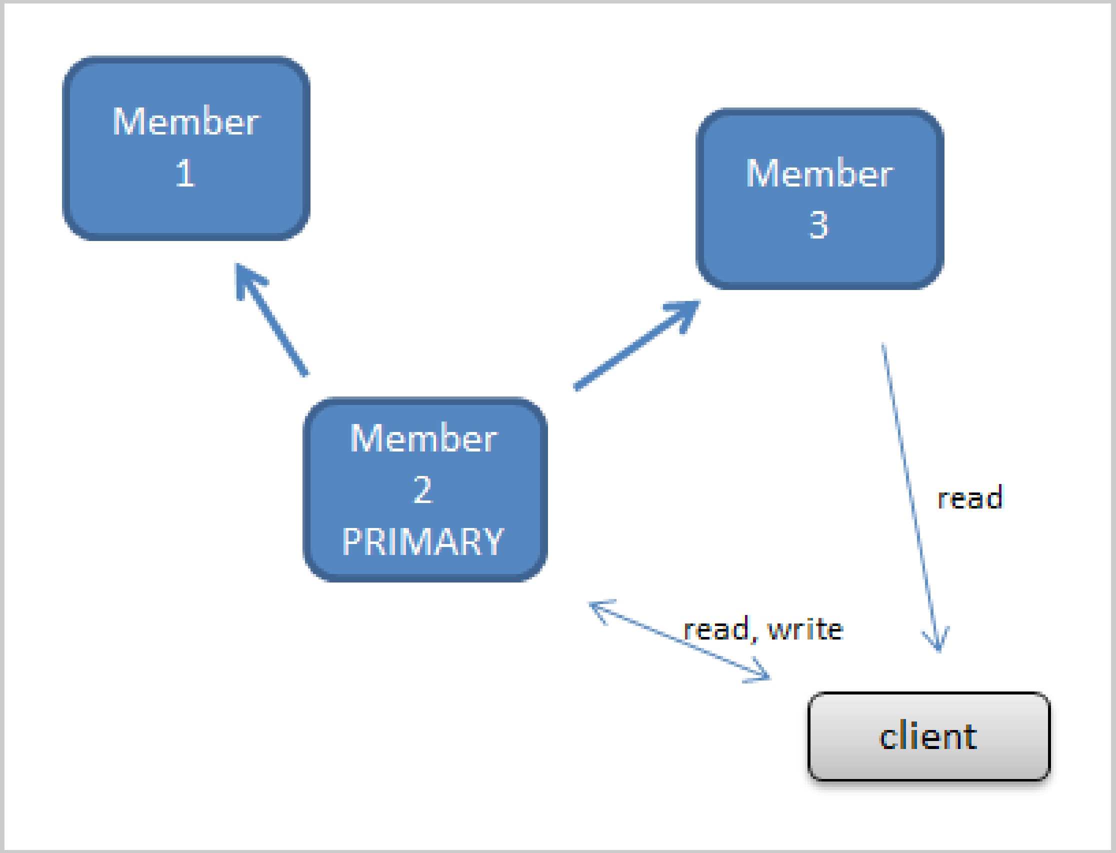
```
<?php ini_set('xdebug.var_display_max_depth', 1);
$m = new Mongo;
$c = $m->demo->elephpants;
$c->drop();

$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

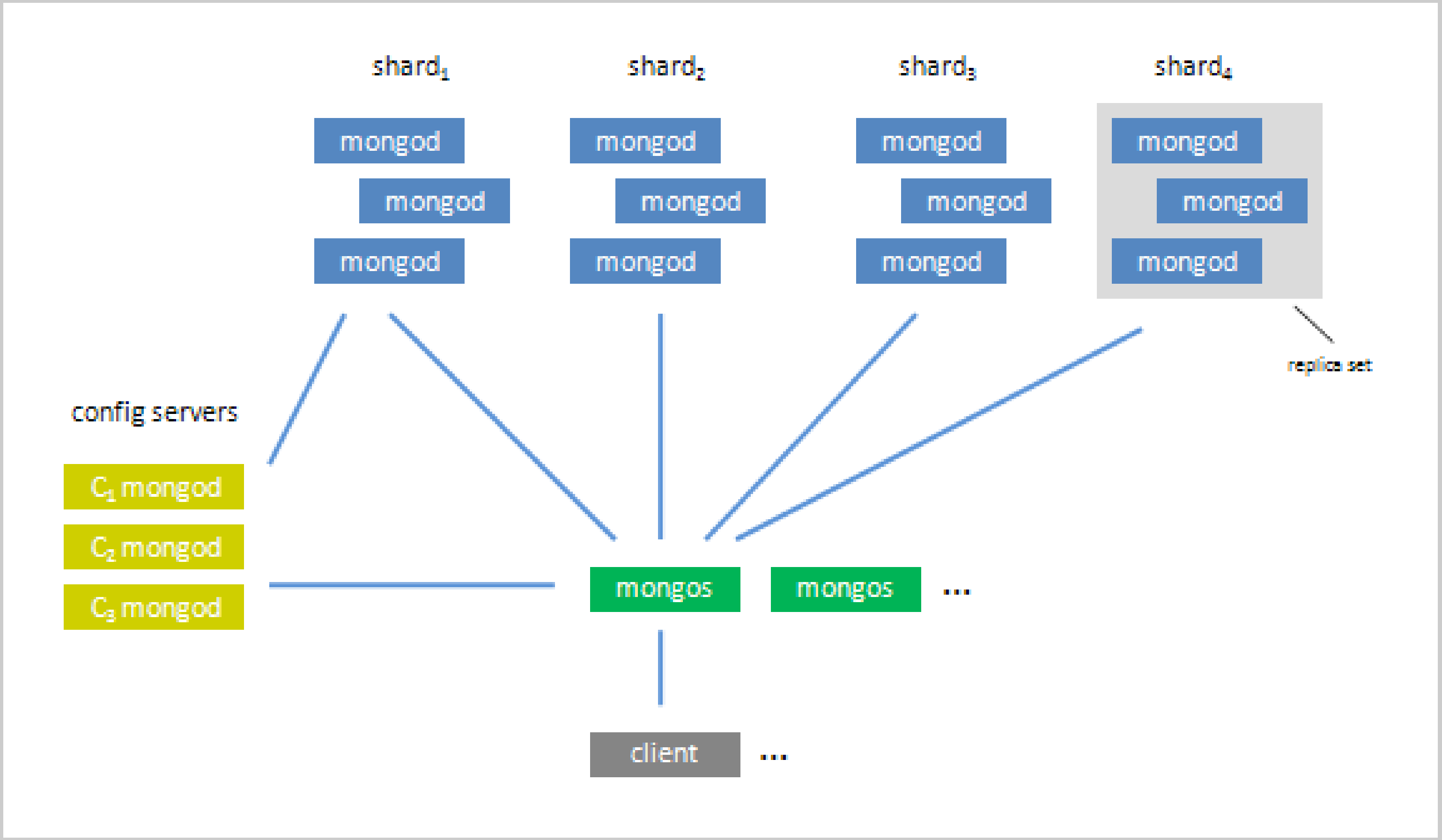
```
<?php
$m = new Mongo; $c = $m->demo->pubs; $c->drop();

$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 );
}
?>
```

Replication



- Failover/Availability
- Scaling reads
- Primaries, secondaries and arbiters
- Odd number to prevent split brain

Sharding

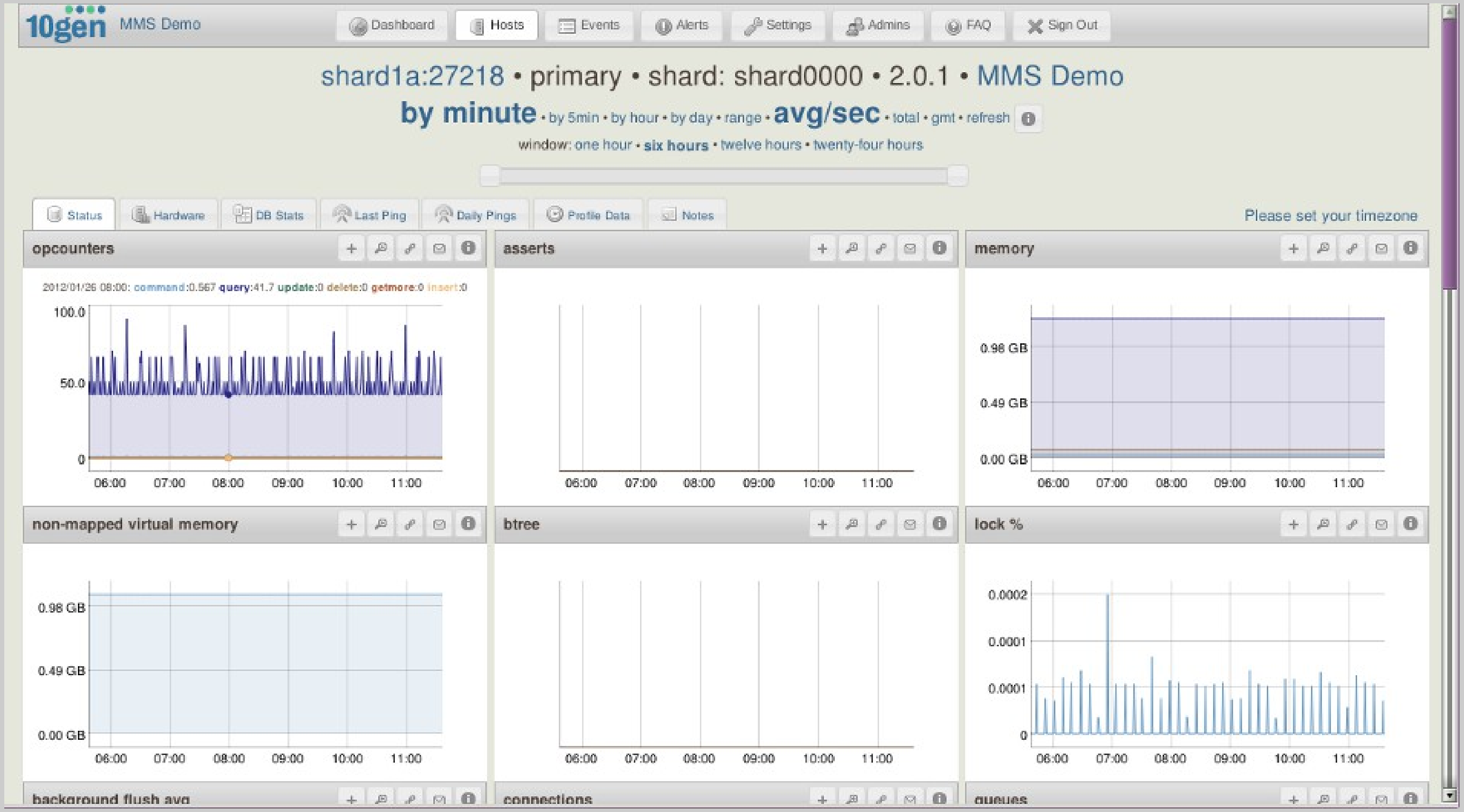


- Scaling writes and reads
- Config servers, router (mongos) and replica sets

Who uses MongoDB?

craigslist
CHECK24
sourceforge
Pinterest
Disney
IGN
SHOPwiki
HM Government
shutterfly
CERN
bit.ly
theguardian
foursquare
The New York Times
Chicago Tribune
Grooveshark
intuit
Flowdock
edelight
Viber
Eventbrite
Events Made Easy
Word²
sk songkick
UBERVU
SAP
the national archives
Trello
CustomInk
Design T-shirts Online
floxee

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
2012/01/26 08:00: command:0.567 query:41.7 update:0 delete:0 getmore:0 insert:0
asserts
memory
non-mapped virtual memory
btree
lock %
background flush avg
connections
queues

MongoDB UGs and Conference

MongoDB UK

- Annual one day conference dedicated to the open source database MongoDB.
- June 20th, 2012 at the Mermaid Conference & Events Centre
- Early Bird ($50) ends May 23rd

Mongo UGs

- MongoDB London (Mar 29th): bi-monthly
- Office Hours (Apr 4th): bi-weekly near Old Street

Resources

- Slides: http://derickrethans.nl/talks/:-:talk_id:-:
- Contact me: Derick Rethans: @derickr, derick@10gen.com
- Feedback: