mongoDB

24.02.2012

Introduction

Vincent van Proosdij, Senior Consultant

Tobias Meier, Senior Consultant (QS)

Agenda

mongoDB

- nosql background
- what is mongodb
- more mongodb details
- socket.io introduction
- extend the website tracking demo

NoSql background

- term for everything non RDBMS/relational

no joins
+ no complex transactions
====================
scalable architectures
& new data models

- webscale - large scale parallel processing of big data

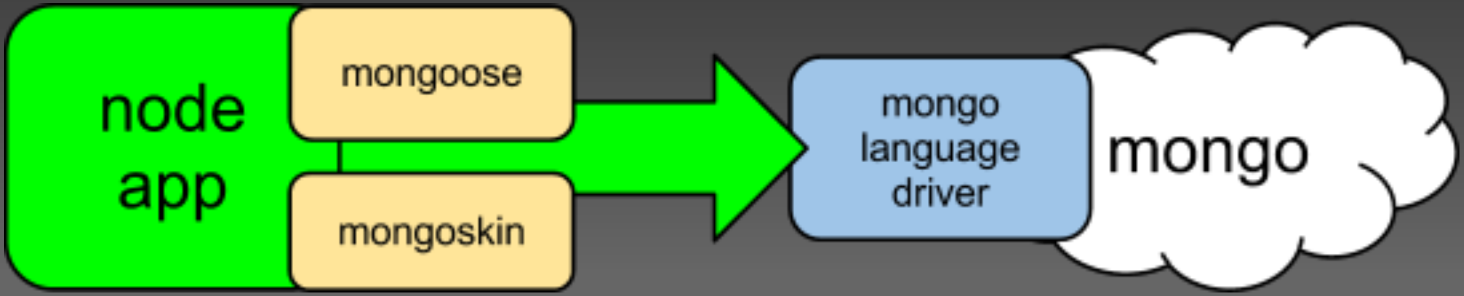
- various types
  - key-value stores (amazon simpledb, azure table storage)
  - document stores (mongodb, couchdb)
  - graph databases (neo4j) > tracking relations

- specialized databases - Different data models are used to solve different problems. graph problem in a graph database etc...

Mongo is:

- started in 2007 by 10gen, 2009 opensource
- non-relational
- document oriented (JSON), schema free
- javascript all the way, written in c++
- high performance (no joins, embedded documents)
- scalable (sharding with read/write distribution)
- has many language drivers (ruby/c#/java/perl/javascript...)

http://www.mongodb.org/display/DOCS/Introduction



Mongo terminology

RDBMS	Mongo
Table	Collection
Rows	Document
Index	Index
Primary key	_id field
Join	Embedding & Linking
Partition	Shard

- regular processes have regular data
    - need RDBMS

Cool features

- map/ reduce
    - aggregate data

- geo-spatial indexes
    - $near

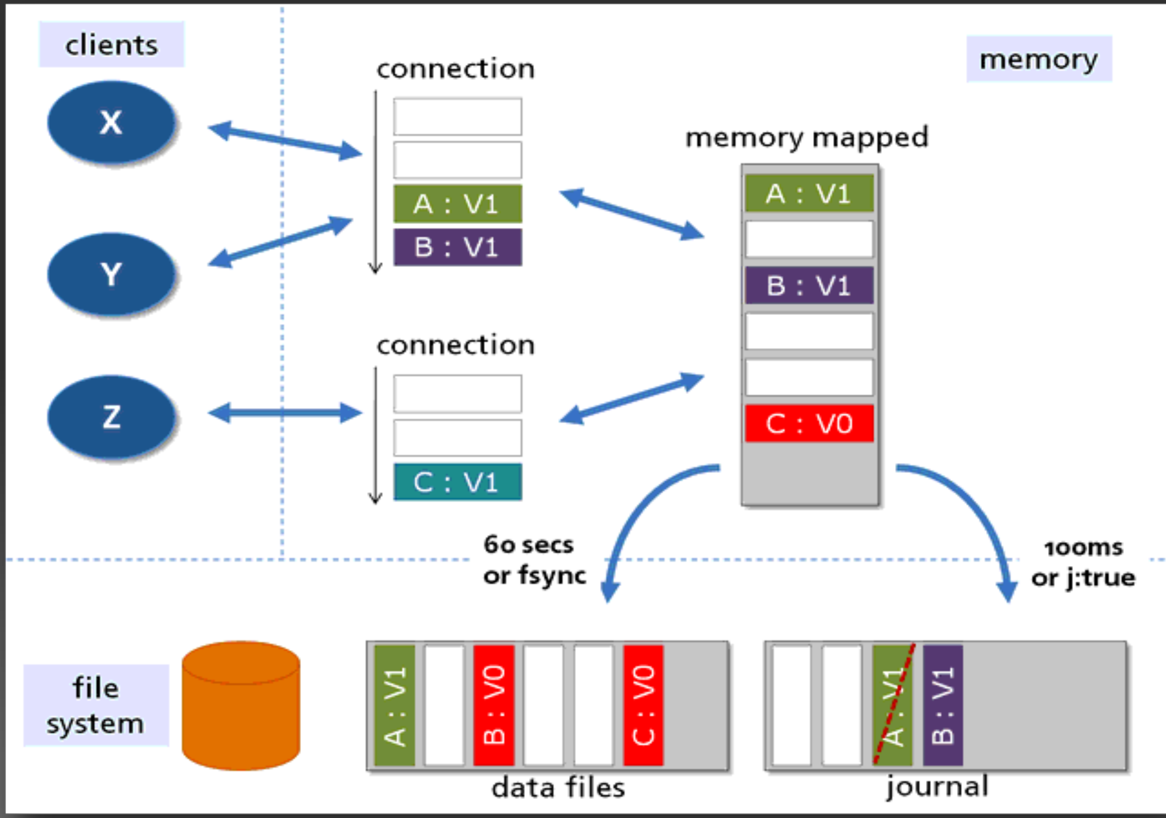
- gridfs
    - store large binary files

- capped collections
    - fixxed space FIFO collections

- upserts
    - more on that later

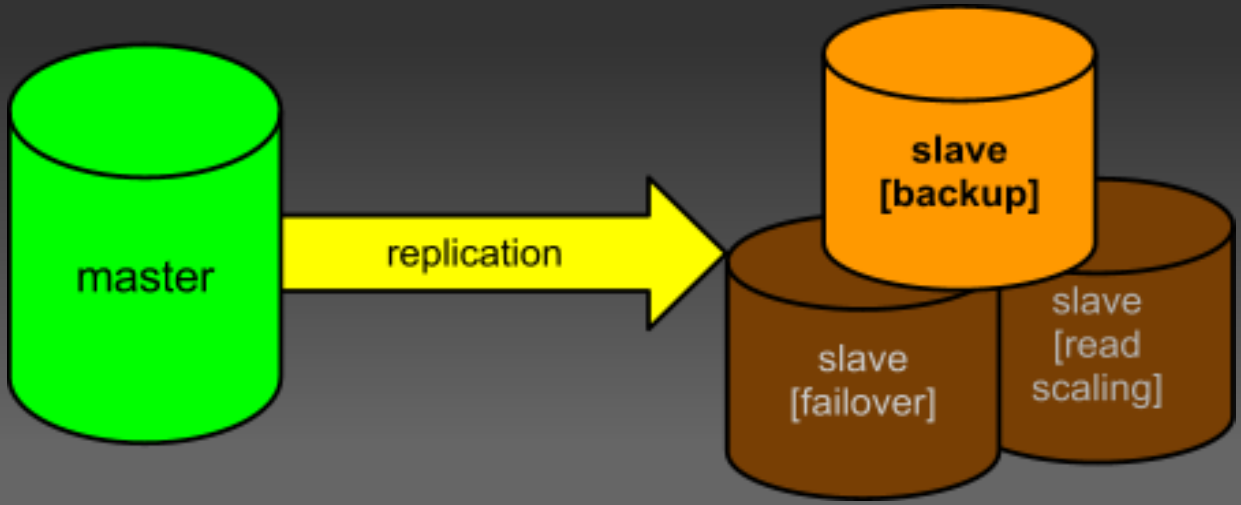
Under the hood

- written in c++
- available on most platforms
- data serialized to BSON (fast)
- extensive use of memory-mapped files (fast)



Backups

- data file backup
  - downtime: shutdown server, then copy files
- mongodump
  - no downtime: flush writes > lock db > dump > unlock
- slave (recommended)
  - a replication slave is in sync with master and can utilize above methods



JSON documents

```
db.posts.find()
{
"_id" : ObjectId("4bfe4946cfbfb01420000011"),
"created_at" : "Thu, 27 May 2012 10:25:40 +0000",
"author" : "vip32",
"text" : "my blogpost text here",
"tags" : ["tag1", "tag2"],
"comments" : [{
"author" : 14825648891,
"text": "my comment text"
}],
}
```

- _id is unique, but can be anything you'd like
- comments is a an embedded document

Queries, it's all javascript

- conditional operators
  - $gt, $lt, $gte, $lte, $ne, $all, $in, $nin, $size, $exists, $type, ...
  - db.posts.find({tags: {$exists: true}})
- regular expressions
  - db.posts.find({author: /^r*/i })
- count, limit, skip, group
  - db.posts.find({author: "vip32"}).count()

Insert/Update

- add document
  - save({author: "vip32", text: "my blog text"})

- embedded documents
  - comment = {author: "you", text: "right on"}

    ```
    db.posts.update({_id:"xxx"},
                          {'$push': {comments: comment}})
    ```

    ^^ great for evolving schemas

Short shell demo

- **is it javascript?**

```
1+1
```

- **insert**

```
for (var i = 1; i <= 24; i++) db.blog.save({author : "vip32", text : "my blog text"});
db.runCommand( "getlasterror" )
```

- **find**

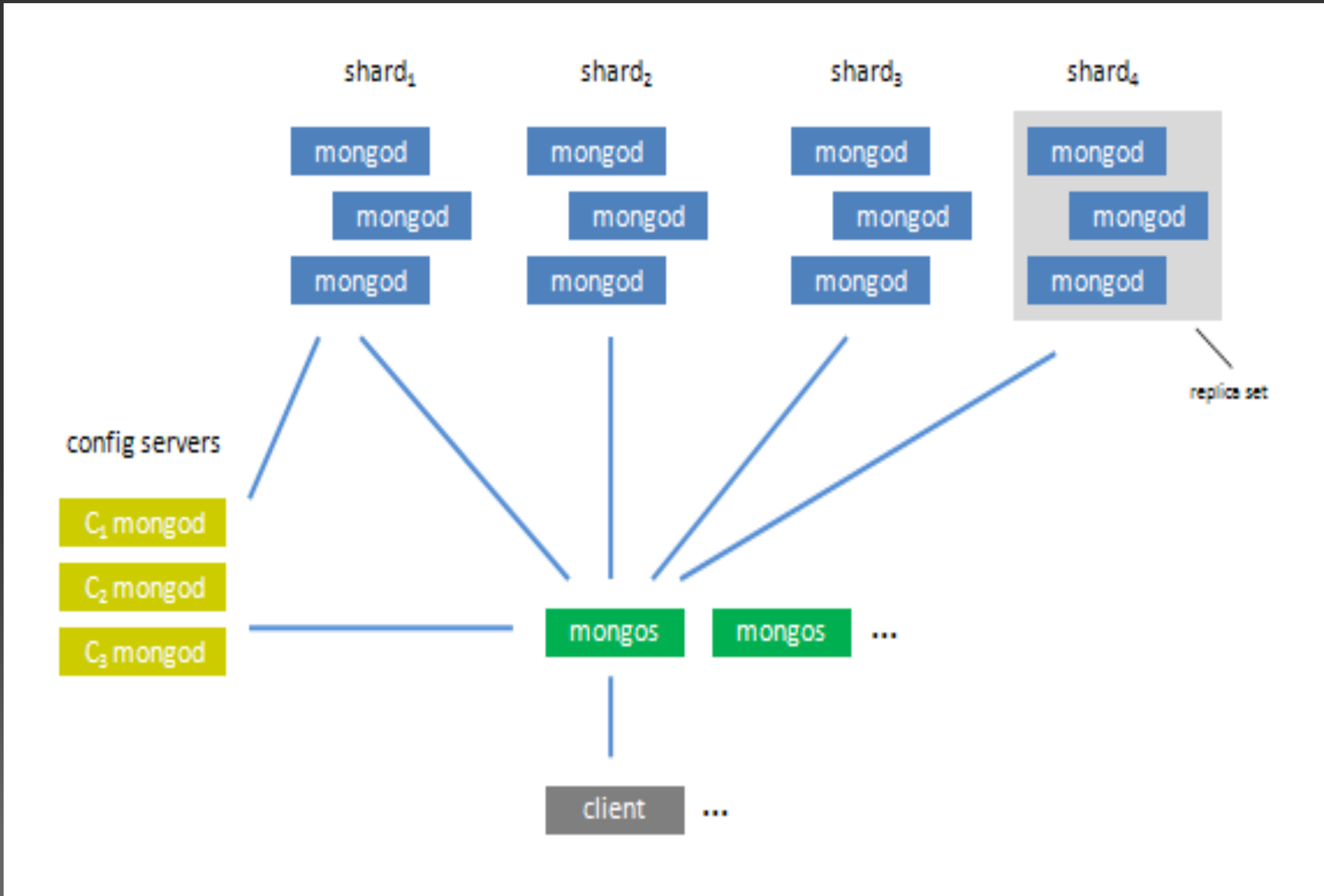
```
db.blog.find()
it
db.blog.find({author: "vip10"})
```

- **add tags**

```
tag = {name: "new"}
db.blog.update({author: "vip10"}, {$push: {tags: tag}}, false, true)
db.blog.find()
```

Sharding

- partitioning
- too big a topic to handle right now

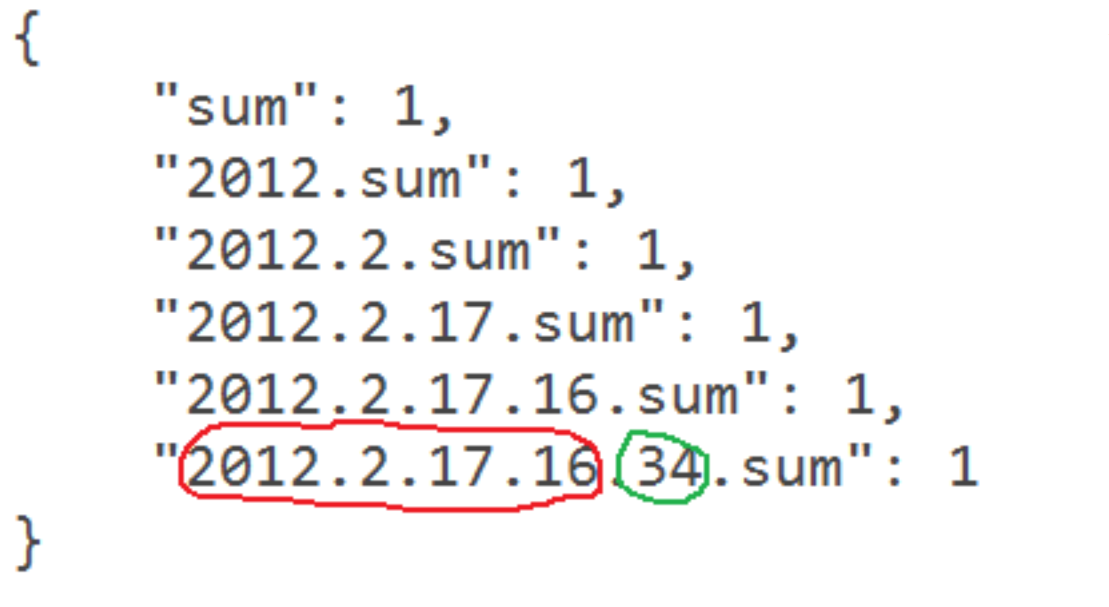


Upserts, something special

- one record updates the **whole** document and embedded documents

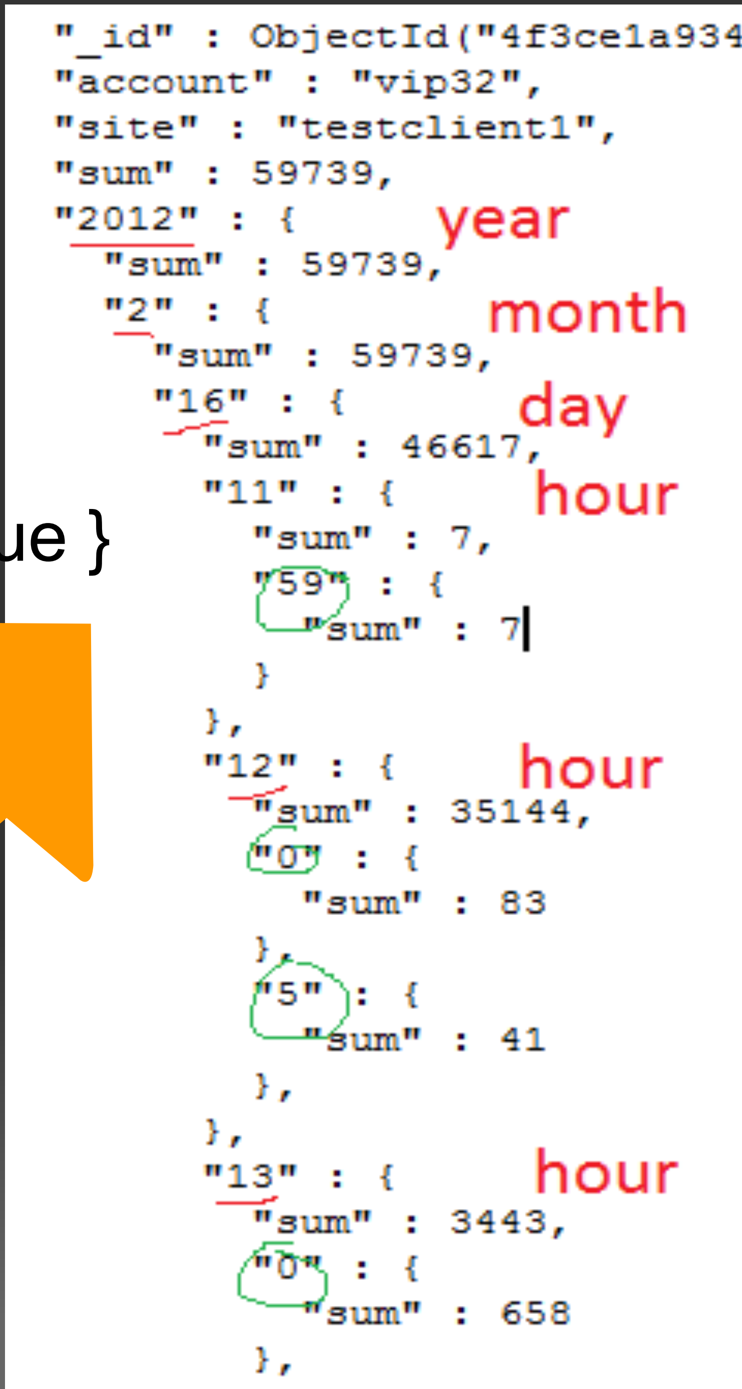
- db.statrecords.**update** {account: "vip32", site: record.site}, {$inc : statrecord}, { upsert: true, safe: true }

record send by application



```
{
    "sum": 1,
    "2012.sum": 1,
    "2012.2.sum": 1,
    "2012.2.17.sum": 1,
    "2012.2.17.16.sum": 1,
    "2012.2.17.16.34.sum": 1
}
```

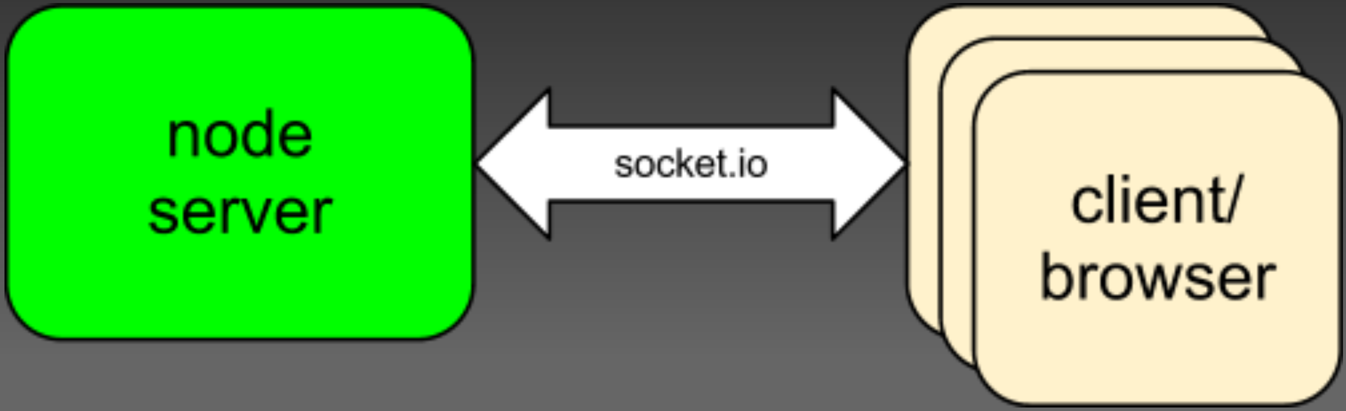
mongo document



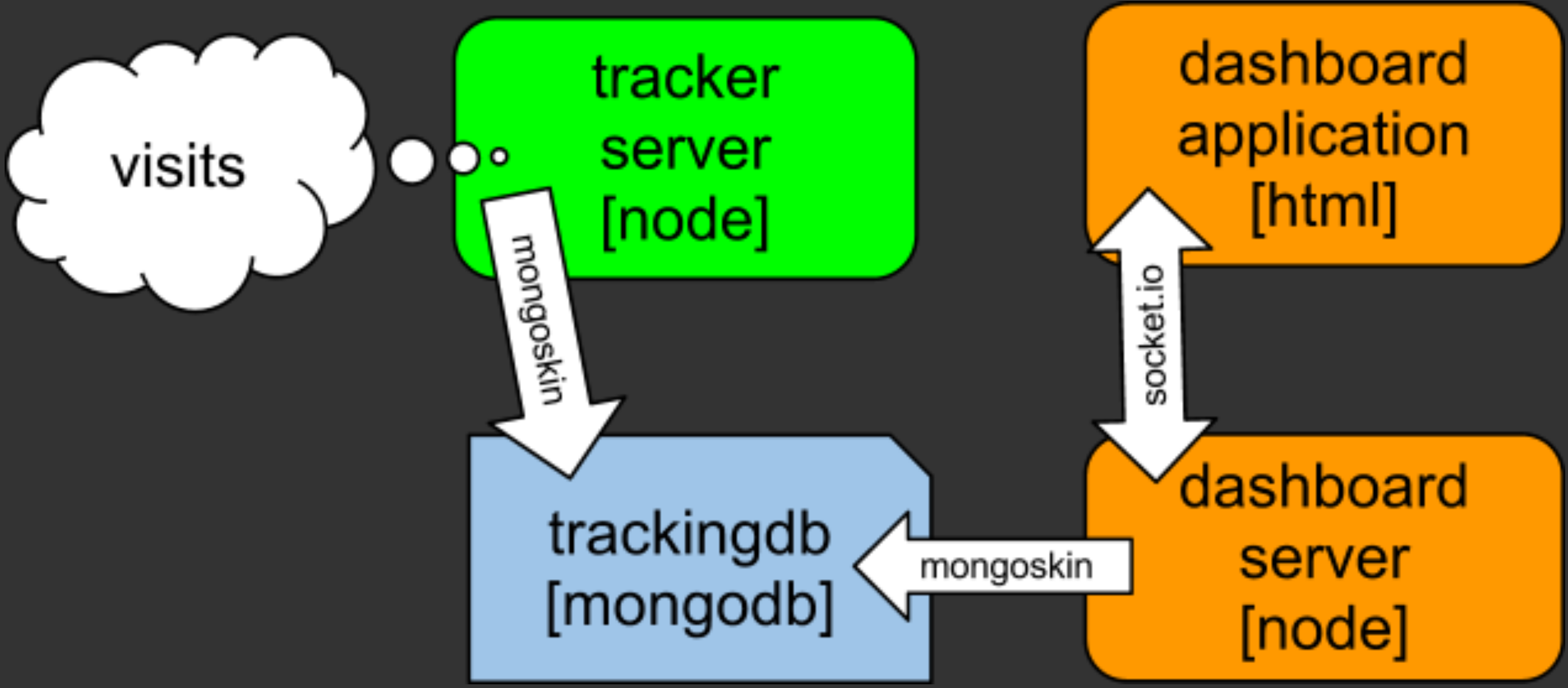
```
"_id" : ObjectId("4f3ce1a934
"account" : "vip32",
"site" : "testclient1",
"sum" : 59739,
"2012" : {
  "sum" : 59739,
  "2" : {
    "sum" : 59739,
    "16" : {
      "sum" : 46617,
      "11" : {
        "sum" : 7,
        "59" : {
          "sum" : 7
        }
      },
      "12" : {
        "sum" : 35144,
        "0" : {
          "sum" : 83
        },
        "5" : {
          "sum" : 41
        },
      },
      "13" : {
        "sum" : 3443,
        "0" : {
          "sum" : 658
        },
```

[Node] Socket.io is:

- a node module (npm install socket.io)
- simplifies asynchronous requests dramaticly
- wraps up websockets and various fallbacks
- has an api that matches node's EventEmitter (emit/on)
- the api is the same for server and client :)



website tracking demo:

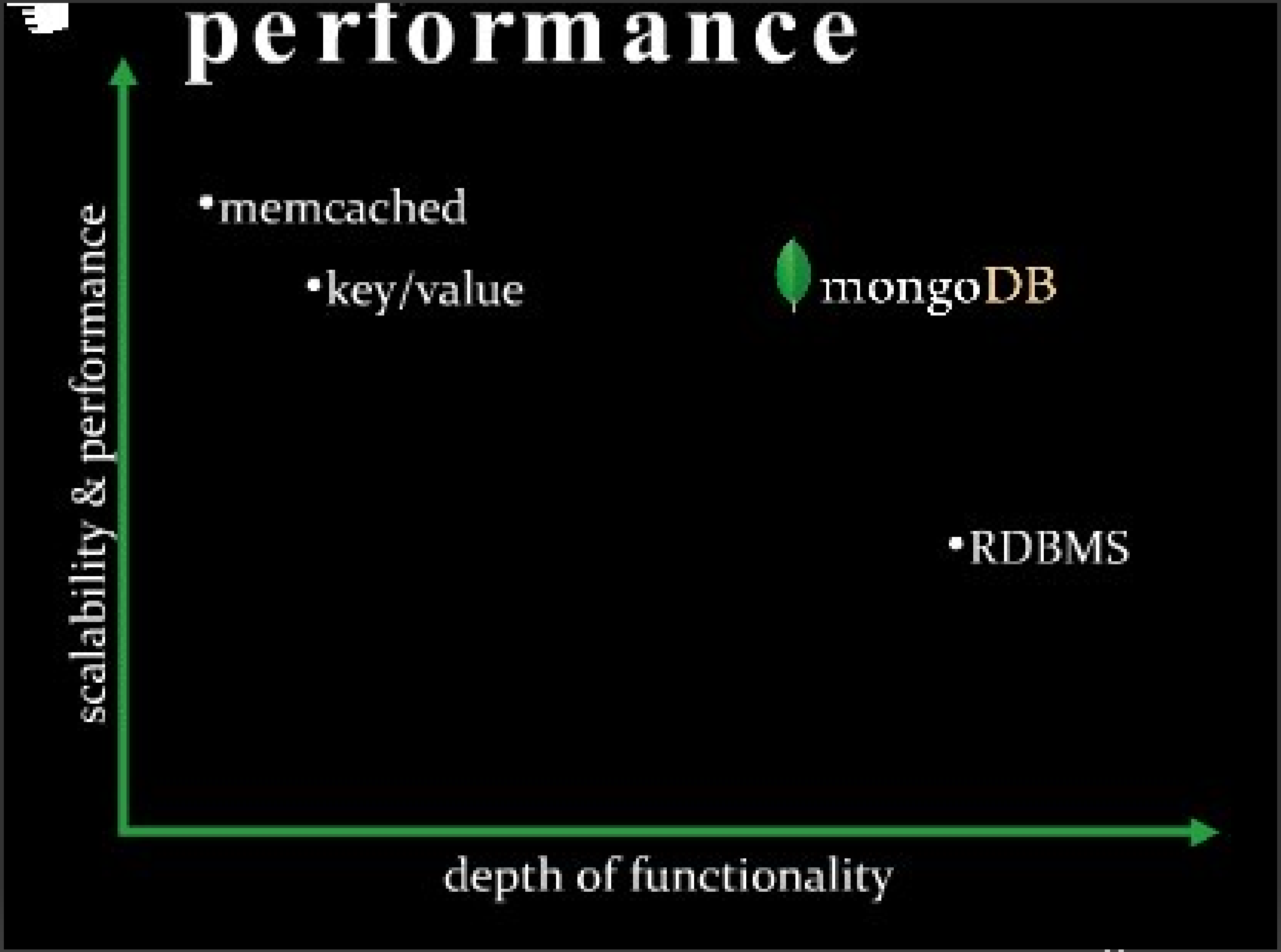


extend the tracker server with persistency and build a dashboard with socket.io connectivity

Fragen ? !

mongoDB

http://github.com/vip32/bit_et2012_1



source: mongodb.org

Wir freuen uns auf Sie.

Standort Mannheim

N7, 5-6
68161 Mannheim

Standort Frankfurt

Solmsstraße 4
60486 Frankfurt

Standort Stuttgart

Königstraße 42
70173 Stuttgart

Standort Karlsruhe

Rüppurrer Straße 4
76137 Karlsruhe

Standort Köln

Richmodstraße 6
50667 Köln