mongoDB

Миграция с MySQL на MongoDB в высоконагруженном проекте.

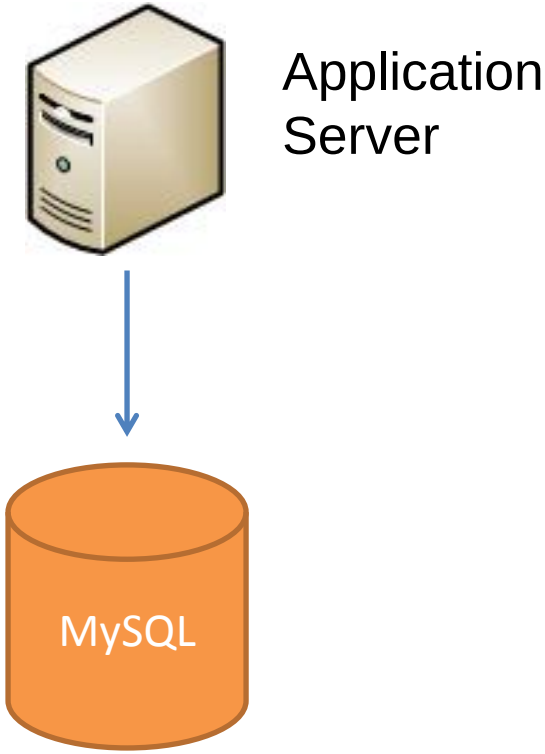
Сергеев Антон
Веб-разработчик

flysoft
mobile developing

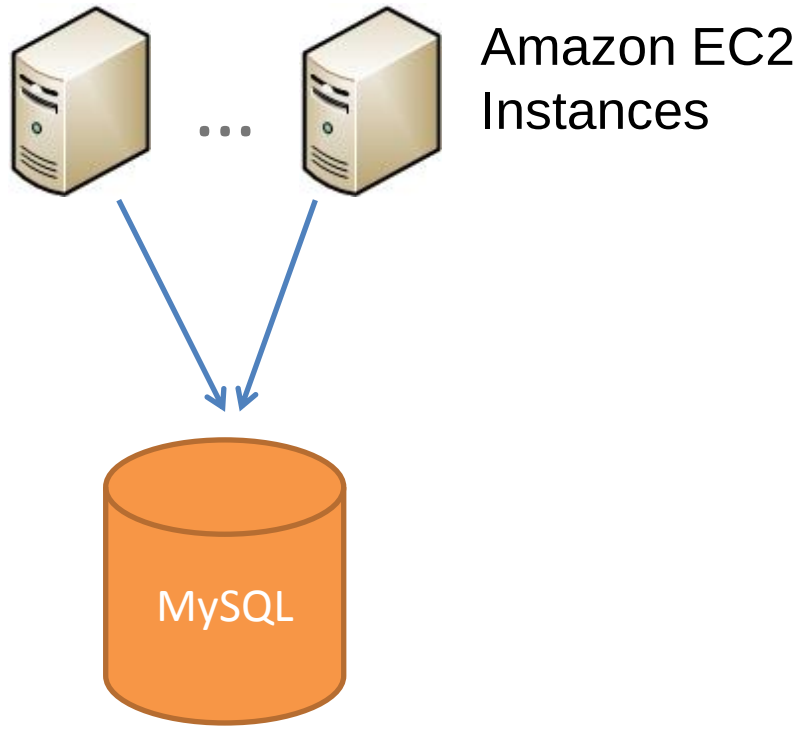
iFunny :)

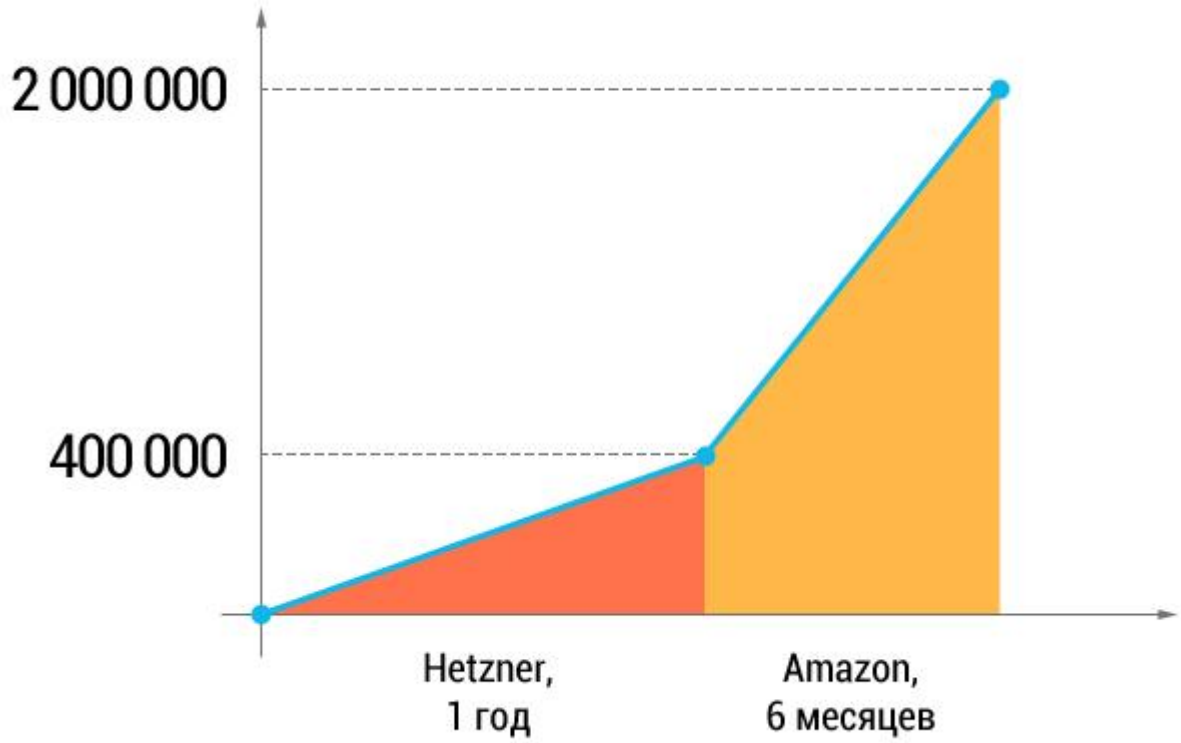
- **2 000 000** пользователей в сутки
- **400 000 000** запросов к базе данных в сутки
- **3 000** операций выборки в секунду
- **7 000** операций добавления/обновления в секунду

HETZNER ONLINE




amazon
web services™


Amazon EC2
Instances
MySQL


2 000 000
400 000
Hetzner,
1 год
Amazon,
6 месяцев

Что же было не так с MySQL?

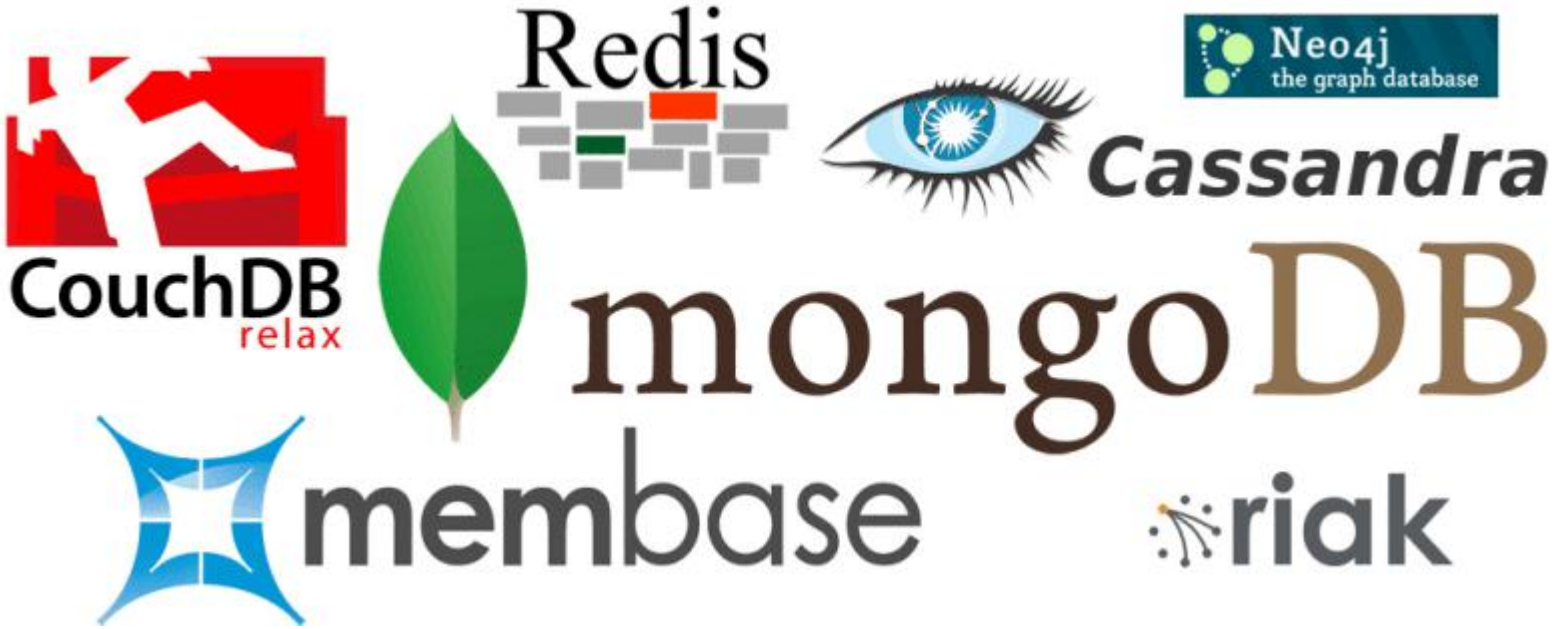
MySQL

Y U So Slow?

Попытки решения проблем на существующей архитектуре

MySQL®

Поиск альтернативы MySQL



mongoDB

http://mongodb.org

Not Only SQL

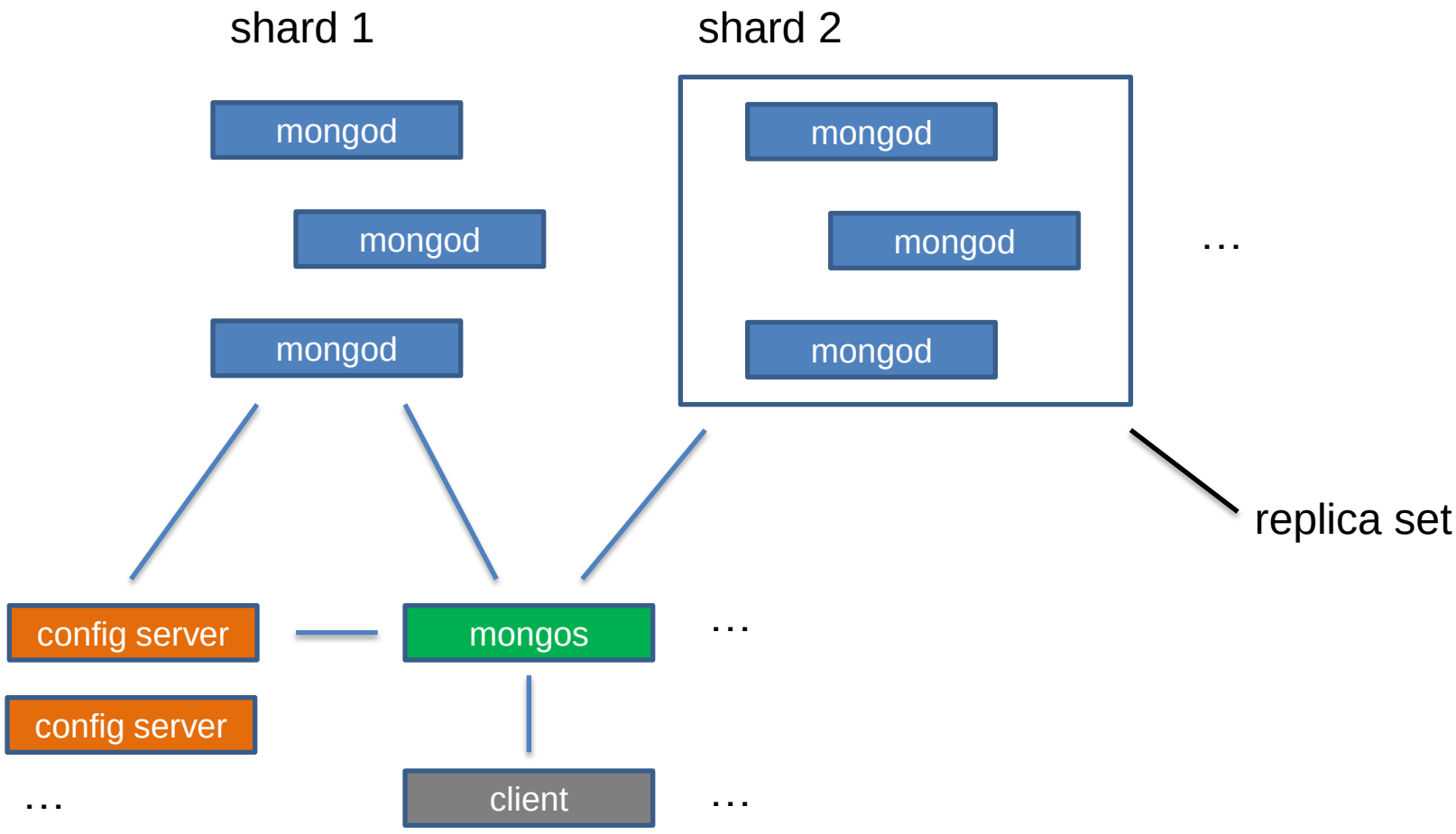
Терминология

MySQL	MongoDB
table	collection
row	BSON document
column	BSON field
join	embedding and linking
primary key	_id field
group by	aggregation

Гибкость хранения и обработки данных

- Schema-less данные
- Оптимизированная запись
- Поддержка атомарных операций
- Индексы
- Capped collections
- MapReduce

Горизонтальное масштабирование



Миграция

Жестко структурированные данные

> SELECT * FROM users

id	first_name	last_name	email	facebook_uid	…
1	Ivan	Ivanov	ivan@ivanov.com	NULL	…
2	Anton	Petrov	NULL	5780342432	…

Schema-less данные

```
> db.users.find();

{
   _id : ObjectId(“4f837427d6b4aba06900b963”),
   name : {first : “Ivan”, last: “Ivanov”},
   email : ivan@ivanov.com
},
{
   _id : ObjectId(“4f837427d6b4aba06900b964”),
   firstName : “Anton”,
   lastName : “Petrov”,
   fbUid : 5780342432
}
```

Оптимизированная запись

```
> db.users.update(
    { name.first: “Ivan” } ,
    { $set: { email: “new@mail.ru" } } ,
     [<upsert>], [<multi>]
);
```

Безопасные обновления

```
> db.runCommand( "getlasterror" );
```

Атомарные операции. FindAndModify

MySQL

> UPDATE user SET num_comments = num_comments + 2 WHERE name = “Anton”;

> SELECT num_comments FROM user WHERE name = “Anton”;

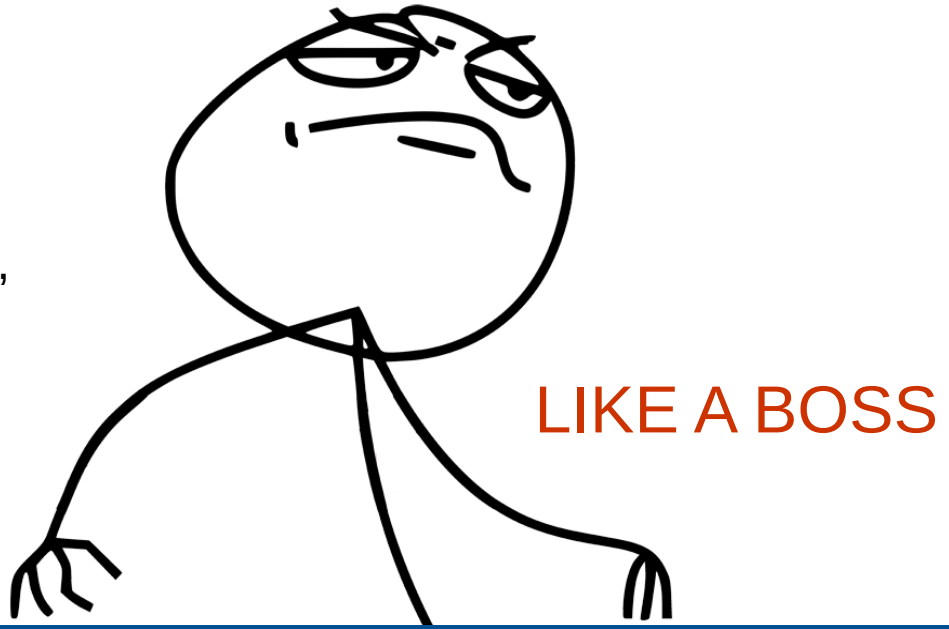
Атомарные операции. FindAndModify

MySQL

```
> UPDATE user SET num_comments = num_comments  + 2 WHERE name =
    “Anton”;
> SELECT num_comments FROM user WHERE name = “Anton”;
```

MongoDB

```
> num = db.users.findAndModify( {
    query: {name: “Anton”},
    update: {$inc: {nComments: 2}},
    new: true,
    upsert: true,
    fields: {nComments: 1}
} );
```



Индексы

Во многом схожи с таковыми в MySQL

> db.users.ensureIndex({name : 1}, {unique: true});

Индексы по встроенным документам

> db.content.ensureIndex({“content.type”: 1})

Sparse-индексы

> db.users.ensureIndex({“lastAt”: -1}, {sparse: true})

Embedding and Linking

MySQL JOINs

```
> SELECT p.id_post, u.nick FROM post AS p
  JOIN user AS u ON u.id_user = p.id_user
  WHERE u.id_user = 25;
```

user

id_user	…
25	…
…	…

post

id_post	id_user	…
2	25	…
…	…	…

Embedding and Linking

MySQL JOINs

```
> SELECT p.id_post, u.nick FROM post AS p
  JOIN user AS u ON u.id_user = p.id_user
  WHERE u.id_user = 25;
```

user

id_user	…
25	…
…	…

post

id_post	id_user	…
2	25	…
…	…	…

MongoDB Embedding

```
> db.users.find(
    {_id: ObjectId("4f4…ce90")},
    {postIds: 1}
);
```

users

```
{
  _id: ObjectId("4f4…ce90"),
  postIds: [2, 15, …]
}
```

Embedding and Linking

MySQL JOINs

```
> SELECT p.id_post, u.nick FROM post AS p
  JOIN user AS u ON u.id_user = p.id_user
  WHERE u.id_user = 25;
```

user

id_user	…
25	…
…	…

post

id_post	id_user	…
2	25	…
…	…	…

MongoDB Embedding

```
> db.users.find(
    {_id: ObjectId("4f4…ce90")},
    {postIds: 1}
);
```

users

```
{
  _id: ObjectId("4f4…ce90"),
  postIds: [2, 15, …]
}
```

MongoDB Linking

```
> db.posts.find(
    {userId: ObjectId("4f4…ce90")},
    {_id: 1}
);
```

users

```
{ _id: ObjectId("4f4…ce90"), … }
```

posts

```
{ _id: 2,
  userId: ObjectId("4f4…ce90") }
```

Autoincrement с integer _id

users

{_id: 1, name: ”Vasya”},

{_id: 2, name: ”Igor”}

Autoincrement с integer _id

users

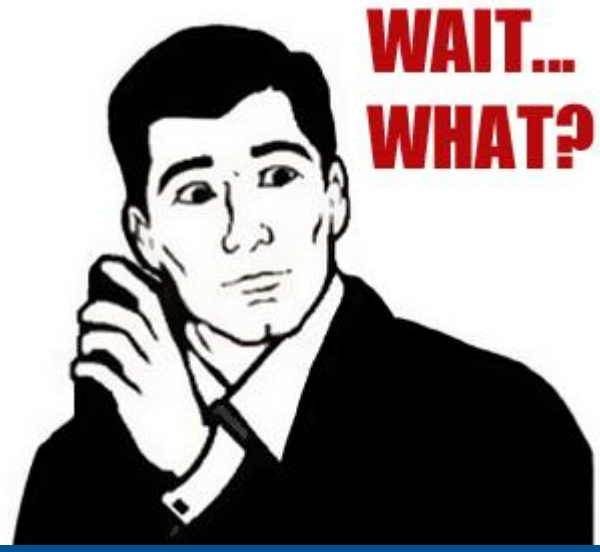
{_id: 1, name: ”Vasya”},

{_id: 2, name: ”Igor”}

Вручную через дополнительную коллекцию “counters”.

counters

{“usersId” : 2}



MapReduce

MySQL

```
> SELECT author, SUM(num_likes) FROM post GROUP BY author;
```

MongoDB

posts

```
{  _id: ObjectId("5a8…be3"),  author: 'mongo',  numLikes: 2 }, {…}
```

1. Map

```
var map = function() {
        emit( this.author, {numLikes: this.numLikes} );
};
```

2. Reduce

```
var reduce = function(key, values) {
        var sum = 0;
        values.forEach(function(doc) {
                sum += doc.numLikes;
        });
        return {numLikes: sum};
};
```

3. Result

```
var group = db.posts.mapReduce(map, reduce);
 > db[group.result].find();
```

MapReduce

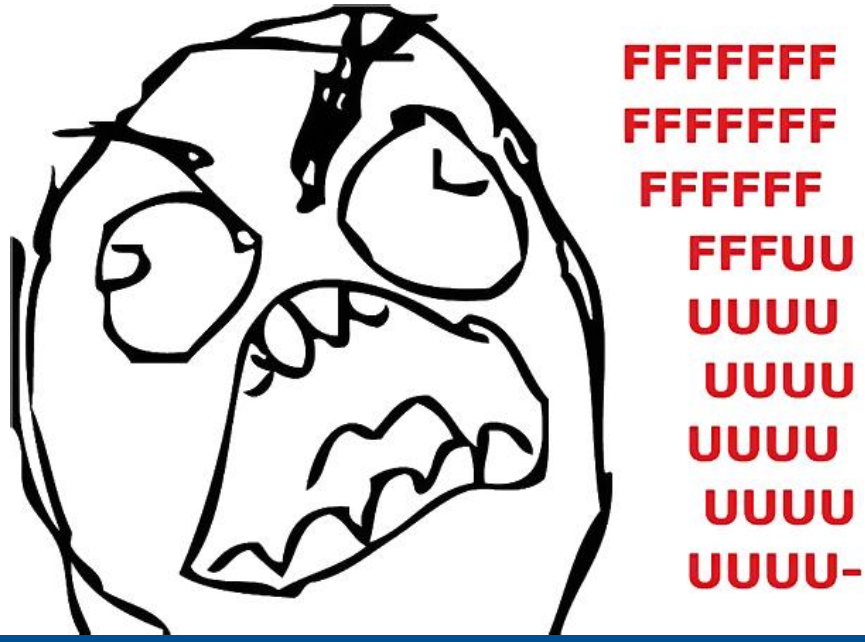
MySQL
```
> SELECT author, SUM(num_likes) FROM post GROUP BY author;
```

MongoDB
posts
```
{  _id: ObjectId("5a8…be3"),  author: 'mongo',  numLikes: 2 }, {…}
```

1. Map
```
var map = function() {
        emit( this.author, {numLikes: this.numLikes} );
};
```

2. Reduce
```
var reduce = function(key, values) {
        var sum = 0;
        values.forEach(function(doc) {
                sum += doc.numLikes;
        });
        return {numLikes: sum};
};
```

3. Result
```
var group = db.posts.mapReduce(map, reduce);
> db[group.result].find();
```



О чем важно не забывать

— Работая с NoSQL, не думайте в духе реляционных БД!

— Смотрите документацию, как принято делать в MongoDB.

— Есть простой и понятный Cookbook.

— Книга “The Little MongoDB Book”.

— Нет изолированности данных.

— Транзакций как таковых нет. Но и это можно обойти.

— Lock ratio. Записывать нужно разумно.

— Ограничение на максимальный размер документа. 16Mb по умолчанию.

— После drop’а коллекций очень желательно делать repairDatabase.

— NoSQL не является заменой реляционных баз данных.

— MongoDB может успешно снижать нагрузку на часть подсистемы хранения данных, сосуществуя с MySQL.

— Возможность постепенного перехода.

— Быстрая дисковая подсистема — хорошо.
Но сразу думайте о горизонтальном масштабировании.

— Весомые трудозатраты при миграции, особенно первое время.

— Инструментарий разработчика хуже развит, чем для MySQL.

Подведем итог

Почему стоит выбирать NoSQL решения

- Гибкие и хорошо масштабируемые
- Обладают компактной структурой хранения данных и отличной производительностью
- Встроенные средства анализа больших объемов данных

Миграция с MySQL на MongoDB в высоконагруженном проекте.

Mongo Manual:
http://www.mongodb.org/display/DOCS/Manual

Mongo Cookbook:
http://cookbook.mongodb.org/

The Little Mongo Book:
http://openmymind.net/2011/3/28/The-Little-MongoDB-Book/

Сергеев Антон
Веб-разработчик
http://twitter.com/hackPNZ

flysoft
mobile developing