## MongoDB Introduction

Ondrej Sika ondrej@ondrejsika.com

Data Script s.r.o., Praha, 29. 3. 2017

https://go.bysika.cz/mongo

#### Agenda

- NoSQL Introduction
- Usage of NoSQL
- MongoDB Overview
  - History, development, present
  - Architecture, Data Types
  - Tools
- MongoDB Installation
- Backup & Restore
- Administration
  - User Management
  - Roles

## **NoSQL Introduction**

#### What is NoSQL

A **NoSQL** database provides a mechanism for storage and retrieval of data which is modeled in means other than the tabular relations used in relational databases. ...

NoSQL databases are increasingly used in **big data** and **real-time** web applications.

#### NoSQL vs RDBMs

NoSQL databases disrupted the database market by offering a more flexible, scalable, and less expensive alternative to relational databases. They also were built to better handle the requirements of Big Data applications.

# Distributed systems

Distributed systems provide horizontal scalability and high availability solutions much less expensive than traditional solutions.

That is a main reason for their rising popularity.

#### **ACID vs BASE**

#### **ACID**

- Atomicity
- Consistency
- Isolation
- Durability

#### **BASE**

- Basically Available
- Soft state
- Eventual consistency

## NoSQL Usage

# Where to use NoSQL

- Big Data
- High Availability Systems
- Unstructured Data

## Advantages Disadvantages

#### Advantages

- Speed
- Scalability
- Cost

#### Disadvantages

- No Transactions
- No Joins

## NoSQL Categories

- Key Value store Redis
- Column oriented
- Graph database Neo4j
- Document oriented MongoDB

## **Key Value Store**

Redis

#### Advantages

- Ultra fast
- Simplicity
- Scalability, Replication

#### Disadvantages

Absence of advanced querying

## **Graph database**

Neo4j

#### Advantages

- Optimized graph queries
- Shortest path algorythms, ...

#### Disadvantages

- Slow on selects throw all the data

# **Document** oriented

MongoDB

#### Advantages

- Document structure

#### Disadvantages

- Data Overhead

# MongoDB

## **MongoDB**

- History
- Architektura
- Data Storage
- Data Types
- Tools

## Historie MongoDB

- 2007 started development by 10gen Inc.
- 2009 Open sourced
- 2013 10gen Inc. renamed to MongoDB Inc.

## MongoDB Architecture

MongoDB stores data as documents in a binary representation called BSON (Binary JSON). Documents that share a similar structure are typically organized as collections.

You can think of collections as being analogous to a table in a relational database: documents are similar to rows, and fields are similar to columns.

## **Data Types**

Every data in MongoDB are serialized to binary JSON - BSON.

### **Data Types**

- Null
- Numbers (ints, longs, doubles)
- String
- Object
- Array
- ObjectId

And more

#### **Tools**

#### Official

- mongostat
- mongotop
- MongoDB Compass
- MongoDB Atlas
- MongoDB Cloud Manager

#### Community

- MongoClient
- AdminMongo

## mongostat

The mongostat utility provides a quick overview of the status of a currently running mongod or mongos instance.

mongostat is functionally similar to the UNIX/Linux file system utility vmstat, but provides data regarding mongod and mongos instances.

## mongotop

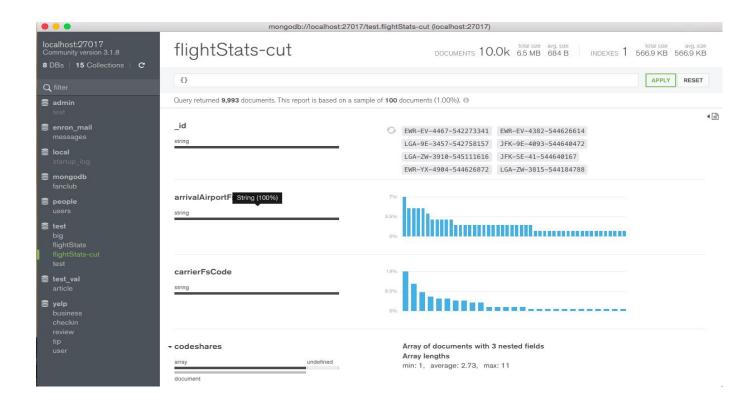
mongotop provides a method to track the amount of time a MongoDB instance spends reading and writing data.

mongotop provides statistics on a per-collection level. By default, mongotop returns values every second.

## MongoDB Compas

**Desktop Client** 

- Desktop client
- All platforms (Linux, OSX, Win)
- Official MongoDB client

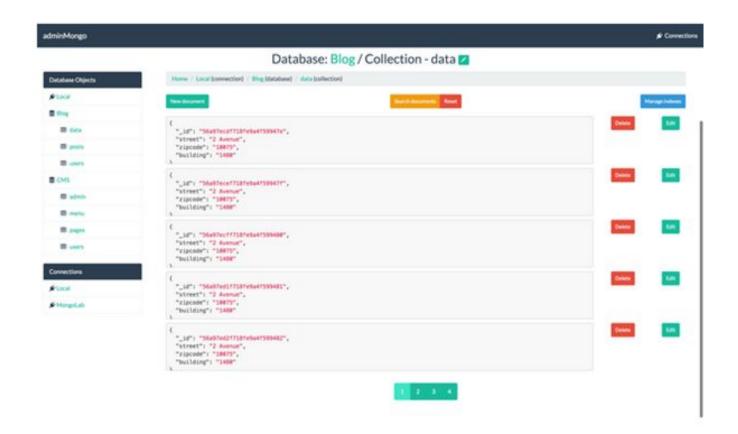


#### MongoDB Compass

## adminMongo

Web Client

- Simple MongoDB Web client
- Responsive User Interface
- Simple monitoring
- Open Source

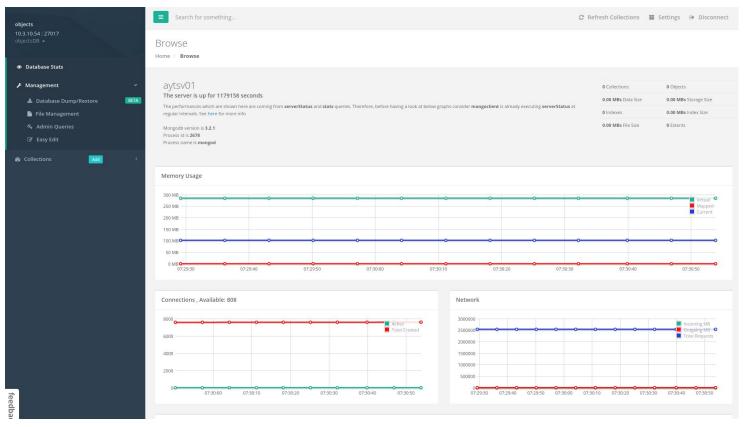


#### adminMongo

## MongoClient

**Advance Web Client** 

- Advance DB management
- Import/Export
- Backup/Restore
- MongoDB Shell (in browser)
- Open Source



#### MongoClient

## Installation & Management

### **Install MongoDB Linux**

From Ubuntu / Debian Repository

\$ sudo apt-get install mongodb

### **Install MongoDB Linux**

From official MongoDB repository

- 1. Add MongoDB repository
- 2. Update
- 3. And install
- \$ sudo apt-get mongodb-org

#### **Install MongoDB on Windows**

Download and install Windows binary.

#### MongoDB & Docker

```
$ docker run --name my-mongo -d mongo
1df0635e48e46102dadd80e732e7d14d77012
```

```
$ docker exec -ti my-mongo mongo
MongoDB shell version v3.4.2
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.4.2
```

>

# Backup & Restore

# **Backup & restore**

Use **mongodump** for backups
Use **mongorestore** for restores

**Don't** use mongoexport and mongoimport for backups!

## **Backup & Restore**

```
Backup
mongodump [<options>]
Restore
mongorestore <backup> [<options>]
```

## **Backup examples**

```
mongodump --out /backup/01
mongodump --archive /backup/01.dump
```

## Restore example

```
mongorestore /backup/01
mongorestore /backup/01.dump
```

## **Import & Export**

**JSON & CSV** 

Need to choose DB and Collection

Select some fields

Can apply query, limits, sorts, ...

## **Import & Export**

```
Backup
mongoexport [<options>]
Restore
mongoimport <file> [<options>]
```

## **Export examples**

```
$ mongoexport --quiet --collection
currencies
{"_id":{"$oid":"58d95db350d5da4475a96853"},"
name":"US Dollar","code":"USD"}
{"_id":{"$oid":"58d95dc150d5da4475a96854"},"
name":"Swiss Franc", "code":"CHF"}
$ mongoexport --collection currencies \
   --out out.json
```

## Import examples

```
$ mongoimport --collection currencies
out.json
```

# User Management

# **Default Settings**

By default **no access policy** are enabled.

You need enable those explicitly by parameter or in config.

#### **Enable Client Access Control**

By parameter:

\$ mongod --auth

Or in config (/etc/mongo.conf)

auth = true

## Roles

#### Basic roles

- read
- readWrite
- root

You can also create own role

#### **Create Admin**

```
use admin
db.createUser({
    user: "root",
    pwd: "root_passwd",
    roles: ["root"]
})
```

#### Connect to auth DB

```
$ mongo
> use admin
> db.auth("root","root_passwd")
>
$ mongo --authenticationDatabase admin
> db.auth("root","root_passwd")
>
```

#### **Create User**

```
db.createUser({
    user: "read", pwd: "read_passwd",
    roles: [{role: "read", "db": "test"}]
})
db.createUser({
    user: "write", pwd: "write_passwd",
    roles: [{role: "readWrite", "db":
"test"}]
```

#### **Get User**

```
> db.getUser("write")
   "_id" : "admin.write",
   "user" : "write",
   "db" : "admin",
   "roles" : [
       {"role":"readWrite", "db":"test"}
```

#### **Grant Role**

#### Revoke Role

## Change password

```
use admin
db.changeUserPassword("some_user", "new")
```

### Resources

- https://docs.mongodb.com
- https://go.bysika.cz/mongo
  - Slides
  - Links to related content, eg.: tools, docs, books, ...

## Thank you & Questions

Ondrej Sika

ondrej@ondrejsika.com @ondrejsika

https://go.bysika.cz/mongo