CSC 369: Distributed Computing

April 2020

Alex Dekhtyar

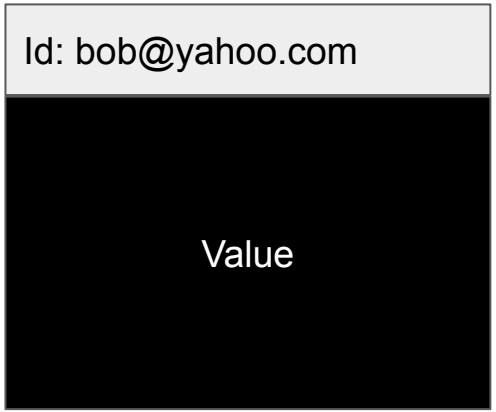
Housekeeping

- We are at 29 people in class
- 5 people on waitlist
- Everyone gets a permission code
  - Shoot me a private message on Slack during office hour/lab for permission code.
- Lab 1 grace period
  - Not certain if I can fully review the outputs today
  - **pandas** on python3.6
- Lab 2 is out
- Lab period - “typealong” demos

MongoDB: Distributed Document Store

Key-Value Stores to Document Stores.

Key Value Stores



Value is obscured

Only **get(key)**

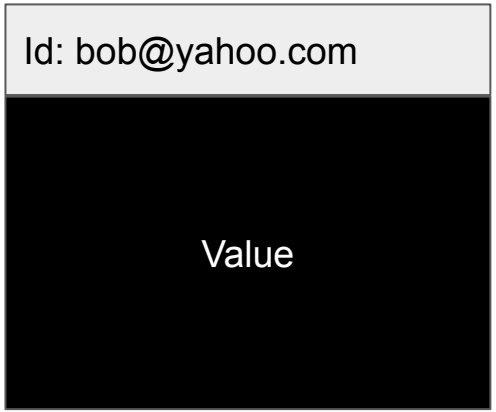
Lightning fast

Easier to Distribute

Gets REALLY annoying REALLY fast

Key-Value Stores to Document Stores.

Key Value Stores



Value is obscured

Only **get(key)**

Lightning fast

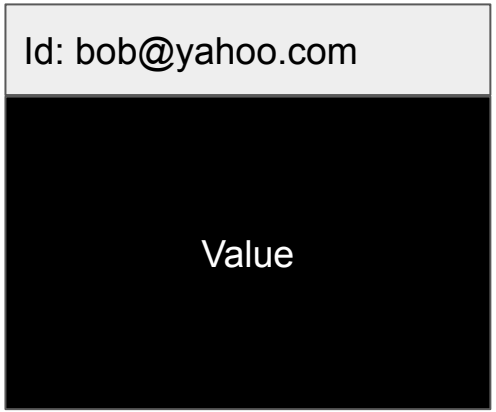
Easier to Distribute

Data stored is NON-relational

Gets REALLY annoying REALLY fast

Key-Value Stores to Document Stores.

Key Value Stores



Value is obscured

Only **get(key)**

Lightning fast

Easier to Distribute

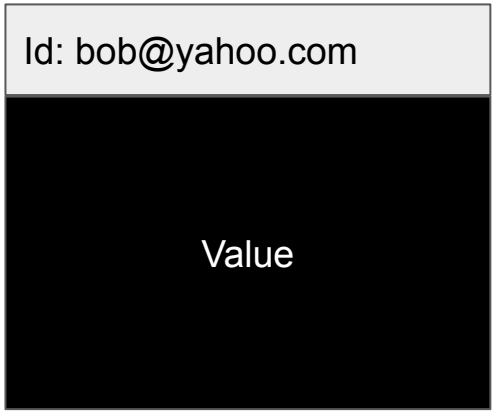
No going back to RDBMS

Data stored is NON-relational

Gets REALLY annoying REALLY fast

Key-Value Stores to Document Stores.

Key Value Stores

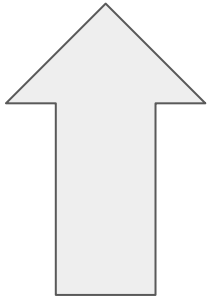


Value is obscured

Only **get(key)**

Lightning fast

Easier to Distribute



No going back to RDBMS

Data stored is NON-relational

Gets REALLY annoying REALLY fast

Key-Value Stores to Document Stores.

Key Value Stores

Id: bob@yahoo.com
Name: {first: Bob, last: Smith} DOB: {day: 22, month: 10, year: 1985} Interests: [skiing, knitting, tigers] Status: single

No going back to RDBMS

Data stored is NON-relational

"Find all single people in Seattle in age group 30-40 whose interests include rock climbing and cats"

"Find how many people residing in the US made posts every day for the last 30 days, and posted at least 20 pictures of their dog"

Enter Distributed Document Stores

So, MongoDB

- Native JSON support
- Databases
- Collections
- No Schema requirements
- Homebrew Query Language
  - Evolving over time
  - … you will spot it

Things we want to do with MongoDB

- Use Interactive client (today!)
- Write Javascript scripts (not too hard for simple things)
- Write Python code (later in the week)

Things we want to do with MongoDB

Authentication

```
> use csc369users
> db.auth(<userName>, passwordPrompt())
```

Things we want to do with MongoDB

HALP!

```
> help
> db.help()
> db.mycoll.help()
> exit
```

Things we want to do with MongoDB

Navigation

```
> show dbs
> use <database>
> show collections
```

Things we want to do with MongoDB

Data Insertion

```
> db.<collection>.insert({<JSON>})
> db.<collection>.insert([{JSON},...])
> db.<collection>.find()
```

Things we want to do with MongoDB

Data Modification

```
> db.<collection>.remove({})
> db.<collection>.remove({<filter>})
> db.<collection>.update({<filter>},{JSON})
```

Things we want to do with MongoDB

Query Document Collections

```
> db.<collection>.find()
> db.<collection>.find({<QueryDoc})
> db.<collection>.find({<QueryDoc>},{ProjectionDoc})
```

Things we want to do with MongoDB

Query Document Collections
Finishing touches

```
> db.<collection>.find(...).count()
> db.<collection>.find(...).limit(N)
> db.<collection>.find(...).skip(N)
> db.<collection>.find(...).sort({sortDoc})
> db.<collection>.find(...).pretty()
```

Things we want to do with MongoDB

Query Documents

```
> db.<collection>.find({“firstname”:”Bob”,
                        “age”: {$lt: 40, $gt:24},
                        “$or”:[{“location”:”Seattle”},
                               {“birthplace”:”Seattle”}],
                        “Hobby”: {$in: [“skiing”, “fishing”]},
                        “Education”: {$not: null}
                       })
```

Things we want to do with MongoDB

Query Arrays

```
> db.<collection>.find({hobbies: “skiing”,
                        languages: [“Java”, “Python”],
                        hobbies: {$size: 4},
                        “Hobbies.1”: “reading”
                       })
```

Things we want to do with MongoDB

Query Embedded Documents

```
> db.<collection>.find({ name:{first:”Bob”,
                               Last:“Smith”},
                         location.city:”Seattle”
                        })
```

Things we want to do with MongoDB

Simple Projections

```
> db.<collection>.find({},{name:1, location:1})

> db.<collection>.find({},{hobbies:0, location:0})

> db.<collection>.find({},{_id:0, name:1, location:1})
```

Limitations

db.<collection>.find():

- Only simple filtering on single collections
- Limited filters:
  - No arithmetics
  - No attribute comparisons
- Simple projection
  - No attribute modifications

db.<collection>.update():

- Updates full documents, rather than individual values
- Not as flexible as `UPDATE <Table> SET x = <Expr> WHERE< Condition>`

So, MongoDB

- Native JSON support
- Databases
- Collections
- No Schema requirements
- Homebrew Query Language
  - Evolving over time
  - … you will spot it

Aggregation Pipelines