MONGODB

STU FIIT

Karol Rástočný

Why NoSQL?

- Semantics over the web based on system tags
  - metadata attached to an part of an web page
- Tags on the Web
  - ~50 billion pages – horizontal salability (distribution)
  - Only basic structure – different content
- A numbers of web information systems
  - Concurrency access
  - Replication

Why MongoDB?

- Unknown data structure
  - Horizontal and vertical differences
- Access by tag ID and ID of tagged document
  - Indexes for multiple attributes
- Read operation to whole objects
  - Each object == One entry in DB
  - Objects are stored in binary format
- Parallelized analysis
  - MapReduce straight over data

Prominent Users

- AppScale, **bit.ly**, Business Insider, **CERN LHC**, craigslist, diaspora, Disney Interactive Media Group, **EA**, foursquare, **GitHub**, MTV Networks, **SAP**, Shutterfly, **SourceForge**, The Guardian, **The New York Times**, Etsy, Thumbtack, Uber, Wordnik, Springer, Chicago Tribune, Viacom, …

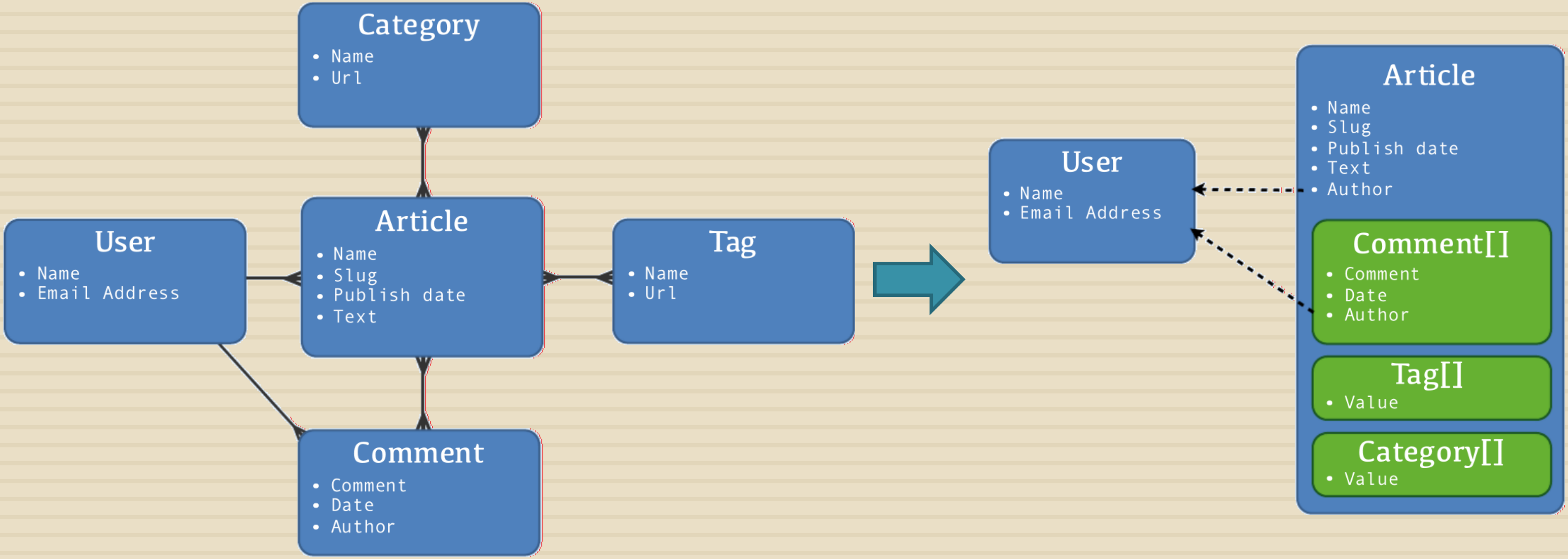
Shutterfly: Photo Metadata

- Problem
  - More flexibility is needed than Oracle delivers
- Results
  - 500% cost reduction
  - 900% performance improvement

Wordnik: Online Dictionary

- Problem
  - MySQL could not scale to handle 5B+ documents
- Results
  - 20x performance improvement

Main Idea – Objects in Objects



Internal Structure

- Databases
- Collections
  - Equivalent to relational database tables
- Documents
  - BSON objects
  - Attributes’ data types: object, object id, string, integer, boolean, double, null, array, date, timestamp, binary data, regular expression, code

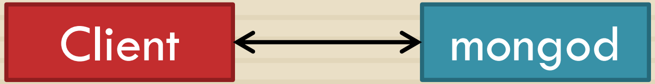
BSON (Binary JSON)

- JavaScript Object Notation

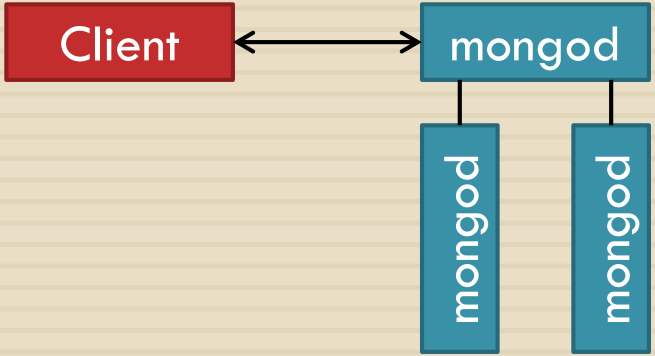
```
{
        "_id" : CSUUID("56a1f2c8-c1fd"),
        "UserID" : 777,
        "MovieID" : 1901,
        "Value" : 4,
        "Timestamp" : NumberLong(975604013)
}
```

Architecture

- □ Single node/database

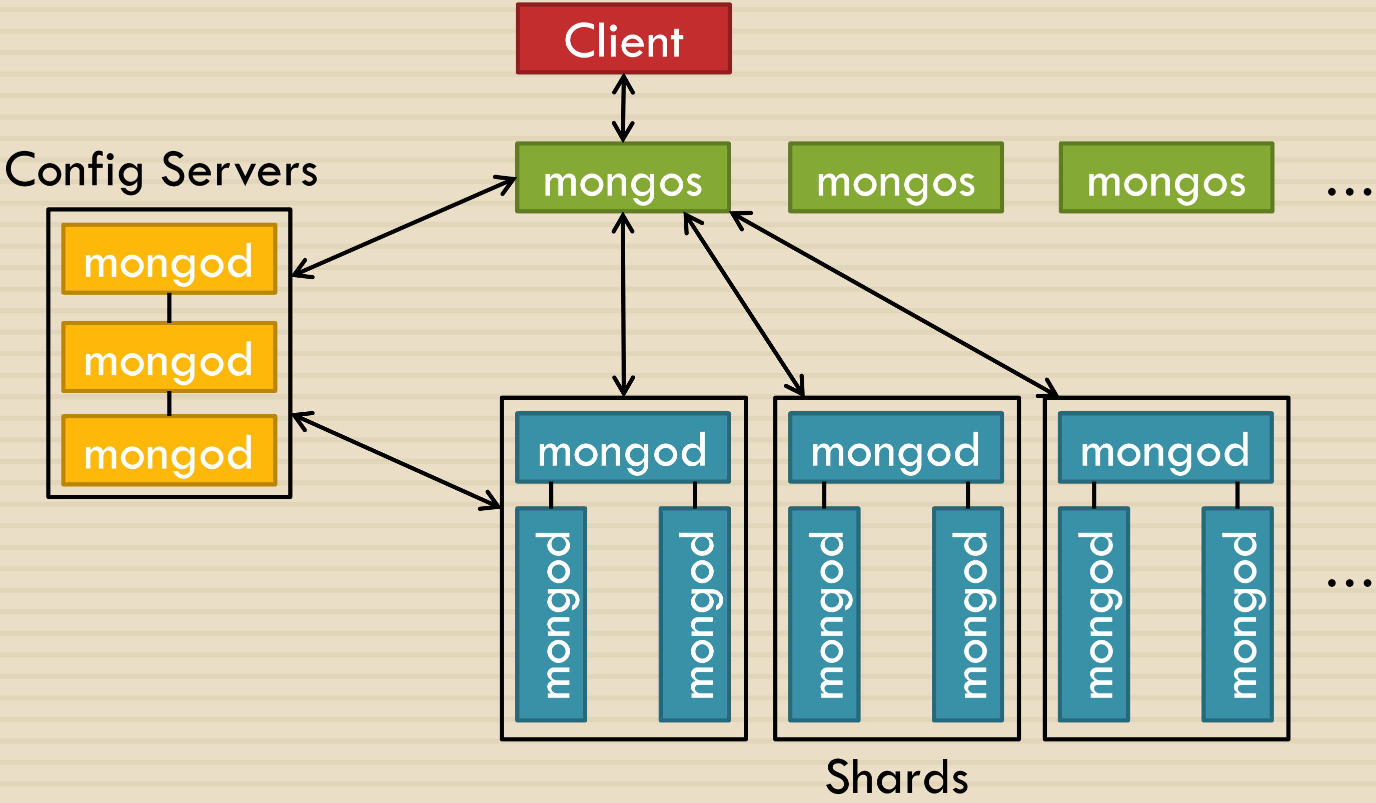


- □ Single database replicated over several nodes



- □ Shared (partitioned) database

Architecture - Partitioned database

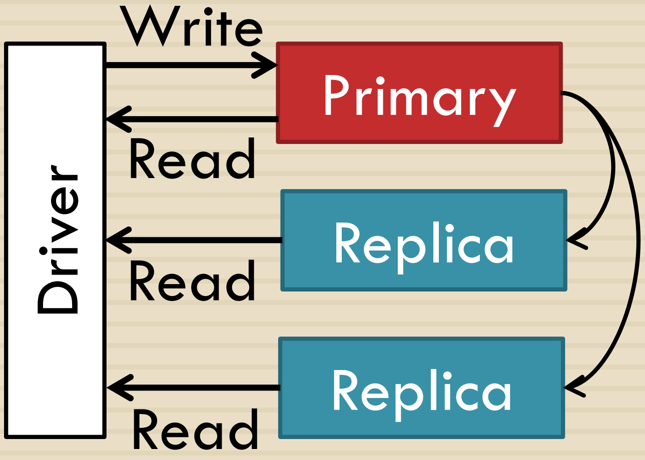


Replication

- Master slave
  - 1 mongod is initialized as master – allows writing
  - ≥ 1 mongods are initialized as slaves – replicated directly from the master
- Replica set
  - Type of Master slave
  - Master (primary node) is elected

Consistency

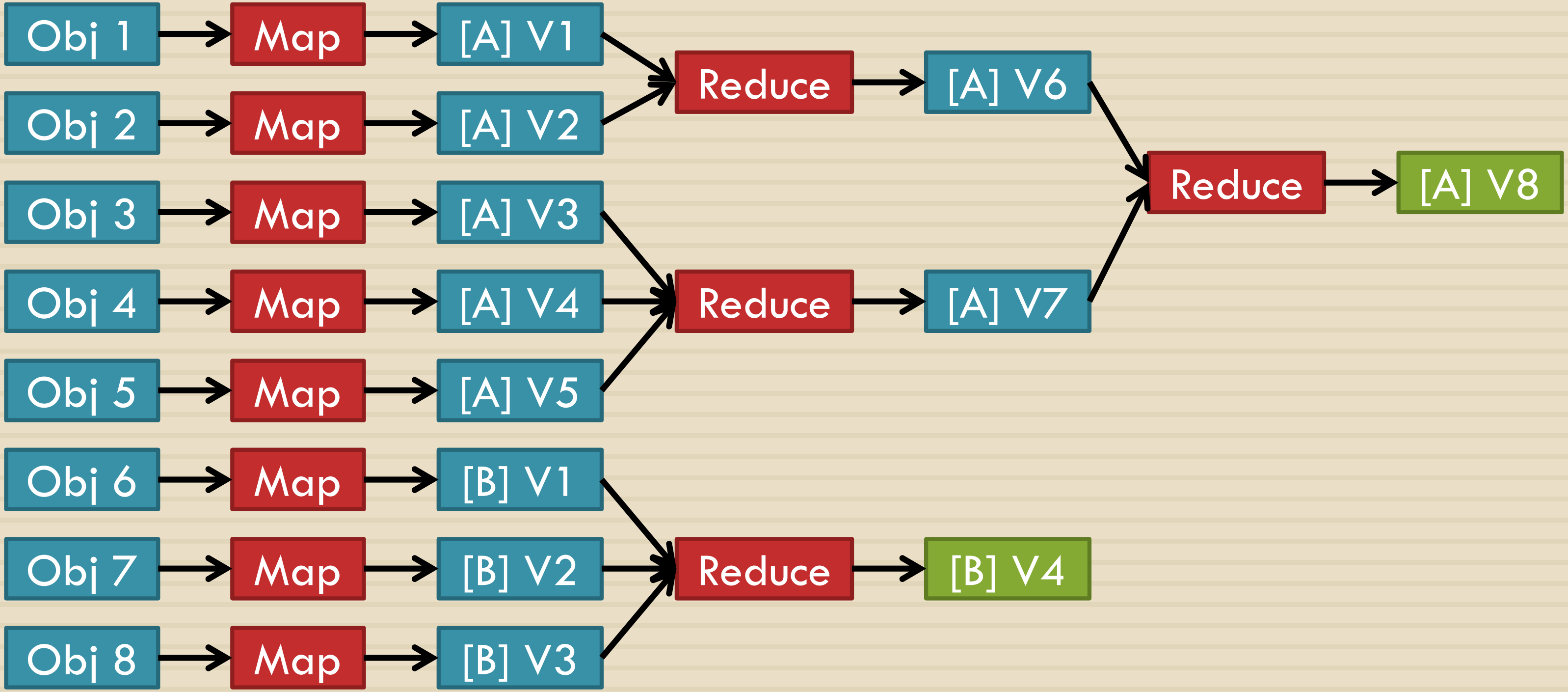
- Managed by drivers
- Strong consistency
  - Reading from the primary node
  - E.g., updating of source code
- Eventual consistency
  - Reading from any live node
  - E.g., wiki page



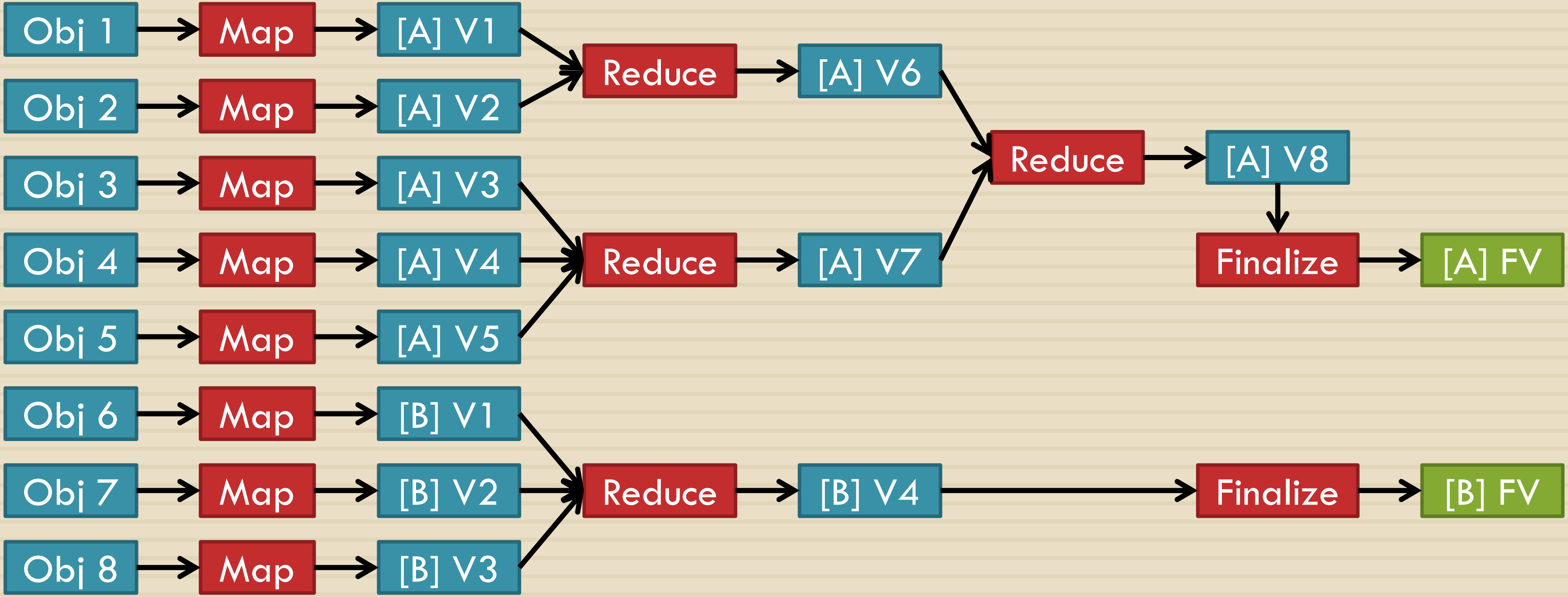
Querying

- Indexing
- Querying objects' attributes
  - Logic expressions, mathematical operations, regexes, …
  - Sort, skip, limit
  - Result: The cursor to the array of queried objects
  - db.foo.find({ name : "bob" , $or : [{ a : 1 } , { b : 2 }] })
- MapReduce + Finalize
  - Result: The view

MapReduce


Obj 1
Map
[A] V1
Obj 2
Map
[A] V2
Reduce
[A] V6
Obj 3
Map
[A] V3
Obj 4
Map
[A] V4
Obj 5
Map
[A] V5
Reduce
[A] V7
Reduce
[A] V8
Obj 6
Map
[B] V1
Obj 7
Map
[B] V2
Obj 8
Map
[B] V3
Reduce
[B] V4

MapReduceFinalize



Updating

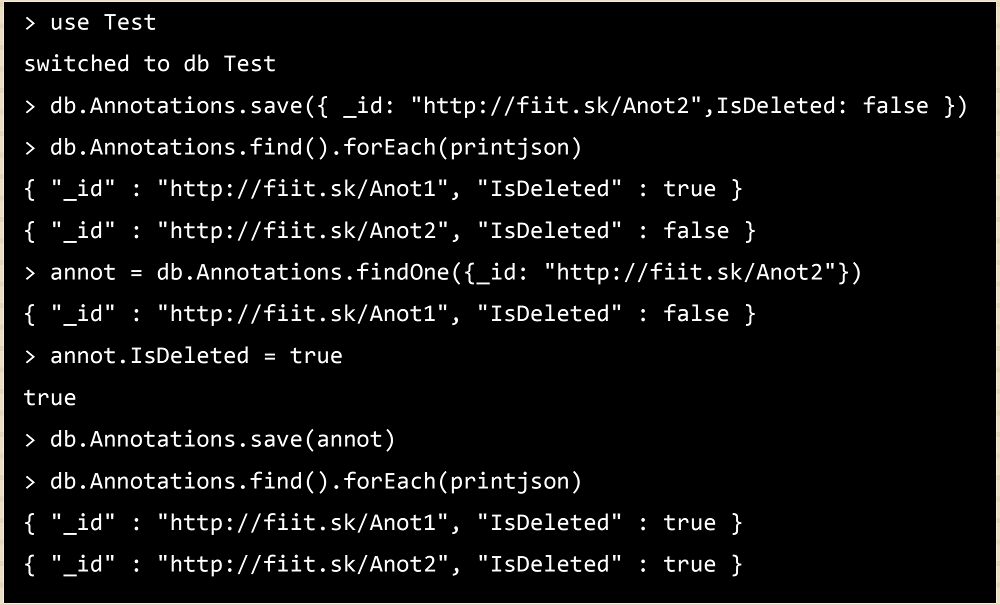
- Several confirmation techniques
- No transactions
- Atomic operations
  - db.shapes.update({ type: "square" }, { $inc: { x: 5 } })
- Blocking – does not support sharding
  - db.students.update({score: {$gt: 60}, $atomic: true}, {$set: {pass: true}}, false, true)
- Find and Modify
  - Returns old/new value

Connectors

- Drivers
  - mongodb.org: C, **C++**, Erlang, Haskell, **Java**, Javascript, .NET (**C#**, PowerShell, etc), Perl, **PHP**, Python, **Ruby**, Scala
  - Community: ActionScript3, Clojure, ColdFusion, D, Delphi, Entity, Factor, Go, Groovy, Lisp, Lua, MatLab, **node.js**, Objective C, Opa, Prolog, R, REST, Racket, Smalltalk
- REST add-on
- JavaScript console

Demonstration

- Dataset: 1M MovieLens
  - ID, UserID, MovieID, Value, TimeStamp
- JavaScript Console
- C# driver



```
> use Test
switched to db Test
> db.Annotations.save({ _id: "http://fiit.sk/Anot2",IsDeleted: false })
> db.Annotations.find().forEach(printjson)
{ "_id" : "http://fiit.sk/Anot1", "IsDeleted" : true }
{ "_id" : "http://fiit.sk/Anot2", "IsDeleted" : false }
> annot = db.Annotations.findOne({_id: "http://fiit.sk/Anot2"})
{ "_id" : "http://fiit.sk/Anot1", "IsDeleted" : false }
> annot.IsDeleted = true
true
> db.Annotations.save(annot)
> db.Annotations.find().forEach(printjson)
{ "_id" : "http://fiit.sk/Anot1", "IsDeleted" : true }
{ "_id" : "http://fiit.sk/Anot2", "IsDeleted" : true }
```

Resources

- Website:
  - http://www.mongodb.org/
- SQL to MongoDB
  - http://www.mongodb.org/display/DOCS/SQL+to+Mongo+Mapping+Chart
- MongoDB&hadoop
  - http://www.slideshare.net/spf13/mongodb-and-hadoop
- Practical Replication Video
  - http://www.10gen.com/presentations/mongosv-2011/a-mongodb-replication-primer-replica-sets-in-practice