

Resources

- <http://nosql-database.org/>
- <http://nosql.mypopescu.com/kb/nosql>
- <http://www.linuxforu.com/developers/up-close-and-personal-with-nosql/>
- <http://opensourcebridge.org/events/2011/sessions>
- <http://jchrisa.net/drl/ApacheCon-Talk-Slides/CouchDB-Intro-EU.pdf>
- <http://guide.couchdb.org/>
- <https://github.com/coderoshi/holy-grail-dbs>

Why?

- Big Data
- Schema-less
- “Programmer friendly”
- Availability
- Scalable
- Low Latency

Blogs
Facebook
Twitter
Web-indexes
.....
do you see a pattern?

Data

- Document Store
- Spatial/location data
- Graphs
- Column store
- Key/Value
- Triple Store
 - subject-predicate-object

CAP

- Consistency, Availability and Partition Tolerance
 - Eric Brewer, 2000
 - distributed databases can choose two.
 - eventual consistency

Software

Document Stores

MongoDB
CouchDB
RavenDB

Graph Databases

Neo4j
HyperGraphDB

Column Oriented

Hadoop Hbase
Apache
Cassandra
HyperTable

Key-Value Stores

Project
Voldemort
Tokyo
Cabinet
Redis
GT.M

Triple Stores

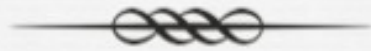
Jena
Sesame
Virtuoso
AllegroGraph

Consistency



- ⌘ A request to any connectable node in the system returns the same data
- ⌘ Strong Consistency
 - ⌘ aka: Strict, Linearizable or Atomic
 - ⌘ When an update completes, subsequent access returns the new result
- ⌘ Weak Consistency
 - ⌘ For most NoSQL purposes, we mean Eventual
 - ⌘ When an update completes, subsequent access will eventually return the new result

Availability



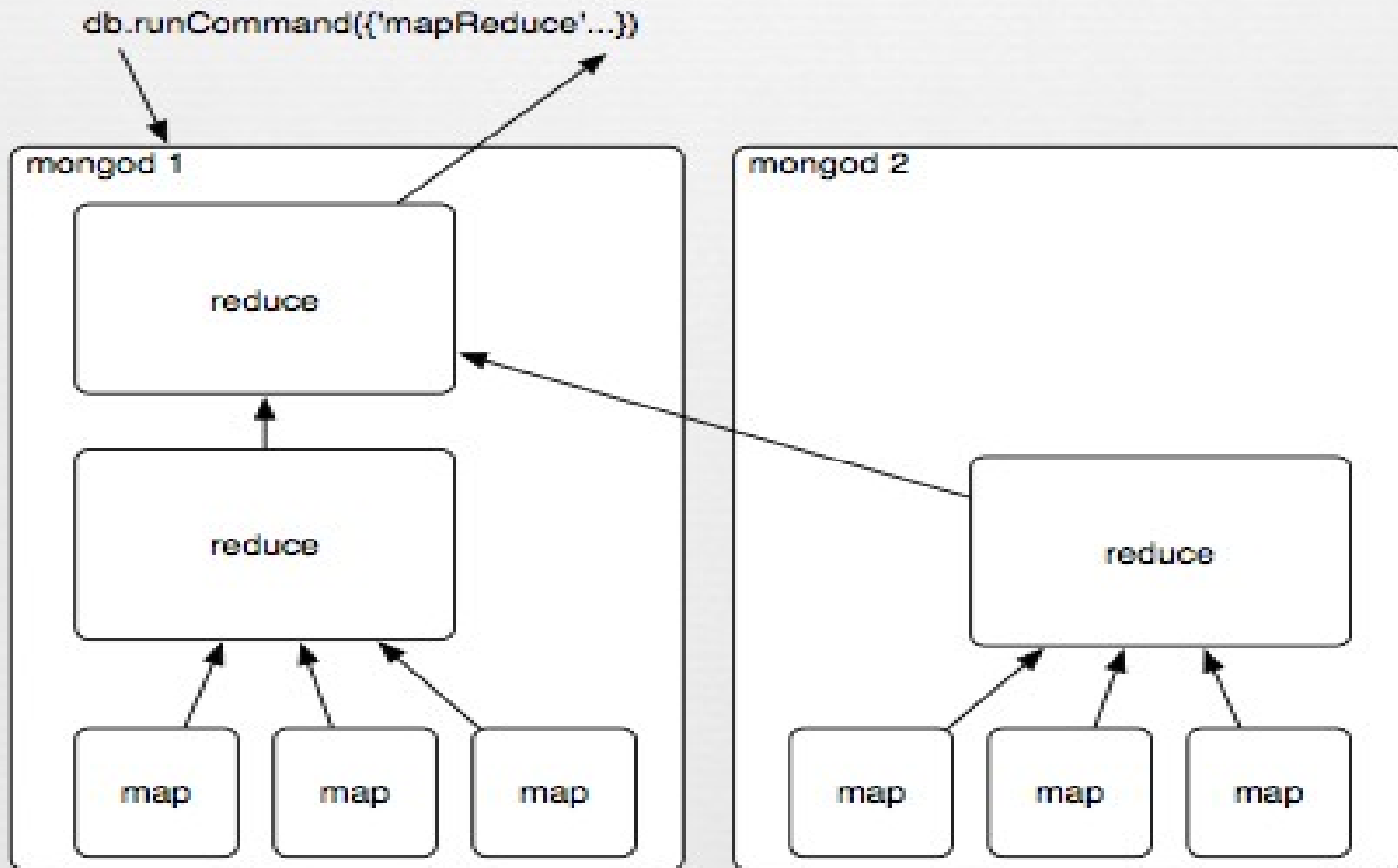
- ❧ Colloquial definition
 - ❧ The data is available when I want it.
 - ❧ Wrong! (latency) It could take forever
- ❧ “Technical-er” definition
 - ❧ Nodes which may sustain pack-loss continue serving requests.
 - ❧ Or: Is it possible to be *unavailable*?

Partition Tolerance



- ⌘ Despite message loss, the DB continues to operate.
- ⌘ A DB is either P or not.
- ⌘ “...the choice is almost always between sequential consistency and high availability”
- ⌘ <http://www.cloudera.com/blog/2010/04/cap-confusion-problems-with-partition-tolerance>

Mapreduce



Document



```
{
  "_id" : ObjectId("4db7ca268e236e5bf9a52224"),
  "_rev" : "2612672603",
  "name" : "Sant Julià de Lòria",
  "country" : "AD",
  "timezone" : "Europe/Andorra",
  "population" : 8022,
  "location" : {
    "latitude" : 42.46372,
    "longitude" : 1.49129
  }
}
```

Sites



- ⌘ <http://nosql-database.org/>
 - ⌘ A great list
- ⌘ <http://sevenweeks.org/>
 - ⌘ The book website (it's a wiki!)
- ⌘ <https://github.com/coderoshi/holy-grail-dbs>
 - ⌘ The project
 - ⌘ The slides

Field	Value
_id	"_design/date_title"
_rev	"36-d3c4e7734d5fdafc8e2c6e1b91f80f33"
✕ language	"javascript"
✕ lists	showall "function(head,req) { var row; while(row = getRow()) { send(row.value);} }" showsome "function(head,req) { var row; var res; var query = req.query.value; send('Content-Type:text/html');while(row = getRow()) { res = row.value...." showsomeF "function(head,req) { var row; var res; var query = req.query.value; var body = ''; start({'headers': {'Content-Type': 'text/html'}});whi..."
✕ shows	title "function(doc, req){ return '<h1>' + doc.title + '</h1>';}"
✕ views	<div> <div> date_title <pre>map "function(doc) { emit(doc.date, doc.title); }"</pre> </div> <div> sum <pre>map "function(doc) { emit(doc.date,1); }" reduce "function(keys, values, rereduce) { return sum(values) }"</pre> </div> <div> findIt <pre>map "function(doc) { if(doc.title) { emit(doc.title, doc.title); } }"</pre> </div> <div> titleContainsB <pre>map "function(doc) { if(doc.title) { var v = doc.title.match(/B/); if(v) { emit(doc.title, null);} } }"</pre> </div> </div>