dealing w/ lots of data

- 20+ billion web pages x 20KB = 400+ terabytes (to just store it -- not to do sth.)
- One computer can read ~50MB/sec from disk
- so 3 months to read from web
- Solution?
Solution

- get 1000’s of computers in cluster
- take 2.5 hrs to process the web
cluster of 1000 computers

4,500 Terabytes (4.5 Petabytes)
8 racks of 64 quad core servers
The only thing needed are network connections, a chilled water supply, and electricity.
Nice Sun Server Box
You: a graduate of UMW
You: a graduate of UMW

I hire you to write a program to handle lots and lots of data.
You: a graduate of UMW

I hire you to write a program to handle lots and lots of data.

3 months run time unacceptable.
☐ You: a graduate of UMW

☐ I hire you to write a program to handle lots and lots of data.

☐ 3 months run time unacceptable.

☐ I buy you 2 nice Sun 20ft server containers
You: a graduate of UMW

I hire you to write a program to handle lots and lots of data.

3 months run time unacceptable.

I buy you 2 nice Sun 20ft server containers

How do you program it?
Solutions

- Map Reduce
  (easy way to write distributed programs)
- Sharding
- node.js
mongo

- document oriented database
- replaces concept of ‘row’ with ‘document’
{
    "_id" : ObjectId("4bddd5bd7a20d38d26e34e7"),
    "rating" : "R",
    "release_year" : 2010,
    "rotten_id" : ObjectId("4d24029da9797eb6a7cfcdab"),
    "title" : "Black Swan",

    "actors" : [
        {
            "name" : "Barbara Hershey",
            "person_id" : ObjectId("4c453f78b6b4c16e051cf25")
        },
        {
            "name" : "Benjamin Millepied",
            "person_id" : ObjectId("4c45b16fb6b4c16e16c6f45")
        }
    ],
    "genres" : [
        "Suspense",
        "Thriller",
        "Psychological Thrillers"
    ]
}
mongoDB

Quickstart
Installation

- On Mac using Homebrew package manager:
  - brew install mongo

- On Linux, see mongodb.org.
  - should install the mongodb-10gen package

- On Windows, download from mongodb.org/downloads
nitrous.io

- parts install mongodb
- parts start mongodb
mongod - the server

Terminal — mongod — 69×22

Last login: Sat Apr 9 22:18:48 on ttys000
Ron-Zacharskis-iMac:~ raz$ mongod
mongod --help for help and startup options
Mon Apr 11 02:42:36 MongoDB starting : pid=48116 port=27017 dbpath=/data/db/ 64-bit
Mon Apr 11 02:42:36 db version v1.6.5, pdfile version 4.5
Mon Apr 11 02:42:36 git version: 0eb017e9b2828155a67c5612183337b89e12e291
Mon Apr 11 02:42:36 sys info: Darwin erh2.10gen.cc 9.6.0 Darwin Kernel Version 9.6.0: Mon Nov 24 17:37:00 PST 2008; root:xnu-1228.9.59~1/RELEASE_I386 i386 BOOST_LIB_VERSION=1_40
Mon Apr 11 02:42:36 [initandlisten] waiting for connections on port 27017
Mon Apr 11 02:42:36 [websvr] web admin interface listening on port 28017
mongod - the server

```
Last login: Sat Apr  9 22:18:48 on ttys000
Ron-Zacharskis-iMac:~ raz$ mongod
mongod --help for help and startup options
Mon Apr 11 02:42:36 MongoDB starting : pid=48116 port=27017 dbpath=/data/db/ 64-bit
Mon Apr 11 02:42:36 db version v1.6.5, pdfile version 4.5
Mon Apr 11 02:42:36 git version: 0eb017e9b2828155a67c5612183337b89e12e291
 MariuszE T386_1386 BOOST UTL VERSION=1 40
Mon Apr 11 02:42:36 [initandlisten] waiting for connections on port 27017
Mon Apr 11 02:42:36 [websvr] web admin interface listening on port 28017
```
mongod Ron-Zacharskis-iMac.local

List all commands | Replica set status

Commands: assertInfo buildInfo cursorInfo features isMaster replSetGetStatus serverStatus top

db version v1.6.5, pdfile version 4.5
git hash: 0eb017e9b2828155a67c5612183337b89e12e291
sys info: Darwin er2.10gen.cc 9.6.0 Darwin Kernel Version 9.6.0: Mon Nov 24 17:37:00 PST 2008; root:xnu-1228.9.59-1/RELEASE_I386
uptime: 162 seconds

low level requires read lock

time to get read lock: 0ms
# databases: 1

replication:
master: 0
slave: 0
initialSyncCompleted: 1

clients

<table>
<thead>
<tr>
<th>Client</th>
<th>Opld</th>
<th>Active</th>
<th>LockType</th>
<th>Waiting</th>
<th>SecsRunning</th>
<th>Op</th>
<th>Namespace</th>
<th>Query</th>
<th>client</th>
<th>msg</th>
<th>progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>initandlisten</td>
<td>0</td>
<td>W</td>
<td></td>
<td></td>
<td></td>
<td>2004</td>
<td>whichflix</td>
<td>{ name: '/^local.temp/' }</td>
<td>0.0.0.0:0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>clientcursormon</td>
<td>0</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td>(NONE)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>websvr</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td>(NONE)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>snapshotthread</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td>(NONE)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

dbtop (occurencespct of elapsed)

<table>
<thead>
<tr>
<th>NS</th>
<th>total</th>
<th>Reads</th>
<th>Writes</th>
<th>Queries</th>
<th>GetMores</th>
<th>Inserts</th>
<th>Updates</th>
<th>Removes</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLOBAL</td>
<td>0</td>
<td>0.0%</td>
<td>0</td>
<td>0.0%</td>
<td>0</td>
<td>0.0%</td>
<td>0</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

write lock % time in write lock, by 4 sec periods
000000000000000000000000000000000000000000000000000
mongo
command line interpreter

Last login: Mon Apr 11 02:48:35 on ttys001
Ron-Zacharskis-iMac:~ raz$ mongo
MongoDB shell version: 1.6.5
connecting to: test
>
> a = 5
5
> b = 10
10
> a + b
15
> contact = {"name": "Cheryl Clark", "phone": "575-541-1360"}
> contact
{ "name" : "Cheryl Clark", "phone" : "575-541-1360" }
> for (i = 0; i < 10; i++) {print('hello');};
hello
hello
hello
commands similar to MySQL
show databases
-> show dbs
use xxx

> use whichflix
switched to db whichflix

>
show tables ->
show collections
Queries

name of collection (i.e., table)

```
db.movies.findOne({})
```

What to match -- like where ...
> db.movies.findOne({})
{
    "_id" : ObjectId("4b9081f449ae805e00627343"),
    "actors" : [
        {
            "name" : "Helena Bonham Carter",
            "person_id" : ObjectId("4c453f57b6b4c16efb0bf250")
        },
        {
            "name" : "Mickey Rourke",
            "person_id" : ObjectId("4c453f57b6b4c16efc0bf250")
        }
    ],
    "amazon" : {
        "updated" : "Thu Aug 26 2010 11:47:18 GMT-0400 (EDT)"
    },
    "directors" : [
        {
            "name" : "Liliana Cavani",
            "person_id" : ObjectId("4c453f57b6b4c16efd0bf250")
        }
    ],
    "genres" : [
db.movies.findOne({"title": "Black Swan")
{
   "_id" : ObjectId("4bddd5bd7a20d38d26e34e7"),
   "actors" : [
      {
         "name" : "Barbara Hershey",
         "person_id" : ObjectId("4c453f78b6b4c16e051cf250")
      },
      {
         "name" : "Benjamin Millepied",
         "person_id" : ObjectId("4c45b16fb6b4c16e16e6f450")
      },
      {
         "name" : "Janet Montgomery",
         "person_id" : ObjectId("4c45a31eb6b4c16e6c93f450")
      },
      {
         "name" : "Kristina Anapau",
         "person_id" : ObjectId("4c454272b6b4c16e878cf250")
      },
      {
         "name" : "Ksenia Solo",
      }
   ]
}
> db.movies.findOne({"actors.name": "Natalie Portman"})
{
   "_id" : ObjectId("4b9081f649ae805e30677343"),
   "actors" : [
      {
         "name" : "Ahmed Best",
         "person_id" : ObjectId("4c453f89b6b4c16eca24f250")
      },
      {
         "name" : "Christopher Lee",
         "person_id" : ObjectId("4c453f73b6b4c16e051bf250")
      },
      {
         "name" : "Ewan McGregor",
         "person_id" : ObjectId("4c453f89b6b4c16ecb24f250")
      },
      {
         "name" : "Frank Oz",
         "person_id" : ObjectId("4c453f55b6b4c16e150bf250")
      }
   ],
   "title" : "Star Wars: Episode II: Attack of the Clones",
   "tmdb" : {
      "updated" : "Thu Sep 09 2010 08:16:35 GMT-0400 (EDT)"
   }
}
Comparison Operators

$lt  $lte  $gt  $gte  $ne

- db.users.find({"age": {"$gte": 18, "$lte": 30}})
- db.users.find("name": {"$ne": "Ann"}})
or

db.movies.find({"actors.name": {$in: ["Angelina Jolie", "Natalie Portman"]}})
Regular Expressions

Joe, joe, Joey, joey

db.users.find({"name": /joey/i})
query demo
save & insert:
the insert equivalents

Collection

db.contacts.save({"name": "Cheryl Clark",
"cell": "575-541-1360"})

What to save
Databases and collections are created automagically.
Databases and collections are created automagically.
Databases and collections are created automagically
How to save and restore MongoDB databases

`mongodump / mongorestore`

for ex.,

`mongodump -d whichflix`

saves to a folder called dump
mongoDB & PHP

the equivalent of mysqli_connect
need to install MongoDB library
pip install pymongo
>>> from pymongo import MongoClient

>>> client = MongoClient()

>>> db = client.abductions

>>> db.implants.find_one()
{u'implant': u'metal cylinder', u'_id': ObjectId('53409e18fed08dc8b0001034'), u'name': u'Dana Scully'}

>>>
more later
replication
(historically master/slave)
writes are only to the primary
secondaries
master slave

- master performs reads and writes
  - mongod --master (default port 27017)
- slaves perform reads not writes
  - mongod --slave --dbpath /data/db/slave --port 10001  --source localhost:27017