

Class project for Introduction to
NoSQL Databases:
1980 US census population
characteristic research

Tomoaki Tanaka

7/9/2016

MongoDB sharding

Set up MongoDB sharded cluster on AWS EC2

- Create three EC 2 instances
- Set up Config server, Query Router, 3 shards on EC2 instances

Use AWS Dataset: 1980 US Census

<https://aws.amazon.com/datasets/1980-us-census/?tag=datasets%23keywords%23economics>

Sharding status:

- No. of documents: 793,340
- No. of chunks: 58

Imported dataset

```
mongos> sh.status(verbose)
2016-07-06T01:51:16.017+0000 E QUERY   ReferenceError: verbose is not defined
    at (shell):1:11
mongos> show dbs
admin      0.016GB
census     7.858GB
config     0.047GB
uscensus   0.156GB
mongos> |
```

Size of dataset

```
switched to db census
mongos> db.counts()
2016-07-06T01:34:16.477+0000 E QUERY   TypeError: Property 'counts' of object census
    at (shell):1:4
mongos> pop.count()
2016-07-06T01:34:24.537+0000 E QUERY   ReferenceError: pop is not defined
    at (shell):1:1
mongos> db.pop.count()
474807
mongos>
```

Number of documents in
Shard0000 imported by
mongoimport

sh.status() before sharding

```
ubuntu@ip-172-31-59-191: ~  
root      2199      1  0 17:49 ?          00:00:00 mongos -f /etc/mongos.conf  
ubuntu    2218    1427  0 17:49 pts/0    00:00:00 grep --color=auto mongo  
ubuntu@ip-172-31-59-191:~$ mongo 172.31.59.191:47017/admin  
MongoDB shell version: 3.0.12  
connecting to: 172.31.59.191:47017/admin  
Welcome to the MongoDB shell.  
For interactive help, type "help".  
For more comprehensive documentation, see  
  http://docs.mongodb.org/  
Questions? Try the support group  
  http://groups.google.com/group/mongodb-user  
Server has startup warnings:  
2016-07-04T17:49:29.465+0000 I CONTROL  ** WARNING: You are running this process  
as the root user, which is not recommended.  
2016-07-04T17:49:29.465+0000 I CONTROL  
mongos> sh.addShard("172.31.59.191:47018")  
{ "shardAdded" : "shard0000", "ok" : 1 }  
mongos> sh.addShard("172.31.54.5:47018")  
{ "shardAdded" : "shard0001", "ok" : 1 }  
mongos> sh.addShard("172.31.57.126:47018")  
{ "shardAdded" : "shard0002", "ok" : 1 }  
mongos> sh.status()  
--- Sharding Status ---  
  sharding version: {  
    "_id" : 1,  
    "minCompatibleVersion" : 5,  
    "currentVersion" : 6,  
    "clusterId" : ObjectId("577a1a90ca002ae037c1560")  
  }  
  shards:  
    { "_id" : "shard0000", "host" : "172.31.59.191:47018" }  
    { "_id" : "shard0001", "host" : "172.31.54.5:47018" }  
    { "_id" : "shard0002", "host" : "172.31.57.126:47018" }  
  balancer:  
    Currently enabled:  yes  
    Currently running:  no  
    Failed balancer rounds in last 5 attempts:  0  
    Migration Results for the last 24 hours:  
      No recent migrations  
  databases:  
    { "_id" : "admin", "partitioned" : false, "primary" : "config" }  
mongos>
```

sh.status() after sharding

```
ubuntu@ip-172-31-59-191: ~  
mongos> sh.status()  
--- Sharding Status ---  
  sharding version: {  
    "_id" : 1,  
    "minCompatibleVersion" : 5,  
    "currentVersion" : 6,  
    "clusterId" : ObjectId("577aala90ca002ae037c1560")  
  }  
  shards:  
    { "_id" : "shard0000", "host" : "172.31.59.191:47018" }  
    { "_id" : "shard0001", "host" : "172.31.54.5:47018" }  
    { "_id" : "shard0002", "host" : "172.31.57.126:47018" }  
  balancer:  
    Currently enabled: yes  
    Currently running: no  
    Failed balancer rounds in last 5 attempts: 0  
    Migration Results for the last 24 hours:  
      77 : Success  
      90 : Failed with error 'could not acquire collection lock for census.pop to migrate chunk [{ : MinKey }, { : MaxKey }]  
:: caused by :: Lock for migrating chunk [{ : MinKey }, { : MaxKey } in census.pop is taken.', from shard0002 to shard0000  
      8 : Failed with error 'data transfer error', from shard0000 to shard0001  
      20 : Failed with error 'moveChunk failed to engage TO-shard in the data transfer: cannot start recv'ing chunk [{ _id:  
MinKey }, { _id: -8903882805028543093 }]' :: caused by :: could not query collection metadata :: caused by :: 11002 socket exception [CO  
NNECT_ERROR] server [172.31.59.191:47019] connection pool error: couldn't connect to server 172.31.59.191:47019 (172.31.59.191), conne  
ction attempt failed', from shard0000 to shard0001  
      17 : Failed with error 'ns not found, should be impossible', from shard0000 to shard0001  
      3564 : Failed with error 'moveChunk failed to engage TO-shard in the data transfer: cannot start recv'ing chunk [{ _id  
: MinKey }, { _id: ObjectId('577b42751fdb23f14b8a2ba6')}]' :: caused by :: could not query collection metadata :: caused by :: 11002 so  
cket exception [CONNECT_ERROR] server [172.31.59.191:47019] connection pool error: couldn't connect to server 172.31.59.191:47019 (172  
.31.59.191), connection attempt failed', from shard0000 to shard0001  
  databases:  
    { "_id" : "admin", "partitioned" : false, "primary" : "config" }  
    { "_id" : "uscensus", "partitioned" : false, "primary" : "shard0002" }  
    { "_id" : "census", "partitioned" : true, "primary" : "shard0000" }  
    census.pop  
      shard key: { "_id" : "hashed" }  
      chunks:  
        shard0000      19  
        shard0001      19  
        shard0002      20  
      too many chunks to print, use verbose if you want to force print  
mongos>
```



sh.status(true)

```
ubuntu@ip-172-31-59-191: ~  
Timestamp(32, 0) { "_id" : NumberLong("353807383836163426") } --> { "_id" : NumberLong("668159398905484360") } on : shard0001  
Timestamp(33, 0) { "_id" : NumberLong("668159398905484360") } --> { "_id" : NumberLong("991858510137594357") } on : shard0002  
Timestamp(34, 0) { "_id" : NumberLong("991858510137594357") } --> { "_id" : NumberLong("1313882720646045759") } on : shard0001  
Timestamp(35, 0) { "_id" : NumberLong("1313882720646045759") } --> { "_id" : NumberLong("1634936765829121262") } on : shard0000  
Timestamp(36, 0) { "_id" : NumberLong("1634936765829121262") } --> { "_id" : NumberLong("1955368854492283565") } on : shard0000  
Timestamp(37, 0) { "_id" : NumberLong("1955368854492283565") } --> { "_id" : NumberLong("2277020325733422214") } on : shard0000  
Timestamp(38, 0) { "_id" : NumberLong("2277020325733422214") } --> { "_id" : NumberLong("2593682239221003785") } on : shard0000  
Timestamp(39, 0) { "_id" : NumberLong("2593682239221003785") } --> { "_id" : NumberLong("2912817508123225439") } on : shard0000  
Timestamp(40, 0) { "_id" : NumberLong("2912817508123225439") } --> { "_id" : NumberLong("3233242649153956436") } on : shard0000  
Timestamp(41, 0) { "_id" : NumberLong("3233242649153956436") } --> { "_id" : NumberLong("3553109082293487956") } on : shard0000  
Timestamp(42, 0) { "_id" : NumberLong("3553109082293487956") } --> { "_id" : NumberLong("3877803580583627472") } on : shard0000  
Timestamp(43, 0) { "_id" : NumberLong("3877803580583627472") } --> { "_id" : NumberLong("4197291875515046228") } on : shard0000  
Timestamp(44, 0) { "_id" : NumberLong("4197291875515046228") } --> { "_id" : NumberLong("4522167404497702413") } on : shard0000  
Timestamp(45, 0) { "_id" : NumberLong("4522167404497702413") } --> { "_id" : NumberLong("4843507449372382979") } on : shard0000  
Timestamp(46, 0) { "_id" : NumberLong("4843507449372382979") } --> { "_id" : NumberLong("5166335042146801559") } on : shard0000  
Timestamp(47, 0) { "_id" : NumberLong("5166335042146801559") } --> { "_id" : NumberLong("5483540799285558182") } on : shard0000  
Timestamp(48, 0) { "_id" : NumberLong("5483540799285558182") } --> { "_id" : NumberLong("5803063510630832238") } on : shard0000  
Timestamp(49, 0) { "_id" : NumberLong("5803063510630832238") } --> { "_id" : NumberLong("6121078733754467974") } on : shard0000  
Timestamp(50, 0) { "_id" : NumberLong("6121078733754467974") } --> { "_id" : NumberLong("6441713431725766293") } on : shard0000  
Timestamp(51, 0) { "_id" : NumberLong("6441713431725766293") } --> { "_id" : NumberLong("6766577458639512353") } on : shard0000  
Timestamp(52, 0) { "_id" : NumberLong("6766577458639512353") } --> { "_id" : NumberLong("7089036087997607603") } on : shard0000  
Timestamp(53, 0) { "_id" : NumberLong("7089036087997607603") } --> { "_id" : NumberLong("7407677582076927112") } on : shard0000
```

Data output by using Pymongo

ubuntu@ip-172-31-59-191: ~

```
ubuntu@ip-172-31-59-191:~$ python uscensus.py
```

DISTRICT OF COLUMBIA 11

MINNESOTA 27

NORTH CAROLINA 37

NORTH DAKOTA 38

DELAWARE 10

IOWA 19

RHODE ISLAND 44

MAINE 23

CONNECTICUT 9

MASSACHUSETTS 25

MONTANA 30

OHIO 39

OKLAHOMA 40

WEST VIRGINIA 54

SOUTH DAKOTA 46

UTAH 49

ILLINOIS 17

MARYLAND 24

IDAHO 16

LOUISIANA 22

NEW HAMPSHIRE 33

INDIANA 18

NEBRASKA 31

MICHIGAN 26

NEVADA 32

NEW YORK 36

FLORIDA 12

MISSOURI 29

NEW MEXICO 35

ARKANSAS 5

COLORADO 8

TENNESSEE 47

OREGON 41

GEORGIA 13

KANSAS 20

HAWAII 15

KENTUCKY 21

NEW JERSEY 34

PENNSYLVANIA 42

ARIZONA 4

TEXAS 48

VERMONT 50

State FIPS State code

uscensus.py retrieved the population by each age range, each state and sex.

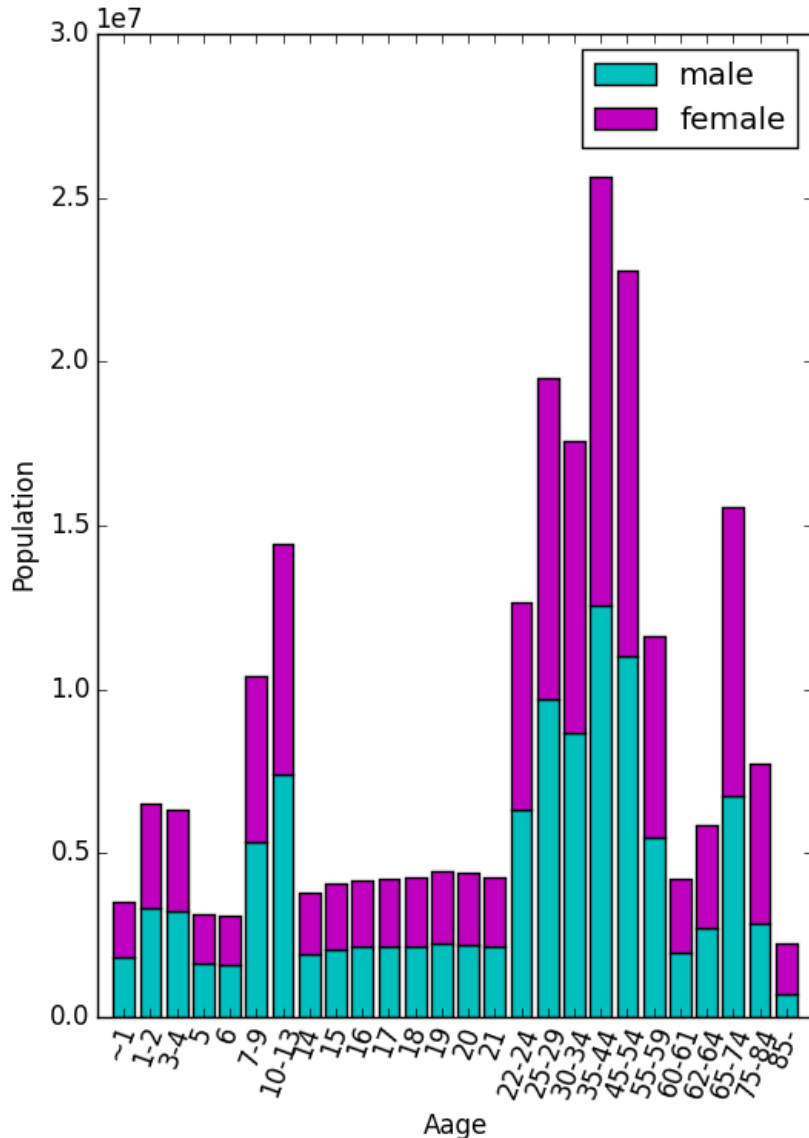


Search the web and Windows

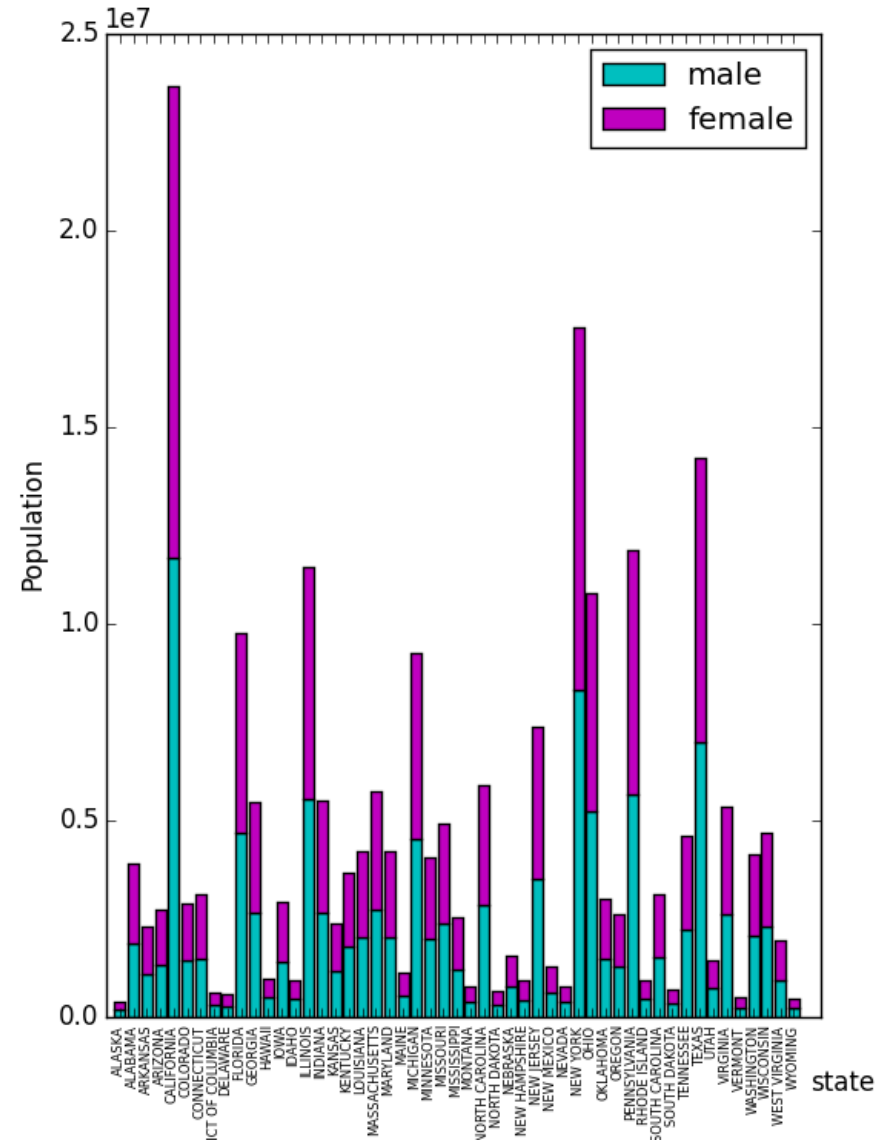


Result: 1980 US population

Total population by age



Total population by state



Result: 1980 US population

Population table for each state

State	population
IDAHO	943935
LOUISIANA	4205900
MINNESOTA	4075970
NEW HAMPSHIRE	920610
NORTH CAROLINA	5881766
UTAH	1461037
ILLINOIS	11426518
INDIANA	5490224
NEBRASKA	1569825
RHODE ISLAND	947154
MICHIGAN	9262078
NEVADA	800493
NEW YORK	17558072
DISTRICT OF COLUMBIA	638333
FLORIDA	9746324
MISSOURI	4916686
NEW MEXICO	1302894
ARKANSAS	2286435
COLORADO	2889964
MAINE	1124660
TENNESSEE	4591120
WEST VIRGINIA	1949644
OREGON	2633105
GEORGIA	5463105
KANSAS	2363679

State	population
DELAWARE	594338
HAWAII	964691
KENTUCKY	3660777
MASSACHUSETTS	5737037
NEW JERSEY	7364823
PENNSYLVANIA	11863895
ARIZONA	2718215
CONNECTICUT	3107576
TEXAS	14229191
NORTH DAKOTA	652717
MONTANA	786690
VERMONT	511456
CALIFORNIA	23667902
OHIO	10797630
MARYLAND	4216975
ALASKA	401851
VIRGINIA	5346818
WYOMING	469557
IOWA	2913808
WASHINGTON	4132156
ALABAMA	3893888
MISSISSIPPI	2520638
OKLAHOMA	3025290
SOUTH DAKOTA	690768
SOUTH CAROLINA	3121820
WISCONSIN	4705767