

Big Data Architecture Lab 2

ZHANG Xin

It is advised to check the HTML version instead in order to save the time of scrolling through long outputs.

Table of Contents

- Task 0: setup
 - 1. Create a directory and launch the server
 - 2. Import the document `moviepeople-10.jsonl` into the server.
 - 3. Launch a client (mongo shell), retrieve all the documents by asking a query in the client.
- Task 1: import and querying
 - 1. Import the documents `moviepeople-3000.jsonl` and `cities.jsonl` into the server (the one set up in Task 0).
 - 2. In the mongo shell client, write queries finding:
 - a) the person named Anabela Teixeira
 - b) the birthplace of Steven Spielberg
 - c) the number of people born in Lisbon
 - d) the people taller than 170 cm
 - e) the names of people whose information contains "Opera"
 - f) for each movie person whose birth place is known, the latitude, longitude and population of that city (if that information exists in the city document).
- Task 2: replication
 - 1. Create working directories for 3 MongoDB servers.
 - 2. Create a replica set for a collection called small-movie.
 - 3. Launch the three MongoDB servers (in different shells) and let them run.
 - Server 1
 - Server 2
 - Server 3
 - 4. Connect a client (mongo) to one server. Through the client, initialize the replication: add one other server as secondary, and add the third one as arbiter.
 - 5. Identify the master from the outputs of the servers' and by requesting replica set information from the servers.
 - Output of server 3:
 - 6. Import `moviepeople-1000.jsonl` through the master. Observe the output of the two other servers.
 - Server 1 Primary
 - Server 2 SECONDARY
 - Server 3 ARBITER
 - 7. Once the synchronization is finished, stop (ctrl-c) the master. Observe the output of the two other servers.
 - from server 1
 - from server 2
- Task 3: sharding

- 1. Start two shard servers.
 - shard server 1
 - shard server 2
 - shard config server
 - starting mongos connected to the config
 - Configure sharding
- 2. Shard the cities by the country.

Task 0: setup

Task 0 and Task 1 outputs are from MongoDB v2.6.12 except [Task1.2.\(f\)](#)

1. Create a directory and launch the server

```
mkdir server_1
mongod --dbpath server_1
```

The port 27017 is used.

2. Import the document `moviepeople-10.jsonl` into the server.

```
mongoimport --db movies --file lab_2_datasets/moviepeople-10.jsonl --collection
movies10
connected to: 127.0.0.1
2019-10-05T17:01:58.792+0200 imported 10 objects
```

3. Launch a client (mongo shell), retrieve all the documents by asking a query in the client.

```
> use movies
switched to db movies
> db.movies10.find()
{ "_id" : ObjectId("5d920f2f1cfd286559094705"), "person-name" : "Cáceres, Luciano", "info" : { "trivia" : [ "Grew up in Buenos Aires's neighborhood Bajo Flores.", "Was classmates of 'Sergio Surraco' (qv), 'Claudio Tolcachir' (qv), 'Lautaro Delgado' (qv) and 'Melina Petriella' (qv) at the drama school.", "Stepfather of 'Ángela Torres (III)' (qv).", "Became a father for the 1st time at age 32 when his wife 'Gloria Carrá' (qv) gave birth to their daughter Amelia Cáceres on October 7, 2009.", "Met wife 'Gloria Carrá' (qv) in 2001, but they didn't become a couple until 2007.", "Is an accomplished stage actor/director.", "Was in a relationship with actress 'Laura Lopez Moyano' (qv) (2005-2007).", "His mother, political activist Haydeé Bello, died of cancer on June 15, 2002.", "Studied acting with 'Alejandra Boero' (qv).", "He started taking acting classes at the age of 9.", "Son of 'Ernesto Cáceres' (qv) and Haydeé Bello." ], "birthnotes" : [ "Autonomous City of Buenos Aires, Argentina" ], "birthdate" : [ "24 January 1977" ], "height" : [ "187 cm" ], "spouse" : [ "'Gloria Carrá' (qv) (27 September 2008 - 2015) (separated); 1 child" ], "nicknames" : [ "Lu" ], "trademark" : [ "Curly hair", "Bright blue eyes" ], "interviews" : [ "Clarín
```

(Argentina), 9 January 2007, by: María Ana Rago, Luciano Cáceres: El teatro en la sangre"] } }

{ "_id" : ObjectId("5d920f2f1cfd286559094706"), "person-name" : "Hughart, Ron", "info" : { "birthnotes" : ["Los Angeles, California, USA"], "birthdate" : ["18 June 1961"], "birthname" : ["Hughart, Ronald P"] } }

{ "_id" : ObjectId("5d920f2f1cfd286559094708"), "person-name" : "Stokely, Charlotte", "info" : { "trivia" : ["Is represented by John Baumgartner and Sandra McCarthy's Nevada-organized talent agency OC Modeling LLC."], "minibiography" : ["Very cute and slim 5'3 blonde Charlotte Stokely was born on August 8, 1986 in Salt Lake City, Utah. Stokely decided to pursue a career in the adult entertainment industry after her stripper roommate was asked to participate in a hardcore shoot and Stokely herself was approached to appear on camera as well. After initially establishing herself in hardcore shoots for primarily Florida-based adult websites, Charlotte subsequently got herself an agent and moved to Los Angeles, California to advance her career by doing features for such top companies as FM Concepts, Sin City, Evil Angel, Wicked Pictures, Zero Tolerance, Evasive Angles, Digital Playground, Pulse Distribution, and Pure Play Media. Stokely was nominated for an AVN Award for Best New Starlet in 2007. Moreover, Charlotte has also modeled for several American Apparel ads. She enjoys playing Dungeons & Dragons in her spare time."], "birthnotes" : ["Utah, USA"], "birthdate" : ["8 August 1986"], "height" : ["160 cm"], "birthname" : ["Vitale, Brittney"] } }

{ "_id" : ObjectId("5d920f2f1cfd286559094709"), "person-name" : "O'Brien, Robert", "info" : { "birthnotes" : ["New York City, New York, USA"], "birthdate" : ["8 May 1918"], "deathdate" : ["7 November 2005"], "spouse" : ["'Lina Romay (I)' (qv) (1992 - 7 November 2005) (his death)"], "deathnotes" : ["Los Angeles, California, USA"] } }

{ "_id" : ObjectId("5d920f2f1cfd28655909470a"), "person-name" : "Wedemeyer, Herman", "info" : { "trivia" : ["Inducted into the Polynesian Football Hall of Fame in 2014 (inaugural class).", "(1950) Played minor league baseball with the Salt Lake City Bees in the Pioneer League (C level).", "Inducted into the Saint Mary's College Athletic Hall of Fame in 1973.", "Inducted into the Hawaii Sports Hall of Fame in 1998.", "Herman is the older brother of Charlie Wedemeyer. Charlie, a former Michigan State football player, eventually (as head football coach at Los Gatos High School in California) gained fame when he continued to coach for many years, after being confined to a wheelchair from contracting 'Lou Gehrig' (qv)'s Disease. He is the subject of _Quiet Victory: The Charlie Wedemeyer Story (1988) (TV)_ (qv).", "Halfback on the football team of St. Mary's College (Moraga, CA), 1943, 1945-1947.", "Inducted into the College Football Hall of Fame in 1979.", "Wedemeyer was elected to the Honolulu City Council as a Republican in 1968, and then elected to the state House of Representatives as a Democrat in 1970. He won re-election in 1972.", "Played halfback/tailback for the Los Angeles Dons (1948) and Baltimore Colts (1949) of the All-America Football Conference (a rival of the National Football League, 1946-1949)."], "birthnotes" : ["Hilo, Hawaii, USA"], "birthdate" : ["20 May 1924"], "height" : ["5' 10"], "deathdate" : ["25 January 1999"], "birthname" : ["Wedemeyer, Herman John"], "nicknames" : ["The Waikiki Wonder", "The Flyin' Hawaiian", "The Hula-hipped Hawaiian", "The Hawaiian Centipede", "Squirmin' Herman"], "deathnotes" : ["Honolulu, Hawaii, USA (heart attack)"] } }

{ "_id" : ObjectId("5d920f2f1cfd28655909470b"), "person-name" : "Goyanes, María José", "info" : { "trivia" : ["Mother of actor 'Javier Collado (I)' (qv).", "Sister of actress 'Mara Goyanes' (qv)", "Daughter of actress 'Mimí Muñoz' (qv)", "Sister of actress 'Vicky Lagos (I)' (qv)", "Sister of actress 'Concha Goyanes' (qv)", "Sister-in-law of actor 'Ismael Merlo' (qv)"], "birthnotes" : ["Spain"], "birthdate" : ["8 December 1948"], "spouse" : ["'Manuel Collado' (qv) (? - ?)"] }

```

], "birthname" : [ "Muñoz, María José Goyanes" ] } }
{ "_id" : ObjectId("5d920f2f1cfd286559094707"), "person-name" : "Hemsworth, Liam",
"info" : { "quotes" : [ "_The Hunger Games (2012)_ (qv) absolutely changed my
life. It's brought up so many different questions about my own personal growth and
where I was heading. I think, before all of it, I really felt like I had it
figured out. Like, you know, it's going to be fine, I'm this person and I know
what I want to be and I know where I want to go. But then it got to the point
where I think I started double-questioning everything. It was like, why am I doing
this? where is it getting me? what if I do this and then I can do this? I wasn't
living in the moment.", "One of the biggest inspirations before I started shooting
came from my brother, when he texted me and said, Hey, fatty, it's called 'The
Hunger Games', not 'The Eating Games'.. So I started working out a lot more and
eating a lot less." ], "trivia" : [ "(2016) Became the first celebrity face of
Diesel Only The Brave fragrance.", "Liam's childhood crush was 'Alyssa Milano'
(qv) from his favorite weekly show _Charmed (1998)_ (qv).", "His uncle was bushman
Rod Ansell, who inspired the film _Crocodile Dundee (1986)_ (qv). Rod was married
to Liam's aunt (Liam's mother's sister), Joanne van Os, who is an author.",
"Cousin Rob Hemsworth, an oyster farmer, competed on Season 6 of _My Kitchen Rules
(2010)_ (qv), representing Victoria.", "Was originally going to appear in _The
Expendables (2010)_ (qv), but was written out due of doing the screen test for
_Thor (2011)_ (qv). He would later appear in _The Expendables 2 (2012)_ (qv).",
"Favorite actors include 'Leonardo DiCaprio' (qv), 'Heath Ledger' (qv), and 'Paul
Newman (I)' (qv) among others.", "Before his career in acting took off, Liam
worked some odd jobs that included nature park ranger, bakery assistant and
bowling alley employee.", "Was considered for the title role in _Thor (2011)_
(qv), which ultimately went to his brother, 'Chris Hemsworth' (qv).", "Favorite
films include _The Goonies (1985)_ (qv), _James Dean (2001) (TV)_ (qv), _A Guide
to Recognizing Your Saints (2006)_ (qv), _The Departed (2006)_ (qv) and _Step
Brothers (2008)_ (qv).", "Ranked the #3 Sexiest Man in 2013 by Glamour, several
places ahead of his older brother 'Chris Hemsworth' (qv) who ranked in at number
7.", "(September 16, 2013) Ended engagement to 'Miley Cyrus' (qv).", "His maternal
grandfather, Martin van Os, is a Dutch immigrant. The rest of Liam's ancestry is
Irish, English, and smaller amounts of German and Scottish.", "Between auditions,
he used to lay floors.", "Younger brother of actor 'Chris Hemsworth' (qv) and
'Luke Hemsworth' (qv).", "Was considered one of the 55 faces of the future by
Nylon Magazine's Young Hollywood Issue.", "Brother-in-law of 'Elsa Pataky' (qv).
"], "minibiography" : [ "Liam Hemsworth was born on January 13, 1990, in Melbourne,
Australia, and is the younger brother of actors 'Chris Hemsworth' (qv) and 'Luke
Hemsworth' (qv). He is the son of Leonie (van Os), a teacher of English, and Craig
Hemsworth, a social-services counselor. He is of Dutch (from his immigrant
maternal grandfather), Irish, English, Scottish, and German ancestry. His uncle,
by marriage, was Rod Ansell, the bushman who inspired the film _Crocodile Dundee
(1986)_ (qv). The Hemsworth family lived primarily on Phillip Island, a small
island located south of Melbourne. Following in the footsteps of his older
brothers, who went into acting in their teens, Liam scored his first audition at
age 16 and appeared on the Australian TV series _Home and Away (1988)_ (qv) and
_McLeod's Daughters (2001)_ (qv) before taking on a recurring character role on
the soap opera _Neighbours (1985)_ (qv), in which his brother Luke had also
appeared. Roles on TV shows _The Elephant Princess (2008)_ (qv) and _Satisfaction
(2007)_ (qv) followed before Liam moved to the United States to pursue a big-
screen career. After suffering two setbacks - his character was written out of
the script for _The Expendables (2010)_ (qv) days before filming and he lost the
title role of _Thor (2011)_ (qv) to his brother Chris - Liam was cast opposite
'Miley Cyrus' (qv) in the 'Nicholas Sparks (I)' (qv) drama _The Last Song (2010)_

```

(qv). The two, who played love interests in the film, soon started dating, and Liam appeared in Cyrus' music video When I Look at You. Following that film's modest commercial success, and the attendant press coverage of his rising career and high-profile romance, he was almost immediately thrust into leading man status, and was cast as Gale Hawthorne in the big-screen adaptation of the best-selling novel *The Hunger Games* (2012)_ (qv). Following the blockbuster success of that film, Liam nabbed a number of roles, including a supporting part in *The Expendables 2* (2012)_ (qv) and leading roles in the war drama *Love and Honor* (2013)_ (qv), the crime drama *Empire State* (2013)_ (qv), and the thriller *Paranoia* (2013/I)_ (qv). He will also star as Ali Baba in a 3D production of *Arabian Nights* (????)_ (qv) and will reprise the role of Gale Hawthorne in *The Hunger Games: Catching Fire* (2013)_ (qv). Hemsworth was engaged to 'Miley Cyrus' (qv) from June 2012 to September 2013."], "birthnotes" : ["Melbourne, Victoria, Australia"], "birthdate" : ["13 January 1990"], "height" : ["6' 3"], "otherworks" : ["(2011) Music video for Zac Brown Band - Colder Weather", "Music video for 'Miley Cyrus' (qv): When I Look At You.", "(2016) TV commercial for Diesel Only the Brave - The Fragrance - Actor with 'Valerie Martinez (II)' (qv). Director: 'Anthony Atanasio' (qv)"], "books" : ["Posy Edwards. *Liam Hemsworth: Against All Odds.* London, England: Orion Publishing, 2012. ISBN 9781409143772"], "trademark" : ["Towering height", "Blue eyes"], "article" : ["USA Today (USA), 20 November 2015, Vol. 34, Iss. 49, pg. 1B-2B, by: Bryan Alexander, 'HUNGER GAMES' TRIO TAKE THEIR FINAL BOW - For these three fast friends, 'Mockingjay - Part 2' marks the end of their adventure", "USA Today (USA), 27 June 2016, Vol. 34, Iss. 202, pg. 2D, by: Brian Truitt, 'Independence Day' sequel isn't worth a worldwide celebration"], "magazinecoverphoto" : ["Entertainment Weekly (USA), 11 October 2013, Iss. 1280"], "pictorial" : ["Vanity Fair (USA), March 2013, Iss. 631, pg. 346, by: Bruce Weber, Bruce Weber's Adventures in Hollywood"] } }

{ "_id" : ObjectId("5d920f2f1cfd28655909470c"), "person-name" : "Rath, Franz", "info" : { "birthdate" : ["22 June 1932"] } }

{ "_id" : ObjectId("5d920f2f1cfd28655909470d"), "person-name" : "van Mello, Luk", "info" : { "trivia" : ["He studied drama at the Koninlijk Conservatorium in Brussels, Belgium. He had some famous teachers, such as 'Nand Buyl' (qv) and 'Leo Dewals' (qv). People like 'Luk De Konink' (qv), 'Jaak Van Assche' (qv), 'Tessy Moerenhout' (qv), 'Luc Springuel' (qv), 'Gilda De Bal' (qv), 'Josse De Pauw' (qv), 'Magda De Winter' (qv), 'Chris Cauwenbergs' (qv), 'Tuur De Weert' (qv) and 'René Verreth' (qv) were his co-students."], "birthnotes" : ["Opbrakel, Flanders, Belgium"], "birthdate" : ["1 September 1950"], "otherworks" : ["Provided a voice in the Dutch version of *Simsala Grimm - Die Märchen der Brüder Grimm* (1999)_ (qv).", "Provided the voice of Luke in the Dutch version of *Jim Knopf* (2000)_ (qv) (Jim Button).", "Provided the voice of Uncle Max in the Dutch version of *The Lion King 1½* (2004) (V)_ (qv).", "Provided the voice of Murphy in the Dutch version of *Spirit: Stallion of the Cimarron* (2002)_ (qv).", "Provided the voice of Hamm in the Dutch version of *Toy Story 2* (1999)_ (qv).", "Provided the voice of Acoliet in the Dutch version of *The Road to El Dorado* (2000)_ (qv).", "Provided a voice in the Dutch version of *Stuart Little* (1999)_ (qv).", "Provided the voice of Kolonel in the Dutch version of *La gabbianella e il gatto* (1998)_ (qv).", "Provided the voice of Coach Fanelli in the Dutch version of *Air Bud: Golden Receiver* (1998)_ (qv).", "Provided the voice of Stekel in the Dutch version of *A Bug's Life* (1998)_ (qv).", "Provided the voice of Hamm in the Dutch version of *Toy Story* (1995)_ (qv)."], "birthname" : ["van Mello, Luc Jean-Marie Joseph"] } }

{ "_id" : ObjectId("5d920f2f1cfd28655909470e"), "person-name" : "Okamoto, Fujita", "info" : { "birthnotes" : ["Yokohama, Japan"], "birthdate" : ["21 November 1946"] } }

Task 1: import and querying

1. Import the documents `moviepeople-3000.jsonl` and `cities.jsonl` into the server (the one set up in Task 0).

```
$ mongoimport --db movies --file lab_2_datasets/moviepeople-3000.jsonl --collection movies3000
connected to: 127.0.0.1
2019-10-05T18:17:24.666+0200 check 9 3000
2019-10-05T18:17:24.740+0200 imported 3000 objects
$ mongoimport --db movies --file lab_2_datasets/cities.jsonl --collection cities
connected to: 127.0.0.1
2019-10-05T18:17:53.733+0200 check 9 99838
2019-10-05T18:17:54.230+0200 imported 99838 objects
```

2. In the mongo shell client, write queries finding:

a) the person named Anabela Teixeira

```
> db.movies3000.find({"person-name":"Teixeira, Anabela"})
{ "_id" : ObjectId("5d9213ed1cfd2865590948c6"), "person-name" : "Teixeira, Anabela", "info" : { "trivia" : [ "Her favorite actor is 'Marcello Mastroianni' (qv) and her favorite actress is 'Juliette Binoche' (qv).", "Partner of musician Frederico Pereira.", "Has two brothers, one biological and one adopted.", "Vice-President of the Portuguese Cinema Academy.", "She was first noticed by the public with the Tv series A Viúva do Enforcado (1992).", "Made her theater debut in 1992 in the play Os Processos of Dostolevsky.", "Did some workshops of dance: dance in movement with Peter Diez and dance classes with Madalena Vitorino.", "Besides Portuguese, she speaks English and French." ], "birthnotes" : [ "Lisbon, Portugal" ], "birthdate" : [ "18 May 1973" ], "birthname" : [ "Teixeira, Anabela Cristina Alves" ], "interviews" : [ "TV Guia (Portugal), 1997, Iss. 960, pg. 24-25, by: Pedro Teixeira", "A Capital (Portugal), 30 April 1998, pg. 55, by: Helena Mata" ] } }
```

b) the birthplace of Steven Spielberg

```
> db.movies3000.find({ "person-name":"Spielberg, Steven" }, { "info.birthnotes": 1, _id:0})
{ "info" : { "birthnotes" : [ "Cincinnati, Ohio, USA" ] } }
```

c) the number of people born in Lisbon


```
> db.movies3000.find({"info.birthnotes":"Lisbon, Portugal"}).count()
171
```

d) the people taller than 170 cm

```
> db.movies3000.find("this.info.height > 170")
{ "_id" : ObjectId("5d9213ee1cfd286559095111"), "person-name" : "Filho, Tarcísio",
"info" : { "trivia" : [ "Son of 'Tarcísio Meira' (qv) and 'Glória Menezes' (qv)",
"Has two half-siblings named Maria Amélia (born in 1957) and João Paulo (born in
1959).", ], "birthnotes" : [ "São Paulo, São Paulo, Brazil" ], "birthdate" : [ "22
August 1964" ], "height" : [ "181" ], "spouse" : [ "'Luzia Dvorek' (? - ?)
(divorced)" ], "birthname" : [ "de Magalhães, Tarcísio Pereira Filho" ] } }
```

e) the names of people whose information contains "Opera"

```
> db.movies3000.find({"info.trivia":/ Opera /},{_id:0, "person-name":1 })
{ "person-name" : "Rolfe, James" }
{ "person-name" : "Mascolo, Joseph" }
{ "person-name" : "Bailey, Laura" }
{ "person-name" : "Christian, Shawn" }
{ "person-name" : "Fulton, Eileen" }
{ "person-name" : "Springer, Jerry" }
{ "person-name" : "Heinle, Amelia" }
{ "person-name" : "Mattson, Robin" }
{ "person-name" : "Egan, Melissa Claire" }
{ "person-name" : "de la Fuente, Cristián" }
{ "person-name" : "O'Shaughnessey, Colleen" }
{ "person-name" : "Bjorlin, Nadia" }
{ "person-name" : "Randall, Tony" }
```

f) for each movie person whose birth place is known, the latitude, longitude and population of that city (if that information exists in the city document).

This command is run under MongoDB version v4.0.12 instead of v2.6.12 in order to use `$lookup` for aggregation.

```
> db.movies3000.aggregate([
  {$project: {city : {$split: [{$arrayElemAt["$info.birthnotes",0]}, " ",
""]}},
  {$lookup: {from: "cities",localField:
"city",foreignField:"name",as:"cit"}},
  {$unwind: "$cit"},
  {$project:{_id : 0, "cit.name":1, "cit.population" : 1, "cit.location" :
1}}
])
```

```

{ "cit" : { "name" : "Chicago", "population" : 2841952, "location" : { "latitude"
: 41.85003, "longitude" : -87.65005 } } }
{ "cit" : { "name" : "Chicago", "population" : 2841952, "location" : { "latitude"
: 41.85003, "longitude" : -87.65005 } } }
{ "cit" : { "name" : "Salford", "population" : 72750, "location" : { "latitude" :
53.48771, "longitude" : -2.29042 } } }
{ "cit" : { "name" : "Eccles", "population" : 37275, "location" : { "latitude" :
53.48333, "longitude" : -2.33333 } } }
{ "cit" : { "name" : "England", "population" : 2950, "location" : { "latitude" :
34.54426, "longitude" : -91.96903 } } }
{ "cit" : { "name" : "Omaha", "population" : 390007, "location" : { "latitude" :
41.25861, "longitude" : -95.93779 } } }
{ "cit" : { "name" : "Brooklyn", "population" : 3855, "location" : { "latitude" :
-41.31667, "longitude" : 174.75 } } }
{ "cit" : { "name" : "Brooklyn", "population" : 1511, "location" : { "latitude" :
39.53921, "longitude" : -86.36916 } } }
{ "cit" : { "name" : "Brooklyn", "population" : 1387, "location" : { "latitude" :
41.73361, "longitude" : -92.44546 } } }
{ "cit" : { "name" : "Brooklyn", "population" : 1247, "location" : { "latitude" :
42.10587, "longitude" : -84.24828 } } }
{ "cit" : { "name" : "Brooklyn", "population" : 2300664, "location" : { "latitude"
: 40.6501, "longitude" : -73.94958 } } }
{ "cit" : { "name" : "New York City", "population" : 8008278, "location" : {
"latitude" : 40.71427, "longitude" : -74.00597 } } }
{ "cit" : { "name" : "Brooklyn", "population" : 11037, "location" : { "latitude" :
41.43977, "longitude" : -81.73541 } } }
{ "cit" : { "name" : "Brooklyn", "population" : 1120, "location" : { "latitude" :
42.85361, "longitude" : -89.3704 } } }
{ "cit" : { "name" : "Philadelphia", "population" : 7266, "location" : {
"latitude" : 32.77152, "longitude" : -89.11673 } } }
{ "cit" : { "name" : "Philadelphia", "population" : 1517550, "location" : {
"latitude" : 39.95234, "longitude" : -75.16379 } } }
{ "cit" : { "name" : "Philadelphia", "population" : 1537, "location" : {
"latitude" : 44.1545, "longitude" : -75.70882 } } }
{ "cit" : { "name" : "Chicago", "population" : 2841952, "location" : { "latitude"
: 41.85003, "longitude" : -87.65005 } } }
{ "cit" : { "name" : "Indianapolis", "population" : 773283, "location" : {
"latitude" : 39.76838, "longitude" : -86.15804 } } }
{ "cit" : { "name" : "Indiana", "population" : 14357, "location" : { "latitude" :
40.62146, "longitude" : -79.15253 } } }
Type "it" for more
>

```

And I've checked, there are really multiple cities named "Brooklyn", "Philadelphia", "Washington"... in the file `cities.jsonl` and the real-life United States, Americans really don't have too much creativity to name the cities they live in...

Task 2: replication

Task 2 is run under MongoDB version v4.0.12

1. Create working directories for 3 MongoDB servers.


```
$ mkdir ./mongo1 ./mongo2 ./mongo3
```

2. Create a replica set for a collection called small-movie.

Because the `--rest` parameter was removed in MongoDB 3.6 as described in this page: [MongoDB Configuration Hardening](#) and this ticket: [SERVER-29000](#), this step is done with the next one.

3. Launch the three MongoDB servers (in different shells) and let them run.

Server 1

```
$ mongod --replSet small-movie --dbpath ./mongo1 --port 27011
2019-10-05T23:18:31.397+0200 I STORAGE [main] Max cache overflow file size custom
option: 0
2019-10-05T23:18:31.400+0200 I CONTROL [main] Automatically disabling TLS 1.0, to
force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
2019-10-05T23:18:31.404+0200 I CONTROL [initandlisten] MongoDB starting :
pid=77052 port=27011 dbpath=./mongo1 64-bit host=lame10.enst.fr
2019-10-05T23:18:31.404+0200 I CONTROL [initandlisten] db version v4.0.12
2019-10-05T23:18:31.404+0200 I CONTROL [initandlisten] git version:
5776e3cbf9e7afe86e6b29e22520ffb6766e95d4
2019-10-05T23:18:31.404+0200 I CONTROL [initandlisten] OpenSSL version: OpenSSL
1.0.0-fips 29 Mar 2010
2019-10-05T23:18:31.404+0200 I CONTROL [initandlisten] allocator: tcmalloc
2019-10-05T23:18:31.404+0200 I CONTROL [initandlisten] modules: none
2019-10-05T23:18:31.404+0200 I CONTROL [initandlisten] build environment:
2019-10-05T23:18:31.404+0200 I CONTROL [initandlisten] distmod: amazon
2019-10-05T23:18:31.404+0200 I CONTROL [initandlisten] distarch: x86_64
2019-10-05T23:18:31.404+0200 I CONTROL [initandlisten] target_arch: x86_64
2019-10-05T23:18:31.404+0200 I CONTROL [initandlisten] options: { net: { port:
27011 }, replication: { replSet: "small-movie" }, storage: { dbPath: "./mongo1" }
}
2019-10-05T23:18:31.420+0200 I STORAGE [initandlisten] wiredtiger_open config:
create,cache_size=257478M,cache_overflow=(file_max=0M),session_max=20000,eviction=
(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=
(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=
(close_idle_time=100000),statistics_log=(wait=0),verbose=(recovery_progress),
2019-10-05T23:18:34.157+0200 I STORAGE [initandlisten] WiredTiger message
[1570310314:157591][77052:0x7f4b0f32a000], txn-recover: Set global recovery
timestamp: 0
2019-10-05T23:18:34.261+0200 I RECOVERY [initandlisten] WiredTiger
recoveryTimestamp. Ts: Timestamp(0, 0)
2019-10-05T23:18:37.058+0200 I CONTROL [initandlisten]
2019-10-05T23:18:37.058+0200 I CONTROL [initandlisten] ** WARNING: Access control
is not enabled for the database.
2019-10-05T23:18:37.058+0200 I CONTROL [initandlisten] ** Read and write
access to data and configuration is unrestricted.
2019-10-05T23:18:37.058+0200 I CONTROL [initandlisten]
2019-10-05T23:18:37.058+0200 I CONTROL [initandlisten] ** WARNING: This server is
bound to localhost.
```

```

2019-10-05T23:18:37.058+0200 I CONTROL [initandlisten] ** Remote systems
will be unable to connect to this server.
2019-10-05T23:18:37.058+0200 I CONTROL [initandlisten] ** Start the
server with --bind_ip <address> to specify which IP
2019-10-05T23:18:37.058+0200 I CONTROL [initandlisten] ** addresses it
should serve responses from, or with --bind_ip_all to
2019-10-05T23:18:37.058+0200 I CONTROL [initandlisten] ** bind to all
interfaces. If this behavior is desired, start the
2019-10-05T23:18:37.058+0200 I CONTROL [initandlisten] ** server with --
bind_ip 127.0.0.1 to disable this warning.
2019-10-05T23:18:37.058+0200 I CONTROL [initandlisten]
2019-10-05T23:18:37.061+0200 I CONTROL [initandlisten]
2019-10-05T23:18:37.061+0200 I CONTROL [initandlisten] ** WARNING: You are
running on a NUMA machine.
2019-10-05T23:18:37.061+0200 I CONTROL [initandlisten] ** We suggest
launching mongod like this to avoid performance problems:
2019-10-05T23:18:37.061+0200 I CONTROL [initandlisten] ** numactl --
interleave=all mongod [other options]
2019-10-05T23:18:37.061+0200 I CONTROL [initandlisten]
2019-10-05T23:18:37.067+0200 I STORAGE [initandlisten] createCollection:
local.startup_log with generated UUID: 221448d6-1702-45aa-b775-fe22fb448951
2019-10-05T23:18:37.281+0200 I FTDC [initandlisten] Initializing full-time
diagnostic data capture with directory './mongo1/diagnostic.data'
2019-10-05T23:18:37.283+0200 I STORAGE [initandlisten] createCollection:
local.replset.oplogTruncateAfterPoint with generated UUID: 5c5cdca6-b1e3-4c08-
b4b8-65e8dccf4d08
2019-10-05T23:18:37.501+0200 I STORAGE [initandlisten] createCollection:
local.replset.minvalid with generated UUID: 315be355-2a14-4a0f-9ebd-451fbe7a42fa
2019-10-05T23:18:37.710+0200 I STORAGE [initandlisten] createCollection:
local.replset.election with generated UUID: f5cec572-4984-4e82-bcda-629f7d764493
2019-10-05T23:18:37.910+0200 I REPL [initandlisten] Did not find local
initialized voted for document at startup.
2019-10-05T23:18:37.910+0200 I REPL [initandlisten] Did not find local
Rollback ID document at startup. Creating one.
2019-10-05T23:18:37.910+0200 I STORAGE [initandlisten] createCollection:
local.system.rollback.id with generated UUID: 1bd96e2e-cf6a-4cdb-9244-aeb9ab9c68b8
2019-10-05T23:18:38.130+0200 I REPL [initandlisten] Initialized the rollback
ID to 1
2019-10-05T23:18:38.130+0200 I REPL [initandlisten] Did not find local replica
set configuration document at startup; NoMatchingDocument: Did not find replica
set configuration document in local.system.replset
2019-10-05T23:18:38.131+0200 I CONTROL [LogicalSessionCacheRefresh] Sessions
collection is not set up; waiting until next sessions refresh interval:
Replication has not yet been configured
2019-10-05T23:18:38.131+0200 I NETWORK [initandlisten] waiting for connections on
port 27011
2019-10-05T23:18:38.132+0200 I CONTROL [LogicalSessionCacheReap] Sessions
collection is not set up; waiting until next sessions reap interval:
config.system.sessions does not exist

```

Server 2

```
$ mongod --replSet small-movie --dbpath ./mongo2 --port 27012
2019-10-05T23:23:53.378+0200 I STORAGE [main] Max cache overflow file size custom
option: 0
2019-10-05T23:23:53.381+0200 I CONTROL [main] Automatically disabling TLS 1.0, to
force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
2019-10-05T23:23:53.385+0200 I CONTROL [initandlisten] MongoDB starting :
pid=78567 port=27012 dbpath=./mongo2 64-bit host=lame10.enst.fr
2019-10-05T23:23:53.385+0200 I CONTROL [initandlisten] db version v4.0.12
2019-10-05T23:23:53.385+0200 I CONTROL [initandlisten] git version:
5776e3cbf9e7afe86e6b29e22520ffb6766e95d4
2019-10-05T23:23:53.385+0200 I CONTROL [initandlisten] OpenSSL version: OpenSSL
1.0.0-fips 29 Mar 2010
2019-10-05T23:23:53.385+0200 I CONTROL [initandlisten] allocator: tcmalloc
2019-10-05T23:23:53.385+0200 I CONTROL [initandlisten] modules: none
2019-10-05T23:23:53.385+0200 I CONTROL [initandlisten] build environment:
2019-10-05T23:23:53.385+0200 I CONTROL [initandlisten] distmod: amazon
2019-10-05T23:23:53.385+0200 I CONTROL [initandlisten] distarch: x86_64
2019-10-05T23:23:53.385+0200 I CONTROL [initandlisten] target_arch: x86_64
2019-10-05T23:23:53.385+0200 I CONTROL [initandlisten] options: { net: { port:
27012 }, replication: { replSet: "movie-small" }, storage: { dbPath: "./mongo2" }
}
2019-10-05T23:23:53.392+0200 I STORAGE [initandlisten] wiredtiger_open config:
create,cache_size=257478M,cache_overflow=(file_max=0M),session_max=20000,eviction=
(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=
(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=
(close_idle_time=100000),statistics_log=(wait=0),verbose=(recovery_progress),
2019-10-05T23:23:55.868+0200 I STORAGE [initandlisten] WiredTiger message
[1570310635:868458][78567:0x7f0c9f8e0000], txn-recover: Set global recovery
timestamp: 0
2019-10-05T23:23:55.980+0200 I RECOVERY [initandlisten] WiredTiger
recoveryTimestamp. Ts: Timestamp(0, 0)
2019-10-05T23:23:58.704+0200 I CONTROL [initandlisten]
2019-10-05T23:23:58.704+0200 I CONTROL [initandlisten] ** WARNING: Access control
is not enabled for the database.
2019-10-05T23:23:58.704+0200 I CONTROL [initandlisten] ** Read and write
access to data and configuration is unrestricted.
2019-10-05T23:23:58.704+0200 I CONTROL [initandlisten]
2019-10-05T23:23:58.704+0200 I CONTROL [initandlisten] ** WARNING: This server is
bound to localhost.
2019-10-05T23:23:58.704+0200 I CONTROL [initandlisten] ** Remote systems
will be unable to connect to this server.
2019-10-05T23:23:58.704+0200 I CONTROL [initandlisten] ** Start the
server with --bind_ip <address> to specify which IP
2019-10-05T23:23:58.704+0200 I CONTROL [initandlisten] ** addresses it
should serve responses from, or with --bind_ip_all to
2019-10-05T23:23:58.704+0200 I CONTROL [initandlisten] ** bind to all
interfaces. If this behavior is desired, start the
2019-10-05T23:23:58.704+0200 I CONTROL [initandlisten] ** server with --
bind_ip 127.0.0.1 to disable this warning.
2019-10-05T23:23:58.705+0200 I CONTROL [initandlisten]
2019-10-05T23:23:58.707+0200 I CONTROL [initandlisten]
2019-10-05T23:23:58.708+0200 I CONTROL [initandlisten] ** WARNING: You are
running on a NUMA machine.
```

```

2019-10-05T23:23:58.708+0200 I CONTROL [initandlisten] **           We suggest
launching mongod like this to avoid performance problems:
2019-10-05T23:23:58.708+0200 I CONTROL [initandlisten] **           numactl --
interleave=all mongod [other options]
2019-10-05T23:23:58.708+0200 I CONTROL [initandlisten]
2019-10-05T23:23:58.713+0200 I STORAGE [initandlisten] createCollection:
local.startup_log with generated UUID: 94307054-2b6e-4759-8a84-6a83eb6fe230
2019-10-05T23:23:59.079+0200 I FTDC [initandlisten] Initializing full-time
diagnostic data capture with directory './mongo2/diagnostic.data'
2019-10-05T23:23:59.080+0200 I STORAGE [initandlisten] createCollection:
local.replset.oplogTruncateAfterPoint with generated UUID: b3e018a8-59a8-481b-
801c-1322b33f50bc
2019-10-05T23:23:59.439+0200 I STORAGE [initandlisten] createCollection:
local.replset.minvalid with generated UUID: 33324978-ae5a-4fec-b0d1-6da90927056a
2019-10-05T23:23:59.700+0200 I STORAGE [initandlisten] createCollection:
local.replset.election with generated UUID: d1c191ea-8c16-40a6-becf-ec112d7d073d
2019-10-05T23:23:59.900+0200 I REPL [initandlisten] Did not find local
initialized voted for document at startup.
2019-10-05T23:23:59.900+0200 I REPL [initandlisten] Did not find local
Rollback ID document at startup. Creating one.
2019-10-05T23:23:59.900+0200 I STORAGE [initandlisten] createCollection:
local.system.rollback.id with generated UUID: 39c8beb4-89ff-4731-a23e-0830ebf69c73
2019-10-05T23:24:00.110+0200 I REPL [initandlisten] Initialized the rollback
ID to 1
2019-10-05T23:24:00.110+0200 I REPL [initandlisten] Did not find local replica
set configuration document at startup; NoMatchingDocument: Did not find replica
set configuration document in local.system.replset
2019-10-05T23:24:00.111+0200 I CONTROL [LogicalSessionCacheRefresh] Sessions
collection is not set up; waiting until next sessions refresh interval:
Replication has not yet been configured
2019-10-05T23:24:00.111+0200 I CONTROL [LogicalSessionCacheReap] Sessions
collection is not set up; waiting until next sessions reap interval:
config.system.sessions does not exist
2019-10-05T23:24:00.111+0200 I NETWORK [initandlisten] waiting for connections on
port 27012
^C2019-10-05T23:27:26.185+0200 I CONTROL [signalProcessingThread] got signal 2
(Interrupt), will terminate after current cmd ends
2019-10-05T23:27:26.185+0200 I NETWORK [signalProcessingThread] shutdown: going
to close listening sockets...
2019-10-05T23:27:26.185+0200 I NETWORK [signalProcessingThread] removing socket
file: /tmp/mongodb-27012.sock
2019-10-05T23:27:26.186+0200 I REPL [signalProcessingThread] shutting down
replication subsystems
2019-10-05T23:27:26.187+0200 I ASIO [Replication] Killing all outstanding
egress activity.
2019-10-05T23:27:26.187+0200 I CONTROL [signalProcessingThread] Shutting down
free monitoring
2019-10-05T23:27:26.188+0200 I FTDC [signalProcessingThread] Shutting down
full-time diagnostic data capture
2019-10-05T23:27:26.208+0200 I STORAGE [signalProcessingThread]
WiredTigerKVEngine shutting down
2019-10-05T23:27:26.208+0200 I STORAGE [signalProcessingThread] Shutting down
session sweeper thread
2019-10-05T23:27:26.208+0200 I STORAGE [signalProcessingThread] Finished shutting

```

```

down session sweeper thread
2019-10-05T23:27:26.370+0200 I STORAGE [signalProcessingThread] shutdown:
removing fs lock...
2019-10-05T23:27:26.372+0200 I CONTROL [signalProcessingThread] now exiting
2019-10-05T23:27:26.372+0200 I CONTROL [signalProcessingThread] shutting down
with code:0
[xizhang@lame10]~/tmp/bda2% mongod --replSet small-movie --dbpath ./mongo2 --port
27012
2019-10-05T23:28:15.253+0200 I STORAGE [main] Max cache overflow file size custom
option: 0
2019-10-05T23:28:15.256+0200 I CONTROL [main] Automatically disabling TLS 1.0, to
force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
2019-10-05T23:28:15.260+0200 I CONTROL [initandlisten] MongoDB starting :
pid=80429 port=27012 dbpath=./mongo2 64-bit host=lame10.enst.fr
2019-10-05T23:28:15.260+0200 I CONTROL [initandlisten] db version v4.0.12
2019-10-05T23:28:15.260+0200 I CONTROL [initandlisten] git version:
5776e3cbf9e7afe86e6b29e22520ffb6766e95d4
2019-10-05T23:28:15.260+0200 I CONTROL [initandlisten] OpenSSL version: OpenSSL
1.0.0-fips 29 Mar 2010
2019-10-05T23:28:15.260+0200 I CONTROL [initandlisten] allocator: tcmalloc
2019-10-05T23:28:15.260+0200 I CONTROL [initandlisten] modules: none
2019-10-05T23:28:15.260+0200 I CONTROL [initandlisten] build environment:
2019-10-05T23:28:15.260+0200 I CONTROL [initandlisten] distmod: amazon
2019-10-05T23:28:15.260+0200 I CONTROL [initandlisten] distarch: x86_64
2019-10-05T23:28:15.260+0200 I CONTROL [initandlisten] target_arch: x86_64
2019-10-05T23:28:15.260+0200 I CONTROL [initandlisten] options: { net: { port:
27012 }, replication: { replSet: "small-movie" }, storage: { dbPath: "./mongo2" }
}
2019-10-05T23:28:15.273+0200 I STORAGE [initandlisten] Detected data files in
./mongo2 created by the 'wiredTiger' storage engine, so setting the active storage
engine to 'wiredTiger'.
2019-10-05T23:28:15.276+0200 I STORAGE [initandlisten] wiredtiger_open config:
create,cache_size=257478M,cache_overflow=(file_max=0M),session_max=20000,eviction=
(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=
(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=
(close_idle_time=100000),statistics_log=(wait=0),verbose=(recovery_progress),
2019-10-05T23:28:19.684+0200 I STORAGE [initandlisten] WiredTiger message
[1570310899:684296][80429:0x7f1f255d5000], txn-recover: Main recovery loop:
starting at 1/23168 to 2/256
2019-10-05T23:28:19.887+0200 I STORAGE [initandlisten] WiredTiger message
[1570310899:887768][80429:0x7f1f255d5000], txn-recover: Recovering log 1 through 2
2019-10-05T23:28:20.058+0200 I STORAGE [initandlisten] WiredTiger message
[1570310900:58766][80429:0x7f1f255d5000], txn-recover: Recovering log 2 through 2
2019-10-05T23:28:20.157+0200 I STORAGE [initandlisten] WiredTiger message
[1570310900:157655][80429:0x7f1f255d5000], txn-recover: Set global recovery
timestamp: 0
2019-10-05T23:28:21.407+0200 I RECOVERY [initandlisten] WiredTiger
recoveryTimestamp. Ts: Timestamp(0, 0)
2019-10-05T23:28:23.416+0200 I CONTROL [initandlisten]
2019-10-05T23:28:23.416+0200 I CONTROL [initandlisten] ** WARNING: Access control
is not enabled for the database.
2019-10-05T23:28:23.416+0200 I CONTROL [initandlisten] ** Read and write
access to data and configuration is unrestricted.
2019-10-05T23:28:23.416+0200 I CONTROL [initandlisten]

```



```

2019-10-05T23:28:23.416+0200 I CONTROL [initandlisten] ** WARNING: This server is
bound to localhost.
2019-10-05T23:28:23.416+0200 I CONTROL [initandlisten] **           Remote systems
will be unable to connect to this server.
2019-10-05T23:28:23.416+0200 I CONTROL [initandlisten] **           Start the
server with --bind_ip <address> to specify which IP
2019-10-05T23:28:23.416+0200 I CONTROL [initandlisten] **           addresses it
should serve responses from, or with --bind_ip_all to
2019-10-05T23:28:23.416+0200 I CONTROL [initandlisten] **           bind to all
interfaces. If this behavior is desired, start the
2019-10-05T23:28:23.416+0200 I CONTROL [initandlisten] **           server with --
bind_ip 127.0.0.1 to disable this warning.
2019-10-05T23:28:23.416+0200 I CONTROL [initandlisten]
2019-10-05T23:28:23.419+0200 I CONTROL [initandlisten]
2019-10-05T23:28:23.419+0200 I CONTROL [initandlisten] ** WARNING: You are
running on a NUMA machine.
2019-10-05T23:28:23.419+0200 I CONTROL [initandlisten] **           We suggest
launching mongod like this to avoid performance problems:
2019-10-05T23:28:23.419+0200 I CONTROL [initandlisten] **           numactl --
interleave=all mongod [other options]
2019-10-05T23:28:23.419+0200 I CONTROL [initandlisten]
2019-10-05T23:28:23.565+0200 I FTDC [initandlisten] Initializing full-time
diagnostic data capture with directory './mongo2/diagnostic.data'
2019-10-05T23:28:23.568+0200 I REPL [initandlisten] Did not find local
initialized voted for document at startup.
2019-10-05T23:28:23.685+0200 I REPL [initandlisten] Rollback ID is 1
2019-10-05T23:28:23.685+0200 I REPL [initandlisten] Did not find local replica
set configuration document at startup; NoMatchingDocument: Did not find replica
set configuration document in local.system.replset
2019-10-05T23:28:23.686+0200 I CONTROL [LogicalSessionCacheRefresh] Sessions
collection is not set up; waiting until next sessions refresh interval:
Replication has not yet been configured
2019-10-05T23:28:23.687+0200 I NETWORK [initandlisten] waiting for connections on
port 27012
2019-10-05T23:28:23.687+0200 I CONTROL [LogicalSessionCacheReap] Sessions
collection is not set up; waiting until next sessions reap interval:
config.system.sessions does not exist

```

Server 3

```

$ mongod --replSet small-movie --dbpath ./mongo3 --port 27013
2019-10-05T23:30:31.680+0200 I STORAGE [main] Max cache overflow file size custom
option: 0
2019-10-05T23:30:31.683+0200 I CONTROL [main] Automatically disabling TLS 1.0, to
force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
2019-10-05T23:30:31.687+0200 I CONTROL [initandlisten] MongoDB starting :
pid=81484 port=27013 dbpath=./mongo3 64-bit host=lame10.enst.fr
2019-10-05T23:30:31.687+0200 I CONTROL [initandlisten] db version v4.0.12
2019-10-05T23:30:31.687+0200 I CONTROL [initandlisten] git version:
5776e3cbf9e7afe86e6b29e22520fffb6766e95d4
2019-10-05T23:30:31.687+0200 I CONTROL [initandlisten] OpenSSL version: OpenSSL

```


1.0.0-fips 29 Mar 2010

```

2019-10-05T23:30:31.687+0200 I CONTROL [initandlisten] allocator: tcmalloc
2019-10-05T23:30:31.687+0200 I CONTROL [initandlisten] modules: none
2019-10-05T23:30:31.687+0200 I CONTROL [initandlisten] build environment:
2019-10-05T23:30:31.687+0200 I CONTROL [initandlisten]     distmod: amazon
2019-10-05T23:30:31.687+0200 I CONTROL [initandlisten]     distarch: x86_64
2019-10-05T23:30:31.687+0200 I CONTROL [initandlisten]     target_arch: x86_64
2019-10-05T23:30:31.687+0200 I CONTROL [initandlisten] options: { net: { port:
27013 }, replication: { replSet: "small-movie" }, storage: { dbPath: "./mongo3" }
}
2019-10-05T23:30:31.693+0200 I STORAGE [initandlisten] wiredtiger_open config:
create,cache_size=257478M,cache_overflow=(file_max=0M),session_max=20000,eviction=
(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=
(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=
(close_idle_time=100000),statistics_log=(wait=0),verbose=(recovery_progress),
2019-10-05T23:30:34.183+0200 I STORAGE [initandlisten] WiredTiger message
[1570311034:183644][81484:0x7fea58f13000], txn-recover: Set global recovery
timestamp: 0
2019-10-05T23:30:34.292+0200 I RECOVERY [initandlisten] WiredTiger
recoveryTimestamp. Ts: Timestamp(0, 0)
2019-10-05T23:30:37.079+0200 I CONTROL [initandlisten]
2019-10-05T23:30:37.079+0200 I CONTROL [initandlisten] ** WARNING: Access control
is not enabled for the database.
2019-10-05T23:30:37.079+0200 I CONTROL [initandlisten] **           Read and write
access to data and configuration is unrestricted.
2019-10-05T23:30:37.079+0200 I CONTROL [initandlisten]
2019-10-05T23:30:37.079+0200 I CONTROL [initandlisten] ** WARNING: This server is
bound to localhost.
2019-10-05T23:30:37.079+0200 I CONTROL [initandlisten] **           Remote systems
will be unable to connect to this server.
2019-10-05T23:30:37.079+0200 I CONTROL [initandlisten] **           Start the
server with --bind_ip <address> to specify which IP
2019-10-05T23:30:37.079+0200 I CONTROL [initandlisten] **           addresses it
should serve responses from, or with --bind_ip_all to
2019-10-05T23:30:37.079+0200 I CONTROL [initandlisten] **           bind to all
interfaces. If this behavior is desired, start the
2019-10-05T23:30:37.079+0200 I CONTROL [initandlisten] **           server with --
bind_ip 127.0.0.1 to disable this warning.
2019-10-05T23:30:37.079+0200 I CONTROL [initandlisten]
2019-10-05T23:30:37.082+0200 I CONTROL [initandlisten]
2019-10-05T23:30:37.082+0200 I CONTROL [initandlisten] ** WARNING: You are
running on a NUMA machine.
2019-10-05T23:30:37.082+0200 I CONTROL [initandlisten] **           We suggest
launching mongod like this to avoid performance problems:
2019-10-05T23:30:37.082+0200 I CONTROL [initandlisten] **           numactl --
interleave=all mongod [other options]
2019-10-05T23:30:37.083+0200 I CONTROL [initandlisten]
2019-10-05T23:30:37.088+0200 I STORAGE [initandlisten] createCollection:
local.startup_log with generated UUID: 1475105d-dec1-4184-a814-e4dfd485e794
2019-10-05T23:30:37.302+0200 I FTDC [initandlisten] Initializing full-time
diagnostic data capture with directory './mongo3/diagnostic.data'
2019-10-05T23:30:37.304+0200 I STORAGE [initandlisten] createCollection:
local.replset.oplogTruncateAfterPoint with generated UUID: 044e8993-4de1-4138-
8d1a-a1567224ec46

```

```

2019-10-05T23:30:37.512+0200 I STORAGE [initandlisten] createCollection:
local.replset.minvalid with generated UUID: 859f5b94-e923-483a-8e2e-e26c5b660a01
2019-10-05T23:30:37.722+0200 I STORAGE [initandlisten] createCollection:
local.replset.election with generated UUID: 17c6e322-b333-4bab-80ed-8e9c5678f408
2019-10-05T23:30:37.952+0200 I REPL [initandlisten] Did not find local
initialized voted for document at startup.
2019-10-05T23:30:37.952+0200 I REPL [initandlisten] Did not find local
Rollback ID document at startup. Creating one.
2019-10-05T23:30:37.952+0200 I STORAGE [initandlisten] createCollection:
local.system.rollback.id with generated UUID: 4c392043-132e-4296-8c03-3db186e34c0c
2019-10-05T23:30:38.162+0200 I REPL [initandlisten] Initialized the rollback
ID to 1
2019-10-05T23:30:38.162+0200 I REPL [initandlisten] Did not find local replica
set configuration document at startup; NoMatchingDocument: Did not find replica
set configuration document in local.system.replset
2019-10-05T23:30:38.163+0200 I CONTROL [LogicalSessionCacheRefresh] Sessions
collection is not set up; waiting until next sessions refresh interval:
Replication has not yet been configured
2019-10-05T23:30:38.163+0200 I NETWORK [initandlisten] waiting for connections on
port 27013
2019-10-05T23:30:38.163+0200 I CONTROL [LogicalSessionCacheReap] Sessions
collection is not set up; waiting until next sessions reap interval:
config.system.sessions does not exist

```

4. Connect a client (mongo) to one server. Through the client, initialize the replication: add one other server as secondary, and add the third one as arbiter.

```

$ mongo localhost:27011
MongoDB shell version v4.0.12
connecting to: mongod://localhost:27011/test?gssapiServiceName=mongod
Implicit session: session { "id" : UUID("41e06f06-911b-4d07-9b5a-fecfc802e6fa") }
MongoDB server version: 4.0.12
Server has startup warnings:
2019-10-05T23:18:37.058+0200 I CONTROL [initandlisten]
2019-10-05T23:18:37.058+0200 I CONTROL [initandlisten] ** WARNING: Access control
is not enabled for the database.
2019-10-05T23:18:37.058+0200 I CONTROL [initandlisten] **           Read and write
access to data and configuration is unrestricted.
2019-10-05T23:18:37.058+0200 I CONTROL [initandlisten]
2019-10-05T23:18:37.058+0200 I CONTROL [initandlisten] ** WARNING: This server is
bound to localhost.
2019-10-05T23:18:37.058+0200 I CONTROL [initandlisten] **           Remote systems
will be unable to connect to this server.
2019-10-05T23:18:37.058+0200 I CONTROL [initandlisten] **           Start the
server with --bind_ip <address> to specify which IP
2019-10-05T23:18:37.058+0200 I CONTROL [initandlisten] **           addresses it
should serve responses from, or with --bind_ip_all to
2019-10-05T23:18:37.058+0200 I CONTROL [initandlisten] **           bind to all
interfaces. If this behavior is desired, start the
2019-10-05T23:18:37.058+0200 I CONTROL [initandlisten] **           server with --
bind_ip 127.0.0.1 to disable this warning.

```

```

2019-10-05T23:18:37.058+0200 I CONTROL [initandlisten]
2019-10-05T23:18:37.061+0200 I CONTROL [initandlisten]
2019-10-05T23:18:37.061+0200 I CONTROL [initandlisten] ** WARNING: You are
running on a NUMA machine.
2019-10-05T23:18:37.061+0200 I CONTROL [initandlisten] **           We suggest
launching mongod like this to avoid performance problems:
2019-10-05T23:18:37.061+0200 I CONTROL [initandlisten] **           numactl --
interleave=all mongod [other options]
2019-10-05T23:18:37.061+0200 I CONTROL [initandlisten]

> rs.initiate({_id: 'small-movie',
... members: [ {_id: 1, host: 'localhost:27011'},
... {_id: 2, host: 'localhost:27012'} ] })
{
  "ok" : 1,
  "operationTime" : Timestamp(1570312541, 1),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1570312541, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}
small-movie:SECONDARY> rs.addArb("localhost:27013")
{
  "ok" : 1,
  "operationTime" : Timestamp(1570312554, 4),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1570312554, 4),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}
small-movie:PRIMARY>

```

5. Identify the master from the outputs of the servers' and by requesting replica set information from the servers.

Output of server 3:

```

2019-10-05T23:55:54.749+0200 I REPL [replexec-0] This node is localhost:27013
in the config
2019-10-05T23:55:54.749+0200 I REPL [replexec-0] transition to ARBITER from
STARTUP
2019-10-05T23:55:54.750+0200 I REPL [replexec-0] Member localhost:27011 is now
in state PRIMARY
2019-10-05T23:55:54.750+0200 I REPL [replexec-0] Member localhost:27012 is now
in state SECONDARY

```

We can see that the **server 1** is the master, and we verify with requesting replica information from *localhost:27011*, alias *server 1*:

```
small-movie:PRIMARY> db.isMaster()
{
  "hosts" : [
    "localhost:27011",
    "localhost:27012"
  ],
  "arbiters" : [
    "localhost:27013"
  ],
  "setName" : "small-movie",
  "setVersion" : 2,
  "ismaster" : true,
  "secondary" : false,
  "primary" : "localhost:27011",
  "me" : "localhost:27011",
  "electionId" : ObjectId("7fffffff0000000000000001"),
  "lastWrite" : {
    "opTime" : {
      "ts" : Timestamp(1570313094, 1),
      "t" : NumberLong(1)
    },
    "lastWriteDate" : ISODate("2019-10-05T22:04:54Z"),
    "majorityOpTime" : {
      "ts" : Timestamp(1570313094, 1),
      "t" : NumberLong(1)
    },
    "majorityWriteDate" : ISODate("2019-10-05T22:04:54Z")
  },
  "maxBsonObjectSize" : 16777216,
  "maxMessageSizeBytes" : 48000000,
  "maxWriteBatchSize" : 100000,
  "localTime" : ISODate("2019-10-05T22:04:56.015Z"),
  "logicalSessionTimeoutMinutes" : 30,
  "minWireVersion" : 0,
  "maxWireVersion" : 7,
  "readOnly" : false,
  "ok" : 1,
  "operationTime" : Timestamp(1570313094, 1),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1570313094, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}
```

We can see from the output, the value `ismaster` is true.

6. Import `moviepeople-1000.jsonl` through the master. Observe the output of the two other servers.

Firstly we import the file to PRIMARY (master)

```
$ mongoimport --file lab_2_datasets/moviepeople-1000.jsonl --host=localhost:27011
--collection small-movie
2019-10-06T00:21:39.754+0200    connected to: localhost:27011
2019-10-06T00:21:40.853+0200    imported 1000 documents
```

Server 1 Primary

```
2019-10-06T00:21:39.753+0200 I NETWORK [listener] connection accepted from
127.0.0.1:37940 #24 (9 connections now open)
2019-10-06T00:21:39.954+0200 I STORAGE [conn24] createCollection: test.small-
movie with generated UUID: 14444f21-8c99-4ed8-a577-68c1ddf74abf
2019-10-06T00:21:40.851+0200 I COMMAND [conn24] command test.small-movie command:
insert { insert: "small-movie", writeConcern: { getLastError: 1, w: "majority" },
ordered: false, $db: "test" } ninserted:1000 keysInserted:1000 numYields:0
reslen:245 locks:{ Global: { acquireCount: { r: 19, w: 19 } }, Database: {
acquireCount: { w: 18, W: 1 } }, Collection: { acquireCount: { w: 18 } } }
storage:{} protocol:op_query 898ms
```

Server 2 SECONDARY

```
2019-10-06T00:21:40.313+0200 I STORAGE [repl writer worker 0] createCollection:
test.small-movie with provided UUID: 14444f21-8c99-4ed8-a577-68c1ddf74abf
2019-10-06T00:21:40.492+0200 I REPL [repl writer worker 0] applied op: command
{ ts: Timestamp(1570314099, 1), t: 1, h: 5483686551903084631, v: 2, op: "c", ns:
"test.$cmd", ui: UUID("14444f21-8c99-4ed8-a577-68c1ddf74abf"), wall: new
Date(1570314100303), o: { create: "small-movie", idIndex: { v: 2, key: { _id: 1 },
name: "_id_", ns: "test.small-movie" } } }, took 180ms
```

We can see that the SECONDARY copied the data set right after it has been created in the PRIMARY.

Server 3 ARBITER

```
2019-10-06T00:20:05.568+0200 I NETWORK [LogicalSessionCacheRefresh] Successfully
connected to localhost:27011 (1 connections now open to localhost:27011 with a 0
second timeout)
2019-10-06T00:20:05.568+0200 I CONTROL [LogicalSessionCacheRefresh] Sessions
collection is not set up; waiting until next sessions refresh interval:
```

```

config.system.sessions does not exist
2019-10-06T00:25:05.566+0200 I NETWORK [LogicalSessionCacheRefresh] Starting new
replica set monitor for small-movie/localhost:27011,localhost:27012
2019-10-06T00:25:05.566+0200 I CONTROL [LogicalSessionCacheReap] Sessions
collection is not set up; waiting until next sessions reap interval:
config.system.sessions does not exist

```

We can see that during this time, there is no single special output in **ARBITER**, which means the replication doesn't take place here.

7. Once the synchronization is finished, stop (ctrl-c) the master. Observe the output of the two other servers.

Before we violently stop the **master**, I've checked the config by

```

> rs.conf()
{
  "_id" : "small-movie",
  "version" : 2,
  "protocolVersion" : NumberLong(1),
  "writeConcernMajorityJournalDefault" : true,
  "members" : [
    {
      "_id" : 1,
      "host" : "localhost:27011",
      "arbiterOnly" : false,
      "buildIndexes" : true,
      "hidden" : false,
      "priority" : 1,
      "tags" : {

      },
      "slaveDelay" : NumberLong(0),
      "votes" : 1
    },
    {
      "_id" : 2,
      "host" : "localhost:27012",
      "arbiterOnly" : false,
      "buildIndexes" : true,
      "hidden" : false,
      "priority" : 1,
      "tags" : {

      },
      "slaveDelay" : NumberLong(0),
      "votes" : 1
    },
    {
      "_id" : 3,
      "host" : "localhost:27013",

```



```

        "arbiterOnly" : true,
        "buildIndexes" : true,
        "hidden" : false,
        "priority" : 0,
        "tags" : {

        },
        "slaveDelay" : NumberLong(0),
        "votes" : 1
    },
    ],
    "settings" : {
        "chainingAllowed" : true,
        "heartbeatIntervalMillis" : 2000,
        "heartbeatTimeoutSecs" : 10,
        "electionTimeoutMillis" : 10000,
        "catchUpTimeoutMillis" : -1,
        "catchUpTakeoverDelayMillis" : 30000,
        "getLastErrorModes" : {

        },
        "getLastErrorDefaults" : {
            "w" : 1,
            "wtimeout" : 0
        },
        "replicaSetId" : ObjectId("5d99115d77d9e6648402a3e4")
    }
}

```

So from the `Settings` section, we can see `"heartbeatTimeoutSecs" : 10`, `"electionTimeoutMillis" : 10000`, so if we directly stop the *master*, the terminals will be flushed with at least 1000 error messages before the election, so before reset the `"electionTimeoutMillis"`, I've tried with `rs.stepDown()` to call the election immediately.

```

small-movie:PRIMARY> rs.stepDown()
2019-10-06T01:02:49.558+0200 I NETWORK [js] DBClientConnection failed to receive
message from localhost:27011 - HostUnreachable: Connection closed by peer
2019-10-06T01:02:49.559+0200 E QUERY [js] Error: error doing query: failed:
network error while attempting to run command 'replSetStepDown' on host
'localhost:27011' :
DB.prototype.runCommand@src/mongo/shell/db.js:168:1
DB.prototype.adminCommand@src/mongo/shell/db.js:186:16
rs.stepDown@src/mongo/shell/utils.js:1444:12
@(shell):1:1
2019-10-06T01:02:49.561+0200 I NETWORK [js] trying reconnect to localhost:27011
failed
2019-10-06T01:02:49.562+0200 I NETWORK [js] reconnect localhost:27011 ok
small-movie:SECONDARY>

```

```

2019-10-06T01:02:49.557+0200 I COMMAND [conn11] Attempting to step down in
response to replSetStepDown command
2019-10-06T01:02:49.557+0200 I REPL [conn11] transition to SECONDARY from
PRIMARY
2019-10-06T01:02:49.558+0200 I NETWORK [conn11] Skip closing connection for
connection # 10
2019-10-06T01:02:49.558+0200 I NETWORK [conn11] Skip closing connection for
connection # 9
2019-10-06T01:02:49.558+0200 I NETWORK [conn11] Skip closing connection for
connection # 5
2019-10-06T01:02:49.558+0200 I NETWORK [conn11] Skip closing connection for
connection # 2
2019-10-06T01:02:49.558+0200 I NETWORK [conn12] end connection 127.0.0.1:41476 (7
connections now open)
2019-10-06T01:02:49.558+0200 I NETWORK [conn14] end connection 127.0.0.1:41482 (6
connections now open)
2019-10-06T01:02:49.558+0200 I NETWORK [conn13] end connection 127.0.0.1:41480 (5
connections now open)
2019-10-06T01:02:49.558+0200 I REPL [conn11] Handing off election to
localhost:27012
2019-10-06T01:02:49.559+0200 I NETWORK [conn11] Error sending response to client:
SocketException: Broken pipe. Ending connection from 127.0.0.1:41460 (connection
id: 11)
2019-10-06T01:02:49.559+0200 I NETWORK [conn11] end connection 127.0.0.1:41460 (4
connections now open)
2019-10-06T01:02:49.561+0200 I NETWORK [listener] connection accepted from
127.0.0.1:41618 #18 (5 connections now open)
2019-10-06T01:02:49.562+0200 I NETWORK [conn18] received client metadata from
127.0.0.1:41618 conn18: { application: { name: "MongoDB Shell" }, driver: { name:
"MongoDB Internal Client", version: "4.0.12" }, os: { type: "Linux", name: "Fedora
release 30 (Thirty)", architecture: "x86_64", version: "Kernel 5.0.10-
200.fc29.x86_64" } }
2019-10-06T01:02:50.529+0200 I REPL [replexec-0] Member localhost:27012 is now
in state PRIMARY
2019-10-06T01:02:51.292+0200 I REPL [rsBackgroundSync] sync source candidate:
localhost:27012
2019-10-06T01:02:51.292+0200 I ASIO [RS] Connecting to localhost:27012
2019-10-06T01:02:51.295+0200 I REPL [rsBackgroundSync] Changed sync source
from empty to localhost:27012

```

from server 2

```

2019-10-06T01:02:49.559+0200 I COMMAND [conn3] Received replSetStepUp request
2019-10-06T01:02:49.559+0200 I REPL [conn3] Starting an election due to step
up request
2019-10-06T01:02:49.559+0200 I REPL [conn3] skipping dry run and running for
election in term 6
2019-10-06T01:02:49.562+0200 I REPL [replexec-1] VoteRequester(term 6)
received a no vote from localhost:27013 with reason "can see a healthy primary
(localhost:27011) of equal or greater priority"; response message: { term: 6,

```

```

voteGranted: false, reason: "can see a healthy primary (localhost:27011) of equal
or greater priority", ok: 1.0 }
2019-10-06T01:02:49.564+0200 I REPL      [replexec-12] VoteRequester(term 6)
received a yes vote from localhost:27011; response message: { term: 6,
voteGranted: true, reason: "", ok: 1.0, operationTime: Timestamp(1570316565, 1),
$clusterTime: { clusterTime: Timestamp(1570316565, 1), signature: { hash:
BinData(0, 0000000000000000000000000000000000000000000000000000000000000000), keyId: 0 } } }
2019-10-06T01:02:49.564+0200 I REPL      [replexec-12] election succeeded, assuming
primary role in term 6
2019-10-06T01:02:49.564+0200 I REPL      [replexec-12] transition to PRIMARY from
SECONDARY
2019-10-06T01:02:49.564+0200 I REPL      [replexec-12] Resetting sync source to
empty, which was localhost:27011
2019-10-06T01:02:49.564+0200 I REPL      [replexec-12] Entering primary catch-up
mode.
2019-10-06T01:02:49.565+0200 I REPL      [replexec-16] Member localhost:27011 is
now in state SECONDARY
2019-10-06T01:02:49.565+0200 I REPL      [replexec-16] Caught up to the latest
optime known via heartbeats after becoming primary. Target optime: { ts:
Timestamp(1570316565, 1), t: 5 }. My Last Applied: { ts: Timestamp(1570316565, 1),
t: 5 }
2019-10-06T01:02:49.565+0200 I REPL      [replexec-16] Exited primary catch-up
mode.
2019-10-06T01:02:49.565+0200 I REPL      [replexec-16] Stopping replication
producer
2019-10-06T01:02:49.566+0200 I REPL      [rsBackgroundSync] Replication producer
stopped after oplog fetcher finished returning a batch from our sync source.
Abandoning this batch of oplog entries and re-evaluating our sync source.
2019-10-06T01:02:50.697+0200 I REPL      [rsSync-0] transition to primary complete;
database writes are now permitted

```

from server 3

```

2019-10-06T01:02:50.479+0200 I REPL      [replexec-0] Member localhost:27012 is now
in state PRIMARY
2019-10-06T01:02:50.479+0200 I REPL      [replexec-1] Member localhost:27011 is now
in state SECONDARY

```

We can see that after server 1 steps down, server 2 is selected as PRIMARY after. So Now we try to shut down server 1 by **Ctrl+C**, so between numerous error messages as below on both servers:

```

2019-10-06T13:07:15.192+0200 I ASIO      [Replication] Failed to connect to
localhost:27011 - HostUnreachable: Error connecting to localhost:27011
(127.0.0.1:27011) :: caused by :: Connection refused
2019-10-06T13:07:15.192+0200 I CONNPOOL [Replication] Dropping all pooled
connections to localhost:27011 due to HostUnreachable: Error connecting to
localhost:27011 (127.0.0.1:27011) :: caused by :: Connection refused
2019-10-06T13:07:15.194+0200 I ASIO      [Replication] Connecting to
localhost:27011

```

```

2019-10-06T13:07:15.195+0200 I ASIO      [Replication] Failed to connect to
localhost:27011 - HostUnreachable: Error connecting to localhost:27011
(127.0.0.1:27011) :: caused by :: Connection refused
2019-10-06T13:07:15.195+0200 I CONNPOOL [Replication] Dropping all pooled
connections to localhost:27011 due to HostUnreachable: Error connecting to
localhost:27011 (127.0.0.1:27011) :: caused by :: Connection refused

```

From **SERVER 2**, alias **SECONDARY** we can find

```

2019-10-06T13:07:15.191+0200 I REPL      [replexec-5] Starting an election, since
we've seen no PRIMARY in the past 10000ms
2019-10-06T13:07:15.191+0200 I REPL      [replexec-5] conducting a dry run election
to see if we could be elected. current term: 2
2019-10-06T13:07:15.191+0200 I ASIO      [Replication] Connecting to
localhost:27011
2019-10-06T13:07:15.192+0200 I REPL      [replexec-4] VoteRequester(term 2 dry run)
received a yes vote from localhost:27013; response message: { term: 2,
voteGranted: true, reason: "", ok: 1.0 }
2019-10-06T13:07:15.192+0200 I REPL      [replexec-4] dry election run succeeded,
running for election in term 3
2019-10-06T13:07:15.195+0200 I REPL      [replexec-1] VoteRequester(term 3) failed
to receive response from localhost:27011: HostUnreachable: Error connecting to
localhost:27011 (127.0.0.1:27011) :: caused by :: Connection refused
2019-10-06T13:07:15.206+0200 I REPL      [replexec-6] VoteRequester(term 3)
received a yes vote from localhost:27013; response message: { term: 3,
voteGranted: true, reason: "", ok: 1.0 }
2019-10-06T13:07:15.206+0200 I REPL      [replexec-6] election succeeded, assuming
primary role in term 3
2019-10-06T13:07:15.206+0200 I REPL      [replexec-6] transition to PRIMARY from
SECONDARY
2019-10-06T13:07:15.206+0200 I REPL      [replexec-6] Resetting sync source to
empty, which was :27017
2019-10-06T13:07:15.206+0200 I REPL      [replexec-6] Entering primary catch-up
mode.
2019-10-06T13:07:15.210+0200 I REPL      [replexec-7] Caught up to the latest
optime known via heartbeats after becoming primary. Target optime: { ts:
Timestamp(1570360013, 1000), t: 1 }. My Last Applied: { ts: Timestamp(1570360013,
1000), t: 1 }
2019-10-06T13:07:15.210+0200 I REPL      [replexec-7] Exited primary catch-up mode.
2019-10-06T13:07:15.210+0200 I REPL      [replexec-7] Stopping replication producer
2019-10-06T13:07:16.400+0200 I REPL      [rsSync-0] transition to primary complete;
database writes are now permitted

```

and from **SERVER 3**, alias **ARBITER**:

```

2019-10-06T13:07:15.775+0200 I REPL      [replexec-1] Member localhost:27012 is now
in state PRIMARY

```

I think now it's better to use `rs.stepDown()` to test the election mechanism, for the **error messages do not stop** after the new PRIMARY has been elected.

Task 3: sharding

Task 3 is run under MongoDB version v2.6.12

1. Start two shard servers.

```
$ mkdir ./mongo4 ./mongo5
```

shard server 1

```
$ mongod --shardsvr --dbpath ./mongo4 --port 27014 --smallfiles
2019-10-06T14:59:41.669+0200 [initandlisten] MongoDB starting : pid=39532
port=27014 dbpath=./mongo4 64-bit host=lame13.enst.fr
2019-10-06T14:59:41.669+0200 [initandlisten]
2019-10-06T14:59:41.669+0200 [initandlisten] ** WARNING: You are running on a NUMA
machine.
2019-10-06T14:59:41.669+0200 [initandlisten] **           We suggest launching
mongod like this to avoid performance problems:
2019-10-06T14:59:41.669+0200 [initandlisten] **           numactl --
interleave=all mongod [other options]
2019-10-06T14:59:41.670+0200 [initandlisten]
2019-10-06T14:59:41.670+0200 [initandlisten] db version v2.6.12
2019-10-06T14:59:41.670+0200 [initandlisten] git version:
d73c92b1c85703828b55c2916a5dd4ad46535f6a
2019-10-06T14:59:41.670+0200 [initandlisten] build info: Linux
build5.ny.cbi.10gen.cc 2.6.32-431.3.1.el6.x86_64 #1 SMP Fri Jan 3 21:39:27 UTC
2014 x86_64 BOOST_LIB_VERSION=1_49
2019-10-06T14:59:41.670+0200 [initandlisten] allocator: tcmalloc
2019-10-06T14:59:41.670+0200 [initandlisten] options: { net: { port: 27014 },
sharding: { clusterRole: "shardsvr" }, storage: { dbPath: "./mongo4", smallFiles:
true } }
2019-10-06T14:59:41.772+0200 [initandlisten] journal dir=./mongo4/journal
2019-10-06T14:59:41.772+0200 [initandlisten] recover : no journal files present,
no recovery needed
2019-10-06T14:59:41.870+0200 [initandlisten] allocating new ns file
./mongo4/local.ns, filling with zeroes...
2019-10-06T14:59:42.119+0200 [FileAllocator] allocating new datafile
./mongo4/local.0, filling with zeroes...
2019-10-06T14:59:42.120+0200 [FileAllocator] creating directory ./mongo4/_tmp
2019-10-06T14:59:42.127+0200 [FileAllocator] done allocating datafile
./mongo4/local.0, size: 16MB, took 0.003 secs
2019-10-06T14:59:42.220+0200 [initandlisten] build index on: local.startup_log
properties: { v: 1, key: { _id: 1 }, name: "_id_", ns: "local.startup_log" }
2019-10-06T14:59:42.220+0200 [initandlisten] added index to empty collection
2019-10-06T14:59:42.220+0200 [initandlisten] command local.$cmd command: create {
create: "startup_log", size: 10485760, capped: true } ntoreturn:1 keyUpdates:0
```

```
numYields:0 reslen:37 350ms
2019-10-06T14:59:42.221+0200 [initandlisten] waiting for connections on port 27014
```

shard server 2

```
mongod --shardsvr --dbpath ./mongo5 --port 27015 --smallfiles
2019-10-06T15:00:05.458+0200 [initandlisten] MongoDB starting : pid=39693
port=27015 dbpath=./mongo5 64-bit host=lame13.enst.fr
2019-10-06T15:00:05.459+0200 [initandlisten]
2019-10-06T15:00:05.459+0200 [initandlisten] ** WARNING: You are running on a NUMA
machine.
2019-10-06T15:00:05.459+0200 [initandlisten] **           We suggest launching
mongod like this to avoid performance problems:
2019-10-06T15:00:05.459+0200 [initandlisten] **           numactl --
interleave=all mongod [other options]
2019-10-06T15:00:05.459+0200 [initandlisten]
2019-10-06T15:00:05.459+0200 [initandlisten] db version v2.6.12
2019-10-06T15:00:05.459+0200 [initandlisten] git version:
d73c92b1c85703828b55c2916a5dd4ad46535f6a
2019-10-06T15:00:05.459+0200 [initandlisten] build info: Linux
build5.ny.cbi.10gen.cc 2.6.32-431.3.1.el6.x86_64 #1 SMP Fri Jan 3 21:39:27 UTC
2014 x86_64 BOOST_LIB_VERSION=1_49
2019-10-06T15:00:05.459+0200 [initandlisten] allocator: tcmalloc
2019-10-06T15:00:05.459+0200 [initandlisten] options: { net: { port: 27015 },
sharding: { clusterRole: "shardsvr" }, storage: { dbPath: "./mongo5", smallFiles:
true } }
2019-10-06T15:00:05.563+0200 [initandlisten] journal dir=./mongo5/journal
2019-10-06T15:00:05.566+0200 [initandlisten] recover : no journal files present,
no recovery needed
2019-10-06T15:00:05.662+0200 [initandlisten] allocating new ns file
./mongo5/local.ns, filling with zeroes...
2019-10-06T15:00:05.871+0200 [FileAllocator] allocating new datafile
./mongo5/local.0, filling with zeroes...
2019-10-06T15:00:05.872+0200 [FileAllocator] creating directory ./mongo5/_tmp
2019-10-06T15:00:05.877+0200 [FileAllocator] done allocating datafile
./mongo5/local.0, size: 16MB, took 0.003 secs
2019-10-06T15:00:06.006+0200 [initandlisten] build index on: local.startup_log
properties: { v: 1, key: { _id: 1 }, name: "_id_", ns: "local.startup_log" }
2019-10-06T15:00:06.006+0200 [initandlisten] added index to empty collection
2019-10-06T15:00:06.006+0200 [initandlisten] command local.$cmd command: create {
create: "startup_log", size: 10485760, capped: true } ntoreturn:1 keyUpdates:0
numYields:0 reslen:37 344ms
2019-10-06T15:00:06.007+0200 [initandlisten] waiting for connections on port 27015
```

shard config server

```
$ mkdir ./mongoconfig
$ mongod --configsvr --dbpath ./mongoconfig --port 27016
```



```

2019-10-06T14:46:32.984+0200 I STORAGE [main] Max cache overflow file size custom
option: 0
2019-10-06T14:46:32.987+0200 I CONTROL [main] Automatically disabling TLS 1.0, to
force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
2019-10-06T14:46:32.991+0200 I CONTROL [initandlisten] MongoDB starting :
pid=45965 port=27016 dbpath=./mongoconfig 64-bit host=lame10.enst.fr
2019-10-06T14:46:32.991+0200 I CONTROL [initandlisten] db version v4.0.12
2019-10-06T14:46:32.991+0200 I CONTROL [initandlisten] git version:
5776e3cbf9e7afe86e6b29e22520ffb6766e95d4
2019-10-06T14:46:32.991+0200 I CONTROL [initandlisten] OpenSSL version: OpenSSL
1.0.0-fips 29 Mar 2010
2019-10-06T14:46:32.991+0200 I CONTROL [initandlisten] allocator: tcmalloc
2019-10-06T14:46:32.991+0200 I CONTROL [initandlisten] modules: none
2019-10-06T14:46:32.991+0200 I CONTROL [initandlisten] build environment:
2019-10-06T14:46:32.991+0200 I CONTROL [initandlisten] distmod: amazon
2019-10-06T14:46:32.991+0200 I CONTROL [initandlisten] distarch: x86_64
2019-10-06T14:46:32.991+0200 I CONTROL [initandlisten] target_arch: x86_64
2019-10-06T14:46:32.991+0200 I CONTROL [initandlisten] options: { net: { port:
27016 }, sharding: { clusterRole: "configsvr" }, storage: { dbPath:
"./mongoconfig" } }
2019-10-06T14:46:32.998+0200 I STORAGE [initandlisten] wiredtiger_open config:
create,cache_size=257478M,cache_overflow=(file_max=0M),session_max=20000,eviction=
(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=
(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=
(close_idle_time=100000),statistics_log=(wait=0),verbose=(recovery_progress),
2019-10-06T14:46:35.098+0200 I STORAGE [initandlisten] WiredTiger message
[1570365995:98320][45965:0x7f8e9124c000], txn-recover: Set global recovery
timestamp: 0
2019-10-06T14:46:35.209+0200 I RECOVERY [initandlisten] WiredTiger
recoveryTimestamp. Ts: Timestamp(0, 0)
2019-10-06T14:46:38.074+0200 I CONTROL [initandlisten]
2019-10-06T14:46:38.074+0200 I CONTROL [initandlisten] ** WARNING: Access control
is not enabled for the database.
2019-10-06T14:46:38.074+0200 I CONTROL [initandlisten] ** Read and write
access to data and configuration is unrestricted.
2019-10-06T14:46:38.074+0200 I CONTROL [initandlisten]
2019-10-06T14:46:38.074+0200 I CONTROL [initandlisten] ** WARNING: This server is
bound to localhost.
2019-10-06T14:46:38.074+0200 I CONTROL [initandlisten] ** Remote systems
will be unable to connect to this server.
2019-10-06T14:46:38.074+0200 I CONTROL [initandlisten] ** Start the
server with --bind_ip <address> to specify which IP
2019-10-06T14:46:38.074+0200 I CONTROL [initandlisten] ** addresses it
should serve responses from, or with --bind_ip_all to
2019-10-06T14:46:38.074+0200 I CONTROL [initandlisten] ** bind to all
interfaces. If this behavior is desired, start the
2019-10-06T14:46:38.074+0200 I CONTROL [initandlisten] ** server with --
bind_ip 127.0.0.1 to disable this warning.
2019-10-06T14:46:38.074+0200 I CONTROL [initandlisten]
2019-10-06T14:46:38.077+0200 I CONTROL [initandlisten]
2019-10-06T14:46:38.077+0200 I CONTROL [initandlisten] ** WARNING: You are
running on a NUMA machine.
2019-10-06T14:46:38.077+0200 I CONTROL [initandlisten] ** We suggest
launching mongod like this to avoid performance problems:

```

```

2019-10-06T14:46:38.077+0200 I CONTROL [initandlisten] **                               numactl --
interleave=all mongod [other options]
2019-10-06T14:46:38.077+0200 I CONTROL [initandlisten]
2019-10-06T14:46:38.078+0200 I STORAGE [initandlisten] createCollection:
admin.system.version with provided UUID: abcfe4ac-16ca-4f3b-b9f9-5c492fb687fc
2019-10-06T14:46:38.298+0200 I COMMAND [initandlisten] setting
featureCompatibilityVersion to 4.0
2019-10-06T14:46:38.305+0200 I STORAGE [initandlisten] createCollection:
local.startup_log with generated UUID: 2fa6b2ca-90f0-49a7-a89b-720248257319
2019-10-06T14:46:38.518+0200 I FTDC [initandlisten] Initializing full-time
diagnostic data capture with directory './mongoconfig/diagnostic.data'
2019-10-06T14:46:38.526+0200 I SHARDING [thread1] creating distributed lock ping
thread for process ConfigServer (sleeping for 30000ms)
2019-10-06T14:46:38.528+0200 I STORAGE [replSetDistLockPinger] createCollection:
config.lockpings with generated UUID: 0da45068-131c-4f85-b77e-059a604db3fb
2019-10-06T14:46:38.529+0200 I SH_REFR [ConfigServerCatalogCacheLoader-0] Refresh
for database config took 0 ms and found { _id: "config", primary: "config",
partitioned: true }
2019-10-06T14:46:38.529+0200 I NETWORK [initandlisten] waiting for connections on
port 27016

```

starting mongos connected to the config

```

mongos --configdb localhost:27016 --chunkSize 1 --port 27020
2019-10-06T15:00:39.480+0200 warning: running with 1 config server should be done
only for testing purposes and is not recommended for production
2019-10-06T15:00:39.523+0200 [mongosMain] MongoDB version 2.6.12 starting:
pid=39838 port=27020 64-bit host=lame13.enst.fr (--help for usage)
2019-10-06T15:00:39.523+0200 [mongosMain] db version v2.6.12
2019-10-06T15:00:39.523+0200 [mongosMain] git version:
d73c92b1c85703828b55c2916a5dd4ad46535f6a
2019-10-06T15:00:39.523+0200 [mongosMain] build info: Linux build5.ny.cbi.10gen.cc
2.6.32-431.3.1.el6.x86_64 #1 SMP Fri Jan 3 21:39:27 UTC 2014 x86_64
BOOST_LIB_VERSION=1_49
2019-10-06T15:00:39.523+0200 [mongosMain] allocator: tcmalloc
2019-10-06T15:00:39.524+0200 [mongosMain] options: { net: { port: 27020 },
sharding: { chunkSize: 1, configDB: "localhost:27016" } }
2019-10-06T15:00:39.535+0200 [LockPinger] creating distributed lock ping thread
for localhost:27016 and process lame13.enst.fr:27020:1570366839:1804289383
(sleeping for 30000ms)
2019-10-06T15:00:39.766+0200 [LockPinger] cluster localhost:27016 pinged
successfully at Sun Oct 6 15:00:39 2019 by distributed lock pinger
'localhost:27016/lame13.enst.fr:27020:1570366839:1804289383', sleeping for 30000ms
2019-10-06T15:00:39.891+0200 [mongosMain] distributed lock
'configUpgrade/lame13.enst.fr:27020:1570366839:1804289383' acquired, ts :
5d99e5778f4a75f6f53558fa
2019-10-06T15:00:39.892+0200 [mongosMain] starting upgrade of config server from
v0 to v5
2019-10-06T15:00:39.893+0200 [mongosMain] starting next upgrade step from v0 to v5
2019-10-06T15:00:39.893+0200 [mongosMain] about to log new metadata event: { _id:
"lame13.enst.fr-2019-10-06T13:00:39-5d99e5778f4a75f6f53558fb", server:

```

```

"lame13.enst.fr", clientAddr: "N/A", time: new Date(1570366839893), what:
"starting upgrade of config database", ns: "config.version", details: { from: 0,
to: 5 } }
2019-10-06T15:00:39.894+0200 [mongosMain] creating WriteBackListener for:
localhost:27016 serverID: 00000000000000000000000000000000
2019-10-06T15:00:39.906+0200 [mongosMain] writing initial config version at v5
2019-10-06T15:00:39.914+0200 [mongosMain] about to log new metadata event: { _id:
"lame13.enst.fr-2019-10-06T13:00:39-5d99e5778f4a75f6f53558fd", server:
"lame13.enst.fr", clientAddr: "N/A", time: new Date(1570366839914), what:
"finished upgrade of config database", ns: "config.version", details: { from: 0,
to: 5 } } }
2019-10-06T15:00:39.921+0200 [mongosMain] upgrade of config server to v5
successful
2019-10-06T15:00:39.922+0200 [mongosMain] distributed lock
'configUpgrade/lame13.enst.fr:27020:1570366839:1804289383' unlocked.
2019-10-06T15:00:39.973+0200 [mongosMain] scoped connection to localhost:27016 not
being returned to the pool
2019-10-06T15:00:39.973+0200 [Balancer] about to contact config servers and shards
2019-10-06T15:00:39.974+0200 [mongosMain] waiting for connections on port 27020
2019-10-06T15:00:39.974+0200 [Balancer] config servers and shards contacted
successfully
2019-10-06T15:00:39.974+0200 [Balancer] balancer id: lame13.enst.fr:27020 started
at Oct 6 15:00:39
2019-10-06T15:00:39.978+0200 [Balancer] distributed lock
'balancer/lame13.enst.fr:27020:1570366839:1804289383' acquired, ts :
5d99e5778f4a75f6f5355900
2019-10-06T15:00:39.979+0200 [Balancer] distributed lock
'balancer/lame13.enst.fr:27020:1570366839:1804289383' unlocked.

```

Configure sharding

```

$ mongo localhost:27020/admin
MongoDB shell version: 2.6.12
connecting to: localhost:27020/admin
mongos> db.runCommand( { addshard : "localhost:27014" } )
{ "shardAdded" : "shard0000", "ok" : 1 }
mongos> db.runCommand( { addshard : "localhost:27015" } )
{ "shardAdded" : "shard0001", "ok" : 1 }
mongos> db.runCommand({ enablesharding: "test"})
{ "ok" : 1 }

```

2. Shard the cities by the country.

Import through the admin port:

```

mongoimport --db test --collection cities --file lab_2_datasets/cities.jsonl --
port 27020 --drop

connected to: 127.0.0.1:27020

```

```

2019-10-06T15:25:30.672+0200 dropping: test.cities
2019-10-06T15:25:33.230+0200          Progress: 3786881/14333341      26%
2019-10-06T15:25:33.230+0200          26400    8800/second
2019-10-06T15:25:37.672+0200          Progress: 6393010/14333341      44%
2019-10-06T15:25:37.672+0200          44700    6385/second
2019-10-06T15:25:42.246+0200          Progress: 9071728/14333341      63%
2019-10-06T15:25:42.246+0200          63500    5291/second
2019-10-06T15:25:46.680+0200          Progress: 11710098/14333341     81%
2019-10-06T15:25:46.680+0200          82000    5125/second
2019-10-06T15:25:48.984+0200 check 9 99838
2019-10-06T15:25:55.047+0200 imported 99838 objects

```

Shard the collection imported by city:

```

mongos> use test
switched to db test
mongos> db.cities.ensureIndex({"country":"hashed"})
{
  "raw" : {
    "localhost:27014" : {
      "createdCollectionAutomatically" : false,
      "numIndexesBefore" : 1,
      "numIndexesAfter" : 2,
      "ok" : 1
    }
  },
  "ok" : 1
}
mongos> use admin
switched to db admin
mongos> db.runCommand( { shardcollection : "test.cities", key : {"country" :
"hashed"} } )
{ "collectionsharded" : "test.cities", "ok" : 1 }

```

check the result

```

mongos> sh.status()
--- Sharding Status ---
  sharding version: {
    "_id" : 1,
    "version" : 4,
    "minCompatibleVersion" : 4,
    "currentVersion" : 5,
    "clusterId" : ObjectId("5d99e5778f4a75f6f53558fc")
  }
  shards:
    { "_id" : "shard0000", "host" : "localhost:27014" }
    { "_id" : "shard0001", "host" : "localhost:27015" }
  databases:

```

```

{ "_id" : "admin", "partitioned" : false, "primary" : "config" }
{ "_id" : "test", "partitioned" : true, "primary" : "shard0000" }
    test.cities
        shard key: { "country" : "hashed" }
        chunks:
            shard0000      18
            shard0001      17
        too many chunks to print, use verbose if you want to force

print
mongos> use test
switched to db test
mongos> db.cities.getShardDistribution()

Shard shard0000 at localhost:27014
  data : 22.85MiB docs : 99838 chunks : 18
  estimated data per chunk : 1.26MiB
  estimated docs per chunk : 5546

Shard shard0001 at localhost:27015
  data : 8.6MiB docs : 37607 chunks : 17
  estimated data per chunk : 518KiB
  estimated docs per chunk : 2212

Totals
  data : 31.45MiB docs : 137445 chunks : 35
  Shard shard0000 contains 72.63% data, 72.63% docs in cluster, avg obj size on
shard : 240B
  Shard shard0001 contains 27.36% data, 27.36% docs in cluster, avg obj size on
shard : 240B

mongos>

```

From the result above, we can see that the **cities** collection is successfully sharded on two servers with **"country"** as the key.