

# Extra Credit Work

Jiarong Ye

September 21, 2018

## Environment:

- Ubuntu 18.04
- python 2.7.15

## Set up Cassandra

### Step 1: Make sure java is installed

In [4]: !java -version

```
java version "1.8.0_181"  
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)  
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)
```

### Step 2: Downloading and Installing Cassandra

Add the software repository to verify that the package is available for the system:

```
karen@karen-Lenovo-ideapad-320-15ABR:~/workspace/data_science/Datasci_Fall_18/ds220/Extra/twissandra$ echo 'deb http://www.apache.org/dist/cassandra/debian 36x main' | sudo tee -a /etc/apt/sources.list.d/cassandra.list  
deb http://www.apache.org/dist/cassandra/debian 36x main
```

Add a public key to your Cassandra repository file:

```
karen@karen-Lenovo-ideapad-320-15ABR:~/workspace/data_science/Datasci_Fall_18/ds220/Extra/twissandra$ gpg --keyserver pgp.mit.edu --recv-keys 749D6EEC0353B12C  
gpg: key 749D6EEC0353B12C: 1 signature not checked due to a missing key  
gpg: key 749D6EEC0353B12C: "T Jake Luciani <jake@apache.org>" not changed  
gpg: Total number processed: 1  
gpg: unchanged: 1  
karen@karen-Lenovo-ideapad-320-15ABR:~/workspace/data_science/Datasci_Fall_18/ds220/Extra/twissandra$ gpg --export --armor 749D6EEC0353B12C | sudo apt-key add -  
OK
```

Update the Cassandra repositories:

```
karen@karen-Lenovo-ideapad-320-15ABR:~/workspace/data_science/Datasci_Fall_18/ds220/Extra/twissandra$ sudo apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Ign:3 http://dl.google.com/linux/chrome/deb stable InRelease
Hit:4 http://dl.google.com/linux/chrome/deb stable Release
Hit:5 https://cran.rstudio.com/bin/linux/ubuntu xenial/ InRelease
Get:6 http://security.ubuntu.com/ubuntu bionic-security InRelease [83.2 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Hit:8 https://cloud.r-project.org/bin/linux/ubuntu bionic-cran35/ InRelease
Hit:11 https://cloud.r-project.org/bin/linux/ubuntu xenial-cran35/ InRelease
Hit:9 https://packagecloud.io/slacktechnologies/slack/debian jessie InRelease
Get:13 http://us.archive.ubuntu.com/ubuntu bionic-updates/main i386 Packages [293 kB]
Get:14 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [328 kB]
Get:15 http://us.archive.ubuntu.com/ubuntu bionic-updates/universe i386 Packages [199 kB]
Get:16 http://us.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [199 kB]
0% [Connecting to dl.bintray.com (75.126.118.188)] [Connecting to ppa.launchpad.net (91.189.95.83)]
```

Install Cassandra:

```
karen@karen-Lenovo-ideapad-320-15ABR:~/workspace/data_science/Datasci_Fall_18/ds220/Extra/twissandra$ sudo apt-get install cassandra
Reading package lists... Done
Building dependency tree
Reading state information... Done
cassandra is already the newest version (3.6).
```

Make sure Cassandra is running:

```
karen@karen-Lenovo-ideapad-320-15ABR:~$ cassandra -f
OpenJDK 64-Bit Server VM warning: Cannot open file /var/log/cassandra/gc.log due to Permission denied
15:07:51.807 |INFO| in ch.qos.logback.classic.LoggerContext[default] - Found resource [logback.xml] at [file:/etc/cassandra/logback.xml]
15:07:51.990 |INFO| in ch.qos.logback.classic.joran.action.ConfigurationAction - debug attribute not set
15:07:52.005 |INFO| in ReconfigureOnChangeFilter[InvocationCounter=0] - Will scan for changes in [/etc/cassandra/logback.xml] every 60 seconds.
15:07:52.005 |INFO| in ch.qos.logback.classic.joran.action.ConfigurationAction - Adding ReconfigureOnChangeFilter as a turbo filter
15:07:52.016 |INFO| in ch.qos.logback.classic.joran.action.JMXConfiguratorAction - begin
15:07:52.194 |INFO| in ch.qos.logback.core.joran.action.ShutdownHookAction - About to instantiate shutdown hook of type [ch.qos.logback.core.hook.DelayingShutdownHook]
15:07:52.201 |INFO| in ch.qos.logback.core.joran.action.AppenderAction - About to instantiate appender of type [ch.qos.logback.core.rolling.RollingFileAppender]
15:07:52.209 |INFO| in ch.qos.logback.core.joran.action.AppenderAction - Naming appender as [SYSTEMLOG]
15:07:52.323 |INFO| in ch.qos.logback.core.rolling.FixedWindowRollingPolicy@5afa04c - Will use zip compression
15:07:52.399 |INFO| in ch.qos.logback.core.joran.action.NestedComplexPropertyIA - Assuming default type [ch.qos.logback.classic.encoder.PatternLayoutEncoder] for [encoder] property
```

Step 3: Create the schema

```
karen@karen-Lenovo-ideapad-320-15ABR:~/workspace/data_science/Datasci_Fall_18/ds220/Extra/twissandra$ python manage.py sync_cassandra
All done!
```

Start up the webserver

```
karen@karen-Lenovo-ideapad-320-15ABR:~/workspace/data_science/Datasci_Fall_18/ds220/Extra/twissandra$ python manage.py runserver
Validating models...
0 errors found
September 21, 2018 - 17:31:09
Django version 1.5.5, using settings 'twissandra.settings'
Development server is running at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
/usr/local/lib/python2.7/dist-packages/django/conf/urls/defaults.py:3: DeprecationWarning: django.conf.urls.defaults is deprecated; use django.conf.urls instead
  DeprecationWarning)
[21/Sep/2018 17:31:14] "GET / HTTP/1.1" 200 2205
[21/Sep/2018 17:31:15] "GET /media/css/reset.css HTTP/1.1" 200 648
[21/Sep/2018 17:31:15] "GET /media/css/text.css HTTP/1.1" 200 418
[21/Sep/2018 17:31:15] "GET /media/css/screen.css HTTP/1.1" 200 1070
[21/Sep/2018 17:31:15] "GET /media/css/960.css HTTP/1.1" 200 5336
[21/Sep/2018 18:28:49] "GET /auth/login/ HTTP/1.1" 200 3305
[21/Sep/2018 18:28:49] "GET /media/css/reset.css HTTP/1.1" 304 0
[21/Sep/2018 18:28:49] "GET /media/css/text.css HTTP/1.1" 304 0
[21/Sep/2018 18:28:49] "GET /media/css/screen.css HTTP/1.1" 304 0
```

## Public Timeline

There are no tweets yet. Make sure to post one!

Twissandra is an example project, created to learn and demonstrate how to use Cassandra. Running the project will present a website that has similar functionality to Twitter.

View the code for this site [on GitHub](#)

[Log in to post a tweet](#)

## Create 2 user accounts

## Sign Up

Username:

Password1:

Password2:

## Sign Up

Username:

Password1:

Password2:

Connect the users as friends and tweet from each of the users

## Add Friends

Hooray, your friend richard\_hendricks is on the site!

Twissandra

[Home](#) [Public](#) [Find Friends](#) [Sign out of richard\\_hendricks](#)

### Public Timeline

[richard\\_hendricks](#) Great! just reach \$2.51 USD!

[bertram\\_gilfoyle](#) Hey richard. How's pipercoin doing on the market now?

Twissandra is an example project, created to learn and demonstrate how to use Cassandra. Running the project will present a website that has similar functionality to Twitter.

View the code for this site [on GitHub](#)

## Cql Queries

### describe tables

```
cqlsh> describe tables;

Keyspace system_schema
-----
tables      triggers  views  keyspaces  dropped_columns
functions  aggregates indexes  types      columns

Keyspace system_auth
-----
resource_role_permissions_index  role_permissions  role_members  roles

Keyspace system
-----
available_ranges  peers  batchlog  transferred_range
batches           compaction_history  size_estimates  hints
prepared_statements  sstable_activity  built_views
"IndexInfo"         peer_events      range_xfers
views_builds_in_progress  paxos            local

Keyspace system_distributed
-----
repair_history  view_build_status  parent_repair_history

Keyspace system_traces
-----
events  sessions

Keyspace twissandra
-----
users  timeline  followers  tweets  userline  friends
```

### describe keyspaces

```
cqlsh> describe keyspaces;

system_schema  system      system_traces
system_auth    system_distributed  twissandra
```

### show all users

```
cqlsh> select * from twissandra.users;

  username  | password
-----+-----
  bertram_gilfoyle | 12345
  richard_hendricks | 12345
(2 rows)
```

show all tweets

```
cqlsh> select * from twissandra.tweets;
```

tweet_id	body	username
7b91438c-efa9-412d-9b7c-90aaefd371f	Great! just reach \$2.51 USD!	richard_hendricks
321fa215-77bd-4cec-88fa-6545441f4d47	Hey richard. How's pipercoin doing on the market now?	bertram_gilfoyle

show all friends

```
cqlsh> select * from twissandra.friends;
```

username	friend	since
richard_hendricks	bertram_gilfoyle	2018-09-22 00:17:59.495000+0000

## Challenges when setting up

### Cassandra Version

```
karen@karen-Lenovo-ideapad-320-15ABR:~/workspace/data_science/datasci_Fall_18/ds220/Extra/twissandra$ python manage.py sync_cassandra  
InvalidRequest: Error from server: code=2200 [Invalid query] message="unconfigured table schema_keyspaces"
```

**Reason:** Migration from Cassandra 2.x to 3.x: schema\_keyspaces table is gone

**Fix:**

In twissandra/tweets/management/commands/sync\_cassandra.py, change one line as follows:

- before

```
rows = session.execute(  
    "SELECT * FROM system.schema_keyspaces WHERE keyspace_name='twissandra'")
```

- after

```
rows = session.execute(  
    "SELECT * FROM system_schema.keyspaces WHERE keyspace_name='twissandra'")
```

## Fail to display tweets on Public Timeline

### TypeError at /

object of type 'ResultSet' has no len()

```
Request Method: GET
Request URL: http://127.0.0.1:8000/
Django Version: 1.5.5
Exception Type: TypeError
Exception Value: object of type 'ResultSet' has no len()
Exception Location: /home/karen/workspace/data_science/Datasci_Fall_18/ds220/Extra/twissandra/cass.py in _get_line, line 81
Python Executable: /usr/bin/python
Python Version: 2.7.15
Python Path: ['/home/karen/workspace/data_science/Datasci_Fall_18/ds220/Extra/twissandra',
              '/usr/lib/python2.7',
              '/usr/lib/python2.7/plat-x86_64-linux-gnu',
              '/usr/lib/python2.7/lib-tk',
              '/usr/lib/python2.7/lib-old',
              '/usr/lib/python2.7/lib-dynload',
              '/home/karen/.local/lib/python2.7/site-packages',
              '/usr/local/lib/python2.7/dist-packages',
              '/usr/lib/python2.7/dist-packages']
Server time: Fri, 21 Sep 2018 19:45:34 -0500
```

**Fix:** In `twissandra/cass.py`, at line 76, add one line as follows:

- before

```
results = session.execute(query, params)

if not results:
    return [], None
```

- after

```
results = session.execute(query, params)
results = list(results)
if not results:
    return [], None
```