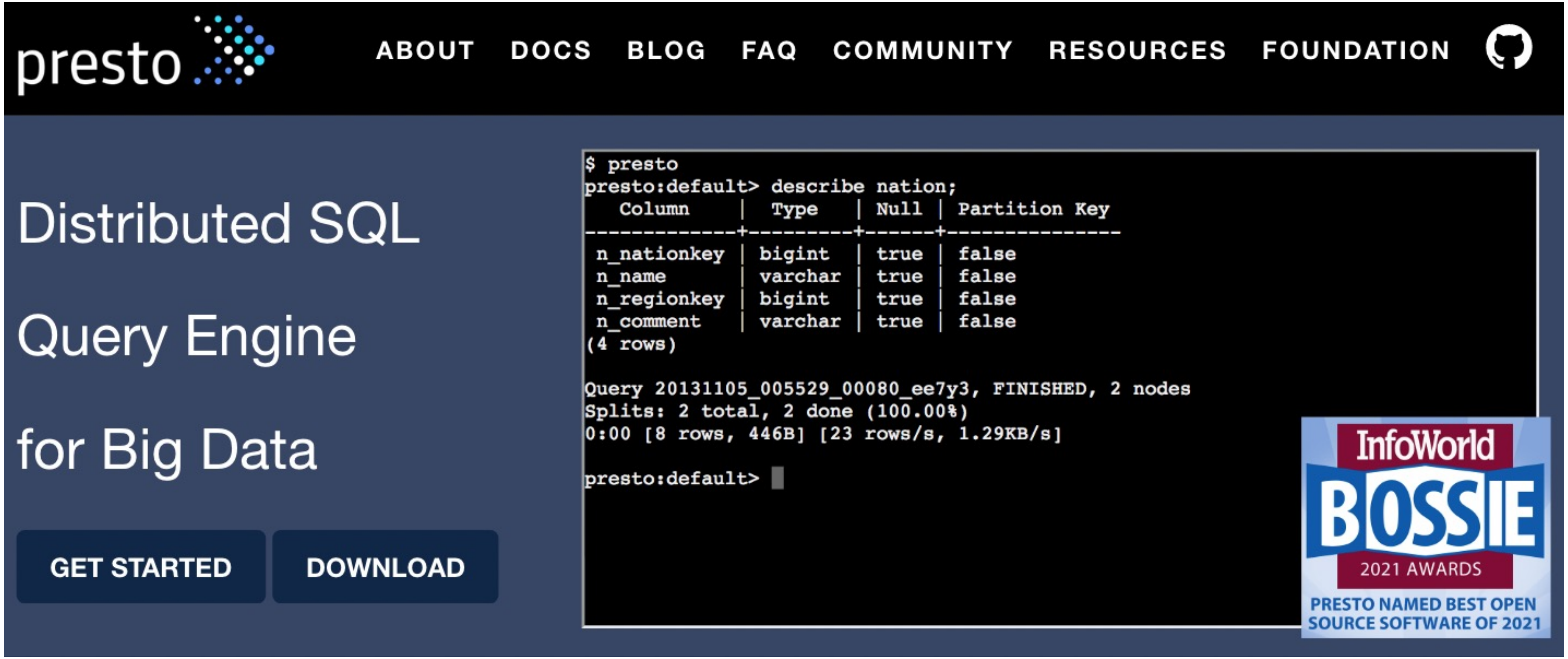
PRESTO

Distributing queries across different data stores

Bernard Lee Kok Bang

Query engines; they act like a middle man between data stores as well as SQL query



Distributed SQL Query Engine for Big Data

```
$ presto
presto:default> describe nation;
   Column     |  Type   | Null | Partition Key
--------------+---------+------+--------------
 n_nationkey  | bigint  | true | false
 n_name       | varchar | true | false
 n_regionkey  | bigint  | true | false
 n_comment    | varchar | true | false
(4 rows)

Query 20131105_005529_00080_ee7y3, FINISHED, 2 nodes
Splits: 2 total, 2 done (100.00%)
0:00 [8 rows, 446B] [23 rows/s, 1.29KB/s]

presto:default> 
```

https://prestodb.io/

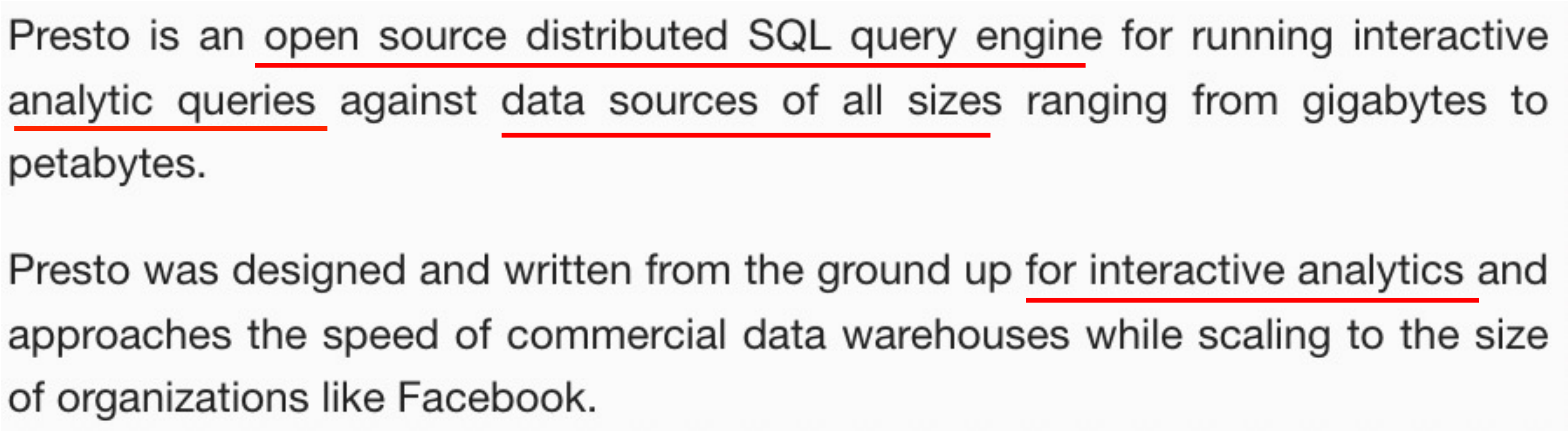
Developed by FaceBook;
Cassandra is a FaceBook-sponsored project

Presto

- Conceptually like Drill
  - Can connect to many different "big data" databases and data stores at once, and issue queries across them [MongoDB, Cassandra, Hive] **local files**
  - Provide single point of access for issuing queries against data no matter where the data might locate in your organization, and join them together
  - Handy for more complex queries from different data stores
  - Act as a layer between SQL queries and various distributed data stores
  - Offer familiar SQL syntax
  - Optimized for OLAP – analytical queries, data warehousing
- Developed, and still partially maintained by Facebook
- Exposes JDBC driver, Command-Line, Tableau interfaces

OLTP - online transaction processing -> much faster; Phoenix

What is Presto? **meant for analytics queries**



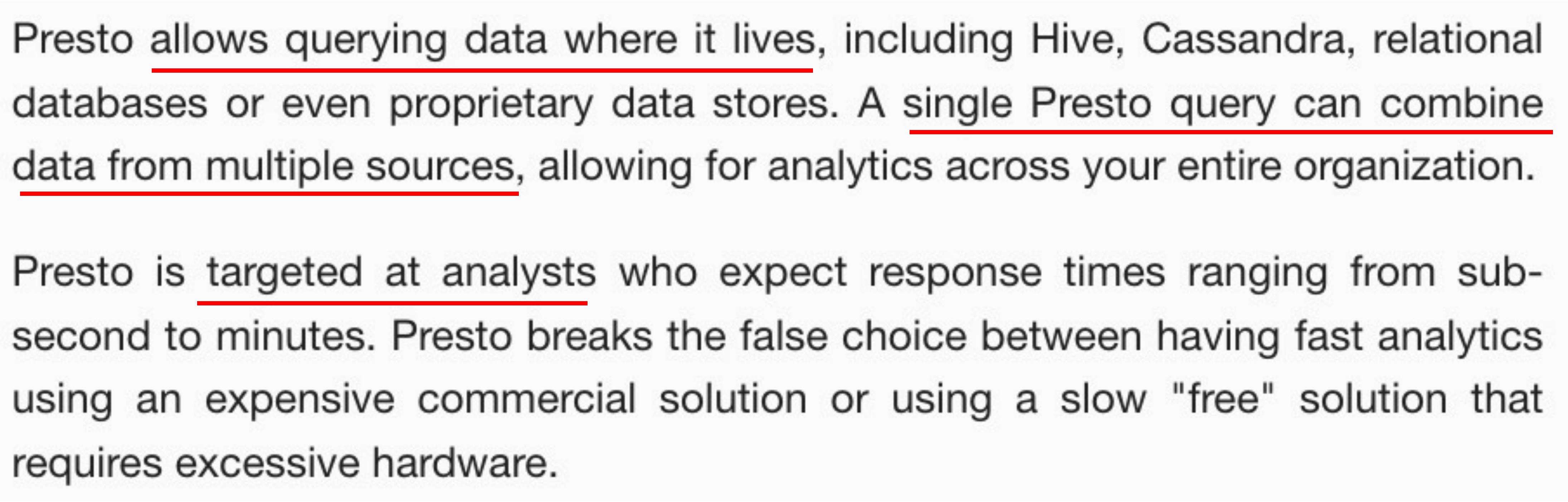
Presto is an open source distributed SQL query engine for running interactive analytic queries against data sources of all sizes ranging from gigabytes to petabytes.

Presto was designed and written from the ground up for interactive analytics and approaches the speed of commercial data warehouses while scaling to the size of organizations like Facebook.

https://prestodb.io/

What can Presto do?

Developed FaceBook; open-sourced



Presto allows querying data where it lives, including Hive, Cassandra, relational databases or even proprietary data stores. A single Presto query can combine data from multiple sources, allowing for analytics across your entire organization.

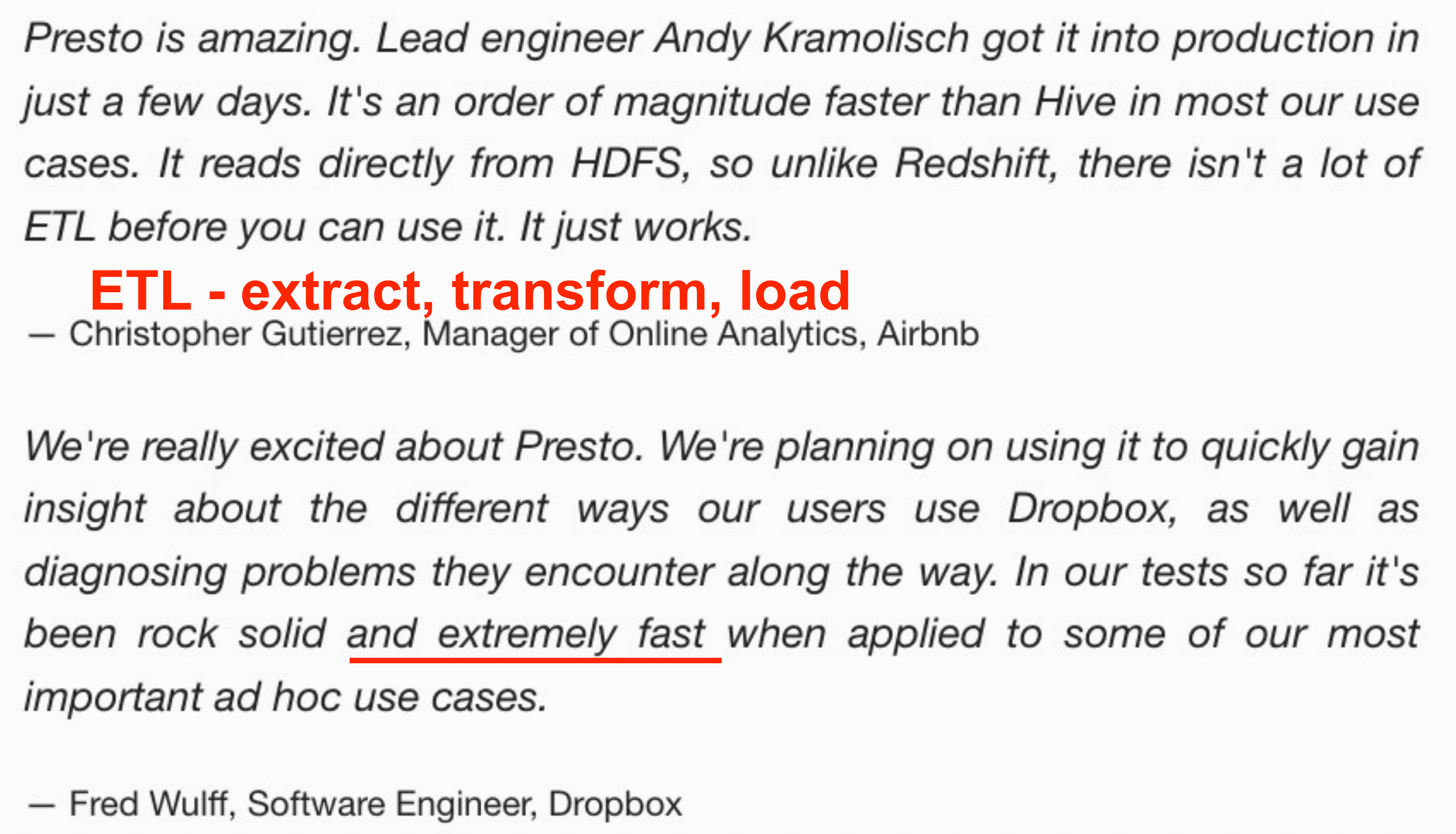
Presto is targeted at analysts who expect response times ranging from sub-second to minutes. Presto breaks the false choice between having fast analytics using an expensive commercial solution or using a slow "free" solution that requires excessive hardware.

Why Presto?

Drill can also connect to Cassandra

- Vs. Drill? It has a stable Cassandra connector
- If it’s good enough for Facebook…
  - - “Facebook uses Presto for interactive queries against several internal data stores, including their 300PB data warehouse. Over 1,000 Facebook employees use Presto daily to run more than 30,000 queries that in total scan over a petabyte each per day”

What leading internet companies said about Presto?



Presto is amazing. Lead engineer Andy Kramolisch got it into production in just a few days. It's an order of magnitude faster than Hive in most our use cases. It reads directly from HDFS, so unlike Redshift, there isn't a lot of ETL before you can use it. It just works.

ETL - extract, transform, load

— Christopher Gutierrez, Manager of Online Analytics, Airbnb

We're really excited about Presto. We're planning on using it to quickly gain insight about the different ways our users use Dropbox, as well as diagnosing problems they encounter along the way. In our tests so far it's been rock solid and extremely fast when applied to some of our most important ad hoc use cases.

— Fred Wulff, Software Engineer, Dropbox

Redshift is a data warehouse product which forms part of the larger cloud-computing platform AWS

What can Presto connect to?

- Cassandra (Facebook-sponsored project)
- Hive
- MongoDB
- MySQL
- Local files

Let’s set up Presto

- Word of caution: Quite troublesome to set up!!!
- Can easily go wrong!!!

Relational DB

- Query Hive *ratings* table using Presto
- Spin Cassandra back up, and query *users* table in Cassandra using Presto **non-relational DB**
- Execute a query that joins *users* in Cassandra with *ratings* in Hive

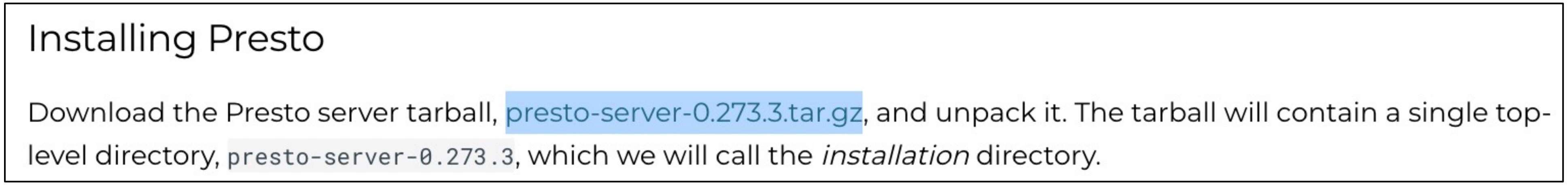
from movielens dataset

Install Presto and Query Hive

Bernard Lee Kok Bang

Install Presto

- Login to Hortonworks SandBox through puTTY
- Login as root
  - *su root*
- Get the link to download Presto from *prestodb.io* website, navigate to Docs > Installation > Deploying Presto
- Download the Presto server tarball file and unpack it [right click and choose *copy link address*]



Installing Presto

Download the Presto server tarball, presto-server-0.273.3.tar.gz, and unpack it. The tarball will contain a single top-level directory, `presto-server-0.273.3`, which we will call the *installation* directory.

Install Presto (cont…)

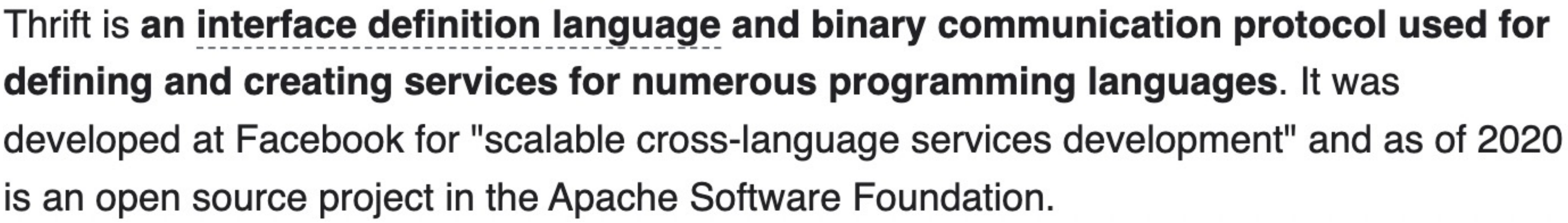
- make sure you are in *maria_dev* home directory: */home/maria_dev/*
    - *wget + right click to paste the link*
- Uncompress the tarball file
    - *tar –xvf xxx.tar.gz file*
- Download configuration file for Presto [not provided in Presto installation, need to install ourselves, under a folder called *"etc"*]
- Configuration parameters will change depending on *where are your different servers for different databases, what port you want to run* and so on…

Install Presto (cont…)

/home/maria_dev/presto-server-xxx/

- Download Presto configuration file from the net
- make sure you are in presto server directory
  - *wget http://media.sundog-soft.com/hadoop/presto-hdp-config.tgz*
- Unpack the .tgz file to create *etc* directory
- tar –xvf xxx.tgz

What is Thrift?



Thrift is **an interface definition language and binary communication protocol used for defining and creating services for numerous programming languages**. It was developed at Facebook for "scalable cross-language services development" and as of 2020 is an open source project in the Apache Software Foundation.

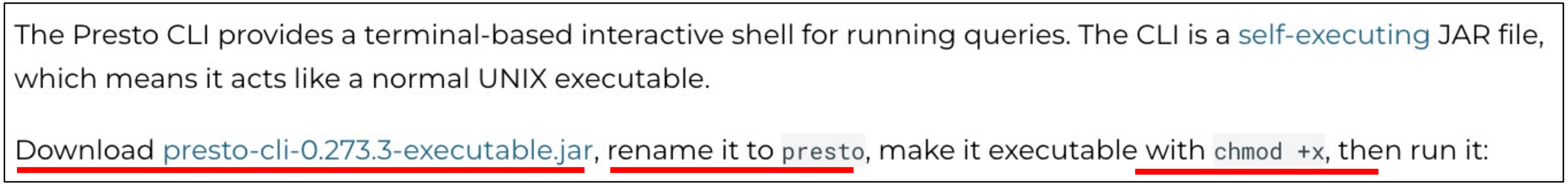
- **Thrift protocol is like a universal language that allows computer programs written in different languages to communicate with each other effectively**

https://bit.ly/3HFoHMP

- **acts as a translator by providing a set of rules and conventions for data exchange, making it easier for diverse systems to work together smoothly.**

Install Presto (cont…)

- Download and install CLI for Presto; navigate back to prestodb.io > Docs > Installation > Command Line Interface



The Presto CLI provides a terminal-based interactive shell for running queries. The CLI is a self-executing JAR file, which means it acts like a normal UNIX executable.

Download presto-cli-0.273.3-executable.jar, rename it to `presto`, make it executable with `chmod +x`, then run it:

- Make sure you are in *bin* folder
  - *wget + right click to paste the link address*
- Change the name of the jar file and make it executable
  - mv xxx.jar presto
  - *chmod +x presto*

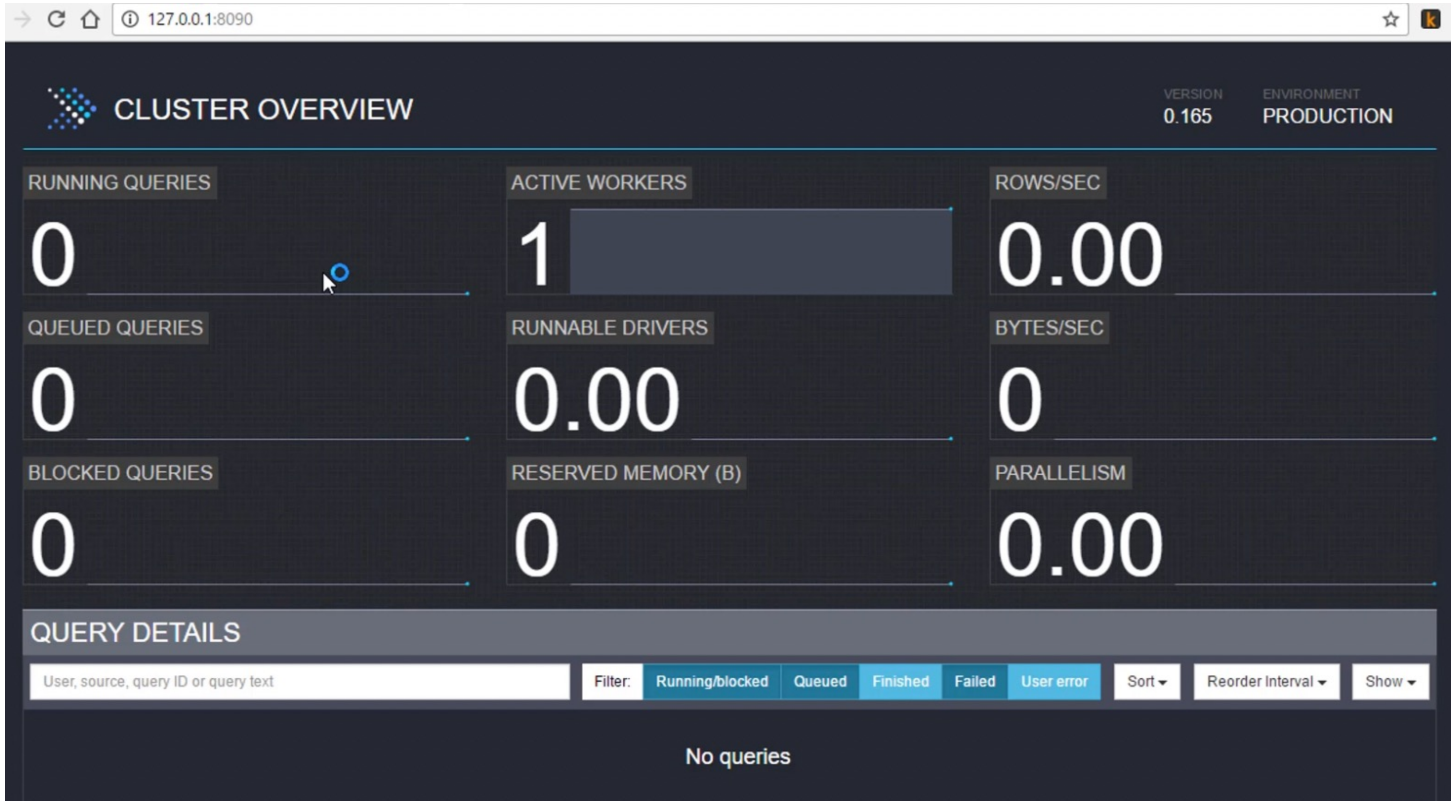
Prerequisite:

- Make sure you are using Java 8
- Python version 2.7

Launch Presto

- Make sure you are in *Presto server* directory
- To start Presto server
  - *bin/launcher start*
- Connect to web interface for Presto
  - *127.0.0.1:8090*

Presto operational dashboard UI



CLUSTER OVERVIEW

VERSION 0.165

ENVIRONMENT PRODUCTION

RUNNING QUERIES 0

ACTIVE WORKERS 1

ROWS/SEC 0.00

QUEUED QUERIES 0

RUNNABLE DRIVERS 0.00

BYTES/SEC 0

BLOCKED QUERIES 0

RESERVED MEMORY (B) 0

PARALLELISM 0.00

QUERY DETAILS

No queries

- *Spikes in Running Queries section whenever SQL statements were issued*

Presto CLI

- Make sure you are in Presto server file
- Set up CLI to connect to Hive database
- bin/presto --server 127.0.0.1:8090 --catalog hive
- Assuming you still have the *ratings* table under the *default* database in Hive [Ambari]
  - *show tables from default;*
  - *select * from default.ratings limit 10;*
  - *select * from default.ratings where rating = 5 limit 10;*
  - *select count(*) from default.ratings where rating=1;*
  - *quit # if you want to quit the presto CLI*
  - *bin/launcher stop # to stop the presto server*

Query both Cassandra and Hive using Presto

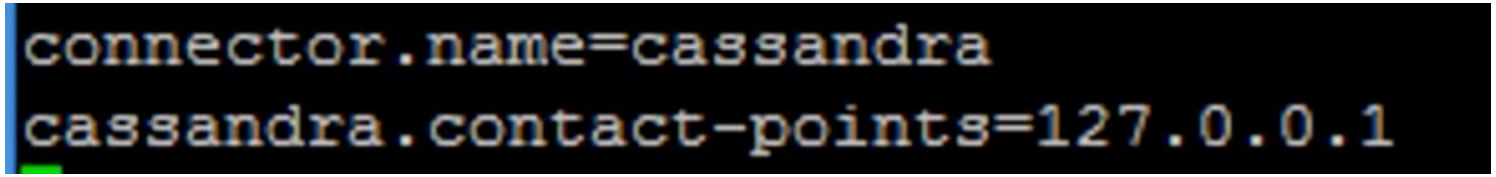
Bernard Lee Kok Bang

Starting Cassandra

  - *service cassandra start*
- # Enable thrift service on Cassandra - to enable Presto to communicate with Cassandra
  - *nodetool enablethrift*
- #Check Cassandra still have the *u.user* data
  - *cqlsh*
  - *describe keyspaces;*
  - *use movielens;*
  - *describe tables;*
  - *select * from users limit 10;*
  - *quit*

Setting up configuration file for Presto

- Navigate to Presto server directory
- #Go into etc/catalog subdirectory
  - *cd /etc/catalog*
- # Configure properties for Cassandra
  - vi cassandra.properties



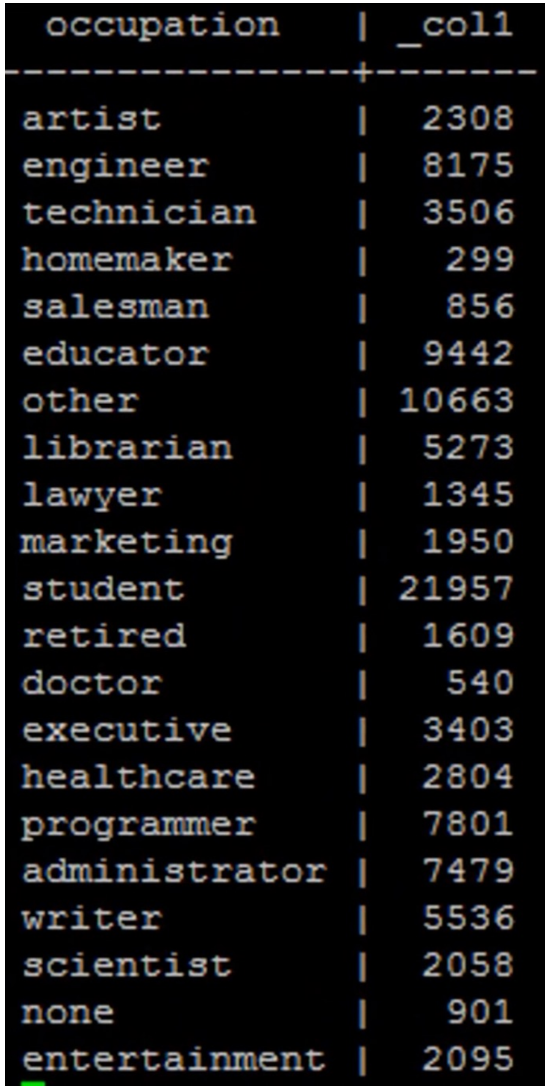
```
connector.name=cassandra
cassandra.contact-points=127.0.0.1
```

Setting up configuration file for Presto (cont…)

- # back to the Presto server file
  - *cd ../../*
  - *bin/launcher start*
- # Connect Presto to Hive and Cassandra
  - *bin/presto --server 127.0.0.1:8090 --catalog hive,cassandra*
  - *show tables from cassandra.movielens;*
  - *describe cassandra.movielens.users;*
  - *select * from cassandra.movielens.users limit 10;*
  - *select * from hive.default.ratings limit 10;*

Join both data from Cassandra and Hive using Presto

- *select u.occupation, count(*) from hive.default.ratings r join cassandra.movielens.users u on r.user_id = u.user_id group by u.occupation*



```
  occupation    | _col1
----------------+-------
 artist         |  2308
 engineer       |  8175
 technician     |  3506
 homemaker      |   299
 salesman       |   856
 educator       |  9442
 other          | 10663
 librarian      |  5273
 lawyer         |  1345
 marketing      |  1950
 student        | 21957
 retired        |  1609
 doctor         |   540
 executive      |  3403
 healthcare     |  2804
 programmer     |  7801
 administrator  |  7479
 writer         |  5536
 scientist      |  2058
 none           |   901
 entertainment  |  2095
```

Clean up the mess!!!

- quit
- bin/launcher stop
- service cassandra stop