Perbandingan Performa Apache Impala Dengan Apache Spark Dalam Mengeksekusi Kueri

Pratama Arjan Rangkuti[1], Muhammad Zihni Athallah[2], Tasya Harwani Barus[3], Nadila Afisa Rani[4], Muhammad Ikhsan Setiawan[5]

[1]Universitas Sriwijaya, Sistem Komputer
[2]Universitas Sriwijaya, Sistem Komputer
[3]Universitas Sriwijaya, Sistem Komputer
[4]Universitas Sriwijaya, Sistem Komputer
[5]Universitas Sriwijaya, Sistem Komputer

Penulis Korespondensi: Tasya Harwani Barus (tasyabarus87@gmail.com)
Nadila Afisa Rani (nadylaafisarani12@gmail.com)

Diterima	:			URL	:	
Disetujui	:			DOI	:	
Diterbitkan	:			ISSN	:	

ABSTRAK

Perkembangan teknologi komputasi modern membawa transformasi besar dalam pengelolaan dan analisis data. Saat ini, kita memiliki akses ke teknologi komputasi yang kuat, yang memudahkan pengolahan dan analisis data. Apache Impala dan Apache Spark adalah dua platform yang memainkan peran penting dalam ekosistem analisis data saat ini. Keduanya adalah bagian integral dari Hadoop, sebuah framework open-source yang efisien dalam penyimpanan dan pemrosesan data besar.Apache Impala adalah query engine yang memungkinkan eksekusi query SQL secara interaktif terhadap data yang tersimpan di HDFS atau HBase. Di sisi lain, Apache Spark adalah platform yang mengintegrasikan pemrosesan data streaming dan batch dalam satu kerangka kerja yang fleksibel. Kedua platform ini memiliki keunggulan dan perbedaan yang perlu dipahami dengan baik.Penelitian ini bertujuan untuk membandingkan performa eksekusi kueri antara Apache Impala dan Apache Spark. Eksperimen dilakukan dengan merancang kueri yang serupa dan menjalankannya pada kedua platform dengan dataset yang ada. Waktu eksekusi kueri dicatat dan dianalisis untuk memahami kinerja relatif dari masing-masing platform dalam menanggapi permintaan kueri.Hasil dari penelitian ini memberikan wawasan berharga tentang kecepatan dan efisiensi eksekusi kueri antara Apache Impala dan Apache Spark. Pemahaman ini dapat membantu organisasi dan praktisi data dalam memilih platform yang sesuai dengan kebutuhan dan konteks analisis data mereka.

Kata Kunci: ***Apache Impala, Apache Spark, Waktu Eksekusi Kueri, Pemrosesan Data***

ABSTRACT

The development of modern computing technology brought about a major transformation in data management and analysis. Today, we have access to powerful computing technology, which facilitates data processing and analysis. Apache Impala and Apache Spark are two platforms that play an important role in today's data analytics ecosystem. Both are integral parts of Hadoop, an open-source framework that is efficient in the storage and processing of big data. Apache Impala is a query engine that enables interactive execution of SQL queries against data stored in HDFS or HBase. On the other hand, Apache Spark is a platform that integrates streaming and batch data processing in one flexible framework. Both of these platforms have advantages and differences that need to be well understood. This study aims to compare query execution performance between Apache Impala and Apache Spark. Experiments were conducted by designing similar queries and running them on both platforms with existing datasets. Query execution time is recorded and analyzed to understand the relative performance of each platform in response to query requests. The results of this study provide valuable insights into the speed and efficiency of query execution between Apache Impala

and Apache Spark. This understanding can assist organizations and data practitioners in choosing a platform that suits their data analysis needs and context.

Keywords: Apache Impala, Apache Spark, Query execution time, Data processing

1. PENDAHULUAN

Latar Belakang

Dalam konteks analisis data yang semakin penting, performa dan efisiensi eksekusi kueri merupakan aspek utama dalam memilih platform yang tepat. Dua platform terkemuka yang sering digunakan untuk menganalisis data secara real-time adalah Apache Impala dan Apache Spark. Apache Impala memungkinkan eksekusi kueri SQL interaktif terhadap data yang disimpan, sementara Apache Spark memfasilitasi pemrosesan data yang terdistribusi. Penelitian ini bertujuan untuk membandingkan performa eksekusi kueri antara Apache Impala dan Apache Spark. Penelitian ini melibatkan uji coba eksekusi kueri pada data yang dihasilkan melalui pembuatan tabel dan pengisian konten. Hasil penelitian menunjukkan perbandingan performa antara kedua platform dalam merespon kueri. Penelitian ini memberikan wawasan yang berharga bagi para praktisi teknologi dan pengambil keputusan dalam memilih platform yang paling sesuai untuk kebutuhan analisis data yang efektif.

Analisis perbandingan merupakan teknik analisis suatu aplikasi atau software yang dilakukan dengan cara memasukkan proses terhadap suatu aplikasi dan membandingkan antara satu menggunakan yg lain, serta memberikan keterangan atau data lain berupa kecepatan memproses data atau besar memori yang dibutuhkan untuk memproses sebuah data pada suatu aplikasi.

Apache Spark adalah suatu framework terdistribusi buat pengolahan data menggunakan standar yang kegunaannya mirip dengan Map-Reduce milik Apache Hadoop apalagi dipergunakan buat pengolahan sebagai parallel. Suatu komponen yang ada Apache Spark yang tak dipunyai sang platform yang lain ialah RDD (Resilient Distributed Dataset). RDD ialah memori tererovolusi yang bisa dilihat ketika kluster dijalankan sebagai parallel (Pan 2016). Pada saat pengoperasian di setiap RDD, nilai pembatas pada dalamnya akan menunjukkan kualitas paralelisme.[1]

Apache Spark tersusun atas sebagian susunan yaitu (1) Spark SQL merupakan evolusi modern sesudah Shark SQL yang awalnya dipakai pada Spark. Spark SQL bisa mengerjakan pengarsipan memori atau pada Bahasa (In-Memory Columnar Storage) berasal dari Shark, Spark SQL bisa compatible menggunakan Hive suatu Query yang ada di Hadoop (Ivanov et al. 2014). Oleh sebab itu Spark SQL Ini mempunyai kemajuan besar pada hal pengumpulan data serta menumbuhkan kinerja (Han and Zhang 2016). (2) Spark Streaming ialah framework stream yang bergerak pada Apache Spark yang ada pada API serta spontan terhubung menggunakan streaming menjadi terhubung. Pada dasarnya Spark Streaming ini bisa menghandle kekurangan yang ada di Apache Hadoop pada bagian pemaparan pada streaming data secara realtime.[1]

Apache Impala menyediakan kueri SQL berperforma tinggi dan berlatensi rendah pada data yang disimpan dalam format file Apache Hadoop yang populer. Respons cepat terhadap kueri memungkinkan eksplorasi interaktif dan penyesuaian kueri analitik, dibandingkan pekerjaan batch panjang yang biasanya

dikaitkan dengan teknologi SQL-on-Hadoop. (Anda akan sering melihat istilah "interaktif" diterapkan pada kueri cepat semacam ini dengan waktu respons skala manusia.[2]

Impala terintegrasi dengan database metastore Apache Hive, untuk berbagi database dan tabel antara kedua komponen. Integrasi tingkat tinggi dengan Hive, dan kompatibilitas dengan sintaksis HiveQL, memungkinkan Anda menggunakan Impala atau Hive untuk membuat tabel, mengeluarkan kueri, memuat data, dan sebagainya.[2]

Berikut ini adalah beberapa keunggulan utama Impala:

- Impala terintegrasi dengan ekosistem CDH yang ada , artinya data dapat disimpan, dibagikan, dan diakses menggunakan berbagai solusi yang disertakan dengan CDH . Hal ini juga menghindari silo data dan meminimalkan perpindahan data yang mahal.
- Impala menyediakan akses ke data yang disimpan di CDH tanpa memerlukan keterampilan Java yang diperlukan untuk pekerjaan MapReduce. Impala dapat mengakses data langsung dari sistem file HDFS. Impala juga menyediakan front-end SQL untuk mengakses data di sistem database HBase, atau di Amazon Simple Storage System (S3) .
- Impala mengembalikan hasil biasanya dalam hitungan detik atau beberapa menit, bukan beberapa menit atau jam yang sering kali diperlukan untuk menyelesaikan kueri Hive.
- Impala memelopori penggunaan format file Parket, tata letak penyimpanan berbentuk kolom yang dioptimalkan untuk kueri skala besar yang umum digunakan dalam skenario gudang data.[2]

Perbedaan antara apche impala dan spark adalah impala merupakan *Kueri Waktu Nyata untuk Hadoop* . Impala adalah mesin kueri MPP SQL sumber terbuka dan modern untuk Apache Hadoop. Impala dikirimkan oleh Cloudera, MapR, dan Amazon. Dengan Impala, Anda dapat melakukan kueri data, baik yang disimpan di HDFS atau Apache HBase – termasuk fungsi SELECT, JOIN, dan agregat – secara real-time. Sedangkan apache spark merupakan Mesin cepat dan umum untuk pemrosesan data skala besar . Spark adalah mesin pemrosesan cepat dan umum yang kompatibel dengan data Hadoop. Itu dapat berjalan di kluster Hadoop melalui mode mandiri YARN atau Spark, dan dapat memproses data dalam HDFS, HBase, Cassandra, Hive, dan Hadoop InputFormat apa pun. Ini dirancang untuk melakukan pemrosesan batch (mirip dengan MapReduce) dan beban kerja baru seperti streaming, kueri interaktif, dan pembelajaran mesin.

2. TINJAUAN PUSTAKA

Kegagalan perangkat keras adalah hal yang biasa dan bukan pengecualian. Sebuah instansi HDFS bisa terbentuk dari ratusan atau ribuan mesin server, yang masing-masing menyimpan sebagian data sistem arsip. Berita bahwa adanya sejumlah besar komponen dan setiap komponen mempunyai kemungkinan kegagalan yg tak sepele berarti bahwa beberapa komponen HDFS selalu tidak berfungsi. Oleh karena itu, deteksi kesalahan dan pemulihan kesalahan secara cepat dan otomatis merupakan tujuan arsitektur inti HDFS. Pada project ini, kami melakukan perbandingan Apache Impala dengan Apache Spark dalam mengeksekusi kueri.

Hadoop Distributed File System (HDFS) bisa dipaparkan berdasarkan harfiah dapat bermakna Sistem arsip tersalur pada Hadoop. meskipun disebut sistem arsip, tetapi HDFS tak sama dengan sistem arsip tipe sistem operasi, contohnya NTFS, FAT32, serta lain sebagainya. HDFS ialah sistem pengumpulan tersalur, yang melaksanakan metode memecah arsip besar jadi pecahan kecil serta setelah itu mendistribusikannya ke cluster komputer. Cluster ini umumnya terbentuk karena adanya banyak node ataupun computer atau server. Tiap node di cluster ini wajib menginstal Hadoop supaya dapat bekerja. Selaku sistem arsip terdistribusi, HDFS berfungsi buat mengatasi data besar yang ditaruh acak di segala cluster.

Apache Impala mengizinkan pengguna buat mengeksekusi kueri SQL berlatensi rendah pada data yang disimpan di HDFS serta Apache HBase tanpa membutuhkan perpindahan maupun transformasi data berkat pengenalan teknologi database paralel yang bisa diskalakan ke Hadoop. Impala berhubungan dengan Hadoop untuk menggunakan metadata, keamanan, manajemen sumber energi, dan format file serta informasi yang sama semacam Apache Hive, MapReduce, Apache Pig, serta aplikasi Hadoop lainnya. Impala dipasarkan untuk analis serta ilmuwan data untuk memakai SQL ataupun perlengkapan intelijen bisnis buat melaksanakan analisis terhadap data yang ditaruh di Hadoop. Hasilnya, tidak butuh lagi mentransfer kumpulan data ke sistem spesial ataupun format kepemilikan buat melaksanakan analisis. Pencarian interaktif serta pemrosesan informasi skala besar(lewat MapReduce) bisa dicoba secara bertepatan pada sistem yang sama dengan memakai data serta metadata yang sama.

Apache Spark adalah framework yang bisa digunakan buat mengambil informasi dari bermacam sumber, memprosesnya, dan setelah itu menaruh informasi yang sudah diproses di penyimpanan informasi buat dianalisis. Data engineer bisa membuat aplikasi alur pemrosesan Big Data memakai fitur- fitur Apache Spark. Apache Spark terintegrasi dengan beberapa bahasa pemrograman agar dapat mengubah himpunan data yang terdistribusi.

3. METODELOGI PENELITIAN

3.1 Jenis Penelitian

Penelitian pustaka (Library Research), yaitu penelitian yang di lakukan dengan menggunakan beberapa artikel sebagai referensi untuk penulisan.

3.2 Alat dan Bahan Penelitian

3.2.1 perangkat Lunak (Software)

1. Virtual Box
2. Apache Impala
3. Apache Spark
4. Apache Hive
5. Cloudera
6. Hue Manager

3.2.2 Bahan Penelitian

1. covid_19_indonesia_time_series_all.csv

3.2.3 Sistem Operasi Yang Digunakan

1. CentOS 6.7
2. Ubuntu 18.04

3.2.4 Perangkat Keras (Hardware)

1. Processor i5
2. Memory DDR4 8 GB
3. LocalDrive 20 GB

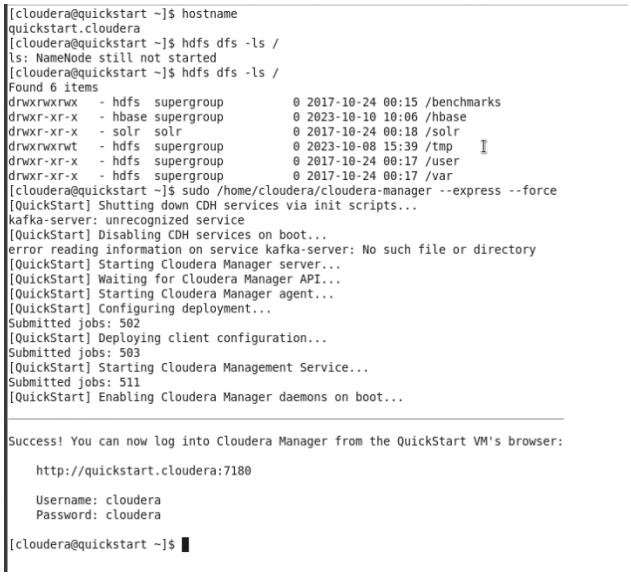
3.3 Jenis dan Pendekatan

Membandingkan performa apache Impala dan apache Spark dalam pengeksekusian kueri berupa data set "covid_19_indonesia_time_series_all" yang berisikan data sebanyak 32823 kolom. Penelitian ini akan menunjukkan perbedaan secara signifikan pada performa antara kedua software tersebut dalam mengeksekusi kueri.

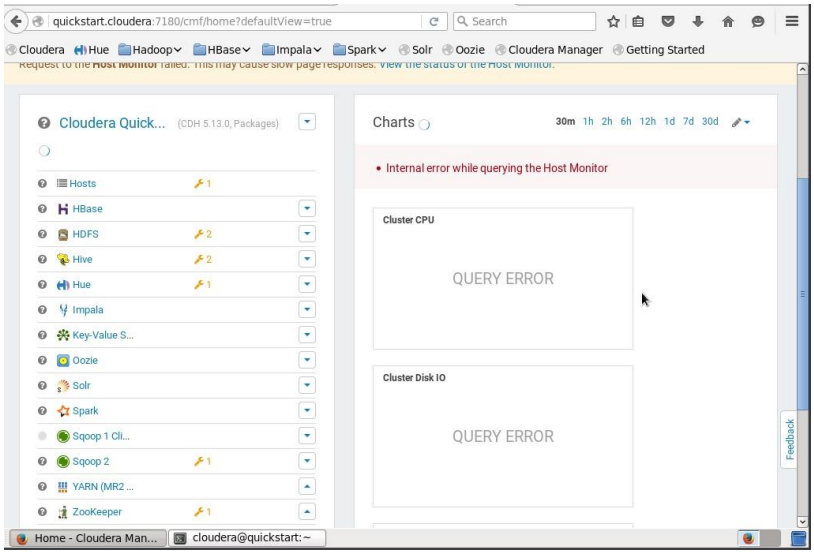
3.4 Tahap Penelitian

Beberapa tahapan yang dilaksanakan selama penelitian, sebagai berikut:

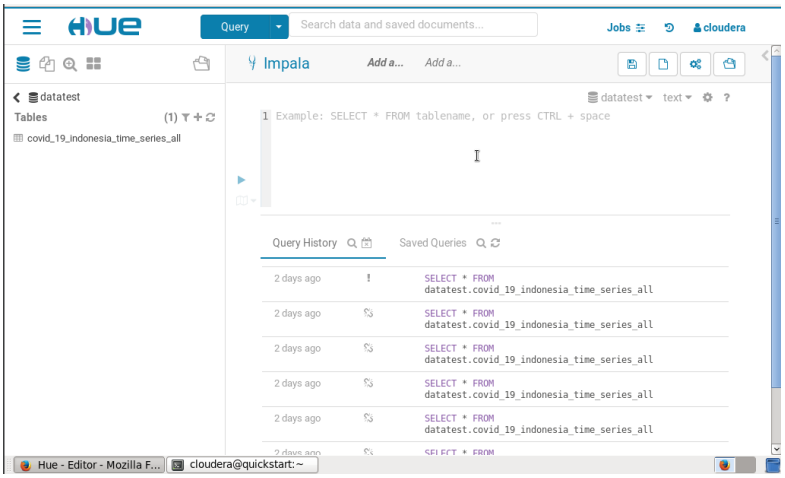
1. Langkah mengeksekusi query menggunakan Apache Impala dengan software cloudera manager



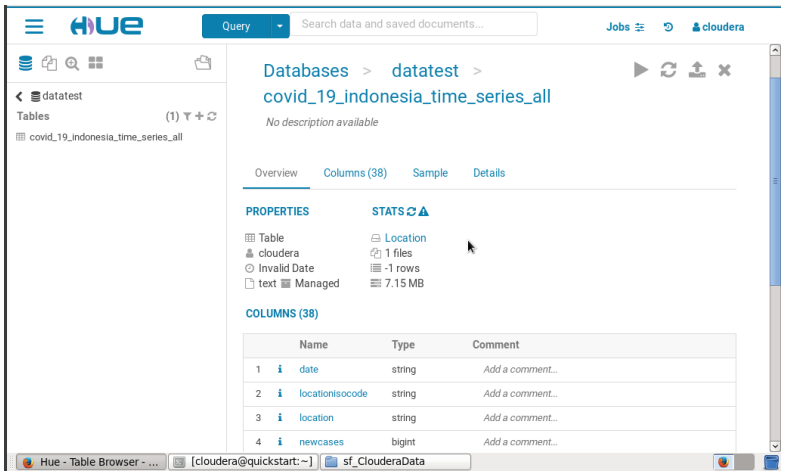
Setelah menyalakan Virtual Machine, kita jalankan command hdfs dfs / untuk memberi permission terhadap sistem. Dilanjutkan dengan command sudo /home/cloudera/cloudera-manager --expess --force untuk mengaktifkan client cloudera manager. Saat client sudah aktif, akan muncul link serta username dan password yang akan kita gunakan untuk login ke cloudera manager.



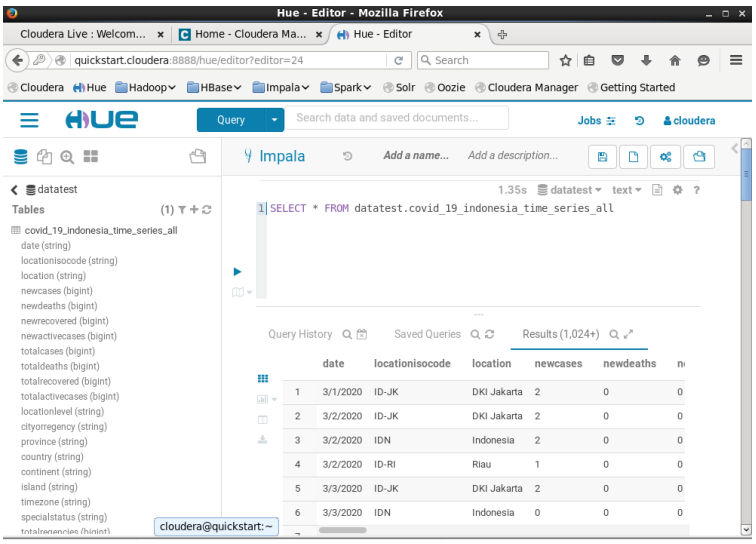
Setelah login, kita harus menjalankan progam HBase, HDFS, Hive, Impala, YARN, Zookeeper dan Hue yang diperlukan untuk mengeksekusi query dengan Impala. Pengeksekusian query akan dilakukan di halaman Hue yang perlu kita buka terlebih dahulu.



Saat kita membuka Hue, akan muncul tampilan seperti diatas. Kolom ditengah berfungsi sebagai tempat menaruh command untuk pengeksekusian query. Kita perlu memasukkan dataset dengan cara menambahkannya kedalam table, tekan simbol "+" pada bagian kiri



Setelah kita menambahkan dataset, akan muncul preview seperti diatas. Kita perlu memberi izin terlebih dahulu kepada sistem untuk bisa mengakses data dalam dataset tersebut dengan menggunakan command $hdfs dfs https//:cloudera/user/cloudera.

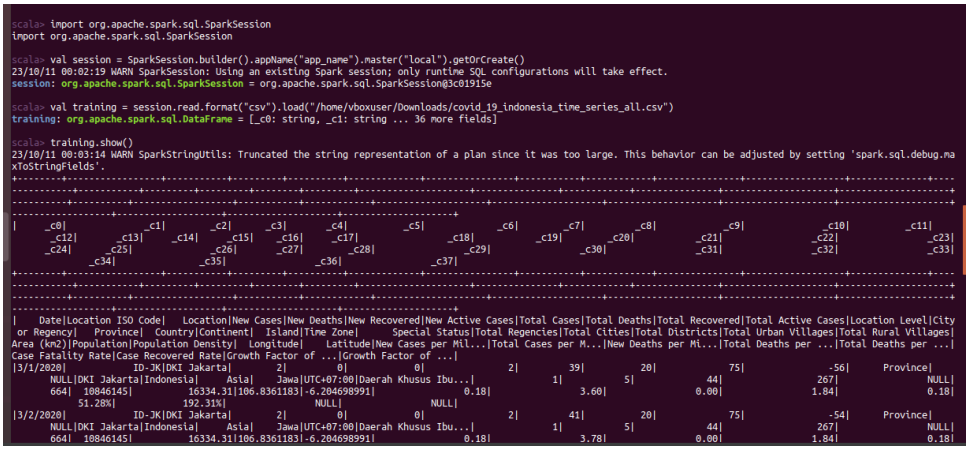


Setelah dataset telah ditambah, kita masukkan command diatas dan tekan tombol run pada bagian kiri kolom command. Lalu sistem akan mulai mengeksekusi data dan memunculkan waktu proses data tersebut.

1. Langkah mengeksekusi query menggunakan Apache Spark pada OS Ubuntu 18.04

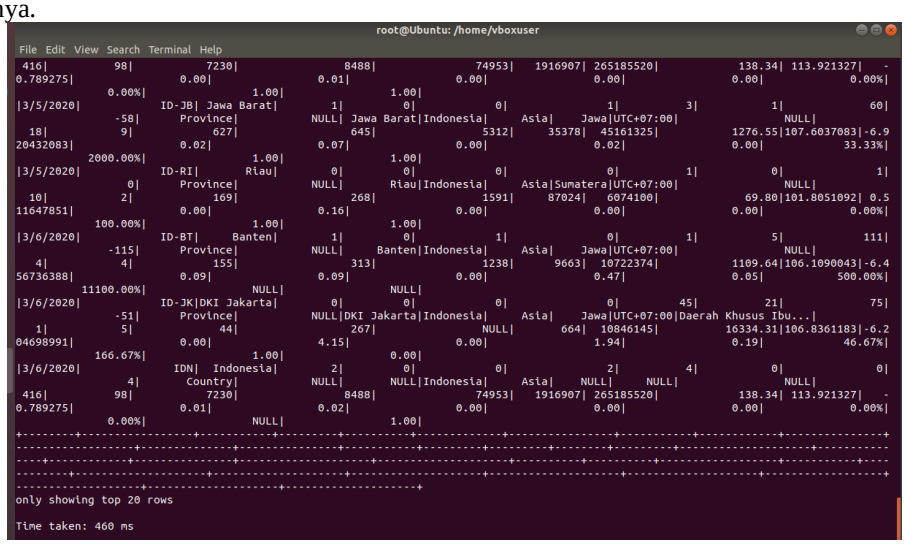
```
vboxuser@Ubuntu:~$ su
Password:
root@Ubuntu:/home/vboxuser# spark-shell
23/10/11 00:46:46 WARN Utils: Your hostname, Ubuntu resolves to a loopback address: 127.0.1.1; using 10.0.2.15 instead (on interface
enp0s3)
23/10/11 00:46:46 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/10/11 00:47:04 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where a
pplicable
Spark context Web UI available at http://10.0.2.15:4040
Spark context available as 'sc' (master = local[*], app id = local-1696956427434).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.5.0
      /_/

Using Scala version 2.12.18 (OpenJDK 64-Bit Server VM, Java 11.0.19)
Type in expressions to have them evaluated.
Type :help for more information.
```

Pertama-tama, kita buka terminal dan masuk ke mode root. Kita cek ke aktifan Apache Spark dengan command spark-shell.



Untuk mulai mengeksekusi query, kita harus memulai spark session terlebih dahulu dengan command import org.apache.spark.sql.SparkSession. Lalu kita build sesi spark dengan command *val session = SparkSession.builder().appName("app_name").master("local").getOrCreate().* Setelah itu spark session akan aktif. Lalu kita masukan dataset yang berupa csv dengan command *val training = session.read.format("csv").load("/home/vboxuser/Downloads/covid_19_indonesia_time_series_all.csv")* dimana "training" adalah nama database kita. Lalu kita panggil database sesuai namanya dengan command

<database>.show(). Setelah itu spark akan mulai mengeksekusi data yang terdapat pada file csv yang telah kita tuliskan pathnya.



Setelah selesai, spark akan menunjukan preview isi data dan waktu yang diperlukan untuk menyelesikan query

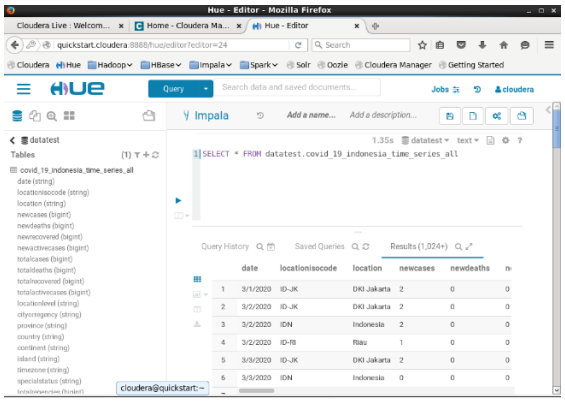
4. HASIL PENELITIAN

Pada penelitian ini, kami melakukan perbandingan performa antara Apache Impala dan Apache Spark dalam mengeksekusi query pada kluster big data. Apache Impala dipilih karena merupakan framework open-source yang efektif untuk menyimpan dataset dalam jumlah besar dan mampu memproses data mulai dari ukuran gigabyte hingga petabyte secara efisien. Sebaliknya, Apache Spark dipilih sebagai salah satu pesaing utama yang juga mampu mengelola dan memproses data besar. Apache Spark dikenal karena kemampuannya dalam memanfaatkan prosesor multi-core dan memproses data secara distribusi, yang dapat mempengaruhi performa eksekusi query secara signifikan. Hasil percobaan menunjukkan bahwa Apache Impala memiliki keunggulan dalam eksekusi query pada dataset dengan skala besar, terutama dalam hal efisiensi dan kecepatan. Namun, Apache Spark juga memberikan kinerja yang baik dalam menangani query pada dataset yang lebih kompleks dan saat memanfaatkan prosesor multi-core. Hasil ini memberikan wawasan penting bagi pengembang dan analis data dalam memilih platform yang sesuai dengan kebutuhan spesifik mereka, terutama dalam konteks pengolahan dan analisis data pada skala besar.

PERCOBAAN

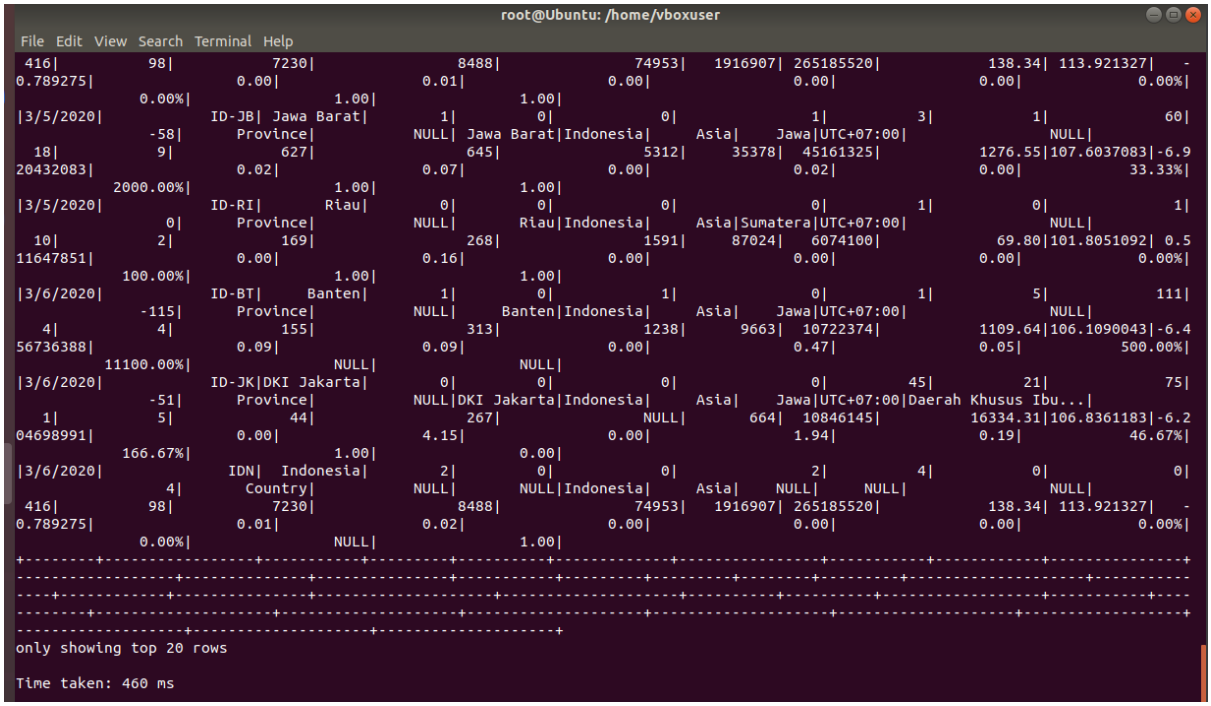
Percobaan ini menggunakan file csv. Disini file csv berisi data “covid_19_indonesia_time_series_all” yang berisikan data sebanyak 32823 kolom.

1.Impala



Waktu yang dibutuhkan untuk menampilkan file csv pada impala dicloudera adalah 1.35s.

2.Spark



Pada spark untuk menghasilkan table sperepti diatas membutuhkan waktu sekitar 460 ms atau sekitar 0,46s untuk mengeksekusinya.

5. KESIMPULAN

Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa Apache Spark mengungguli Apache Impala dalam hal kecepatan eksekusi query pada dataset besar. Apache Spark terbukti lebih cepat dalam mengeksekusi query, terutama ketika memanfaatkan potensi pemrosesan paralel dan distribusi yang optimal. Meskipun Apache Impala efisien dalam menangani dataset besar, Apache Spark menawarkan performa eksekusi yang lebih unggul dan efektif, terutama dalam konteks pemrosesan data dengan skala yang lebih besar. Dalam konteks perbandingan antara Apache Impala dan Apache Spark, Apache Impala memiliki keunggulan dalam mendukung pemrosesan paralel langsung tanpa melalui MapReduce, sehingga meningkatkan efisiensi eksekusi query. Namun, Apache Spark juga memiliki daya saing yang kuat, terutama dalam manfaatkan prosesor multi-core untuk mempercepat pemrosesan query pada dataset besar. Sehingga, kesimpulannya adalah Apache Spark memiliki kinerja eksekusi yang lebih cepat dibandingkan dengan Apache Impala dalam penelitian ini, mempertimbangkan efisiensi dan kecepatan sebagai faktor utama dalam memilih framework untuk menangani query pada data besar.

DAFTAR PUSTAKA

Armbrust, M., Xin, R. S., Lian, C., Huai, Y., Liu, D., Bradley, J. K., ... & Zaharia, M. (2015, May). Spark sql: Relational data processing in spark. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data* (pp. 1383-1394).

Bittorf, M. K. A. B. V., Bobrovytsky, T., Erickson, C. C. A. C. J., Hecht, M. G. D., Kuff, M. J. I. J. L., Leblang, D. K. A., ... & Yoder, M. M. (2015, January). Impala: A modern, open-source SQL engine for Hadoop. In *Proceedings of the 7th biennial conference on innovative data systems research* (pp. 1-10).

Wibawa, C., Wirawan, S., Mustikasari, M., & Anggraeni, D. T. (2022). KOMPARASI KECEPATAN HADOOP MAPREDUCE DAN APACHE SPARK DALAM MENGOLAH DATA TEKS. *Jurnal Ilmiah MATRIK*, *24*(1), 10-20.

https://youtu.be/nRm3NbuS0IA?si=fN5Y86WArRIAeqNd

[1] (Wibawa et al., 2022)Aminudin, A., & Cahyono, E. B. (2019). Pengukuran Performa Apache Spark dengan Library H2O Menggunakan Benchmark Hibench Berbasis Cloud Computing. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTTIK)*, *6*(5), 519-526.

[2] https://docs.cloudera.com/documentation/enterprise/6/6.3/topics/impala.html