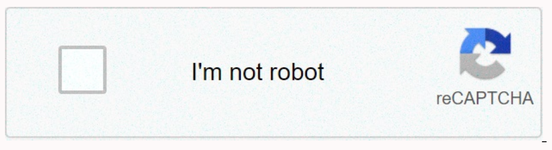

Continue

Apache hive essentials pdf

This book takes you on a fantastic journey to discover the attributes of big data using Apache Hive. Key features Learn the skills you need to write powerful Hive queries to analyze Big Data Discover how Hive can co-exist and work with other tools within the Hadoop ecosystem Using practical, example-oriented scenarios to cover the newly released features of the Apache Hive 2.3.3 book description In this book we prepare you to travel to big data by frstly introducing you to the background of the big data domain. and the process of setting up and learning about the Hive work environment. The book then guides you through discovering and transforming big data values using examples. You can also hone your skills using the Hive language in an effective way. Towards the end, the book focuses on special topics such as performance, security and extensions in Hive, which will guide you to exciting adventures in this value while making big data journeys. By the end of the book, you will be familiar with Hive and able to work effectively to find solutions to big data problems What you will learn about creating and setting up your Hive environment Discover how to use Hive definition language to describe data interesting data from connected and filtering datasets hive transform data using Hive sorting, ordering, and functions Aggregate and sample data in different ways boost Hive query performance and increase the data security of Hive Customize Hive to your needs using user-defined features and integrate it with other tools who make this book ForIf you are a data analyst, developer, or simply someone who wants to quickly get started with Hive to explore and analyze the Big Data Hadoop This book is for you. Since Hive is an SQL-like language, some previous sql experience will be useful to get the most out of this book. This chapter is an overview of big data and Hive, especially in the Hadoop ecosystem. It briefly demonstrates the evolution of big data so that readers know where they are through big data and find out what areas they prefer in the future. This chapter also shows how Hive has become one of the leading tools in the big data ecosystem and why it is still competitive. In this chapter, we cover the following topics: From the database to short history, data repository into big data Introducing Big data databasesBig DataRelal and NoSQL databases over HadoopBatch, real-time, and stream processingHadoop ecosystem overviewHive overview In the 1960s, when computers became a more cost-effective option for businesses, people started using databases to manage the data. Later, in the 1970s, relational databases became increasingly popular for business needs as physical data was easily and closely linked to the logical business. A In the decade, the Structured Query Language (SQL) became the standard query language databases. SQL efficiency and simplicity motivated many people to use databases databases closer to a wide range of users and developers. It was soon observed that people use databases to apply and manage data, and this continued for a long time. After collecting a lot of data, people started thinking about how to handle historical data. Then, in the 1990s, the term data store came up. From then on, people began discussing how to evaluate current performance by reviewing historical data. Various data models and tools have been created to help companies efficiently manage, transform, and analyze historical data. Traditional relational databases have also evolved to provide more advanced aggregation and analyzed functions, as well as optimization for the data store. The leading query language was still SQL, but it was more intuitive and efficient than previous versions. The data is still well structured and the model has normalized. As we entered the 2000s, the Internet gradually became the top industry for creating the majority of data in terms of variety and quantity. Newer technologies, such as social media analytics, web mining and data visualizations, have helped many businesses and companies process large amounts of data to better understand their customers, products, competition and markets. Data volumes have increased and the data format has changed faster than ever before, forcing people to look for new solutions, especially in research and open source areas. As a result, big data has become a hot topic and a challenging area for many researchers and companies. However, there is great potential in every challenge. In the 2010s, Hadoop, which was one of the big data open source projects, received widespread attention thanks to its open source license, active communities and large amount of data management. This was one of the few times an open source project led to a change in technological trends ahead of commercial software products. Soon after, as followers of the NoSQL database, real-time analytics, and machine learning, they quickly became an important component in Hadoop's big data ecosystem. Armed with these big data technologies, companies have been able to review the past, evaluate the current ones, and seize future opportunities. Â Big Data is not simply a large amount of data. Here, the word Big refers to a large range of data. A well-known saying in this area is that it describes big data using three words ranging from the letter V: quantity, speed, and variety. However, data from the analytics and data analytics world saw different data in other dimensions than big data on three Vs, such as reality, variability, volatility, visualization, and value. So far, different Vs explained as follows:Volume: This refers to the amount of data generated in seconds. 90% of today's world's data has been generated in the last two years. Since then, the world's data has doubled, doubled, two years ago. Such large amounts of data are mainly generated by machines, networks, social media and sensors, including structured, semi-structured and unstructured data. Speed: This refers to the speed at which data is generated, stored, analyzed and moved. With internet-connected devices available, wireless or wired machines and sensors transfer their data after they are created. This leads to real-time streaming and helps businesses make valuable and fast decisions. Variety: This applies to different data formats. Data stored .txt, .csv, .dat formats from data sources, such as file systems, spreadsheets, and databases. This type of data, which is located in a field recorded within a record or file, is called structured data. Nowadays, data is not always in the traditional structured format. Newer semi-structured or unstructured data forms are used in different ways, such as email, photos, audio, video, PDFs, SMEs, or even something we have no idea about. These data formats cause problems storing and analyzing data. This is one of the biggest challenges we need to combat in the big data range. Veracy: This applies to the quality of data, such as reliability, bias, noise, and data anomalies. The corrupted data is perfectly normal. It can be due to a number of reasons, such as typo, missing or uncommon abbreviations, recycling, and system errors. However, ignoring malicious data can lead to inaccurate data analysis and ultimately to the wrong decision. Therefore, big data analysis is very important for listening and improving data to make sure that the data is correct. Variability: This applies to changes in data. This means that the same data can have different meanings in different contexts. This is especially important in mood analysis. Analytics algorithms can understand the environment and discover the exact meaning and values of data in this environment. Volatility: This refers to how long the data is valid and can be stored. This is especially important for real-time analysis. The target time window for data needs to be defined so that analysts can focus on specific issues and perform well from the analysis. Visualization: This applies to how the data is well known. A visual doesn't just mean ordinary graphs or pie charts; it also makes huge amounts of data understandable in a multidimensional view that is easily understood. Visualization is an innovative way to see changes in data. Big data analysts and business-domain experts need a lot of interaction, conversation and joint effort to make the visual meaningful. Value: This is based on the knowledge gained from the data analysis of big data The value of big data is how organizations become big data-based companies and use insights from big data analytics in summary, big data is not only about a lot of data, but also the practice of discovering new insights from existing data and guiding the analysis of new data. The big data-driven business will be agile and competitive to overcome challenges and win races. To better understand the differences between the relational database, the NoSQL database, and Hadoop, compare them with the ways to travel. You will be surprised to see many similarities. When people travel or buy cars or airplanes, depending on the distance and costs of travel. For example, when traveling to Vancouver from Toronto, an airplane is always your first choice in terms of travel time and cost. When you travel to Niagara Falls from Toronto, a car is always a good choice. When you travel to Montreal toronto, some people prefer to buy a car to take a plane. The distance and cost here is like the amount of big data and the investment. The traditional relational database is like a car, and the Hadoop big data tool is like an airplane. When dealing with a small amount of data (short distance), the relational database (such as the car) is always the best choice, as it is fast and agile to handle small or medium amounts of data. If you're dealing with large amounts of data (long distance), Hadoop (such as an aircraft) is the best choice because it's more linear, fast and stable if you can handle large amounts of data. You can drive from Toronto to Vancouver, but it takes too long. You can also take a plane from Toronto to Niagara Falls, but it takes more time on the way to the airport and costs more than travelling by car. In addition, you can make a boat or a train. It's like a NoSQL database that offers features and balance from both the relational database and Hadoop in terms of good performance and different data format support for medium to large amounts of data. The batch job is used to process data in batches. It reads data from the input, processes it, and writes it to the output. Apache Hadoop is the best known and most popular open source implementation of the distributed batch processing system using the MapReduce paradigm. The data is stored in a shared and distributed file system, the Hadoop Distributed File System (HDFS), and divides it into divisions that are logical data subdivisions of MapReduce processing. By using the MapReduce paradigm to process the splits, the map job reads the splits and passes all the key/value pairs into a map function, and writes the results to the intermediate files. After the map phase is complete, the reducer reads the intermediate files sent through the random process and passes them to the reduction function. Finally, the reduce task writes the results to the final output files. The advantages of the MapReduce model include simplifying distributed programming, near linear acceleration, and fault tolerance. The disadvantage of this batch processing model is the recursive or iterative tasks. In addition, the obvious batch behavior is that all inputs must be ready to map before the task reduction starts, making MapReduce unsuitable for online and stream processing use cases. Real-time processing is used to process data and get the result almost immediately. This concept was first implemented by Google in Dremel in the field of real-time ad hoc queries related to big data. It uses a new column storage format embedded in structures to quickly index and scale aggregation algorithms to compute query results in parallel instead of batch sequences. These two techniques are the main characters of real-time processing and are used in similar implementations, such as Impala (, Presto (, and Drill (, powered by theÂ columninous storageÂ data format, such as Parquet (, ORC (, Â CarbonDataÂ (, and Arrow (. On the other hand, in-memory computing undoubtedly offers faster solutions for real-time processing. In-memory computing offers very high bandwidth, which is more than 10 gigabytes per second, compared to 200 megabytes per second on the hard disk. In addition, the latency is relatively lower, nanoseconds than in milliseconds than hard drives. With the cost of RAM getting lower and lower every day, in-memory computing is more affordable than a real-time solution, such as Quickly Spark (, which is a popular open source implementation of in-memory computing. Spark is easy to integrate with Hadoop, and the in-memory data structureÂ flexible distributed datasetÂ (RDD)Â can be generated from data sources, such as HDFS and HBase, for efficient caching. Stream processing is used to continuously process and act on live stream data to get results. There are two commonly used general purpose streaming frameworks for streaming processing: Storm (and Flink (. Both frameworks run on the Java virtual machine (JVM) and both processes are key streams. As for the programming model, Storm gives you the basic tools to create a framework, while Flink gives you a well-defined and easy-to-use frame. In addition, Samza (and Kafka Stream (use Kafka for both message caching and conversion. Recently, Spark also provides a kind of stream processing for innovative continuous processing. Hadoop was first released by Apache in 2011 as version 1.0.0, which included only HDFS and MapReduce. A designed from the start as a computing (MapReduce) and storage (HDFS) platform. Due to the growing demand for big data analytics, Hadoop is attracting a number of other software to solve big data issues and melting into a Hadoop-focused big data ecosystem. The next gives a brief overview of the Hadoop big data ecosystem in apache stack: Apache Hadoop ecosystem Hdfsis is still the main option in the current Hadoop ecosystem when using hard drive storage, and Alluxio provides virtually distributed memory alternatives. On top of HDFS, theÂ Parquet, Avro, and ORC data formats could be used in combined withÂ aÂ dynamic compression algorithmÂ for computing and storage optimization. Yarn,Â as the first Hadoop general purpose resource manager, is designed for improved resource management and scalability. Â SparkÂ andIgnite,Â in memory computing engines, are able to run yarn to work with Hadoop tightly as well. Â On the other hand, Â Kafka, Flink, and Storm dominate stream processing. Â HBaseis is a leading NoSQL database, especially for Hadoop clusters. Â With machine learning, it comes as Spark MLlib and Madlib together with a new Mahout.Â SqoopÂ is still one of the leading tools for data exchange between Hadoop and relational databases. Flumeis is a mature, distributed, and reliable log collection tool that you can use to move or collect data to HDFS. ImpalaandDrillÂ are able to start interactive SQL queries directly against the data hadoop. In addition, hive spark/tez over and live long and process (LLAP) allow users to query through different computational frameworks, rather than caching in-memory data in MapReduce. As a result, Hive plays a more important role in the ecosystem than ever before. We are also pleased to see that Ambari, as a new generation of cluster management tools, provides more effective cluster management and coordination in addition to Zookeeper. For scheduling and workflow management, we can use Airflow orÂ Oozie. Finally, there is an open source management and metadata service coming into the picture, Altas, which empowers theÂ compliance and lineage of big data in the ecosystem. Hive is a standard for SQL queries with petabytes of data in Hadoop. Provides SQL-like access to data in HDFS, allowing Hadoop to be used as a data warehouse. The Hive Query Language (HQL) has similar semantics and functions as the relational database to standard SQL, so experienced database analysts can easily get their hands on it. Hive's query language can run on various computing engines, such as MapReduce, Tez, and Spark.Hive metadata structures, providing a high-level, desktop-like structure on HDFS. It supports three main data structures, tables, partitions, and buckets. Tables correspond to HDFS directories and can be divided into partitions where data files can be divided into buckets. Hive's metadata structure is usually the Hadoop schema-on-Read concept, which means you don't need to define the schema in Hive before storing data in HDFS. Applying Hive metadata after storing data gives you more flexibility and efficiency in your data work. Popularity of Hive metadata but describe big data, and many of the big data's tools use it. The following illustration is hive's architectural view of the Hadoop ecosystem. The Hive metadata store (also known as the metatartar) can use embedded, local, or remote databases. The thrift serverÃ is built on Apache Thrift Server technology. With the latest version 2, hiveserver2 can handle multiple concurrent clients, support Kerberos, LDAP, and custom connectable authentication, and provide better options for JDBC and ODBC clients, especially for metadata access. Hive architectureHere are some highlights of Hive that we can keep in mind moving forward: Hive provides a simple and optimized query model with less encoding than MapReduceHQL and SQL similar syntaxHive the query response time tends to be much faster than others with the same amount of big datasetsHive supports running different computational frameworksHive supports ad hoc querying of data hdfs and HBaseHive supports user-defined java/scala features, scripts, and procedure languages to extend the functionality of Unreasonable JDBC and ODBC drivers allow many applications to retrieve Hive data for seamless reportingHive allows users to read data in any format, using SerDes and Input/Output formats Hive is a stable and reliable batch processing tool that is long ready for production Hive has a well-defined architecture for metadata management , authentication, and query optimizationIt is a large community of professionals and developers working and using Hive After going through this chapter, we are now able to understand when and why we should use big data instead of the traditional relational database. We also learned the difference between batch processing, real-time processing, and stream processing. We now know the Hadoop ecosystem, especially the Hive. We traveled back in time and went through the history of databases, data storage, and big data. We also explored some big data terms, the Hadoop ecosystem, the Hive architecture, and the benefit of Hive.In in the next chapter, we'll practice installing Hive, and review all the tools needed to start using Hive in the command line environment. Read more Unlock this book for a free 10-day trial