**INF 6032 Big Data Analytics Assessment**

# Abstract

This article first introduces the definition of big data and big data analysis and then introduces the importance of distributed systems in analyzing big data. Next, the classic distributed system architecture of Hadoop is introduced, and then the appropriate tools are selected for two use cases according to the different performance of each tool to build the Hadoop ecosystem, namely the covid-19 monitoring system and the smart repairing system. Finally, the knowledge learned in the course and the difficulties encountered will be summarized.

Zhibo Liu

Reg No: 210164972

Word count: 3053

Content

[Abstract 1](#_Toc101124387)

[1. Introduction 3](#_Toc101124388)

[2. Use cases 4](#_Toc101124389)

[2.1 covid-19 monitoring platform 4](#_Toc101124390)

[2.1.1 components 5](#_Toc101124391)

[2.1.2 suitable languages 6](#_Toc101124392)

[2.2 Smart Repairing System 8](#_Toc101124393)

[2.2.1 components 9](#_Toc101124394)

[2.2.2 suitable languages 10](#_Toc101124395)

[3. Reflections and lessons learned 12](#_Toc101124396)

[4. Discussion and conclusions 13](#_Toc101124397)

[References 14](#_Toc101124398)

# 1. Introduction

In the past decade, humans have created more data than the entire human history. The boom in big data analysis is sweeping across a wide variety of industries and fields. What fuels this big data movement is that we have new tools to get and have more data about anything. For example, everyone is now generating nearly 200 million terabytes of new data every second without knowing it, including tens of thousands of messages sent in increasingly widely used social media software and emails of all kinds. It also includes that various platforms on the Internet upload new articles, photos, and videos every minute, and currently more than 300 hours of new videos are uploaded to YouTube every minute (Marr, 2016). In addition, various types of sensors, including various types of vehicles, are constantly sending back information. Not only do smartphones have a lot of sensors, but now we have all types of smart homes connected to the Internet, uploading a lot of data to the Internet all the time. This is what we call big data, which refers to a large amount of diverse data that appears at a high speed, that is, a very large and complex data set, whose authenticity and reliability are equally important (Clarke, 2016; Niebel, Rasel & Viete, 2019). Including several characteristics, a large number means various data of up to tens of terabytes mentioned above. High speed means that data is received and processed at high speed. Diversification means that in addition to traditional structured data, various unstructured types of data such as text and video are required.

For these big data, structured, semi-structured and unstructured data cannot continue to be analyzed using the past methods. Therefore, to use big data analysis, big data analysis refers to the use of advanced analysis techniques, such as text analysis, machine learning, etc., on large and diverse data sets (Krasnow, 2014; Tsai et al., 2015). However, the amount of data stored in the database was limited in the past. When the data is large and complex, the analysis will cause the system to run slowly. Therefore, instead of spending a lot of money to upgrade computer hardware with scale-up, we use scale-out, which uses multiple computers to distribute big data and calculations. This method is also a distributed system. To complete the task, this model coordinates multiple computer nodes through the network and shards the data for calculation at each node, so that it can be completed on multiple ordinary single computers. analysis (White, 2015; Marr, 2016).

The most famous of these is the Hadoop ecosystem. Applications in various fields are based on the MapReduce system. For example, in the past, the genome sequencing of biology took 20 million US dollars and several months to complete, but now it can be completed in a few days using MapReduce and several thousand US dollars (Marr, 2016). Another example is the social network Twitter with many users. Twitter has a huge amount of meaningful data based on distributed systems for big data analysis and cooperation with small companies to help various small companies analyze problems or make suggestions (Marr, 2016). These tools using Hadoop and Spark enable engineers to solve various problems that could not be solved before or greatly improve the speed of solving, which is the main reason why distributed systems are needed for big data analysis in the era of big data.

# 2. Use cases

## 2.1 covid-19 monitoring platform

The goal of this use case is to build a system that wants to develop a system that monitors multiple regions and predicts the next likely spread infection and issues an early warning. Thus, assumptions are made about the system. First, it is necessary to obtain accurate data. For example, the covid-19 data of various regions of the United Kingdom on the British government website can be the source of data, including the official data of each region is structured data. Second, after acquiring the data, it is necessary to analyze and visualize the data once a day, so it is required to be able to analyze and visualize the batch data. The third is to use machine learning algorithms to predict areas that are more likely to be infected in the future and send timely notifications to users after getting the results.

### 2.1.1 components

This part will determine the various big data analysis tools to be used in this example through requirements. First, to process big data, the Apache Hadoop ecosystem is used, which is an open-source platform for big data storage and processing (Hadoop, 2015; Dahdouh et al., 2019), which has been widely used in the field of big data. The data source for this case is, for example, the UK's official covid-19 case data, which is structured data, so Sqoop is used. It can transfer covid-19 datasets already stored in relational databases to Distributed File System (HDFS), and efficiently import data into Hadoop storage (Bharti et al., 2019; Landset et al., 2015; Hadoop, 2015). HDFS is the file system to be used, which is distributed to store large amounts of data on multiple hardware nodes and is very suitable for applications with large data sets (Lublinsky, Smith & Yakubovich, 2013; Landset et al., 2015). Since this application collects covid-19 data once a day and processes it every day, that is, many batch processing methods, the calculation engine is selected from MapReduce and Spark engines that are good at batch processing. Apache Spark is an analytical framework for distributed big data processing, which is capable of processing large batches (Bharti et al., 2019). Past studies have compared MapReduce and Spark in Hadoop, which are also good at batch processing, and the results show that Spark's performance is much higher than MapReduce, which solves its lack of efficiency in terms of speed and computing resources (Landset et al., 2015; Bharti et al., 2019). Apache YARN is also used to separate HDFS from Spark, providing job scheduling and cluster resource management capabilities (Hadoop, 2015).

This study needs to predict which regions are likely to have many infections in the future by using data on covid-19 across regions. This can be done by cleaning the data and then using machine learning to build models to predict changes in the infection situation by region in the short term. The fastest increase is in the areas that need to be monitored intensively. So, to complete the machine learning part, choose to use spark's machine learning library Spark MLlib, which makes machine learning scalable and easy. Compared with Mahout, its functions are complete, providing commonly used machine learning algorithms, such as classification, regression, feature extraction, statistics, data processing and other functions (Landset et al., 2015). Then to be able to query data quickly and easily, the hive database is used to query and analyze large amounts of data stored in HDFS using a SQL-like language called HiveQL, which is fast, powerful, and easy to learn and understand (Bharti et al., 2019). And when the result is obtained or a message to improve epidemic prevention measures needs to be sent to a certain region, we use Kafka, a distributed publish-subscribe messaging system, to send messages to users promptly. Finally, under the overall framework, Zookeeper is used to manage the whole, coordinate, and synchronize the services of the distributed system (Landset et al., 2015).

Figure 1 case 1

图示

描述已自动生成

### 2.1.2 suitable languages

The workflow in this use case using the above architecture is shown in Figure 2. The daily covid-19 epidemic data of the national government website is transferred to the distributed file system HDFS through Sqoop. Then it is divided into two uses, the first is to use PySpark on python for visual analysis and machine learning modelling to predict covid-19, and the other is to use HQL to query and analyze data through Hive.

Figure 2

图示

描述已自动生成

The data in this system is structured. The example in table 1 is the covid-19 data that can be collected from the official websites of various countries, including multiple variables

Table 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| continent | location | date | total\_cases | new\_cases | new\_cases\_smoothed | total\_deaths |
| Europe | United Kingdom | 2022/4/14 | 21819851 | 32746 | 34498.143 | 171560 |

First, for the main function of the system to monitor and predict areas where covid-19 cases may increase significantly, based on the results of past research on covid-19 big data analysis, the method of machine learning was selected, and the PySpark package was used to implement on python. Multiple past studies have confirmed that the use of big data analytics technology has played an important role in the fight against the outbreak, indicating that big data analytics tools play a vital role in building covid-19 preventive measures (Ahmed et al., 2021; Azzaoui, Singh & Park, 2021; Agbehadji et al., 2020; Verma & Gazara, 2021; Alsunaidi et al., 2021). And machine learning techniques have good results in predicting the future development of covid-19 (Azzaoui, Singh & Park, 2021), and Ahmed et al. (2021) summarized past research on machine learning prediction of covid-19, and the results show that logistic regression techniques are commonly used for prediction, and the effect is also significant. Therefore, this study uses logistic regression for modelling, and the following is a basic example of modelling in python:

From pyspark.ml.classification import LogisticRegression

mlr = LogisticRegression(maxIter=10, regParam=0.3, elasticNetParam=0.8)

mlrmodel = mlr.fit(covid\_data)

train\_results=mlrmodel.evaluate(training\_covid\_data.predictions)

The following code is an example of data visualization: sns.barplot(data=owid-covid-data,x='date',y='total cases').

Another workflow is to use HQL to query or analyze the stored data through Hive. HQL is the data query language provided by Hive. HQL has been supported and compatible with more and more Hive on Hadoop systems. HQL syntax is very similar to SQL, so in this covid-19 system, it is possible to query data through HQL in Hive. Below is an example of a query that aims to find a descending ranking of the number of new covid-19 cases in each European country as a percentage of the total population on April 15, renamed as the new case rate.

select (new\_cases/population) as new\_case\_rate

from covid\_19\_dataset

where data = "2022-04-15"

and continent = "Europe"

order by desc

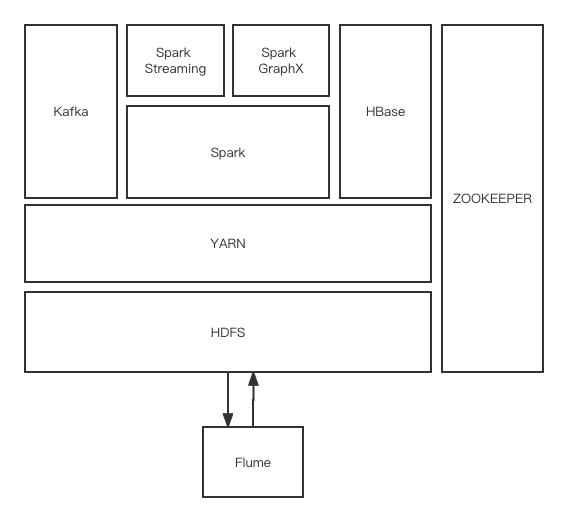
## 2.2 Smart Repairing System

The goal of this use case is to build a smart repair system that wants to remotely find car faults using data generated by the car's numerous sensors. The requirements of the system are mainly several points. First, the system needs to be able to accept a large amount of data of various types, which come from various automotive sensors. The second is to analyze the data in real-time after receiving the data and calculate the nearest garage to the car after obtaining the fault result. Then the third is that the information must be communicated in real-time to the nearest garage and car owner to notify them of the failure. A real-time system must be required to avoid accidents.

### 2.2.1 components

This use case is the collection and analysis of automotive sensor information. In the past, the Hadoop ecosystem has been successfully used to build a big data analysis platform for automotive sensors (Yoo et al., 2020; Kumar & Goel, 2018). This study also uses the Hadoop ecosystem, first, because the sensor is required to report an error in time and then give the customer feedback, it is necessary to use a flow-based method to process it when the data flow is generated (Paunikar et al., 2015). Bharti et al. (2019) and Hoffman (2013) pointed out that Apache Flume can efficiently provide services to collect unstructured streaming data from multiple sources and aggregate them into a Hadoop environment, so it is very suitable for this system. The data transfer between the vehicle and the server is then achieved by opting to use the Apache Kafka subscription-based message distribution system. It was chosen because it has a significantly lower probability of data loss compared to other message brokers during data transmission (Bharti et al., 2019), and can prevent accidents from being lost due to abnormal data. In a Kafka system, messages delivered by producers to consumers go through topics that are brokered by Kafka. Then choose Zookeeper, a distributed process coordination service, to manage all clusters including Kafka for distributed processing (Landset et al., 2015). The transmitted data is not only stored in Hadoop's distributed file system HDFS but also selected to be stored in Apache HBase, which is good at real-time computing for real-time analysis (Hadoop, 2015). It uses HDFS for underlying data storage, and the column-oriented database brings high availability. and scalability. And it is very fast when using multiple data such as SUM, COUNT, MIN, or MAX for real-time calculation (Bharti et al., 2019). In terms of computing engines, since the goal is real-time computing, we choose between Apache storm and Spark. Spark divides the incoming stream into small chunks of data through its wrapper, Spark Streaming, so Spark can be used for batch processing, which can also simulate real-time processing using this kind of micro-batching (Landset et al., 2015; Hadoop, 2015). And compared to Storm, Spark Streaming provides better support for fault-tolerant state computing, even if a node fails, it can still guarantee that each record will be fully processed once (Bharti et al., 2019). Therefore, Spark is chosen as the computing engine and spark GraphX is used to find the nearest path to the repair station. Finally, as with the previous use case, YARN is used to provide job scheduling and management capabilities.

Figure 3 case 2



### 2.2.2 suitable languages

The workflow for the use case of the architecture using these tools is shown in the following figure 4. The various sensor data collected by Kafka from each car are transmitted to the distributed file system HDFS by Flume, and then PySpark and Spark GraphX are used to detect whether there is a problem with the sensor data and calculate which repair station is the shortest distance from the current position of the car and then notify the maintenance station communicates with the owner on time.

Figure 4 workflow

图示

描述已自动生成

The unstructured sensor data in the system. For example, the example in the figure below is the data in JSON format transmitted by a car's speed sensor at 15:30 on April 15, 2022:

{“info”: {“id”: 1,

“name”: “SpeedSensor”

“date”: 202204151530”

},

“data”: {“speed”: 60}}

First, to detect whether the car is faulty, a custom function can be established to detect whether the data of the sensor is within the specified range through the iterative band to realize the detection. Here is an example, define a function through def in python, use a for loop, and judge whether the data is in (the default value range of the sensor) in turn in the for a loop. If there is an error, the result will report FALSE.

Then past research shows that the abundance of sensor data leads to more and more information that can be obtained, especially the automotive sensor data can carry a lot of geographic information (Ganguly & MyiLibrary, 2009; Rathore et al., 2016). And it has been proved that Spark's GraphX ​​is very efficient at computing shortest paths on operations on graphs (Malak & East, 2016; Arfat, Mehmood & Albeshri, 2017; Alazzam, AbuAlghanam & Sharieh, 2021), so Spark GraphX ​​can be used to calculate which vehicle has failed the closest distance between the repair stations, to notify the repair station practice car, owners. The following is an example code for calculating the shortest distance:

vertices = sqlContext.createDataFrame(location data of cars and repair stations)

edges = sqlContext.createDataFrame(the distribution of city roads)

cars\_graph = GraphFrame(vertices, edges)

for x in [(location of all repair stations)

results = cars\_graph.shortestPaths(landmarks=[position of broken-down car, x])

# 3. Reflections and lessons learned

I learned step-by-step how big data analysis works in the classroom part of this course. Also learned the importance of distributed platform in big data analysis and how it solves big data problems. Throughout the course, you will gradually understand the typical distributed ecological Hadoop ecosystem, and gradually understand the functions of ingesting information, storing information, computing engine, application layer above the computing engine, and managing cluster resources in the ecosystem. a tool. As well as what they all apply to and how they differ from each other, these are all very helpful in choosing different infrastructure tools when establishing a specific target task with different needs. In addition to the classroom, a lot of knowledge is also acquired in the practice. Although I spent most of my time studying remotely due to the epidemic, I participated in exercises through Databrick's notebook and watched the videos on encore every time, so I tried my best to participate in the course. Since I am very new to programming in python and want to learn, the content in the exercises is also very useful to me. In addition to learning the basic operations of big data in python, I also learned how to use data frame type data in Pandas, one of the most important libraries for data analysis in python and explained the index that I didn't understand very well in the past. The following exercises also practice the operation of images and the use of data frames in Pyspark. One of the most confusing to me is the User-Defined Functions in SparkR and Pyspark, the output of functions such as Gapply confuses me, and needs more learning and practical use.

# 4. Discussion and conclusions

In conclusion, I learned the use of the Databricks platform throughout the course, and I learned about PySpark's data frame operations and various functions in python for big data analysis, which further improved my python operations. Also learned the importance of using distributed systems for big data analysis, as well as the classic distributed computing Hadoop ecosystem and the use cases for various tools within it. Therefore, the Hadoop system is used in this study to solve two use cases. For the first use case to build a covid-19 monitoring system, use Sqoop to get covid-19 data, use HDFS and Hive to store the data and use YARN to control the cluster. To efficiently process batch data, the Spark computing engine is used, and Spark MLlib is used to predict the dissemination area of ​​machine learning. Hql in the hive can also be used to query data. Finally, users can be notified through Kafka to complete the warning and Zookeeper can be used to control the overall process. The second use case aims to build an intelligent maintenance system. Unlike the first use case, real-time analysis is required. Therefore, Spark streaming is used to achieve real-time analysis on the Spark engine, and HBase is used to store data. And use Flume to stream the data collected by Kafka from the car's sensors into the system. Finally, Spark GraphX calculates the nearest maintenance station to the car and notifies it. Both use cases are using PySpark to extract information on python and convert it into insights to be output by the system. All the course content learning brought me a wealth of knowledge of big data analysis.

# References

Ahmed, Ahmad, M., Jeon, G., & Piccialli, F. (2021). A Framework for Pandemic Prediction Using Big Data Analytics. Big Data Research, 25, 100190–100190. <https://doi.org/10.1016/j.bdr.2021.100190>

Agbehadji, Awuzie, B. O., Ngowi, A. B., & Millham, R. C. (2020). Review of big data analytics, artificial intelligence and nature-inspired computing models towards accurate detection of COVID-19 pandemic cases and contact tracing. International Journal of Environmental Research and Public Health, 17(15), 1–16. <https://doi.org/10.3390/ijerph17155330>

Alsunaidi, Almuhaideb, A. M., Ibrahim, N. M., Shaikh, F. S., Alqudaihi, K. S., Alhaidari, F. A., Khan, I. U., Aslam, N., & Alshahrani, M. S. (2021). Applications of big data analytics to control the covid‐19 pandemic. Sensors (Basel, Switzerland), 21(7), 2282. https://doi.org/10.3390/s21072282

Akshay Paunikar, cover designer.$$QPaunikar, Akshay. (2015). Machine learning with Spark [electronic resource] : create scalable machine learning applications to power a modern data-driven business using Spark. Packt Publishing

Arfat, Y., Mehmood, R., & Albeshri, A. (2017). Parallel shortest path graph computations of United States road network data on apache spark. In International Conference on Smart Cities, Infrastructure, Technologies and Applications (pp. 323-336). Springer, Cham.

Alazzam, AbuAlghanam, O., & Sharieh, A. (2021). Best path in mountain environment based on parallel A algorithm and Apache Spark. The Journal of Supercomputing, 78(4), 5075–5094. <https://doi.org/10.1007/s11227-021-04072-0>

Malak, M., & East, R. (2016). Spark GraphX in action. Simon and Schuster.

Marr, B. (2016). *Big data in practice: how 45 successful companies used big data analytics to deliver extraordinary results*. John Wiley & Sons.

Landset, Khoshgoftaar, T. M., Richter, A. N., & Hasanin, T. (2015). A survey of open source tools for machine learning with big data in the Hadoop ecosystem. Journal of Big Data, 2(1), 1–36. <https://doi.org/10.1186/s40537-015-0032-1>

Bharti, Bajaj, D., Goel, A., & Gupta, S. . (2019). Identifying requirements for big data analytics and mapping to hadoop tools. International Journal of Recent Technology and Engineering, 8(3), 4384–4392. <https://doi.org/10.35940/ijrte.C5524.098319>

Dahdouh, Dakkak, A., Oughdir, L., & Ibriz, A. (2019). Large-scale e-learning recommender system based on Spark and Hadoop. Journal of Big Data, 6(1), 1–23. <https://doi.org/10.1186/s40537-019-0169-4>

EL Azzaoui, Singh, S. K., & Park, J. H. (2021). SNS Big Data Analysis Framework for COVID-19 Outbreak Prediction in Smart Healthy City. Sustainable Cities and Society, 71, 102993–102993. <https://doi.org/10.1016/j.scs.2021.102993>

Hadoop essentials : delve into the key concepts of Hadoop and get a thorough understanding of the Hadoop ecosystem (1st edition.). (2015). Packt Publishing.

Hoffman. (2013). Apache flume [electronic resource] : distributed log collection for Hadoop : stream data to Hadoop using Apache Flume (1st edition.). Packt Pub.

Lublinsky, Smith, K. T., & Yakubovich, A. (2013). Professional hadoop solutions (1st edition.). John Wiley and Sons.

Verma, S., & Gazara, R. K. (2021). Big data analytics for understanding and fighting COVID-19. In Computational intelligence methods in COVID-19: Surveillance, prevention, prediction and diagnosis (pp. 333-348). Springer, Singapore.

White. (2015). Hadoop : the definitive guide (Fourth edition, revised & updated.). O'Reilly.

Niebel, Rasel, F., & Viete, S. (2019). BIG data - BIG gains? Understanding the link between big data analytics and innovation. Economics of Innovation and New Technology, 28(3), 296–316. <https://doi.org/10.1080/10438599.2018.1493075>

Clarke. (2016). Big data, big risks. Information Systems Journal (Oxford, England), 26(1), 77–90. <https://doi.org/10.1111/isj.12088>

Krasnow Waterman. (2014). Big Data analytics: risks and responsibilities. International Data Privacy Law., 4(2), 89–95. <https://doi.org/info:doi/>

Tsai, Lai, C.-F., Chao, H.-C., & Vasilakos, A. V. (2015). Big data analytics: a survey. Journal of Big Data, 2(1), 1–32. <https://doi.org/10.1186/s40537-015-0030-3>

Kumar, S., & Goel, E. (2018). Changing the world of autonomous vehicles using cloud and big data. In 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT) (pp. 368-376). IEEE.

Yoo, Shin, S., Lee, J., & Moon, C. (2020). Implementation of a sensor big data processing system for autonomous vehicles in the C-ITS environment. Applied Sciences, 10(21), 1–16. <https://doi.org/10.3390/app10217858>