



UNIVERSITY of LIMERICK
OLLSCOIL LUIMNIGH

Cloud-based Data Analysis System using Hadoop

Faculty of Science and Engineering

Department of Electronic and Computer Engineering

University of Limerick

Student Name: ZHIKANG TIAN

Student ID: 16060288

Supervisor: Dr. Jacqueline Walker

Course: MEng of Information Network and Security (INS)

Course Code: LM634

Academic Year: 2018/2019

Abstract

As a focus of public opinion, news hotspots can reflect social conditions. The system not only provides journalists with material for news statistics but also helps the government guide and master the direction of public opinion. In this paper, the network hotspot news collecting and analysing system is designed and implemented in detail. As a core component of the system, it realizes the automatic classification of news hot type on the current network and also analyses the news popularity degree. Based on the cloud computing concept and Hadoop framework technology, news published source, news types, comments number, and news released time were analyzed. Finally, the JavaEE application was established, and the final result was presented in the form of data visualization.

The article is divided into two main parts, the literature review part, and the system design and implementation part. Firstly, the article provides a literature review of the techniques and concepts in the system. The background and types of cloud computing and Hadoop frameworks are researched and analyzed, and the HDFS, MapReduce programming frameworks and Hadoop ecosystem are introduced. Second, the article also analyzes other necessary concepts, such as web crawlers, web applications, data visualization, and so forth. Finally, the technical stack of the system was introduced.

Firstly, the system design section describes the system functional requirements, data flow, and system control flow. Three system modules were analyzed and designed: data collection module, data analysis module, and website module. The data collection module solves the problem of the news data source, which is based on web crawler technique with the implementation of HTTPClient and Jsoup. The data analysis module is used to analyze news data. The module is based on the concept of cloud computing, using Hadoop HDFS as a file storage system, and MapReduce mode to implement data analysis. Based on the Spring, Spring-MVC and Mybatis frameworks, the website module creates a website based on the MVC architecture and provides a REST API to facilitate frontend data access and data visualisation. Finally, the module test and system test are carried out and systems are deployed on AWS Cloud Service.

At the end of the article, the results of data analysis are summarized, and some suggestions and criticisms are given for the system.

Acknowledgment

First of all, I would like to thank my supervisor Dr. Jacqueline Walker for her encouragement and patience throughout the year. She helped me choose the research direction and give me a lot of important advice at some key joint points in the project. She gives me great hope on the project, it pushed me as the power to make it better.

Secondly, I would like to thank my course leader Dr. Reiner Dojen. He taught me the distributed system and C++ object-oriented programming, where I picked up lots of knowledge can be applied in web application development.

Then I would like to thank my friend Jianli Wei and classmate Findiland. They are technology geeks, whenever I had a question, they always give me a favor. And I learned a lot from them.

Finally, I would like to thank my parents, they give me financial support for reading master's degrees and moral support for encouragement.

Table of Contents

ABSTRACT	2
1 INTRODUCTION CHAPTER	6
1.1 Research Background and Significance	6
1.2 Research Objectives and Main Work	7
1.3 Report Organization and Structure	8
2 LITERATURE REVIEW	9
2.1 News Hotspots	9
2.1.1 Necessary for Analysing News Hotspot	9
2.1.2 How to Utilize News Hotspots	9
2.2 Cloud Computing	10
2.2.1 History and Development	10
2.2.2 Cloud Computing Types	11
2.3 Hadoop Review	12
2.3.1 History	12
2.3.2 Hadoop Introduction	12
2.3.3 Hadoop Distributed File System	13
2.3.4 MapReduce Engine	14
2.3.5 Hadoop Ecosystem and Products	17
2.4 Web Crawler	18
2.5 Web Application	19
2.6 Data Vistualisation	19
3 ADOPTED FRAMEWORKS	21
3.1 YARN	21
3.2 Amazon Web Service (AWS)	21
3.3 Echats	22
3.4 HTTPClient	22
3.5 Jsoup	22
3.6 SpringBoot	22
3.7 Spring	23
3.8 MyBatis	23
3.9 Maven	23
4 SYSTEM REQUIREMENTS AND DESIGN	24

4.1	Use Case Analysis and Specification	24
4.2	System High-Level Design	26
4.3	Data Collection Module	27
4.4	Hadoop Data Analysis Module	28
4.4.1	Cluster Planning	28
4.4.2	MapReduce Data Analysis Design	30
4.5	Website Module	31
4.5.1	Frontend Design	32
4.5.2	Backend Architecture	33
5	IMPLEMENTATION DETAILS	36
5.1	System Environment Preparations	36
5.1.1	Installing JRE, Hadoop and MySQL	36
5.1.2	Configuring SSH, Management Shell and Network	37
5.1.3	Running Hadoop and MySQL Service	37
5.2	Data Collection	38
5.2.1	Web Spider Implementation	38
5.2.2	Web Spider Business Logic Description	39
5.3	Cloud-based Hadoop Data Analysis	44
5.3.1	Data Cleaning	44
5.3.2	MapReduce Programs	45
5.3.3	AWS Cloud Implementation	48
5.4	Website Implementation	51
5.4.1	Back-end MVC Implementation with SSM	51
5.4.2	Front-end Implementation	53
6	TEST EVALUATION AND DEPLOYMENT	56
7	CONCLUSION	59
	APPENDIX A: MAPREDUCE PROGRAMS FOR DATA ANALYSIS	60
	APPENDIX B: DETAILS OF HADOOP CLUSTER CONFIGURATION	64
	REFERENCES	72

1 Introduction Chapter

1.1 Research Background and Significance

With the rapid development of the Internet, people's way of obtaining information has undergone tremendous changes, and more and more information is flooding into people's eyes. Hereby, it provides us with convenience at every link of life and makes information propagation faster and easier.

Online news not only changes the way people getting information but also provides an important source of information for the public. The popular news shows the public focus, which also makes online news as a barometer for public opinion supervision and a survey system of public opinion. Therefore, how to identify the current news hotspot from the massive data, and analyze the popular news type in news, timely grasp the issues that people are generally concerned about and people's views on news topics, are several important issues. By solving these issues, enterprises can keep abreast of the latest developments in related fields, strategic partners and competitors; governments can grasp the sensational trend in a timely and comprehensive manner, thus leading the public opinion and propaganda. In this context, the analysis of hot topics on the Internet highlights its own importance and will become a direction worthy of further study.

However, these various types of data volumes are large, complex. Although the performance of computers is increasing day by day, single-node computers cannot process massive amounts of data.

For large-scale data processing, the traditional method mainly is parallel computing [1]. However, parallel computing is mainly based on high-performance computers, which are too expensive to be widely used. Moreover, parallel computing processing methods still have shortcomings such as high hardware cost and difficulty in writing parallel programs [2]. On this basis, a variety of big data processing platforms have emerged, and in many large data processing platforms, the open-source big data framework Apache Hadoop is widely recognized for its low cost and high efficiency. It has become the most popular big data processing platform [3].

However, to process big data with a Hadoop cluster, it must deploy a cluster of a certain size. Cloud computing currently is a very popular alternative solution for parallel computing. By publishing the Hadoop cluster to the cloud, it is easy to use the cloud cluster for meeting

needs without having to worry about the cost of cluster deployment and code development. It only needs to pay a certain fee to the cloud computing service provider to enjoy the cluster in the cloud. In this context, this paper designs and implements a news hotspot analysis system based on Hadoop and cloud computing.

The system mainly uses Internet information collection technology and big data analysis technology to realise automatic new collection, topic detection and trend analysis of massive news information on the Internet, and it can visualise analysed data such as charts to provide results to enterprises, governments. The news hotspot collection and analysis system realise the news acquisition by web spiders, news analysis by Hadoop, and analysed data visualisation by web application and so on.

1.2 Research Objectives and Main Work

The purpose for the project is to make the analysis of current news at an authoritative news website and give the visualised result to help government or others social organisation have a better understanding of the current social values.

The article focuses on the online news hotspot under the background of Internet big data, designs and implements the visual news hotspot collection and analysis system, introduces Hadoop data analysis framework and cloud computing concepts, and finally publishes it to the cloud. It mainly completed the following work:

1. Firstly, clarifying the significance of the research. And giving literature reviews on Hadoop, Cloud computing and relevant technologies.
2. Analysing system requirements, detailing system use cases and expected outputs, and designing system modules.
3. Installing the JDK and configuring the Hadoop runtime environment to prepare for subsequent data analysis.
4. Analysing the method and characteristics to collecting online news, designing four key functions to getting data: data collection crawling strategy, scraping fields, dynamic web crawling methods and data storage.
5. Designing and implementing the Hadoop data processing module, including data cleaning, data analysis, and data storage is planned and designed.

6. The data visualization Java web application was designed and implemented, and the results of news hotspots analysis were presented in the form of charts on the webpage.
7. Finally, testing the system with unit testing, and system testing, and deploying the tested system to the cloud computing platform.

1.3 Report Organization and Structure

The first chapter introduces the research background and main research work.

The second chapter, literature review, summarises and reviews major technologies including Hadoop, cloud computing, web crawler, web application and data visualisation. It explains the application and significance of some important technologies.

In chapter 3, the framework and technology that will be adopted in the system will be further elaborated, and some relevant basic knowledge will be introduced as well.

The following chapters adopt the system report document mode. In the fourth chapter, the system requirements are put forward and the system module design is derived according to the requirements. It describes the data flow and designs the data collection module, data analysis module, and website module.

Chapter 5 is the technical implementation of chapter 4, which first describes the configuration of Hadoop cluster and cloud service, then defines the business logic of data collection, the process of MapReduce data analysis program, front and backstage interaction of the website.

Chapter 6 describes details for system testing and module testing, among them the system deployment discussed.

In the final part, the thesis gives a conclusion of the research and summaries the project with a critical review.

2 Literature Review

2.1 News Hotspots

News hotspots refer to the news topic or news type where there are more people concerned about. In most cases, the hot news topic is where the news hotspots located [18].

2.1.1 Necessary for Analysing News Hotspot

From the perspective of sociology, news hotspot generally concerned with social opinion, which may have an influence on social value orientation. In the spread of online news events, the entire process of netizens participating in event discussions is affected by their psychological activities. In the long run, the trend of group psychology has certain guiding significance for the development of a country and the practice of news communication activities [19].

As long as the new values advocated by the news public opinion adapt to social development, they often have great rendering power. It can generate social and emotional resonance and become a new value orientation in social life [20].

Therefore, news hotspots can indirectly reflect social values and social stability. The government can use big news data to carry out active macro-control, and enterprises can make popular products according to the prevailing ideology of society.

2.1.2 How to Utilize News Hotspots

Through the process of analysis, news can be categorized by news type, then a macroscopic statistic made on the news circulating in the current society, so that the current public's concern can be found, so that the type of news occupying the public's psychology can be inferred. It is conducive to the government's management for news information propaganda and the promotion of correct social values.

For the objectivity and authenticity of data analysis, it is necessary to choose high-quality news, so it is necessary to choose some comprehensive news portals. The biggest news website in China is 163 NetEase news. Here the 163 news will be taken as an example to conduct data analysis of news hotspot. Web spider technique can be used for news record collection [7].

2.2 Cloud Computing

Cloud computing, is a kind of computing based on the Internet, in this way, software resources, hardware resources, and information can be shared according to the demand for computers and other devices.

2.2.1 History and Development

The calculation mode has experienced from the initial model that focus task to mainframe as processing (figure 1 (a)), to the distributed model where tasks are distributed to hosts based on the Internet for processing (figure 1 (b)), and finally developed as the more recent, on-demand processing of cloud computing model (figure 1 (c)) [5].

The original single processor pattern had limited processing power, and the requests of client machine needed to wait, which were inefficient. Later, with the continuous development of network technology, the server cluster with high-load configuration will waste and idle resources when it encounters low loading, resulting in an increase of operation and maintenance costs [17].

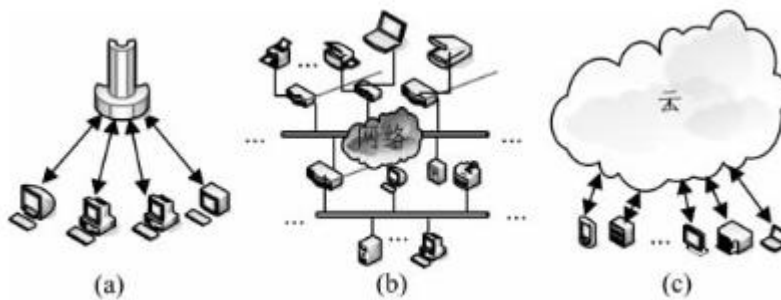


Figure 1. The Development of Task Delivery on Network

Services resources are virtualized by cloud computing, the whole service resource scheduling, and management, are responsible for the specialized maintainer, users no need to care about the realization of the cloud inside, so the cloud computing is essentially like others resource as traditional electricity, water, gas – service in great-demand [2]. Cloud computing has become an important trend in the future and nowadays. Major IT companies in the world, such as Google, IBM, Microsoft, and Amazon, have set up their own "cloud computing platforms" at present. Many companies have developed their own cloud computing services, such as popular Google, Drive, SkyDrive, Dropbox, Amazon Web Services, etc.

Large cloud-based data storage is popular at present, there has been a most Internet companies have begun to use, for example, Amazon, alibaba, baidu, part of the company has taken the Hadoop as their core products technique implementation, such as Intel, IBM, and provide the solution of a big data for part of the work.

2.2.2 Cloud Computing Types

With the increasing demand for Internet-related services, delivery mode, usually it involves using the Internet to provide dynamic easy extension and often virtualized resources. These can implement by cloud services.

In the past, the cloud was often used to represent the telecommunications network, and later, it is an abstraction of the Internet and underlying infrastructure. In the narrow sense, cloud computing refers to the delivery and use mode of IT infrastructure, and refers to obtaining required resources through the network in an on-demand and easily scalable manner. The broad definition of cloud computing is the service delivery and usage pattern, which refers to the availability of required services through the network in an on-demand.

Nowadays, cloud service can be categorized into three types: Infrastructure-as-a-Service, Platform-as-a-Service, and Software-as-a-Service.

➤ IaaS

The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. [5]

➤ PaaS

Users can deploy onto the cloud infrastructure consumer-created applications using programming languages and tools supported by the provider. Consumer has control over the deployed applications and possibly application hosting environment configurations.

➤ SaaS

SaaS uses the provider's applications running on a cloud infrastructure and accessible from various client devices through a thin client interface such as a Web browser, such as web Email.

2.3 Hadoop Framework

Hadoop is a distributed system framework developed by the Apache Foundation, which mainly solves the storage of massive data and the analysis and calculation of massive data

2.3.1 History

Hadoop was designed and developed under the leadership of Doug Cutting, etc at 2003. At the same time, it was recognized by the Apache foundation and became a well-known open-source project. Based on the earlier open source frameworks - Lucene and Nutch, which basically realized Google's own file system and the latest distributed programming idea.

The core idea of that framework is Map/Reduce. Google developed hardware models such as GFS to host the MapReduce model. Map/Reduce is a programming model for mass data operation, and it is also an efficient task scheduling model [6]. In 2007, Google further developed in that.

In 2004, they completed the Hadoop distributed file system and released the earliest version. In 2005, they further developed and increased the number of Nutch clusters. After 2006, with the help of Apache, a part of the Nutch project was moved out and called Hadoop.

At present, Hadoop has attracted deep attention and extensive research and application in the industry and academia.

2.3.2 Hadoop Introduction

Hadoop is based on Java, which allowed Hadoop to be deployed in low-cost computer clusters, without being limited to an operating system. Hadoop a distributed file system for storing big data. It has many excellent features of a distributed system [3]:

1. Fault Tolerance: when designing HDFS, hardware errors are considered as normal events. HDFS can automatically save multiple copies of data, and automatically redistribute failed tasks
2. High Reliability: because Hadoop assumes failure of computing elements and storage, it maintains multiple copies of working data and can reload the process of failed nodes.
3. High Scalability: allocating task data among clusters, which can easily expand thousands of nodes.
4. High Efficiency: under the idea of MapReduce, Hadoop works in parallel to speed up task processing

Hadoop main contains four modules: HDFS, MapReduce, YARN, Common [3].

1. Hadoop HDFS: a distributed file system with high reliability and throughput.

2. Hadoop MapReduce: a distributed offline parallel computing framework.
3. Hadoop YARN: a framework for job scheduling and cluster resource management.
4. Hadoop Common: tool module supporting other modules.

2.3.3 Hadoop Distributed File System

HDFS is a file system used to store mass data files and whose method to locate files is finding by directory tree; Second, its structure is distributed. Many nodes joining together to implement the mass storage functions, and the node in the cluster have their own roles.

HDFS is designed for single-write, multiple-read scenarios. Thus, it does not support file modification in the file system. So, it is suitable used for data analysis, not suitable for network disk applications.

Files in HDFS are stored in block, and the block size can be specified by the configuration parameter. The default size is 128M in hadoop2.x and 64M in older versions. HDFS blocks are larger than harddisk blocks in order to minimize addressing time.

If the block is set too large, the time taking to transfer a block from HDFS is significantly greater than the time taking to addressing the block's location. As a result, the efficiency of file transfer is not high, and the storage of files also causes a certain degree of waste. Therefore, 128MB is an average value for storing data.

● HDFS Node Types

There are three types of HDFS cluster nodes: NameNode, DataNode and Secondary NameNode.

1. The namenode is the HDFS system management node. Generally, only one NameNode will be deployed in an HDFS cluster. There are two files on a namenode: the log file and the namespace mirror file. The former holds the system logs, the latter stores the file system file tree and files directories in the file system. The namenode also stores the information of the datanode where the file resides in the HDFS.
2. The secondary NameNode exists to avoid some problems of namenode - due to the uniqueness of namenode in HDFS cluster, once namenode fails, the operation of the entire cluster will be affected. Under normal circumstances, secondary namenode will be deployed and run on another machine separated from namenode. In order to prevent namenode from making mistakes, it will keep communication with namenode at all times. Meanwhile, secondary namenode will also periodically save snapshots of

data files. Once the namenode fails, the files in the file system will not be lost. When the namenode is restarted after failure, according to the previous configuration, the snapshot of the saved data files can be obtained from secondary namenode.

3. The datanode is responsible for specific data storage. Before storing files in the datanode, the system will split them firstly. Therefore, the data blocks in the datanode will eventually be stored. In order to prevent data loss due to node corruption, data blocks that need to be saved should have multiple backups, which will be stored in different datanodes. A datanode registers to all namenodes for HDFS fault tolerance and it periodically sends block heartbeat information to all namenodes to report their status.

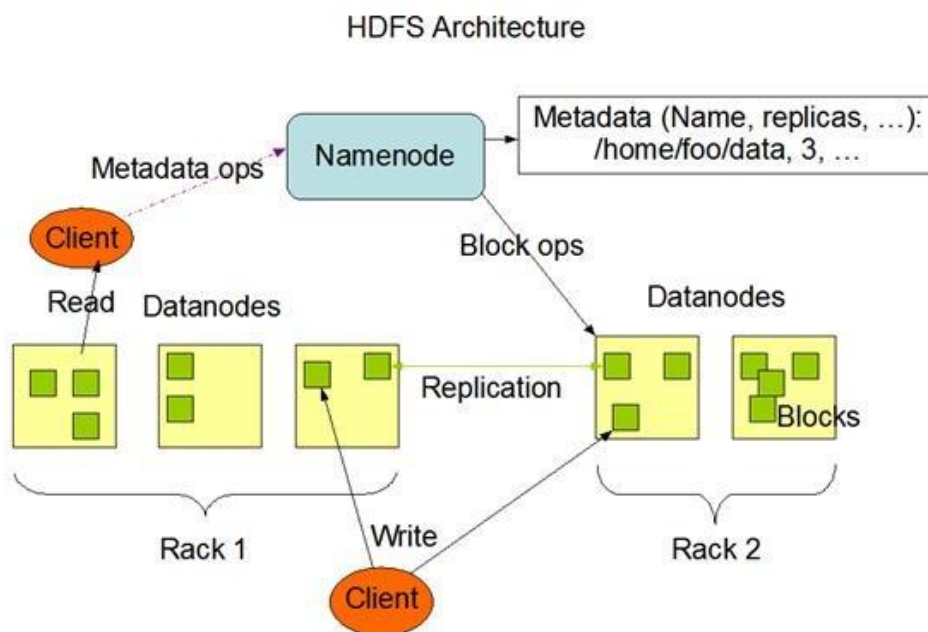


Figure 2. HDFS Architecture.

● HDFS Operation

HDFS provides command-line model for file system management, which is similar to the command to manage file in Linux. And it also provides the client operation method for file system management, which can be used in Java programming. For the client operation, it provides two methods: API for management and Input/Output (IO) streaming.

2.3.4 MapReduce Engine

Mapreduce is a programming framework for distributed computing programs and also is the core for developing data analysis applications based on Hadoop.

As a single machine is not competent to massive data processing due to hardware resource limitations. Once the single-node-version program is extended to cluster to run distributed, it will greatly increase the complexity and development difficulty of the program. With the introduction of the MapReduce framework, developers can focus most of their efforts on the development of business logic, leaving the complexity of distributed computing to the framework [9].

The core function of Mapreduce is to integrate user-written business logic code and built-in default components into a complete distributed computing program running concurrently on a Hadoop cluster. The MapReduce encapsulates many common functionalities in distributed applications.

A complete MapReduce program has three types of instance processes when running in distributed operation:

1. MrAppMaster: responsible for process scheduling and state coordination of the whole program
2. MapTask: responsible for the whole data processing process of mapping stage
3. ReduceTask: responsible for the whole data processing process of reducing phase

● MapReduce Processing Procedure

Distributed computing programs often need to be divided into at least three phases: map, shuffle and reduce [23].

1. In the first phase, concurrent map task instances running completely in parallel and irrelevant to each other. The map program reads the file line by line into the map program to execute the business code and then outputs the result into the shuffle process in accordance with the user-defined key-value pairs as the output format.
2. The shuffle phase refers to the phase that mapper creates the intermediate <key,value> pair and transfers them to the reduce task. With the help of shuffling procedure, data can be divided into different partitions, sorted and combined by key values. By default, system assigning partition number as one part. And system sorts <key,value> pairs by the hashcode of key values so that the reduce task can easily understand when a new reducing task will be started by the sorted <key,value> pairs.
3. In the reduce phase, all concurrent reduce task instances are irrelevant, and their input data depending on the outputted <key, value> pairs of all map tasks in the previous phase, executing reducing code and finally generating new <key,value> pairs on HDFS.

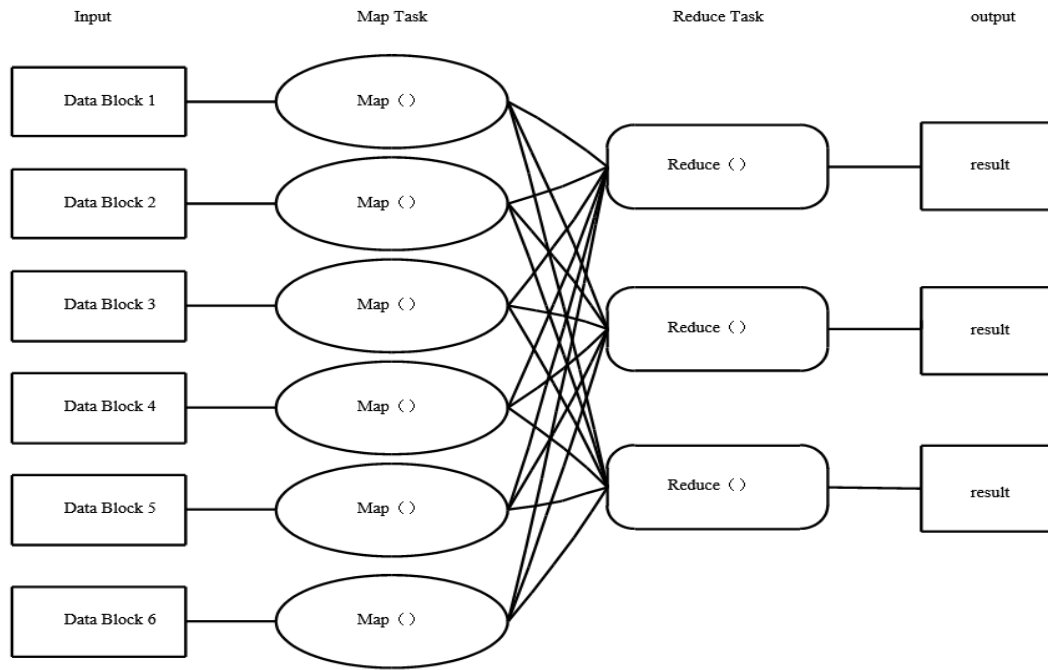


Figure 3. MapReduce Model

● MapReduce Overall Process

The concrete executing steps of MapReduce is complexed, that are listed as follows:

1. Store corresponding files in the input directory where the MapReduce program reads files.
2. Before the submit() method is executed, the client program obtains the data to be processed and then forms a task assignment plan according to the configuration of parameters in the cluster.
3. The client submits job.split, jar file, job.xml and other files to yarn, and yarn resource manager starts the MRAppMaster.
4. After MRAppMaster starts, it calculates the number of required map task instances according to the description information of this job, and then starts the corresponding number of map task processes.
5. Map tasks read data by the customer input format and form input KV pairs.
6. Map tasks pass the input KV pairs to the map() method defined by customer for logical operation.
7. Key-value pairs are collected into map task cache after map() operation.
8. Key-value pairs in map task cache are written to HDFS disk files after K partition sorting.
9. After the completion of all map task process tasks, the MRAppMaster starts the corresponding number of reduce task processes according to the parameters specified by the customer and informs reduce task process which data partition it will deal with.
10. After the start of reduce task, according to the information from MRAppMaster, process will get several map tasks output result files from several map task machines, and

rearrange the result locally, then call the `reduce()` method defined by the customer to carry out logical operations in accordance with KV with same key.

11. When reduce tasks finish calculation, output the result data to the external storage in customer specified output format.

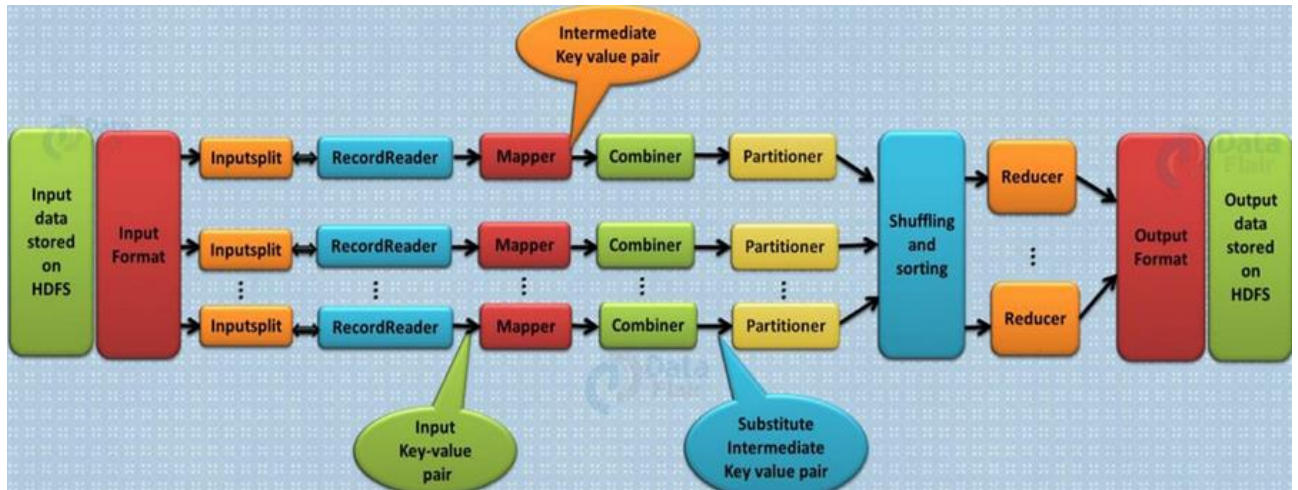


Figure 4. Whole Process of MapReduce Framework

2.3.5 Hadoop Ecosystem and Products

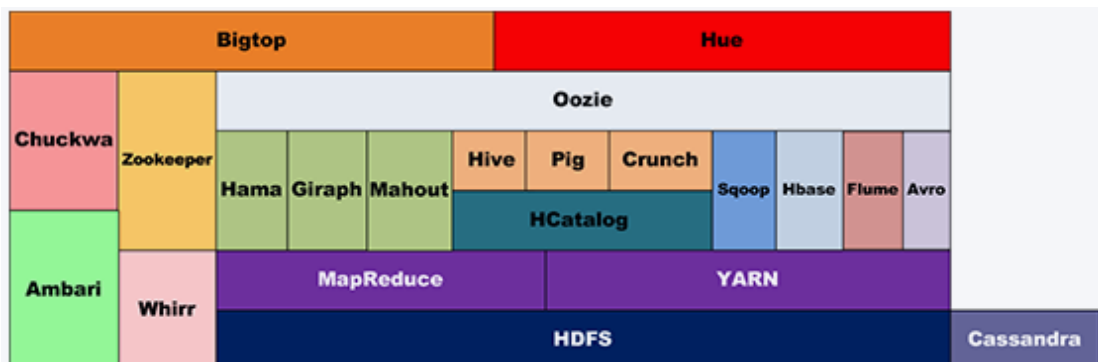


Figure 5. Hadoop Ecosystem Structure

Hadoop is not a stand-alone software project, but a framework. In the early period of this project, some companies and organization contributed their work to enrich the Hadoop project usability and extensibility. Among them, there are Apache, Inter, Cloudera, etc. But the most famous and impressive is Apache. The picture shows the Hadoop ecosystem derived from apache [6]. The following are some very common integrated modules.

Apache Hive: it is a data warehouse tool based on Hadoop. It can map structured data files into a database table. Simple MapReduce statistics can be quickly implemented through SQL-like statements.

Apache HBase: it is a highly reliable, high-performance, column-oriented and scalable distributed storage system. By using HBase technology, a large-scale structured storage cluster can be built on a cheap PC Server.

Apache Zookeeper: it is a distributed, open-source coordination service designed for distributed applications. It is mainly used to solve some data management problems frequently encountered in distributed applications, simplify the coordination and management of distributed applications, and provide high-performance distributed services

Apache Flume: a distributed, reliable, highly available mass logging aggregation system that can be used for logging data collection, logging data processing, and logging data transmission.

2.4 Web Crawler

Generally, a web crawler is a program that can automatically retrieve web pages, parses webpage contents and inner contents, download and store web content for some intents such as searching, web classification, data, and information extraction [11].

Using crawler technology, we can quickly obtain information on the site and perform a series of extractions on information that is of interest to users. This is a common method for acquiring information in the current time, and it also is an important method for collecting data and keeping up to date with the rapidly expanding Internet [12].

Roughly, categorized by the crawler structure and technologies, there are four main types of crawler – general purpose web crawler, focused crawler, incremental crawler, and deep web crawler.

The general-purpose web crawler usually starts from some of the seed URL until it covers the entire Web or meets the stop condition, the collecting data mainly from portals and large web service providers [13]. Focused crawler also named as topic web crawler, that is based on some certain webpage or applied analysis algorithms to focus the link relevant with the given topic, keep the theme of the link and put it into the URL queue to be crawled. For incremental crawler, it is like a periodical crawler for building fresh new data set when it is necessary to refresh the old collection [11]. For the data that can be directly gotten by a single HTTP request, they are in the surface web. Versa, the data that can not be gotten by a single request

like generated by some web shell or stored in databases, they are deep web data. The web spider crawling for this type of data is a deep web crawler [14].

In addition, the 'anti-crawler' strategy is a situation that will be considered in most systems, because crawlers give pressure on the server compared to normal users, resulting in bad performance. Therefore, when designing a crawler, it is necessary to consider the strategy so that crawler robust enough, avoiding being banned [8].

2.5 Web Application

Web application is an application that can be accessed through the Web. The biggest benefit of the program is that it is easy for users to access the application. Users only need to have a browser and do not need to install other software. But in functionality, the web applications are not fundamentally different from programs written in standard programming languages such as C, C++, etc.

A Web application is composed of various web components that perform specific tasks and present services to users via the Web. In practice, Web applications are composed of multiple servlets, JSP pages, HTML files, and image files. All of these components work together to provide the user with a complete set of services.

In general, a web application is divided into frontend and backend. The frontend is mainly responsible for the webpage presentation, while the backend is the implementation of the business logic. The common techniques for frontend mainly are HTML5, CSS3, and JavaScript. For the backend, it consists of database and business logic code. At present, there are lots of frameworks can help developers saving time of implementing system handler to improve their work efficiency. Nowadays as Java has become a mature backend development language, lots framework implemented by it and one of the common is SpringMVC.

2.6 Data Visualisation

Data visualization aims to convey information clearly and effectively by means of graphics. It's about to show data result in an easy way to understand and so to discover the value in the data set. Chart is a common means of data visualisation, the basic chart - bar chart, line chart,

pie chart and so on - is the most commonly used. The following describes some common charts used in data visualisation [21].

Bar charts are the most common and easiest to read. It is used in two-dimensional data sets, but only one dimension needs to be compared. The human eye is sensitive to height differences, and the recognition effect is very good. The applicable scene of bar charts are limited to small-sized and medium-sized data sets.

The Scatter chart is a two-dimensional data visualization using (x,y) dots to represent the values obtained for two different variables, which are used to show the relationship between two variables. Sometimes it will be called correlation plots because they show how two variables are correlated.

A pie chart is a circular statistical chart with several sectors to describe the relative relationship between quantities, frequencies, or percentages and its represented data set must from the same row or column in the data table.

The linear chart is suitable for two-dimensional large data set and is used to reflect the trend of a single data set. It can show continuous data that varies over time, so it is very suitable for showing trends of data at equal intervals.

3 Adopted Frameworks

3.1 YARN

Yarn is a general-purpose resource scheduling platform for providing computing resources for server, and also act as a distributed operating system platform.

Yarn does not know the mechanics of user-submitted programs. It only provides the scheduling of computing resources - when a user application requests a resource from a Yarn, it allocates the resources. The chief role in Yarn is ResourceManager, which responsible for computing resource scheduling, while the role of NodeManager is to provide computing resources. In this way, Yarn is completely decoupled from the running user program, which means that all kinds of distributed computing programs (MapReduce is just one of them) can be run on Yarn, such as MapReduce, storm, spark, and so on.

3.2 Amazon Web Service (AWS)

AWS is one of the most popular cloud computing service providers today. It offers a broad set of global computing, storage, database, analytics, application, and deployment services that help organizations migrate faster, reduce IT costs, and scale applications. Its service includes EC2, S3, VPC, RDS, Lambda and many other modules [29] .

EC2 is an Elastic Computing Cloud. Elastic computing is the flexible computing power – when user needs change computing resources it can be scaled up and down. These computing resources can be processing power, storage, bandwidth, and so forth.

IAM is Identity and Access Management. The IAM defines and manages the roles and access rights of users from a network. Its core aim is to give each user an identity.

VPC is Virtual Private Cloud, which is a dynamic configuration pool of public cloud computing resources that uses encryption protocols, tunneling protocols, and other security procedures to transfer data between user and cloud service providers

The full name of S3 is Simple Storage Service, which is an object-based storage technology. In S3, each file is considered as an object and it shall be identified by a key. A container storing file is called as bucket. Hereby, the AWS provides an API web service used to get objects in S3 by S3 bucket and object key.

3.3 Echats

Echats, a short for Enterprise Charts, is a commercial open source chart technology developed by Baidu. It supports multiple browser versions, including IE, Chrome, Firefox, etc.

Its core uses HTML5 Canvas technology, which is a library based on JavaScript. Its bottom layer uses Zrender technology (Zlevel Render, which is a lightweight Canvas library), providing rich data visualization charts, including pie charts, bar charts, line charts, scatter chart, and other types. In addition, it allows the client to make asynchronous requests to the server, supports static JSON data manipulation, and provides remote Ajax request calls [30].

3.4 HTTPClient

HttpClient is a subproject under Apache Jakarta Common, which is used to provide an efficient client programming toolkit that supports the HTTP protocol. But it does not contain all the expected functionality of a browser application. Because the HttpClient has no user interface, the browser has a rendering engine to display the page and interpret user input. So HTTPClient can only be used programmatically to transmit and receive messages content of HTTP through the API, and cannot execute JavaScript, so it is completely agnostic about the content [28].

3.5 Jsoup

Jsoup is a simple HTML code parser, which can directly parse the content of the specified URL address. Its built-in code parser will convert the source code string into a DOM object, and achieve data fetching through DOM like Javascript. [27]

3.6 SpringBoot

Spring Boot provides a minimum-configured platform to develop a stand-alone and production-grade spring application, which no need to integrated with other service modules such as Tomcat. It can be run as a java application as it includes the build-in Servlet container. It consists of two modules: spring and spring MVC [15]. In the project, SpringBoot is used to implement the Java Web application, which also reduces the complexity of development and simplifies the tedious process of file configuration.

3.7 Spring

The Spring Framework is a lightweight non-intrusive solution for building enterprise applications, whose core functionality is implemented by the Java reflection mechanism. The core concepts are Inversion of Control (IoC) and Aspect-Oriented Programming (AOP). Different from the normal programming way, the framework inverts the control of an object's lifecycle from the developer to the core container [16]. In other words, the developer will just use a configuration file to replace the 'new' keyword. For creating the instance, Spring can automatically find the appropriate object in the core container and inject the object to the aspect where needed [22].

Spring Framework Runtime Diagram The proposed system will adopt a Spring framework for managing dependency between objects so that it reduces the complexity of programming and promotes the loose coupling implementation of the system. Because Spring reads the configuration file from the server, if little changes upon system configuration, it is no need to redeploy system again. [26]

3.8 MyBatis

MyBatis is an excellent persistence layer framework that supports customized SQL, stored procedures, and advanced mapping. It avoids almost all of the JDBC code and manual parameter setting and result set extraction, just using simple XML or annotations to configure and map base bodies, mapping interfaces and Java POJOs(Plain Old Java Objects) to data [31].

3.9 Maven

Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting, and documentation from a central piece of information. The main purpose of using Maven is to managing the thrid-parts jar dependencies and packaging project into jar file [32].

4 System Requirements and Design

The project is going to create a data analysis system, which is based on Hadoop cluster on cloud, its results will be displayed on a web application.

4.1 Use Case Analysis and Specification

The system requirement is clear, and there is only one function to the system: when the user opens the webpage, the analysed news data will be displayed on webpage in charts, as shown in the following figure.

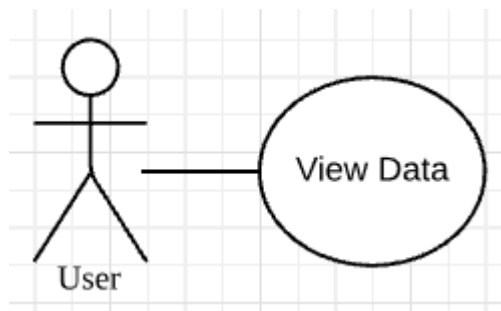


Figure 6. Use Case Diagram

- **Brief Description**

This use case allows the user to view the analysed result on the webpage.

- **Participator**

User

- **Use Case Event**

User open webpage.

- **Special Requirement**

The user must use a web browser to open the website.

- **Precondition**

Analysed result of news data must be in the web application database.

- **Postcondition**

Data representation should be in a manner of charts or diagrams.

- **Activity Chart for Use Case**

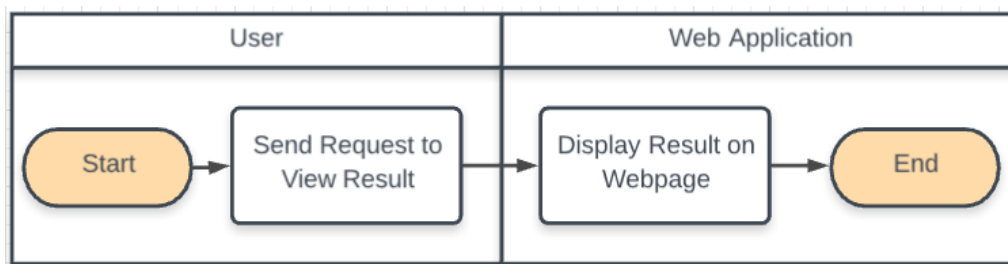


Figure 7. Activity Diagram on Use Case 1: Element Update Notification

● Activity Chart for Web Application

The activity flow diagram shows backend modular interaction between web application and database.

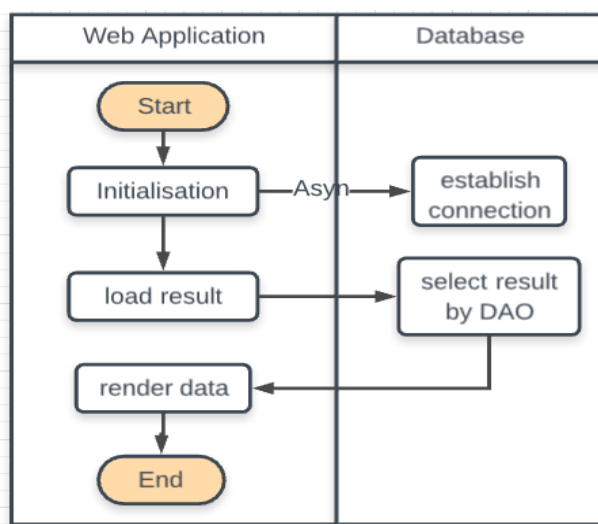


Figure 8. Control Flow at Web Application Side

4.2 System High-Level Design

The system is designed as three modules: news data collecting module, Hadoop data analysis module, and website module.

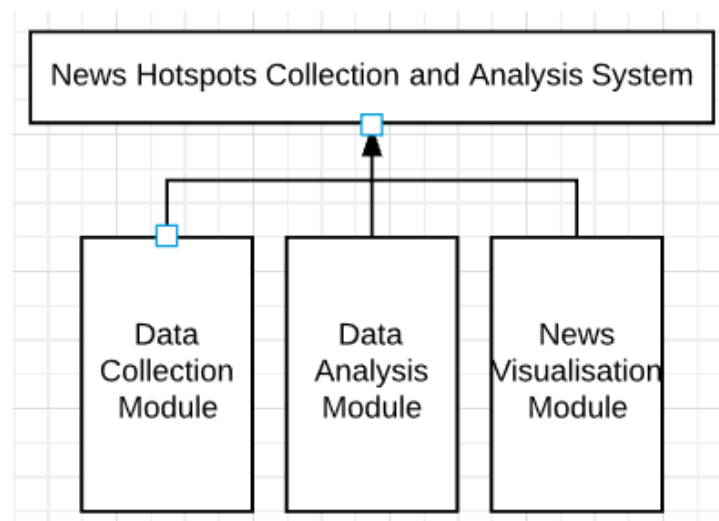


Figure 9. System High-level Design

Collecting module and analysing module communicate by the HDFS. web spider in collecting download news record data into a file on HDFS, then, the analysis module will load data from that file.

The database is a bridge between the analysis module and website module. After data analysis, the result will be written into database. And website can easily load data from it by DAO object. In addition, this way can reduce modular dependency between them.

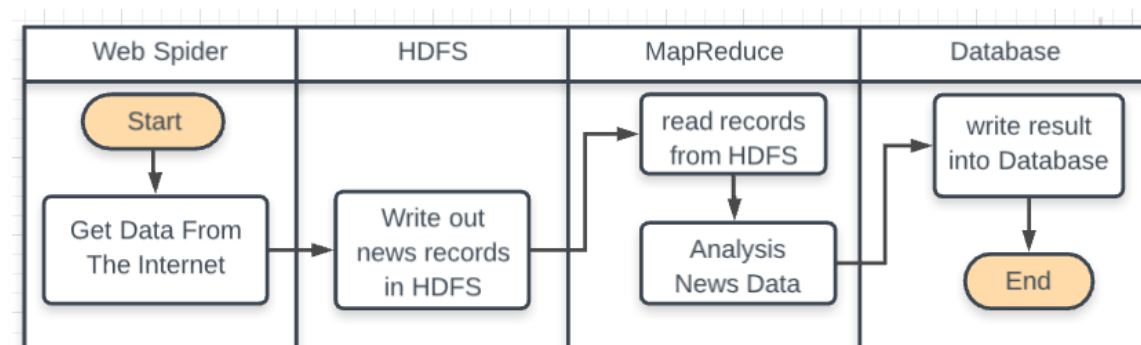


Figure 10. System Data Flow

4.3 Data Collection Module

Just as its name implies, the primary work for the module is collecting data. Here we adopt the web spider method for news data collection.

The target webpage is the 163 news rank page (<http://news.163.com/rank/>), which lists the current hot news by news type. There are two reasons for using this webpage: first of all, 163 news is an authoritative news website in China and secondly, news are listed by group can cut a lot of work for news type recognition, which may cut off the work for artificial intelligence recognition and clustering analysis.

First, the crawler will analyze the hot news leaderboard page, extract all news types and links, then obtain the data of each news in detail, and finally download the data into HDFS.

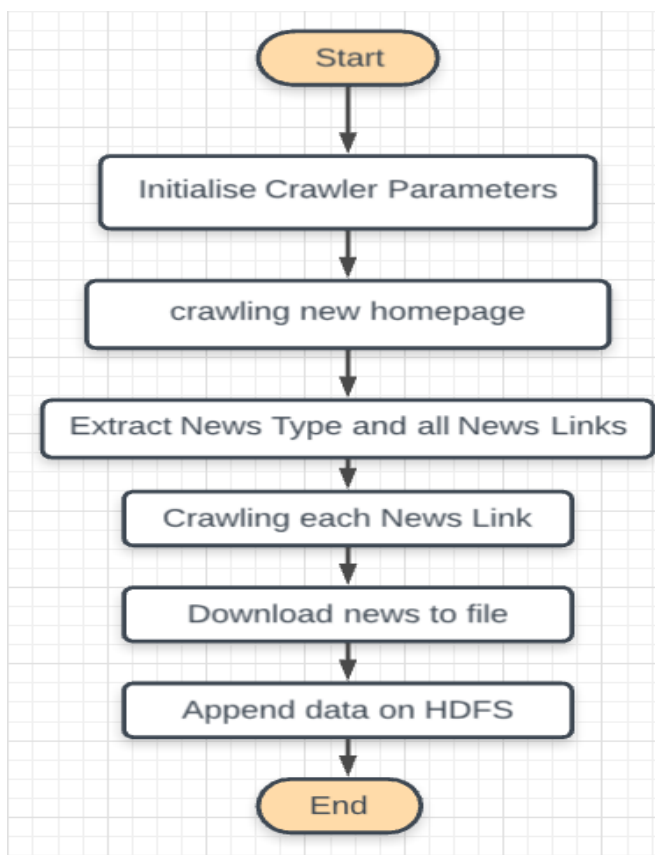


Figure 11. Data Flow in the Data Collection Module

The data collecting module has two components: Web Spider and HDFS Writing. The former is designed for executing data collection work, and the latter is used for writing data to HDFS.

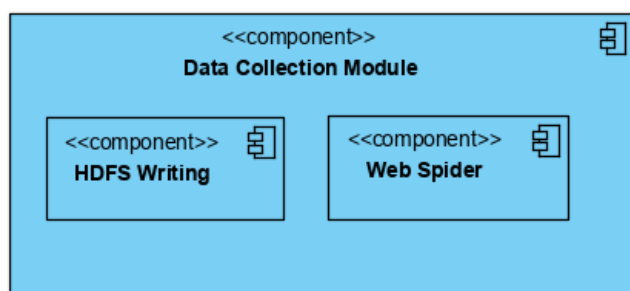


Figure 12. Data Collection Module Components

4.4 Hadoop Data Analysis Module

The main work of data analysis module is analysing data, thus the Hadoop MapReduce model plays a vital role in this part.

Data firstly will be loaded from HDFS and will be formatted. Then it will be inputted into MapReduce model for analysis and finally outputted to database.

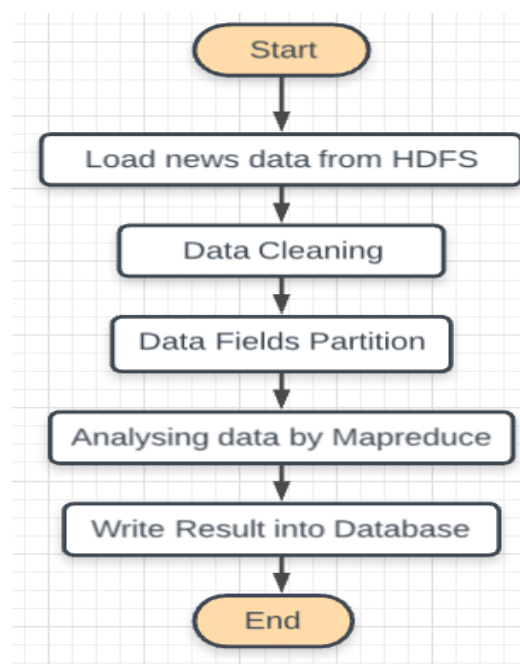


Figure 13. Data Process Flow in The Module

4.4.1 Cluster Planning

Hadoop can manage nodes in three types of modes:

1. Local Mode (default mode): No configuration file is required to be changed, and Hadoop uses a local file system instead of HDFS. It can only be used to debug the logic of the MapReduce program.
2. Fully Distributed Mode: Multiple nodes run together making it a real production environment.

3. Pseudo-distributed Mode: In logic, it equals the fully distributed model, but only one node is used – namenode, datanode, resource manager and node manager are running at one machine but a different process. This pattern is often used to develop and test whether Hadoop programs containing HDFS execute correctly.

Obviously, the project cluster must run as a fully distributed mode. But during modular MapReduce development period, local mode is enough.

The system cluster is planned as following, where four nodes form a computing cluster in the same local area network.

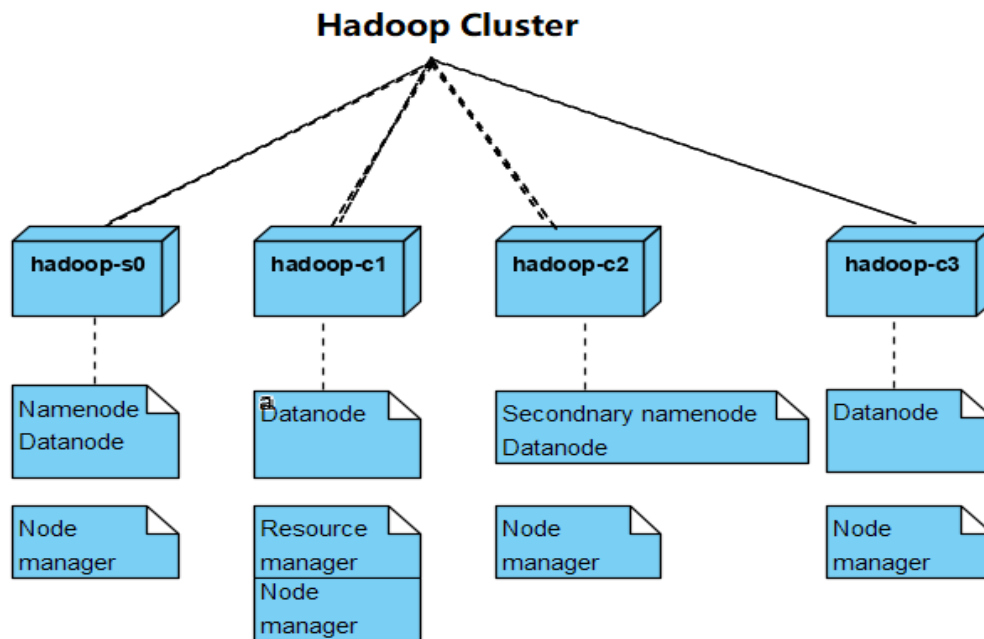


Figure 14. Cluster Plainning

In general, namenode needs to store all datanode information and send heartbeat request for checking datanode status. Thus more computing resources will be occupied compared with datanodes. Similarly, the secondary namenode is an auxiliary program of namenode, that will periodically backup namenode data onto it. Resource manager is responsible for all resources management and allocation in cluster. It receives resources from individual node managers and delivers to individual applications on YARN.

Overall, it is best not put namenode or secondary namenode together with resource manager on one machine. Thus, the namenode is configured on **hadoop-s0**, resource manager configured on **hadoop-c1**, secondary namenode configured on **hadoop-c2**, and only data node on **hadoop-c3**.

4.4.2 MapReduce Data Analysis Design

The new hot spot may reflect the current public's concern, and the comment number can directly reflect the hotspot degree of news. Thus, the helpful news data fields for each news are defined as news type, news comment number, news published time and news published source. As the data flow shows, it needs data cleaning, partitioning, analysis, and formatting output.

Thus, the data cleaning job is needed to format data content for the subsequence analysis, and a partition job also needed for partitioning a news record into four parts as the above. Each part should be stored in a file. These two jobs can be integrated by one MapReduce program [4].

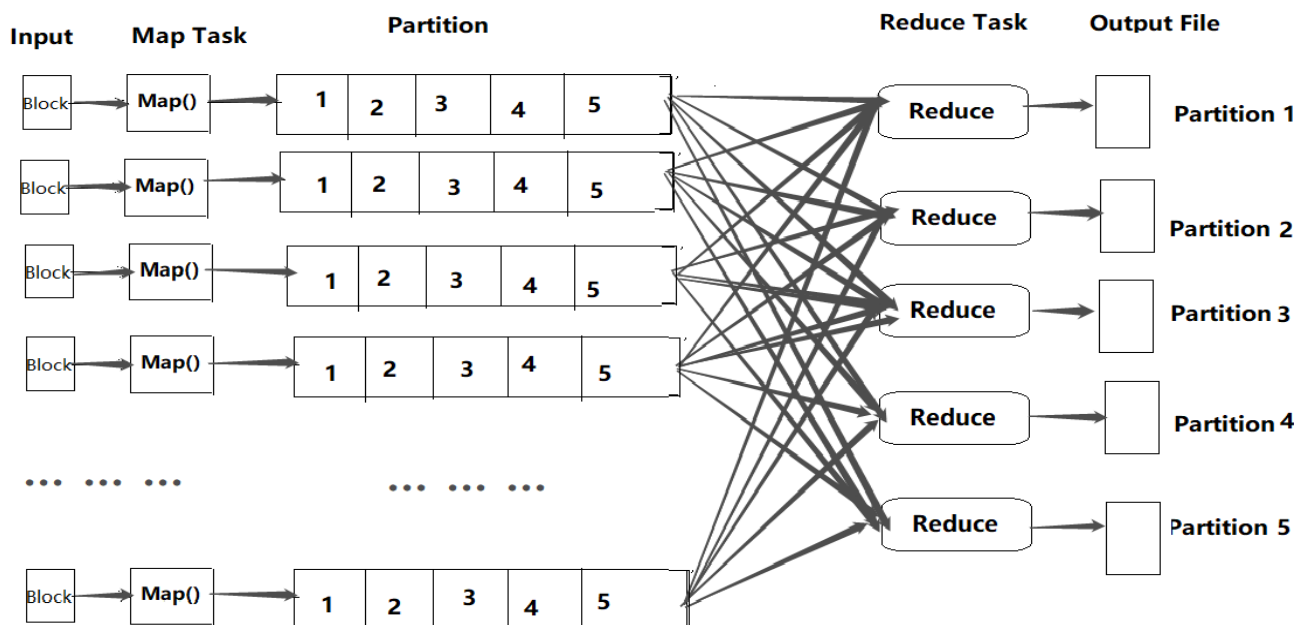


Figure 15. Map Reduce with Partition Process.

The work of the job is similar to the select-command in SQL: image the inputted file is a big table, while each output file shall be all the data from each column of the table.

Then, a word count MapReduce job shall run for counting the frequency of news type, news published time and news published source. For counting the comment for each news type, a CountMergeJob can be implemented.

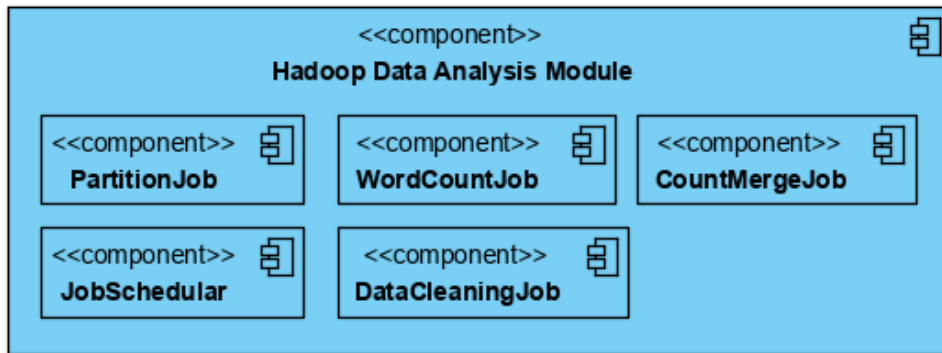


Figure 16. Data Analysis Module Components

However, the MapReduce programming model can only contain one map phase and one reduce phase. If the user's business logic is very complex, only several MapReduce programs can be run in serial. Therefore, it needs a job scheduler as the MapReduce driver for scheduling job execution sequence and configuring map, shuffle, reduce tasks among those jobs.

4.5 Website Module

The web site module takes the data from previous steps and displays the results from the database onto the front webpage.

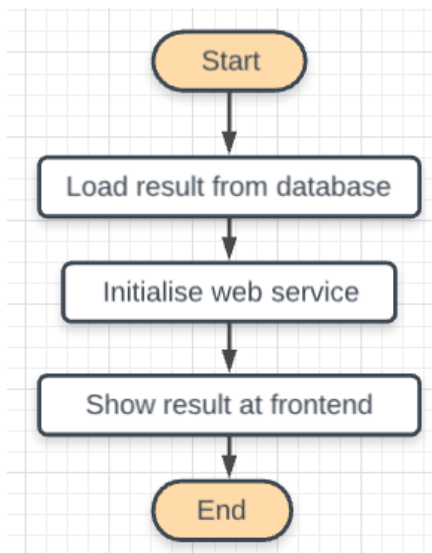


Figure 17. Execution Flow of Website Module

There are two main categories of website modules: frontend and backend. The frontend only has one component: Echats, which is used for data visualization.

The design concept of the background is based on the SSM framework, which is the integration of Spring, Spring MVC and Mybatis. Thus, it encapsulates controls of user's accessing

request, leaving all the underlying server tasks to the framework such as handling HTTP requests forwarding, enabling developers to focus on the development of business code.

Therefore, it only needs to implement three components, Controller, Service and Mapper, to make the SSM framework work.

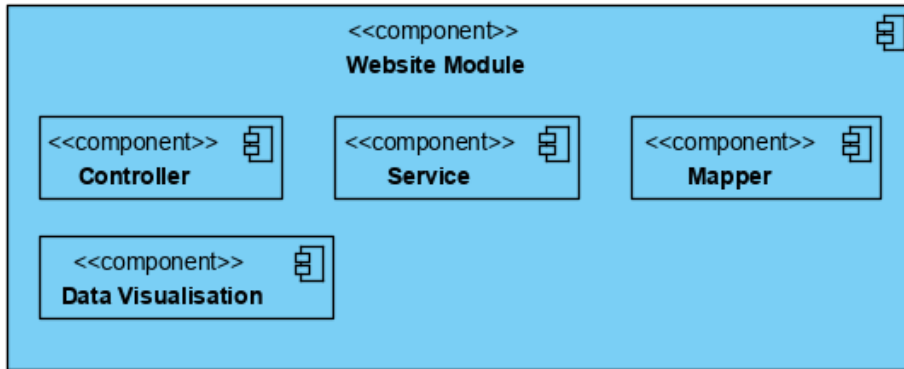


Figure 18. Website Module Components

4.5.1 Frontend Design

There is only one function for the web frontend, which is to display the result in a user-friendly manner. This part corresponds to the Data Visualisation Component at the module design diagram, which consisted of two main parts: Ajax asynchronisation request and echats.js data visualisation framework.

● Ajax Request

The full name of AJAX is Asynchronous JavaScript and XML, which is a browser-side web development technology – a mashup of web technologies. It can be used in the web application for achieving partial refresh, because it makes an asynchronous request, by XMLHttpRequest object, to the server for retrieving the data from the server, and then uses javascript to manipulate the DOM for updating webpage. Nowadays, it has become a common method for sending HTTP request in Javascript browser programming [33].

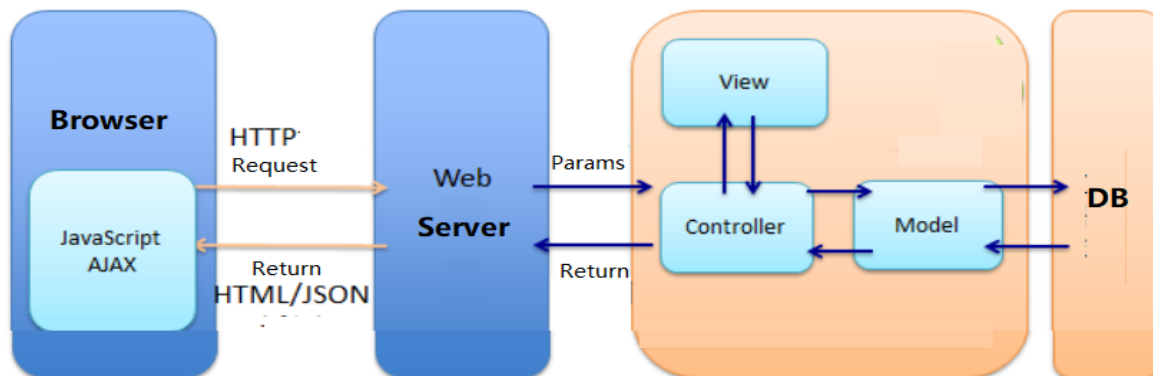


Figure 19. Process of AJAX and Backend Interaction

AJAX procedure just needs to invoke the server API to achieve data, and the API will be provided by web controllers in the module. Therefore, each AJAX procedure should correspond to the API provided by controllers.

● Data Visualisation

The webpage will display the analysed data in a graphic format, and here is a design for which type of graph to be used for each type of news.

The Linear chart is good at representing data trends at time intervals, so it graphing news published time is applicable.

By using the pie chart to represent the news type data, it can clearly see the proportion of each news type in the total news type, thus discovering the news type with the largest proportion.

The total number of comments for each type of news can be shown in a bar chart to find the most popular types of news. Similarly, news published sources can be represented by a bar chart as well.

4.5.2 Backend Architecture

● Spring MVC Introduction

The website is based on spring MVC, which is a web framework based on MVC, MVC refers to the design concept of control layer (C), module layer(M) and view layer (V).

Spring MVC's request process includes the following steps, which also are typical steps for MVC components interaction [16].

- Step 1: Make a request to the frontier controller (DispatcherServlet)
- Step 2: The frontier controller requests HandlerMapping to find the Handler, which can be found based on XML configuration and annotations
- Step 3: HandlerMapping returns Handler to the frontier controller
- Step 4: The frontier controller calls the HandlerAdapter to execute the Handler
- Step 5: The HandlerAdapter performs the Handler
- Step 6: Handler completes execution and returns ModelAndView to the adapter, ModelAndView is an underlying object of the spring MVC framework, including Model and view
- Step 7: HandlerAdapter returns ModelAndView to the frontier controller.
- Step 8: The frontier controller requests the view parser to parse the view, and then parses the logical view name into a real view (JSP)
- Step 9: The View parser returns the View to the frontier controller
- Step 10: The front controller renders the view. View rendering putting the model data (in the ModelAndView object) into the request field
- Step 11: The frontier controller responds to the users

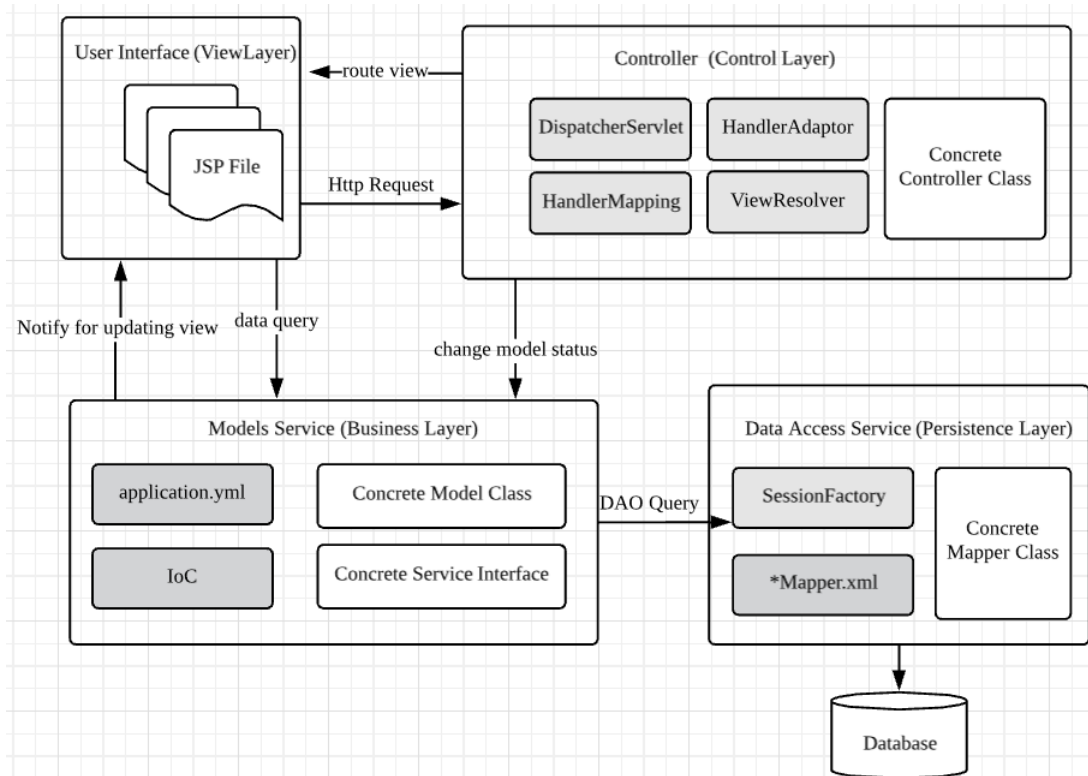


Figure 20. Backend Three Layer Interaction Based on Spring MVC

● Web API Design

However, we are not worried about the implementation of the entire framework process, as these works are left to the SpringMVC framework.

In order to implement sending data results to front-end AJAX, it must develop REST API to provide web services. And it is the only thing need to do at website background.

API URL	HTTP	Function
/index	GET	Return index.html
/comment/list	GET	Return top 10 list of news types by comment count number
/source/list	GET	Return top 10 list of news publish press by the count number
/publish/list	GET	Return the list of all news published time and count
/type/list	GET	Return the list of all news types and count number of the type.
/type/sum	GET	Return the sum of news

Table 1. REST API Specification

● Database Design

The system adopts relational database MySQL which is a Relational Database Management System (RDBMS) and it is open source and free to use. MySQL database is apply to store structured data, which means its format should be defined before saving it, rather the unstructured or semi-structured data downloaded from web crawler. And also, it is why MySQL database can not be used to store massive big data.

There are four tables defined in the database. They are:

- **t_publish:** storing news published time by two integer properties time_hour and publish_cnt, the former is the published time of a news and the latter is the count number for that news.
- **t_comment:** Type is news type, a varchar type data. Comment_sum is integer type data storing the comment number for each of news type.
- **t_type:** Type is news type, a varchar type data. Count is an integer type which means the number of news types that fall into this category.
- **t_source:** Type is news published source, a varchar type data Source_cnt is an integer type data, used to store the count of news from one publish press.

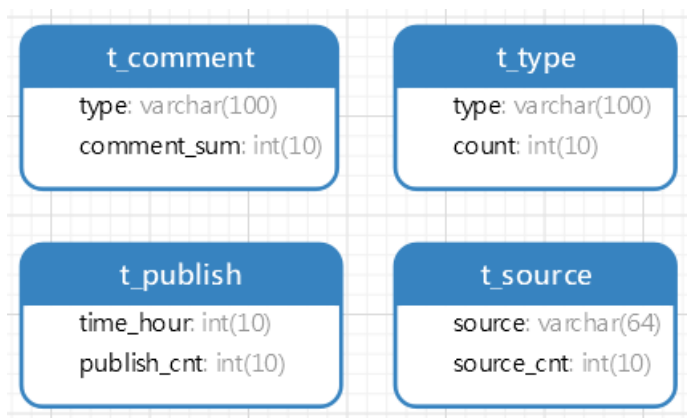


Figure 21. Designed Tables at Database.

5 Implementation Details

5.1 System Environment Preparations

Hadoop distributed system can run at multiple operating systems (OS), including windows, Linux, Unix and so forth. The OS adopted in this project is Linux CentOS 6, which is open source and free to use.

5.1.1 Installing JRE, Hadoop and MySQL

Hadoop is developed by Java, the Java Runtime Environment (JRE) should be installed on Centos operating system ahead of installing Hadoop. Its installing package can be downloaded on Oracle official website.

After downloading the package, whose extension generally end of '.tar.gz', executing the tar -zxvf -C command can unpackage all JRE files. And then, the system environment variable should be configured at file '/etc/profile'. After that, executing 'source /etc/profile' command to renew the system variable and Java programs can be run at the machine.

The next step is to install Hadoop, whose process is similar to the JRE. Download Hadoop source file from Apache Hadoop mirrors website and unzip files in the centos system. Then, append the installing path to 'profile' file and 'source' again. The Hadoop is installed.

```
#JAVA_HOME
export JAVA_HOME=/opt/module/jdk1.7.0_79
export PATH=$PATH:$JAVA_HOME/bin

#HADOOP_HOME
export HADOOP_HOME=/opt/module/hadoop-2.7.2
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin

[hadoop@hadoop-c3 ~]$ source /etc/profile
[hadoop@hadoop-c3 ~]$ echo $HADOOP_HOME
/opt/module/hadoop-2.7.2
[hadoop@hadoop-c3 ~]$ echo $JAVA_HOME
/opt/module/jdk1.7.0_79
```

Figure 22. System Environment Variables

MySQL can be gotten by 'yum' at centos by executing the following command:

```
yum install -y mysql mysql-server mysql-devel
```

MySQL will automatically be installed by yum.

5.1.2 Configuring SSH, Management Shell and Network.

SSH is the Secure Shell Protocol, which guarantees data security and reliability in data transferring. Hadoop requires SSH access as the communication between datanodes should be secure and reliable. It implemented by RSA asymmetric encryption algorithm, a private key and a public key will be generated after initialisation. The public key will be known by all cluster nodes, but the private is only store in local account.

Under normal production environment, it is not a single node to run Hadoop – it must above three machines. Thus, if there is a new node added into the cluster, to modify the configuration file on each node is tedious work. To solve the problem, a shell can be written for file synchronising on each cluster node. Here, the ‘xsync’ and ‘xcall’ shell are used for synchronising file and executing command on each cluster node. (Details on Appendice B.)

Till now, all the required software are installed on one machine, the next step is to clone this one for 3 duplications. However, a problem is the cloned machine has more than one MAC address. Therefore, network configuration is necessary for the cluster. Firstly, the MAC address should be changed – delete the matrix machine MAC and use the new one on file ‘/etc/udev/rules.d/70-persistent-net.rules’. The next step is to configure each hostname, IP address and host file so that to form a cluster within a local area network. Finally use ‘xsync’ command to deliver every configuration file on each node.

5.1.3 Running Hadoop and MySQL Service

However, so far it still can not run Hadoop cluster, as configuration files have not been modified yet. To modify the following files under path of \$HADOOP/etc/hadoop: “core-site.xml”, “hdfs-site.xml”, “mapred-site.xml”, “yarn-site.xml”, and “salves”. The “salves” file defines the hostname of datanode. After the above steps, it needs to initialise Hadoop cluster by executing command:

\$HADOOP_HOME/bin/hadoop namenode format

As yarn resource manager and namenode are configured at a different machine, we can use ‘start-all.sh’ to start Hadoop and Yarn as usual. Firstly, run the following command to start Hadoop distributed file system:

\$HADOOP_HOME/bin/start-dfs.sh

When console window prints out namenode, datanode and secondary namenode are successfully started, we run the command to start Yarn for running MapReduce.

`$HADOOP_HOME/bin/start-yarn.sh`

After resource manager and node manager successfully started, the whole Hadoop cluster is fully running. Details for configuration can be founded at Appendice B.

To start the mysql service, just to execute command at Linux: ‘service mysqld start’. First time run the command may have a process for initialisation, and a root account with password will be initialised as well.

5.2 Data Collection

In this section, class organisation structure and data collection business logic are described.

5.2.1 Web Spider Implementation

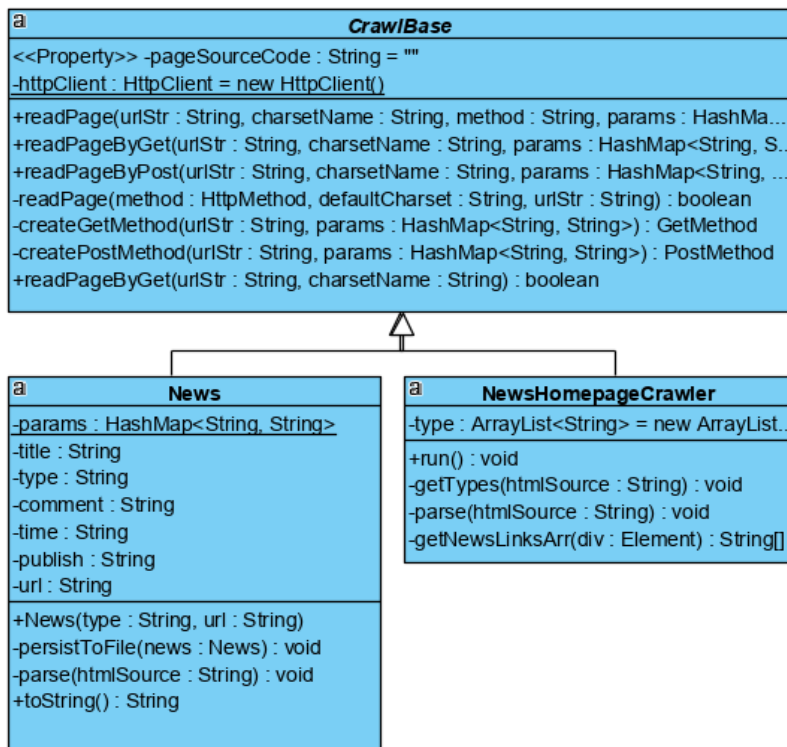


Figure 23. Class Diagram of Web Spider

CrawlBase class is an abstract class defining methods for getting the HTML source code of an online webpage. It encapsulates the usage of `HttpClient` with `readPageByGet` method and `readPageByPost` method. The field ‘pageSourceCode’ is used to store the downloaded webpage source code.

NewsHomepageCrawler and **News** are subclasses of web spider class ‘**CrawlBase**’, where the method `parse(String htmlSource)` defines the concrete business logic and specific crawling

data for each webpage. The former defines the business logic for the homepage web spider, which will get all news links on the homepage and initiate a News object for the later data crawling. Class News not only defines the logic for crawling data for news but also stands for a news record entity.

5.2.2 Web Spider Business Logic Description

To write a vertical crawler, the first step is to determine what kind of data the crawler needs to download. Therefore, it is necessary to analyse the source code of the target webpage.

For convenience, enter the URL of the HTTP in a browser, and open the browser's built-in web source code inspector. By analysing the strings in the web source code, summarise the uniform rules in text to extract the satisfactory text information.

The following will detail how to extract data from the news homepage and a news page:

● Web Spider *NewHomepageCrawler*

Open the news homepage, which contains all news types and news links of the website.

Through these links, crawlers can obtain data from the news list, and all data fields can be obtained. Open the browser web source code inspector, which is as follows:



Figure 24. Webpage Source Analysis with Browser Inspector

After analysing the code structure of the web page, it is found that the news type is enclosed in the red box. The DOM element of the news can be expressed as ".titlebar >h2" in CSSQuery.

In this way, all types of news are available to be obtained. Then continue to look at the web page, found that the news DOM element at news list can be represented, starting with each "class" attribute "titleBar" div, it can be represented as: "div>div>div>table>tbody>tr>td>a".

Jsoup can either directly execute CcssQuery to locate elements, or it can use JavaScript like DOM relationships to locate elements. A Java code for locating news DOM element can be expressed as the following.

```
private String[] getNewsLinksArr(Element div) {
    ArrayList<String> urls = new ArrayList<String>();

    Element bigdiv = div.nextElementSibling();
    Elements aEles = bigdiv.select("div>div>div>table>tbody>tr>td>a");
    for (Element a : aEles) {
        String url = a.attr("href");
        urls.add(url);
    }

    return urls.toArray(new String[urls.size()]);
}
```

Figure 25. Implementation of Extracting All News Link

```
private void parse(String htmlSource) {
    Document html = Jsoup.parse(htmlSource);
    Elements types_divEles = html.getElementsByClass("titleBar");
    int length = types_divEles.size();
    for(int i=TypeConst.type_boundary_index_start ; i<TypeConst.type_boundary_index_end; i++) {
        String [] urlList = getNewsLinksArr(types_divEles.get(i));
        for (String url : urlList) {
            News aNews = new News(type.get(i), url);
        }
    }
}
```

Figure 26. Web Spider Logic for Processing News Homepage

Within the parse(String) method, news crawler will be created and send request to news website server. It probably the website server backend anti-crawler program may detect the crawler and ban it because of the fake HTTP Request and high-frequency accessing.

Open the 'network' tab on the browser inspector, the whole network traffic will be intercepted and displayed. Then, open the one that requests to GET the webpage, the details of HTTP

header are displayed. With the help of these header fields, it can make an HTTP Header spoofing anti-crawler program.

Details of a specific HTTP request are shown in the following figure:



Figure 27. HTTP Request Header for Getting Webpage

● Web Spider News

Open a news details webpage, the required data field displayed on the page are news title, comment number, publish time and publish source.

The second step is to open browser inspector viewing and analysing HTML code structure and summarise rules to locate target data fields.



Figure 28. Inspector on News Details Page

The big div DOM with the id of 'epContentLeft' contains all needed fields. Similar to the above web spider class, CSS Query and Jsoup DOM element can accurately locate these fields.

```
private void parse(String htmlSource) throws HttpException, IOException {
    Document html = Jsoup.parse(htmlSource);

    Elements divEles = html.getElementsByClass("post_content_main");
    if (divEles.size() == 1) { // if it is not empty (it is exist)
        Element bigdivEle = divEles.get(0); // bigdiv element contains title time and source of publish
        Elements title_h1_eles = bigdivEle.getElementsByTag("h1");

        if (title_h1_eles.size() == 1) {
            setTitle(title_h1_eles.get(0).text());
        }

        Elements time_source_divEles = bigdivEle.getElementsByClass("post_time_source");
        if (time_source_divEles.size() == 1) {
            Element time_source_divEle = time_source_divEles.get(0);
            String innerHTML = time_source_divEle.text();
            String [] time_ = innerHTML.split("来源");
            setTime(time_[0]);

            Element sourceEle= time_source_divEle.getElementById("ne_article_source");
            setPublish(sourceEle.text());
        }
    }
    setComment(htmlSource);
}
```

Figure 29. Web Spider Logic for Processing News Details Page

However, the comment data on webpage – 10849, cannot be directly grabbed by web spider as the result from crawler is 0. After analysis network traffics, it finds the data is returned by a javascript callback function where the following link is invoked by an AJAX asynchronous call: [http://comment.news.163.com/api/v1/products/"productKey"/threads/"docId"](http://comment.news.163.com/api/v1/products/)

The productKey and docId refers to the 163 product ID and news ID, which can be found in the HTML source code of news details webpage. By sending request to the API, a JSON Object will as return, where the comment number is defined.

The screenshot shows a web browser with a news article. The article title is "首富儿媳生儿奖200万?小叔子:嫁进10年不大声说话". The article is from 2019-08-23 09:27:43, source: 红星新闻. The article content includes a video player and a comment section. The comment section shows a comment count of 10849. The network tab shows the JSON callback data for the comment, which includes the comment count (10849) and other metadata.

JSON Object

```

{
  "cmtAgainst": 6,
  "cmtCount": 10849,
  "cmtVote": 10409,
  "createTime": "2019-08-23 09:27:44",
  "docId": "EN8LOM0P0001899N"
}

```

Figure 30. AJAX Callback of News Comment

The following defines steps to extract comment field in JSON.

```

private void setComment(String productKey, String docId) throws HttpException, IOException {
    if (!readPageByGet(CmtCountHelper.getNewsCommentsApi(productKey, docId), "utf-8")) {
        this.comment = "0";
    } else {
        String jsons = getPageSourceCode();
        JsonObject json = (JsonObject) new JsonParser().parse(jsons);
        Object cmt = json.get("cmtCount");
        this.comment = "" + cmt;
    }
}

```

Figure 31. Logic for Getting News Comment

5.3 Cloud-based Hadoop Data Analysis

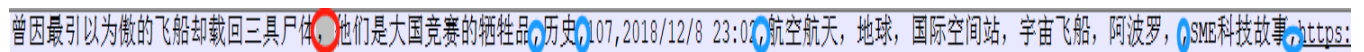
The section introduces the process of data analysis and cloud cluster setup. Data will be loaded from HDFS and first step to do on that is cleaning, and then MapReduce programs will make analysis, the result shall be outputted into MySQL database, which can be specified in Hadoop job configuration. And finally, Hadoop cluster will be put on AWS cloud computing platform.

5.3.1 Data Cleaning

The data from web spiders can not be sure it can be ready to use. The data cleaning work should be ahead of data analysis. When we make data analysis, it is not necessary to use every field in the table. Just some valuable fields are extracted.

The following shows all data fields for a news record, each field is split by a comma, which is circled by blue pen. In sequence, they are news title, news comments number, news type, news published time, news tags and the news published source.

In this context, the four valuable columns are news comments number, news type, news published time and the source of that.



曾因最引以为傲的飞船却载回三具尸体, 他们是大国竞赛的牺牲品, 历史, 107,2018/12/8 23:02, 航空航天, 地球, 国际空间站, 宇宙飞船, 阿波罗, SME科技故事, https:

Although we can see the original data is nearly to formatted data, however, data pre-processing is indispensable. The formatted data should be easy to be splited.

But there are some two-bits-size comma ' , ', casuing ambiguous spliting - as it may appears in the first field to mix up splited data fields, it is necessary to replace all two-bits-size comma with the one-bit-size comma.

```

private void replaceComma() throws FileNotFoundException, IOException {
    String lines[] = readFile();
    for (String line: lines) {
        int line_num = 0;
        if( Pattern.matches("[\\u4e00-\\u9fa5]{1,},[\\u4e00-\\u9fa5]{1,}", line)){
            int lastCommaIndex = line.lastIndexOf(",");
            int commaCount = 1;
            while (commaCount < 6) {
                lastCommaIndex = line.lastIndexOf(",", lastCommaIndex);
                commaCount++;
            }
            String newTitle = line.substring(0,lastCommaIndex).replace(",",". ");
            lines[line_num] = newTitle+line.substring(lastCommaIndex);
        }
        line_num ++;
    }
    String fileStr = "";
    for(String aLine : lines) {
        fileStr += aLine;
    }
    writeFile(fileStr);
}

```

Figure 32. Data Cleaning – Comma Replacement

5.3.2 MapReduce Programs

In this section, each component of the data analysis module will be discussed. It mainly consists of few MapReduce programs and a job scheduler for managing these program. The main logic for map and reduce task and some important details will be covered here.

5.3.2.1 JobScheduler

Job scheduler defines data analysis flow: Firstly, the file shall be partitioned for the subsequence analysis work, so to create an ExtractDriver instance with input file and the output file path for storing result files. In case of code be executed in multi-threads, ‘while’ statement here is a synchronisation. Then, news type, news published time, news published source will be analysed by word count job, and the comment for each news type shall be analysed finally. When all the jobs are finished, true as the return and the analysed return will be stored in database.

```

public class JobShedular {
    public boolean analysis() throws Exception {
        boolean isFinishedParition = new ExtractDriver()
            .run(FileIPathConst.InputfilePathStr,FileIPathConst.tempDir);
        while(!isFinishedParition) {} // block the thread until it finish

        boolean isFinishedWordCount_type = new WordCountDriver().run(FileIPathConst.NewsTypeFileDir);
        boolean isFinishedWordCount_source = new WordCountDriver().run(FileIPathConst.NewsSourceFileDir);
        boolean isFinishedWordCount_time = new WordCountDriver().run(FileIPathConst.NewsTimeFileDir);
        boolean isFinishedMerge_comment = new JoinDriver().run(FileIPathConst.NewsTypeFileDir);

        return isFinishedWordCount_type && isFinishedWordCount_source
            && isFinishedWordCount_time
            && isFinishedMerge_comment;
    }
}

```

Figure 33. Scheduling Flow for MapReduce Jobs

5.3.2.2 PartitionJob

In partition job, the input key-value is <LongWritable, Text> which stands for the line number and line text of the current line mapper program reading. The business logic code is defined within the map () method. We will use a comma ',' as a splitter to extract the four fields at each line. And a partition program is implemented for extracting four columns into four files separately.

```
public class ColumnsPartition extends Partitioner<NullWritable, Column>{
    @Override // 1-type; 2-comment; 3-time; 4-source;
    public int getPartition(NullWritable id, Column field, int numPartitions){
        if (field.getCol_type()==1) {
            return 1;
        } else if (field.getCol_type()==2) {
            return 2;
        } else if (field.getCol_type()==3) {
            return 3;
        } else if (field.getCol_type()==4) {
            return 4;
        } else {
            return 0;
        }
    }
}
```

Figure 34.Partition Planning

The output key-value is < NullWritable, Column >, where the key type is NullWritable as there is no need to be sorted, and Column is a light-weighted serialised bean implemented to carry the four fields information for each news, which is also the input key-value format of reduce task.

```
public class Column implements Writable{
    private int col_type; // 1-type; 2-comment; 3-time; 4-source;
    private String field_1;
    private String field_2;

    @Override
    public void readFields(DataInput in) throws IOException {
        this.field_1 = in.readUTF();
        this.field_2 = in.readUTF();
        this.col_type = in.readInt();
    }

    @Override
    public void write(DataOutput out) throws IOException {
        out.writeUTF(field_1);
        out.writeUTF(field_2);
        out.writeInt(col_type);
    }

    @Override
    public String toString() {
        if (null == field_2 || field_2.isEmpty())
            return field_1;
        else
            return field_1+","+field_2;
    }
}
```

Figure 35. Column Bean

Reduce task only output Column object into file. The key-value pair format is <Column, NullWritable>.

5.3.2.3 WordCountJob

Word count program is used to count up the frequency for each word appeared in a file. It can be utilised for counting news types, published time and news published sources. The input key-value of map task is <LongWritable, Text> as well and the output format of that is <Text, IntWritable>, where Text is a word, IntWritable is 1. Because the map task each time read a line is actually read a word, and its count is one, then the word will be outputted from map task. During the shuffle process, key-value pair will be partitioned, sorted and merged, and then transferred to reduce task. Reduce task will accumulate all the same keys, and then output the accumulation results to the database.

5.3.2.4 MergeCountJob

This is the core part of news statistics. The job counts total comments for each news type. Its functionality is similar to the SQL command:

```
select count(comment_num) from `news` group by news_type;
```

As previously said, each field of news are partitioned into different files, thus before the map task execution, two fields (news type and news comment) should concatenate together as the input key and value of map task. The following override the setup(Context c) method in map task, for adding file of news type as the distributed cache, which achieves multiple files as map task input.

```
public class JoinMapper extends Mapper<LongWritable , Text , Text, IntWritable>{
    private ArrayList<String> comment = new ArrayList<String>();
    @Override
    protected void setup(Context context) throws IOException, InterruptedException {
        // read all comments in the part-r-00002 file, and cache them in list.
        BufferedReader reader = new BufferedReader(
            new InputStreamReader(new FileInputStream(new File("part-r-00002"))));
        String line;
        while(StringUtils.isNotEmpty(line = reader.readLine())) {
            comment.add(line);
        }
        reader.close();
    }
}
```

Figure 36. Adding File as Map Task Distributed Cache

5.3.3 AWS Cloud Implementation

After correctly coding MapReduce programs on the local development environment, the cluster running at virtual machines will be pushed on the cloud so to make the cluster as a truly distributed cloud computing system.

The specific steps to push the cluster on AWS are: creating S3 Object Storage Bucket, installing AWS CLI, configuring IAM, exporting local mirror to AWS S3 bucket, creating an EC2 instance from S3 mirror.

● Creating S3 Bucket

It requires the Amazon S3 bucket to store the exported mirror in the area where the AWS VM is to be imported.

● Installing AWS CLI

AWS Command Line (CLI) is a python-based Vmware export/import tool. Before installing AWS CLI, python shall be installed.

To Install AWS CLI, just run the command:

```
pip3 install awscli --upgrade
```

● Configuring IAM

Each of the specific actions performed in an AWS account requires configuration permissions, so it needs a role that can create a disk image that performs the download from the Amazon S3 bucket. According to the AWS specification, the role must be created with the name of 'vmimport' and attaching a policy document stating the trust relationship, and then the IAM policy shall be appended to that role.

The following JSON file should be named as 'trust-policy.json', which specifies the trust relationship policy for 'vmimport'.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "Service": "vmie.amazonaws.com" },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:Externalid": "vmimport"
        }
      }
    }
  ]
}
```

Figure 37. Code of trust-policy.sjon

Then, executing the following command: Creating a role named 'vmimport' and provide VM Import/Export access permission to that role.

```
aws iam create-role --role-name vmimport --assume-role-policy-document trust-policy.json
```

Create a file called role-policy.json and write the following policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::mirror-bucket",
        "arn:aws:s3:::mirror-bucket/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:ModifySnapshotAttribute",
        "ec2:CopySnapshot",
        "ec2:RegisterImage",
        "ec2:Describe*"
      ],
      "Resource": "*"
    }
  ]
}
```

Figure 38. Code of role-policy.json

Use the following put-role-policy command to mount the policy to the role 'vmimport':

```
aws iam put-role-policy --role-name vmimport --policy-name vmimport --policy-document \
import\role-policy.json
```

● Pushing Cluster on AWS

Frisky, exporting local virtual machine mirror as '.ova' format file and open AWS web console to upload the file on S3 bucket.

And then, creating a JSON file named as 'containers.json', specifying the file format, file description and target S3 bucket location:

```
[
  {
    "Description": "hadoop disk",
    "Format": "ova",
    "UserBucket": {
      "S3Bucket": "the-hadoop-bucket",
      "S3Key": "hadoop-s0.ova"
    }
  }
]
```

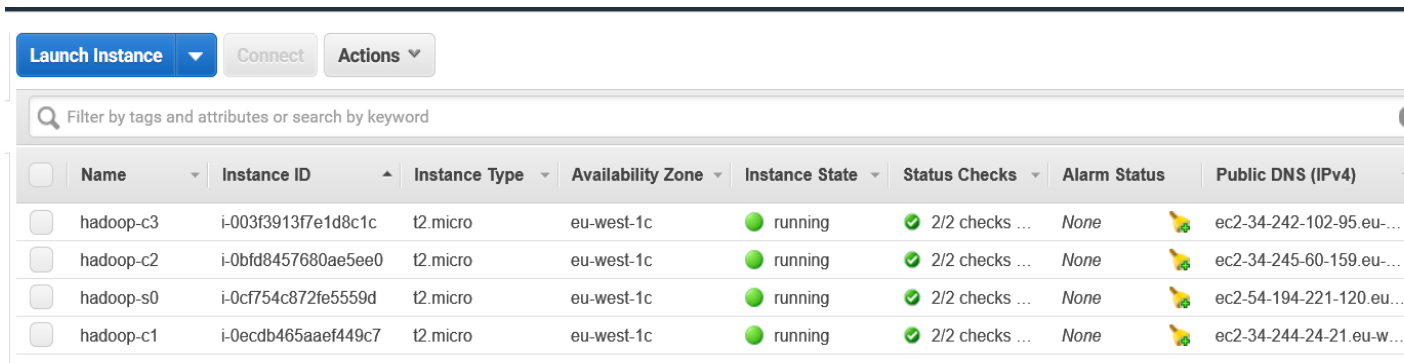
Figure 39. Code for containers.json

And run the following command to creating an EC2 instance.

```
aws ec2 import-image --description "hadoop-s0" --disk-containers containers.json
```

Then, an Amazon Mirror Image (AMI) is created. The next step is creating EC2 instances as usual, the only difference is to use the AMI just created by virtual machine mirror.

Here is an example for push hadoop-s0 mirror onto the AWS cloud. For other nodes, hadoop-c1,2,3 are in a similar fashion. Finally open AWS EC2 console, we will get running cluster on AWS.



	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
<input type="checkbox"/>	hadoop-c3	i-003f3913f7e1d8c1c	t2.micro	eu-west-1c	running	2/2 checks ...	None	ec2-34-242-102-95.eu-...
<input type="checkbox"/>	hadoop-c2	i-0bfd8457680ae5ee0	t2.micro	eu-west-1c	running	2/2 checks ...	None	ec2-34-245-60-159.eu-...
<input type="checkbox"/>	hadoop-s0	i-0cf754c872fe5559d	t2.micro	eu-west-1c	running	2/2 checks ...	None	ec2-54-194-221-120.eu-...
<input type="checkbox"/>	hadoop-c1	i-0ecdb465aaef449c7	t2.micro	eu-west-1c	running	2/2 checks ...	None	ec2-34-244-24-21.eu-w...

Figure 40. EC2 Console Window

When creating an EC2 instance, AWS will ask the user which VPC the instance shall located at. Here the four nodes should be in one LAN as they form a cluster, thus they should be in a similar VPC network. And it does not need to manually configure network as each node IP address shall be assigned automatically by AWS VPC. The only thing needs to do is to re-configure SSH for each instance as discussed in section 5.1.2.

5.4 Website Implementation

5.4.1 Back-end MVC Implementation with SSM

From the class diagram below, the system implements four MVC-based services. The API controller calls the service layer interface which is implemented by specific concrete classes internally, thus to achieve the purpose of decoupling between the control layer and service layer. And the service layer calls the underlying data layer, for reducing the dependence of the business logic layer and the data layer.

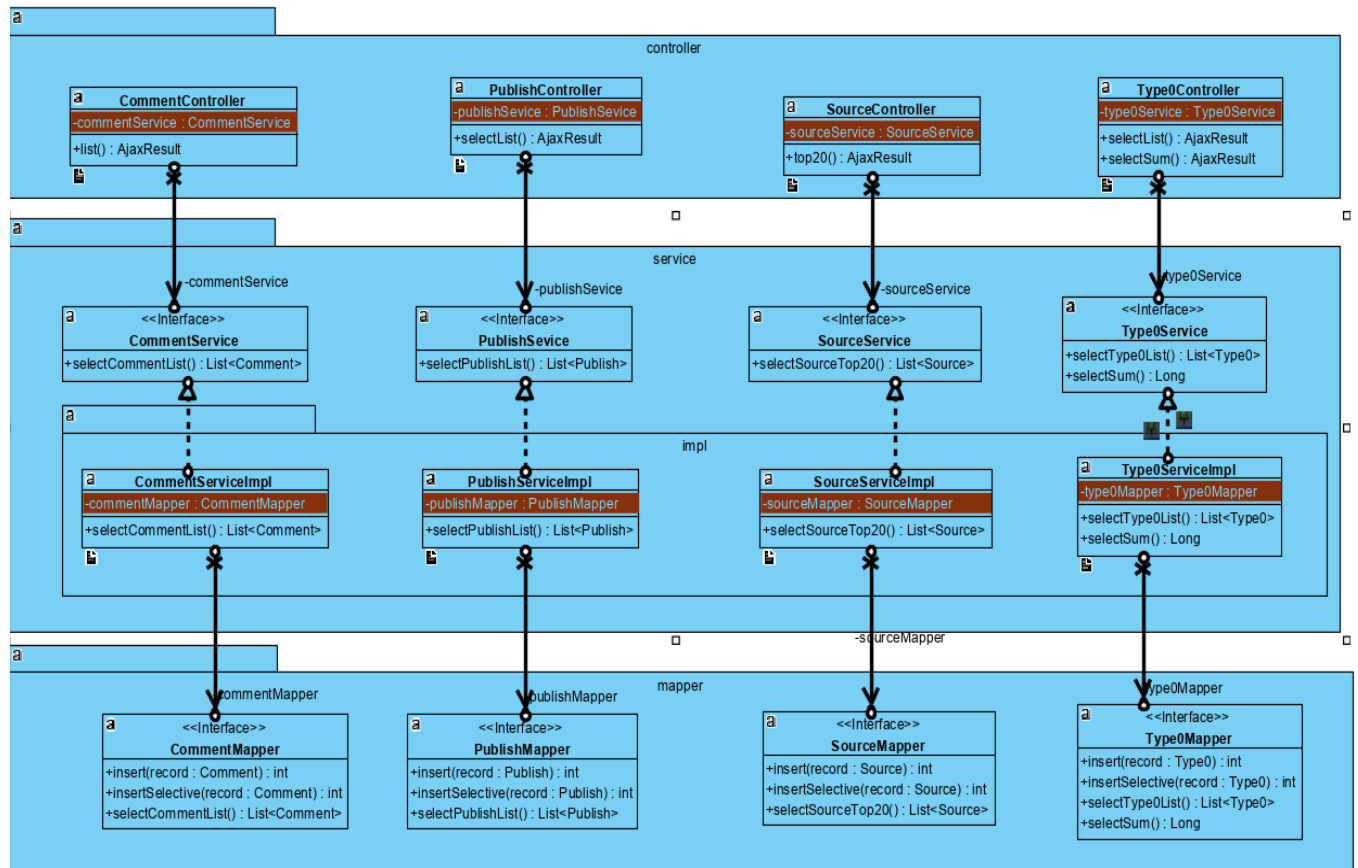


Figure 41. Class Diagram for Website Module

● Controller

All controllers in the module should return an `AjaxResult` object as the frontend callback result. For example, when an HTTP GET request to the location of `/comment/list`, the `list()` method within `CommentController` Class will be automatically invoked by SpringMVC framework, then the business logic code will be executed, and delegate service execution to corresponding bean in Spring container. Finally, the result will be put in the JSON with field name of `'comments'`.

```

@Controller
@RequestMapping("/comment/")
public class CommentController {
    @Autowired
    private CommentService commentService;
}
@GetMapping("list")
@ResponseBody
public AjaxResult list() {
    List<Comment> comments = commentService.selectCommentList();
    return AjaxResult.success().put("comments", comments);
}
}

```

Figure 42. Controller Examples - CommentController

● Service

The service package defines the system interface between controller layer and persistence layer. For each interface, it needs at least one class to implement it. Service defined in the module are used to invoke DAO for database querying.

```

public interface CommentService {
    List<Comment> selectCommentList();
}

```

Figure 43. Service Interface Example - CommentService

```

@Service
public class CommentServiceImpl implements CommentService {
    @Autowired private CommentMapper commentMapper;
    @Override
    public List<Comment> selectCommentList() { return commentMapper.selectCommentList(); }
}

```

Figure 44. Service Interface Implementation Example - CommentServiceImpl

● Mapper

In general, the package defining database operating is the persistence layer. In Mybatis, each table should have a XML file for describing its structure and defining interfaces of data accessing. These interfaces should be implemented by SQL and explicit java interface.

For example, the CommentMapper defines a interface to select top 10 records according to the news comment count.

```

<mapper namespace="com.hot.hotshow.mapper.CommentMapper">
  <resultMap id="BaseResultMap" type="com.hot.hotshow.domain.Comment">
    <result column="type" jdbcType="VARCHAR" property="type" />
    <result column="comment_sum" jdbcType="INTEGER" property="commentSum" />
  </resultMap>
  <select id="selectCommentList" resultMap="BaseResultMap">
    select type,comment_sum from t_comment ORDER BY comment_sum desc limit 10;
  </select>
</mapper>

```

Figure 45. Mapper XML File Configuration

```

public interface CommentMapper { List<Comment> selectCommentList();}

```

Figure 46. Mapper Interface

5.4.2 Front-end Implementation

The front end of the site has only one page: index.html. Its contents are four data graphs. And graph data are fetched by AJAX requests and passed to the Echarts framework to visualise the data.

● AJAX

By introducing the JQuery library, an AJAX request can be implemented concisely.

\$. ajax () defines the implementation details of an ajax request: the URL property is the URL of target API, and the success property defines a callback function which will automatically execute when the request is successfully returned.

The following AJAX request defines that when drawing the comment table, where the callback function defines the data initialization content before drawing.

```

$.ajax({
  type: "GET",
  url: "/source/list",
  data: '',
  contentType: 'application/json',
  success: function(res) {
    if(res.code == 0) {
      console.log(res.sources);
      var cms = res.sources;
      $.each(cms, function (itemIndex, item) {
        sourceArr.push(item['source']);
        countArr.push(item['sourceCnt'])
      });
      console.log(sourceArr);
      echart_4(sourceArr, countArr);
    }
  },
  error: function (res) {
    alert(res.msg);
  }
});

```

Figure 47. Example of AJAX Implementation By JQuery

● Echats Implementation

To using Echats.js, the first step need to import echats.js file in HTML.

```
<script src="echarts.min.js"></script>
```

Before drawing, it needs to prepare a DOM container with width and height for ECharts. In general, a <div> DOM will be used for the container.

```
<div id="main" style="width: 600px;height:400px;"></div>
```

Then a Javascript should be imported for Echarts initialisation, where Echarts instances initialised by echarts.init method and generates a simple chart using the setOption method, where the parameter 'option' is a JSON object defining attributes of the chart. Common attributes are listed as the following.

- color:** the color for block color and line color.
- grid:** specify the grid style.
- tooltip:** when the mouse triggers the chart, what event should response.
- xAxis:** the style of the x-axis.
- yAxis:** the style of the y-axis.
- series:** specify chart type and data source.

The following shows the complete code for implementing comments.

```

function echart_1(x,y) {
    var xAxisData= x;
    var serieData = y;
    var myChart = echarts.init(document.getElementById('chart_1'));
    var colors = ["#2EF7F3", "#FAD860"];
    var option = {
        color: colors,
        grid: { left: '1%', right: '5%', top: '10%', bottom: '6%', containLabel: true },
        tooltip: { trigger: 'axis', axisPointer: { type: 'shadow' } },
        xAxis: [{
            type: 'category',
            axisLine: { show: true, lineStyle: {color: '#6173A3'} },
            axisLabel: {interval: 0, rotate: 40, textStyle: {color: 'fff', fontSize: 12}},
            axisTick: { show: false },
            data: xAxisData,
        }, ],
        yAxis: [{
            axisTick: {show: false},
            splitLine: {show: false},
            axisLabel: { textStyle: {color: 'fff', fontSize: 12} },
            axisLine: {show: true, lineStyle: {color: '#6173A3'}},
        }, {
            type: 'value',
            name: '',
            axisLine: {show: true, lineStyle: {color: '#FAD860'}}
        }],
        series: [{
            type: 'bar',
            data: serieData
        }, {
            type: 'line',
            yAxisIndex: 1,
            data: serieData
        }]
    };
    myChart.setOption(option);
}

```

Figure 48. Example Implementation for A Chart

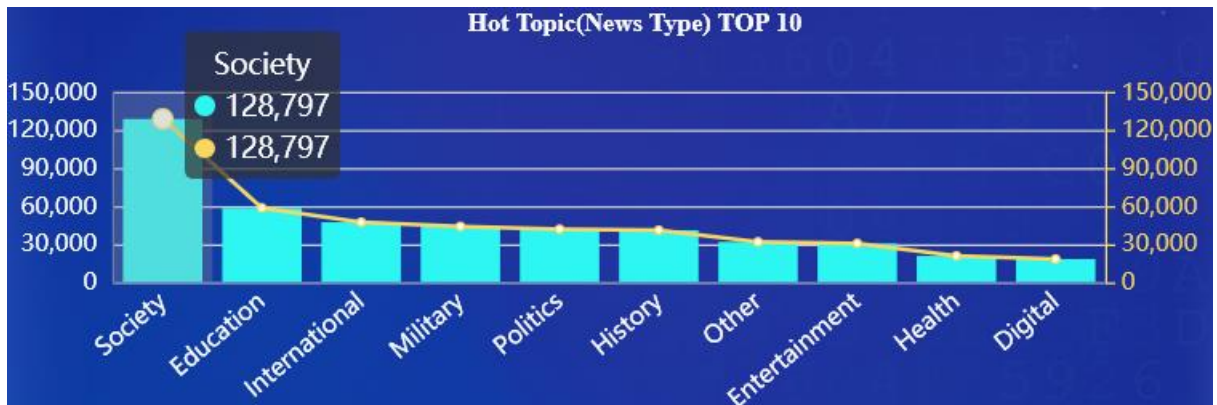


Figure 49. Hot Topic Data in Data Visualisation

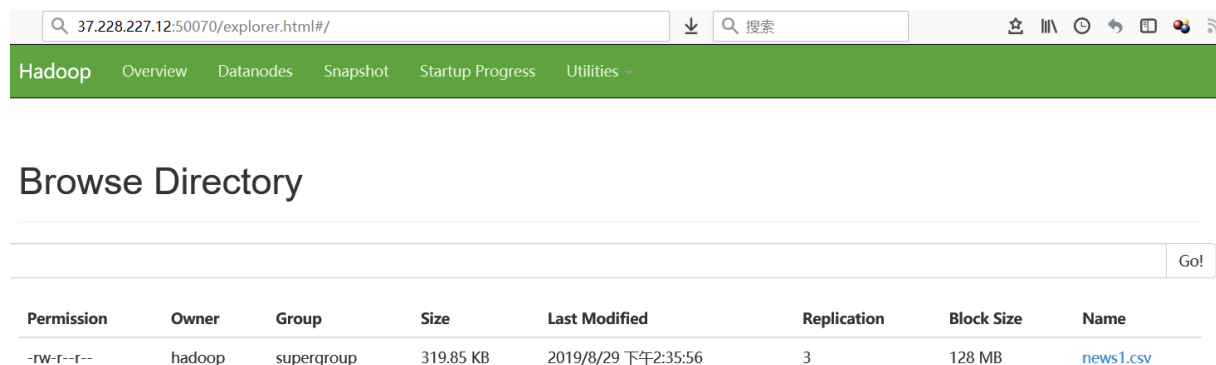
6 Test Evaluation and Deployment

This section describes the testing and deployment of the system. The test is divided into three test steps, each of which tests a module of the system. Finally, the system will be deployed to AWS to implement the designed functions.

● Modular Testing

The data collection module shall be tested firstly. This module cannot be tested using Junit because the test process is blocked when the crawler procedure starts, but we can check the files on HDFS to see if the module is correct.

Open HDFS browsing tool, there is a file named news1.csv in the root directory, the size is not zero, according to the code implementation, the module implementation is correct.



Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	hadoop	supergroup	319.85 KB	2019/8/29 下午2:35:56	3	128 MB	news1.csv

Figure 50. HDFS Root Directory

The second module to be tested is the data analysis module. The following is the test case written in Junit, which defines the MapReduce program tested according to the data analysis process flow. According to the Junit execution results on the left side of the window, all MapReduce program paths are covered and executed correctly. And the module passes its module test.

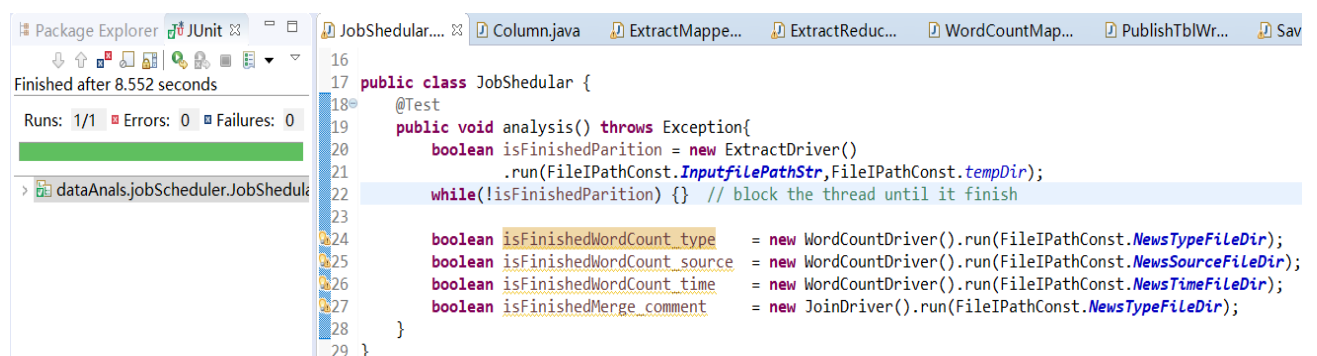


Figure 51. Test for Data Collection Module

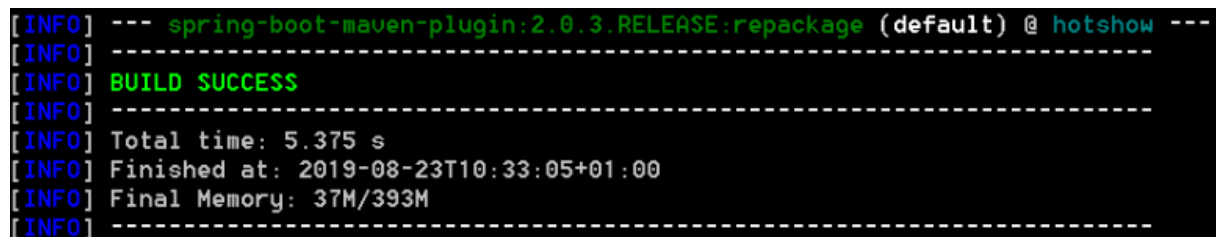
For verifying the correctness of the system, the use case defined in chapter 4 shall be verified, which also is the functional testing of the system. Before the testing, it should deploy the website on the cloud.

● Website Deployment

Firstly configuring project by POM.xml file and then going to project directory using the following command to and building project by Maven:

mvn clean package

If the SpringBoot web project will be package successfully, there is a ‘.jar’ file will be created under target directory, which is the web application.

A terminal window with a black background and green and white text. The text shows the output of a Maven command. It starts with '[INFO] --- spring-boot-maven-plugin:2.0.3.RELEASE:repackage (default) @ hotshow ---', followed by '[INFO] BUILD SUCCESS' in green. Below this, it shows '[INFO] Total time: 5.375 s', '[INFO] Finished at: 2019-08-23T10:33:05+01:00', and '[INFO] Final Memory: 37M/393M'. The terminal output is enclosed in dashed lines.

```
[INFO] --- spring-boot-maven-plugin:2.0.3.RELEASE:repackage (default) @ hotshow ---
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 5.375 s
[INFO] Finished at: 2019-08-23T10:33:05+01:00
[INFO] Final Memory: 37M/393M
[INFO]
```

Figure 52. Package Project by Maven

The next step is to copy the jar file to the Linux server on the cloud and execute the following command for running the website at background.

nohup java -jar sell.jar > /dev/null 2>&1 &

Finally, open a web browser at another machine to access the website, and the webpage will be shown as the next page.



Figure 53. System Result

The system now is deployed on cloud and all functions are successfully tested.

7 Conclusion

The system finally uses 2000 pieces of news data for analysis. According to the results in the webpage, the following four points can be summarised:

1. It is found that the most popular types of news are public-related aspects, followed by education. For the government, it can start with the most popular issues. Obviously, education is a very high-profile topic.
2. For the 163 news platform, most of the news types published were related to international, military and political, which accounted for more than half of the total news, which reflects the characteristics of the news platform.
3. In general, news is released during the day, and 10 am and 5 pm are the peak of the press release.
4. There are many sources of news that are unknown, so the authority of the data needs further confirmation. Therefore, the amount of data needs to be further increased and news records with unknown source should be filtered out by web spiders.

For the design and implementation of the system, the data collection module can be extended as a distributed system that can be deployed on the cluster, thereby to improve the efficiency of data collection. The data analysis module can integrate components later, simplifying the process of data analysis, such as using Hive-SQL, which can save a lot of MapReduce programs. Finally, according to the system design, the MVC architecture of the web module is too big for the system, but it provides extensibility for subsequent business development on it.

Appendix A: MapReduce Programs for Data Analysis

● WordCount MapReduce Program

```

public class WordCountMapper extends Mapper<LongWritable , Text , Text, IntWritable >{
    @Override
    protected void map(LongWritable key, Text value,Context context)
        throws IOException, InterruptedException {
        String line = value.toString();
        String[] words = line.split(" ");
        for(String word:words){
            context.write(new Text(word), new IntWritable(1));
        }
    }
}

public class WordCountReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
    @Override
    protected void reduce(Text key, Iterable<IntWritable> values, Context context)
        throws IOException, InterruptedException {

        int count = 0;
        for(IntWritable value:values){
            count +=value.get();
        }
        context.write(key, new IntWritable(count));
    }
}

public class WordCountDriver {
    public boolean run(String inputFilePath, String outputFilePath)
        throws IllegalArgumentException, IOException,
            ClassNotFoundException,
            InterruptedException {
        Configuration configuration = new Configuration();
        configuration.set("mapreduce.framework.name", "yarn");
        configuration.set("fs.defaultFS", "hdfs://" + Const.hadoop_c3_namenode);
        Job job = Job.getInstance(configuration, "Count");
        job.setJarByClass(dataAnals.wordCount.WordCountDriver.class);
        job.setMapperClass(dataAnals.wordCount.WordCountMapper.class);
        job.setReducerClass(dataAnals.wordCount.WordCountReducer.class);
        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(IntWritable.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        FileInputFormat.setInputPaths(job ,new Path(inputFilePath));
        FileOutputFormat.setOutputPath(job,new Path(outputFilePath));
        return job.waitForCompletion(true);
    }
}

```

● Partition MapReduce Program

```

public class ExtractMapper extends Mapper<LongWritable , Text , NullWritable, Column>{
    @Override
    protected void map(LongWritable key, Text value,Context context)
        throws IOException, InterruptedException {
        int id = 0;
        String line = value.toString();
        String[] fields = line.split(",");
        if(fields.length > 3) {
            Column type = new Column (1, fields[1]);
            Column comment = new Column (2, fields[1], fields[2]);
            String time_s = fields[3];
            int delimiter = time_s.indexOf(" ");
            int simecolom = time_s.lastIndexOf(":");
            time_s = time_s.isEmpty() ? time_s: time_s.substring(delimiter+1, simecolom);
            Column time = new Column(3, time_s);
            Column source = new Column (4, fields[5]);
            Column [] arr = {type,comment,time,source};
            for (Column col: arr) {
                context.write(NullWritable.get(), col);
            }
        }
    }
}

public class ExtractReducer extends Reducer<NullWritable, Column , Column , NullWritable> {
    @Override
    protected void reduce(NullWritable id_key, Iterable<Column > fields, Context context)
        throws IOException, InterruptedException {
        for(Column aColumn: fields) {
            context.write(aColumn,NullWritable.get());
        }
    }
}

public class ExtractDriver {
    public boolean run(String inputFilePath, String outputFilePath)
        throws IllegalArgumentException, IOException,
            ClassNotFoundException, InterruptedException {
        Configuration configuration = new Configuration();
        Job job = Job.getInstance(configuration, "Partition");
        configuration.set("mapreduce.framework.name", "yarn");
        configuration.set("fs.defaultFS", "hdfs://" + Const.hadoop_c3_namenode);
        job.setJarByClass(dataAnals.partition.ExtractDriver.class);
        job.setMapperClass(dataAnals.partition.ExtractMapper.class);
        job.setMapOutputKeyClass(NullWritable.class);
        job.setMapOutputValueClass(dataAnals.partition.Column.class);
        job.setOutputKeyClass(dataAnals.partition.Column.class);
        job.setOutputValueClass(NullWritable.class);
        FileInputFormat.setInputPaths(job ,new Path(inputFilePath));
        FileOutputFormat.setOutputPath(job,new Path(outputFilePath));
        job.setPartitionerClass(dataAnals.partition.ColumnsPartition.class);
        job.setNumReduceTasks(5);
        return job.waitForCompletion(true);
    }
}

```

● MergeCount MapReduce Program

```

public class JoinMapper extends Mapper<LongWritable, Text, Text, IntWritable>{
    private ArrayList<String> comment = new ArrayList<String>();
    @Override
    protected void setup(Context context) throws IOException, InterruptedException {
        // read all comments in the part-r-00002 file, and cache them in list.
        BufferedReader reader = new BufferedReader(
            new InputStreamReader(new FileInputStream(new File("part-r-00002"))));
        String line;
        while(StringUtils.isNotEmpty(line = reader.readLine())) {
            comment.add(line);
        }
        reader.close();
    }

    private Text type = new Text();
    private IntWritable aComment = new IntWritable();
    private int index_count = 0;
    @Override
    protected void map(LongWritable key, Text value, Context context)
        throws IOException, InterruptedException {
        type.set(value.toString());
        int comment_value = comment.get(index_count).isEmpty() ?
            0 : Integer.parseInt( comment.get(index_count));
        aComment.set(comment_value);
        index_count++;
        context.write(type, aComment);
    }
}

public class JoinReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
    @Override
    protected void reduce(Text aNewsType, Iterable<IntWritable> values, Context context)
        throws IOException, InterruptedException {
        int sumForANewsType = 0;
        for(IntWritable value:values){
            sumForANewsType += value.get();
        }
        context.write(aNewsType, new IntWritable(sumForANewsType));
    }
}

```

```
public class JoinDriver {
    public boolean run(String inputFilePath, String outputFilePath)
        throws IllegalArgumentException, IOException, ClassNotFoundException,
            InterruptedException, URISyntaxException {
        Configuration configuration = new Configuration();
        configuration.set("mapreduce.framework.name", "yarn");
        configuration.set("fs.defaultFS", "hdfs://" + Const.hadoop_c3_namenode);
        Job job = Job.getInstance(configuration, "MergeCount");
        job.setJarByClass(JoinDriver.class);
        job.setMapperClass(JoinMapper.class);
        job.setReducerClass(JoinReducer.class);
        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(IntWritable.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        FileInputFormat.setInputPaths(job, new Path(inputFilePath));
        FileOutputFormat.setOutputPath(job, new Path(outputFilePath));
        job.addCacheFile(new URI("file://" + FileIPathConst.OutputPartitionDirPathStr
            + FileIPathConst.NewsCommentFilename));
        return job.waitForCompletion(true);
    }
}
```

Appendix B: Details of Hadoop Cluster Configuration

In this appendix, it shows the steps for making a full distributed Hadoop cluster in details.

● Installing JRE

To install the Java Runtime Environment, another easy way is using yum to execute the following command:

yum install java

```
[hadoop@hadoop-c3 ~]$ yum install java
Loaded plugins: fastestmirror, refresh-packagekit, security
You need to be root to perform this command.
[hadoop@hadoop-c3 ~]$ su
Password:
[root@hadoop-c3 hadoop]# yum install java
Loaded plugins: fastestmirror, refresh-packagekit, security
Setting up Install Process
Determining fastest mirrors
 * base: moztart.ee.ic.ac.uk
 * extras: moztart.ee.ic.ac.uk
 * updates: moztart.ee.ic.ac.uk
base                                                    | 3.7 kB      00:00
extras                                                  | 3.4 kB      00:00
updates                                                 | 3.4 kB      00:00
updates/primary_db                                     | 5.7 MB      00:02
Resolving Dependencies
--> Running transaction check
--> Package java-1.8.0-openjdk.x86_64 1:1.8.0.222.b10-0.el6_10 will be installed
--> Processing Dependency: java-1.8.0-openjdk-headless = 1:1.8.0.222.b10-0.el6_10 for package: 1:java-1.8.0-openjdk-1.8.0.222.b10-0.el6_10.x86_64
```

... ..

```
java-1.8.0-openjdk.x86_64 1:1.8.0.222.b10-0.el6_10
Dependency Installed:
java-1.8.0-openjdk-headless.x86_64 1:1.8.0.222.b10-0.el6_10      tzdata-java.noarch 0:2019b-2.el6
Complete!
```

Finally, the console window will print complete information as the above. And testing by the command

java -version

```
[root@hadoop-c3 hadoop]# java -version
openjdk version "1.8.0_222"
OpenJDK Runtime Environment (build 1.8.0_222-b10)
OpenJDK 64-Bit Server VM (build 25.222-b10, mixed mode)
```

If the above information printed out, the JRE is successfully installed.

● Installing Hadoop

To install Hadoop, we can find the Hadoop package at the apache official website



CELEBRATING 20 YEARS OF COMMUNITY-LED DEVELOPMENT
"THE APACHE WAY"


[Projects](#)
[People](#)
[Community](#)
[License](#)
[Sponsors](#)

We suggest the following mirror site for your download:

<http://mirrors.whoishostingthis.com/apache/hadoop/common/hadoop-2.7.7/hadoop-2.7.7.tar.gz>

Other mirror sites are suggested below.

It is essential that you [verify the integrity](#) of the downloaded file using the PGP signature ([.asc](#) file) or a hash ([.md5](#) or [.sha*](#) file).

Please only use the backup mirrors to download KEYS, PGP signatures and hashes (SHA* etc) -- or if no other mirrors are working.

Using 'wget' command to download Hadoop, as shown the following:

```
[hadoop@hadoop-c3 ~]$ wget http://mirrors.whoishostingthis.com/apache/hadoop/common/hadoop-2.7.7/hadoop-2.7.7.tar.gz
--2019-08-29 14:56:08-- http://mirrors.whoishostingthis.com/apache/hadoop/common/hadoop-2.7.7/hadoop-2.7.7.tar.gz
Resolving mirrors.whoishostingthis.com... 54.246.233.92
Connecting to mirrors.whoishostingthis.com|54.246.233.92|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 218720521 (209M) [application/x-gzip]
Saving to: "hadoop-2.7.7.tar.gz"

15% [=====>] 8,925,786 2.32M/s eta 87s
```

After the .tar.gz file has been downloaded, using the command

```
[hadoop@hadoop-c3 ~]$ tar -zxvf hadoop-2.7.7.tar.gz -C /opt/module/
```

Then going to directory '/opt/module' executing 'll' command to check files in the directory.

```
[hadoop@hadoop-c3 ~]$ cd /opt/module/
[hadoop@hadoop-c3 module]$ ll
total 4
drwxr-xr-x. 9 hadoop hadoop 4096 Jul 18 2018 hadoop-2.7.7
[hadoop@hadoop-c3 module]$
```

Next step is to modify /etc/profile file and add \$HADOOP_HOME environment variable

```
#JAVA_HOME
export JAVA_HOME=/usr/bin/java
export PATH=$PATH:$JAVA_HOME/bin

#HADOOP_HOME
export HADOOP_HOME=/opt/module/hadoop-2.7.7
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
```

Then save it and source it. Echo the \$HADOOP_HOME if its directory are outputted, then the Hadoop successfully installed.

```
[hadoop@hadoop-c3 hadoop-2.7.7]$ source /etc/profile
[hadoop@hadoop-c3 hadoop-2.7.7]$ echo $HADOOP_HOME
/opt/module/hadoop-2.7.7
```

● Configure Host File

In the virtual machine environment, nodes should be configured in a LAN as it's best to make IP address are continuous series as the following:

```
192.168.91.113 hadoop-c3
192.168.91.111 hadoop-c1
192.168.91.112 hadoop-c2
192.168.91.110 hadoop-s0
```

If the hostname is not been configured, modify the hostname in file '/etc/sysconfig/network'

```
[hadoop@hadoop-c3 hadoop-2.7.7]$ sudo vi /etc/sysconfig/network
[sudo] password for hadoop:
NETWORKING=yes
NETWORKING_IPV6=no
HOSTNAME=hadoop-c3
~
```

The name of each node should be end of a series of continuous number, which is valuable as the following shell can be used to manage the cluster. xcall is used to execute a command at each node in the cluster, and xsync is used to deliver file to each node in the cluster:

■ Shell: xcall

```
#!/bin/bash
pcount=$#
if(($pcount==0)); then
echo no args; please input command for hosts executing: hadoopc-1, 2,3, hadoop-s0
exit;
fi

echo

for((i=1;i<=3;i++)); do
    echo -----hadoop-c$i-----
    ssh hadoop-c$i $@
    echo
done

for((i=0;i<1;i++)); do
    echo -----hadoop-s$i-----
    ssh hadoop-s$i $@
    echo
done
```

■ Shell: xsync

```
#!/bin/bash
#1 get parameter, if no input , exit
pcount=$#
if(($pcount==0)); then
echo no args input, please input the filename or dirctory name for synchronize to host:
hadoopc-1,2,3, hadoop-s0
exit;
fi

#2 get filename
p1=$1
fname=`basename $p1`
#echo File/Folder:$p1 Inputed

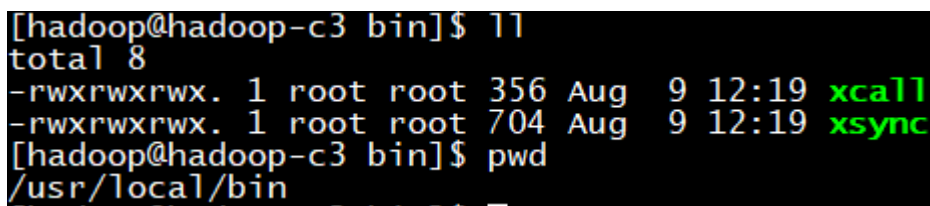
#3 get parents dir
pdir=`cd -P $(dirname $p1); pwd`

#4 get username
user=`whoami`

#5 make synchronization for slave-clusters
for((i=1;i<=3;i++)); do
    echo -----hadoop-c$i is synchronizing...-----
    rsync -rvl $pdir/$fname $user@hadoop-c$i:$pdir
done

for((i=0;i<1;i++)); do
    echo -----hadoop-s$i is synchronizing...-----
    rsync -rvl $pdir/$fname $user@hadoop-s$i:$pdir
```

Those shell should be under the directory '/usr/local/bin'



```
[hadoop@hadoop-c3 bin]$ ll
total 8
-rwxrwxrwx. 1 root root 356 Aug  9 12:19 xcall
-rwxrwxrwx. 1 root root 704 Aug  9 12:19 xsync
[hadoop@hadoop-c3 bin]$ pwd
/usr/local/bin
```

So far for a node, all basic configurations for the node are finished.

● Configuring IP Address For Each Node

For implementing a cluster, the fast way to do that is clone the node has done for three copies.

After that, the MAC address and IP address should be modified in case of collision.

```
[root@hadoop-c3 module]# vi /etc/udev/rules.d/70-persistent-net.rules

# This file was automatically generated by the /lib/udev/write_net_rules
# program, run by the persistent-net-generator.rules rules file.
#
# You can modify it, as long as you keep each rule on a single
# line, and change only the value of the NAME= key.

# PCI device 0x8086:0x100f (e1000)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:0c:29:1c:22:3e", ATTR{type}=="1",
KERNEL=="eth*", NAME="eth0"

# PCI device 0x8086:0x100f (e1000)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:0c:29:7c:c7:5e", ATTR{type}=="1",
KERNEL=="eth*", NAME="eth1"
```

They are two MAC address here, the first one should be deleted and the name of second one should be renamed. Then change the file content as the following:

```
#_This file was automatically generated by the /lib/udev/write_net_rules
# program, run by the persistent-net-generator.rules rules file.
#
# You can modify it, as long as you keep each rule on a single
# line, and change only the value of the NAME= key.

# PCI device 0x8086:0x100f (e1000)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:0c:29:7c:c7:5e", ATTR{type}=="1",
KERNEL=="eth*", NAME="eth0"
```

Opening the network card configuration file, changing the IP address as the designed.

```
[root@hadoop-c3 module]# vi /etc/sysconfig/network-scripts/ifcfg-eth0

CE="eth0"
BOOTPROTO="static"
HWADDR="00:0C:29:7C:C7:5E"
IPV6INIT="no"
NM_CONTROLLED="yes"
ONBOOT="yes"
TYPE="Ethernet"
UUID="dd69c4bb-971d-4230-8499-244f19faa468"
IPADDR="192.168.91.113"
NETMASK="255.255.255.0"
GATEWAY="192.168.91.2"
BROWSER_ONLY="no"
DNS1="8.8.8.8"
```

And finally, reboot the system to reload network.

● Configuring SSH

As data transferring among nodes in Hadoop should be secured and reliable, the Hadoop is based on SSH. Thus, each node in cluster should configure SSH. Here an example showing the details to configure SSH on node 'hadoop-s0'. Firstly generating an SSH key by RSA key, and then copy the public key to others node to achieve the password-less ssh login.

```
[hadoop@hadoop-s0 .ssh]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa.
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub.
The key fingerprint is:
a5:63:27:66:ac:b6:78:1d:ef:ea:4a:14:ee:67:81:a9 hadoop@hadoop-s0
The key's randomart image is:
+--[ RSA 2048 ]-----+
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
|
+-----+
[hadoop@hadoop-s0 .ssh]$
```

```
[hadoop@hadoop-s0 .ssh]$ ssh-copy-id hadoop-c1
hadoop@hadoop-c1's password:
Now try logging into the machine, with "ssh 'hadoop-c1'", and check in:

    .ssh/authorized_keys

to make sure we haven't added extra keys that you weren't expecting.

[hadoop@hadoop-s0 .ssh]$ ssh-copy-id hadoop-c2
hadoop@hadoop-c2's password:
Now try logging into the machine, with "ssh 'hadoop-c2'", and check in:

    .ssh/authorized_keys

to make sure we haven't added extra keys that you weren't expecting.

[hadoop@hadoop-s0 .ssh]$ ssh-copy-id hadoop-s0
hadoop@hadoop-s0's password:
Now try logging into the machine, with "ssh 'hadoop-s0'", and check in:

    .ssh/authorized_keys

to make sure we haven't added extra keys that you weren't expecting.

[hadoop@hadoop-s0 .ssh]$ _
```

Then the hadoop-s0 node is done. For the other node should repeat the above step.

● Modify Hadoop Configuration

Change the following configuration files on one node.

1. core-site.xml

```

<configuration>

<!-- The path of name node for HDFS -->
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://hadoop-s0:8020</value>
</property>

<!-- The path of temp running file while hadoop running -->
<property>
  <name>hadoop.tmp.dir</name>
  <value>/opt/module/hadoop-2.7.2/data/tmp</value>
</property>

```

2. hdfs-site.xml

```

<configuration>
  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>
  <property>
    <name>dfs.http.address</name>
    <value>hadoop-s0:50070</value>
  </property>
  <property>
    <name>dfs.namenode.secondary.http-address</name>
    <value>hadoop-c2:50090</value>
  </property>
</configuration>

```

3. slaves

```

hadoop-c1
hadoop-c2
hadoop-s0

```

4. yarn-site.xml

```

<configuration>
<!-- Site specific YARN configuration properties -->
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>hadoop-c1</value>
  </property>
</configuration>

```

5. mapred-site.xml

```

<configuration>

  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>

</configuration>

```

Executing the xsync shell for delivering these configuration files to others node.

```
cd /opt/module/hadoop-2.7.7/etc/hadoop
xsync /opt/module/hadoop-2.7.7/etc/hadoop/core-site.xml
xsync /opt/module/hadoop-2.7.7/etc/hadoop/yarn-site.xml
xsync /opt/module/hadoop-2.7.7/etc/hadoop/slaves
```

Up to now, the full distributed Hadoop cluster is configured.

To run the cluster just needs to execute the following command.

● Run Hadoop Cluster

1. Format namenode

```
[hadoop@hadoop-s0 ~]$ xcall rm -rf $HADOOP_HOME/data $HADOOP_HOME/logs
-----hadoop-s0-----
-----hadoop-c1-----
-----hadoop-c2-----
-----hadoop-c3-----
```

```
[hadoop@hadoop-s0 ~]$ $HADOOP_HOME/bin/hdfs namenode -format
```

2. Run HDFS

```
[hadoop@hadoop-s0 ~]$ $HADOOP_HOME/sbin/start-dfs.sh
Starting namenodes on [hadoop-s0]
hadoop-s0: starting namenode, logging to /opt/module/hadoop-2.7.7/logs/hadoop-hadoop-namenode-hadoop-s0.out
hadoop-c1: starting datanode, logging to /opt/module/hadoop-2.7.7/logs/hadoop-hadoop-datanode-hadoop-c1.out
hadoop-c2: starting datanode, logging to /opt/module/hadoop-2.7.7/logs/hadoop-hadoop-datanode-hadoop-c2.out
hadoop-s0: starting datanode, logging to /opt/module/hadoop-2.7.7/logs/hadoop-hadoop-datanode-hadoop-s0.out
Starting secondary namenodes [hadoop-c2]
hadoop-c2: starting secondarynamenode, logging to /opt/module/hadoop-2.7.7/logs/hadoop-hadoop-secondarynamenode-hadoop-c2.out
```

3. Run YARN

```
[hadoop@hadoop-c1 ~]$ $HADOOP_HOME/sbin/start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /opt/module/hadoop-2.7.7/logs/yarn-hadoop-resourcemanager-hadoop-c1.out
hadoop-s0: starting nodemanager, logging to /opt/module/hadoop-2.7.7/logs/yarn-hadoop-nodemanager-hadoop-s0.out
hadoop-c1: starting nodemanager, logging to /opt/module/hadoop-2.7.7/logs/yarn-hadoop-nodemanager-hadoop-c1.out
hadoop-c2: starting nodemanager, logging to /opt/module/hadoop-2.7.7/logs/yarn-hadoop-nodemanager-hadoop-c2.out
```

References

- [1] Armbrust, M., Fox, A., Grith, R., et al. 2009, "Above the clouds: A Berkeley View of Cloud Computing" [R]. UCB/EECS-2009-28. Berkeley, USA: Electrical Engineering and Computer Sciences, University of California at Berkeley.
- [2] Yuntao, L., 2006, "Web-based Hadoop Massive Data Analysis System", Harbin University of Technology Computer and Technology School Report, pp44-pp52.
- [3] Yuan L., 2013, "Design and Implementation of Massive Data Analysis System Based on Hadoop", Dalian University of Technology Academic Periodical.
- [4] Jeffrey D. and Sanjay G., 2008, "MapReduce: Simplified Data Processing on Large Clusters", Communications of The ACM Januar, vol.51(1), [online] available: <https://www.cs.amherst.edu/~ccmcgeoch/cs34/papers/p107-dean.pdf> [accessed at: 16-Nov-2018]
- [5] Peter Mell and Tim Grance, "The NIST Definition of Cloud Computing", National Institute of Standards and Technology, vol.15, available: <http://www.nist.gov/itl/cloud/upload/cloud-def-v15.pdf>
- [6] Apache, 2018, [online] available: <http://hadoop.apache.org/docs> [accessed at: 16-Nov-2018]
- [7] Ai S., Liu J. and Liu X. 2018, Design of Network Hot News Collection and Analysis System for Radio Stations, New Medium, Vol(331), pp.71-74, Available: <http://oa.shxyj.org/UploadFile/20130926008/2015-08-24/Issue/mnn1q1cc.pdf>
- [8] Choudhary S, Dincturk ME, Mirtaheri SM, Moosavi A, vonBochmann G, Jourdan G-V, Onut I-V (2012) Crawling rich internet applications: the state of the art. In: CASCON. pp 146–160
- [9] Deng Qian Ni, Chen whole. Cloud computing and its key technology. "Development and application of high performance computing," the first part of 2009
- [10] sanjay Ghemawat; Howard Gobioff et.al The Google file system. Proceedings of the nineteenth ACM symposium on Operating systems principles. Oct. 2003
- [11] Jumgo C. and Hector G., 2000, "The evolution of the Web and implications for an incremental crawler", Proc. Of VLDB Conf..
- [12] Mini,S.A., Jatinder S.B. and Varnica, 2014, "Web Crawler: Extracting the Web Data", International Journal of Computer Trends and Technology (IJCTT), volume 13 (3), pp 132-137
- [13] Glez-Peña, D., Lourenço, A., López F. H., Reboiro J.M. and Fdez R.F. (2014) 'Web Scraping Technologies in an API World'. Briefings in Bioinformatics, 15 (5): 788- 797, available: <http://doi.org/10.1093/bib/bbt026>
- [14] Dhiraj K., Satish K., 2012, IJCSMS International Journal of Computer Science & Management Studies, Vol. 12, Issue 01, ISSN (Online): 2231 –5268. Available: www.ijcsms.com
- [15] Spring Boot Tutorials Point, pp3-4. Available: https://www.tutorialspoint.com/spring_boot/spring_boot_tutorial.pdf
- [16] Deinum M. and Serneels K., 2012, Pro spring MVC: With web flow, vol. 9781430241560.

- [17] Wang L., 2012, "A Review of Parallel Computing Techniques", Information Technology, Vol (10), pp112 - 115.
- [18] Liu X., He T., Long H. and Chen L., 2008, Design of Web Event Detection System, Journal of Chinese Information Processing, Vol(22), No.6, pp81. Available: <https://www.ixueshu.com/document/b00777111d9684a2d0.html>
- [19] Yu G., 2009, Characteristics and Reference of Government Intervention in Public Opinion Hotspots, News and Writing, Vol (6), pp.92, Available: <http://www.cqvip.com/qk/81506x/200906/30587039.html>
- [20] Yan Y., 2004, Construction of Pre-warning Management System of Social Stability, Sociology Research, Vol(3), pp.2, Available: <http://oa.shxyj.org/UploadFile/20130926008/2015-08-24/Issue/1gn2b3bc.pdf>
- [21] HuSpot & Visage, Data Visualization 101:How To Design Charts and Graphs, available: https://cdn2.hubspot.net/hub/53/file-863940581-pdf/Data_Visualization_101_How_to_Design_Charts_and_Graphs.pdf
- [22] Hui Li, GuiJun Xu , Mingji Zhou, Lingling Si, "Aspect-oriented Programming for MVC Framework" , IEEE Paper Dated 2010.
- [23] Chen X., Qi X. and Wang Y., 2016, Big Data System and Analysis Technology Overview, Software Journal, Vol.25(9): pp1889 -1908.
- [24] Huang S. and Ge M., 2013, The Application of Hadoop Platform in Big Data Processing, Modern Computer Science, vol.10(12)
- [25] Vaquero, L., Rodero-Marino, L., Caceres, J., et al., 2009, "Break in the Clouds: Towards a Cloud Definition" [J]. SIGCOMM Computer Communication Review, 39(1): 50-5
- [26] Ankur B., Prashant S., Vinayak K. and Meshram B., 2012, Integration of Struts, Spring and Hibernate for an University Management System, International Journal of Emerging Technology and Advanced Engineering, vol. 2(6), available: <https://pdfs.semanticscholar.org/4097/1737c87bf248abe35e2ea41c9b8836eb7872.pdf>
- [27] JSoup Organization, Jsoup: Java HTML Parser, available: <https://jsoup.org/>
- [28] Jakarta Commons, HTTP Client Introduction, available: <https://hc.apache.org/httpclient-3.x/>
- [29] AWS, AWS General Reference,Reference guide version 1.0, available: <https://docs.aws.amazon.com/general/latest/gr/aws-general.pdf>
- [30] Baidu, Echarts Tutorials, available: <https://www.echartsjs.com/en/>
- [31] Tutorialspoint, Mybatis persistence framework, Tutorialspoint (I) Pvt. Ltd., available: https://www.tutorialspoint.com/mybatis/mybatis_tutorial.pdf
- [32] Apache Maven Project, Maven Tutorial, available: <https://maven.apache.org/>
- [33] Jesse J., 2005, Ajax: A New Approach to Web Applications, pp.2-4, available : <https://pdfs.semanticscholar.org/c440/ae765ff19ddd3deda24a92ac39cef9570f1e.pdf>