IP Network Traffic Analysis Based on Big Data

Hanqi Yin[1(✉)], Jianguo Sun[1], Yiqi Shi[2], and Liu Sun[1]

[1] Department of Computer Science and Technology, Harbin Engineering University, Harbin, China
{yinhanqi,sunjianguo}@hrbeu.edu.cn, sunliuhrbeu@163.com
[2] Harbin University of Commerce, Harbin 150028, China
740165656@qq.com

Abstract. Big data is a hot topic in the current academia and industry circles, which is influencing people's daily lifestyles, work habits and ways of thinking. Due to the complexity of data itself and the huge amount of data, big data faces many problems in the process of collection, storage and use. It requires a new processing model to have greater decision making, insight and process optimization capabilities to accommodate massive, high growth rates and diverse information. The strategic significance of big data is not to master huge data information, but to conduct specialized analysis and processing of these meaningful data. This paper focuses on the analysis of IP network traffic under big data, and studies the sources of existing network traffic, the purpose of traffic analysis, and the common analysis methods for big data traffic. The structure and usability of Hadoop-based traffic analysis framework are mainly studied, and a new prospect is proposed for the future development direction.

Keywords: Big data · Traffic analysis · Hadoop

1 Introduction

With the development of Internet application technology and the expansion of the Internet scale, a large number of mobile terminals access the network for resource sharing and information communication. The operating mechanism and complex behavioral characteristics of the Internet make it difficult to control the data or behavior in the Internet. Since network traffic can reflect the complex dynamic characteristics of the Internet, people can improve the performance of the Internet by controlling, scheduling, shaping the network traffic, helping people to understand the behavior of the network more deeply.

From the perspective of network architecture, network traffic is the foundation of all studies, so all research on the characteristics of network applications

This work is supported by the Fundamental Research Funds for the Central Universities (HEUCFG201827, HEUCFP201839).

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2019
Published by Springer Nature Switzerland AG 2019. All Rights Reserved
S. Liu and G. Yang (Eds.): ADHIP 2018, LNICST 279, pp. 382–391, 2019.
https://doi.org/10.1007/978-3-030-19086-6_43

and the behavior of the network itself can be obtained through research on network traffic. By analyzing the traffic characteristics carried on the network, it is possible to find an effective way to explore the internal operating mechanism of the network. Network traffic can directly reflect the performance of the network. In the network, if the traffic received by the network exceeds its actual carrying capacity, it will cause network performance degradation. Throughput is an important indicator of network performance. An ideal network should accept all traffic until its maximum throughput limit. However, in an actual network, if network traffic is poorly controlled or network congestion occurs, throughput will decrease and network performance will decrease. The relationship between network traffic and throughput is shown in Fig. 1. It can be seen from Fig. 1 that in order to further improve the network performance, it is necessary to study the network traffic, extract the parameters that can characterize the network traffic, and find the controllable performance parameters by modeling, simulating and analyzing the network traffic, to achieve effective control of traffic, improve and optimize network performance.

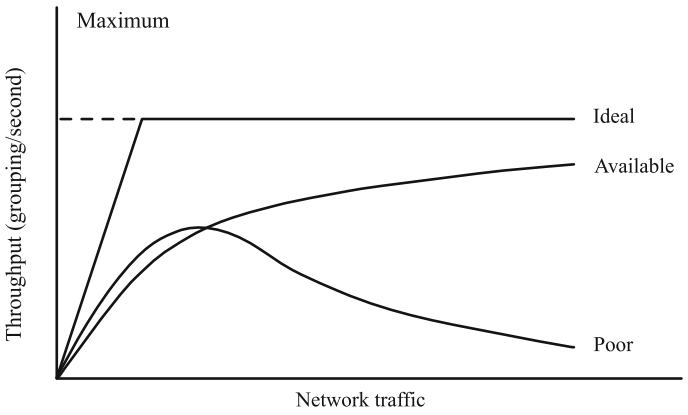

Fig. 1. The relationship between network traffic and throughput.

2 Big Data Analysis in the Network

2.1 The Goal of Big Data Analysis

At present, big data analysis is applied in various fields of science, medicine and commerce. Although its use is very wide, its main objectives are reflected in the following parts:

(1) *Acquire knowledge and predict trends*

People have been doing data analysis for a long time, and the primary and most important purpose is to acquire and utilize knowledge. Because big data contains a lot of original and real information, big data analysis can effectively abandon individual differences and help people grasp the law behind things more accurately. Based on the extracted knowledge, it is possible to predict natural or social phenomena more accurately.

(2) *Analyze personalized features*

Individual activities meet the characteristics of some groups, but also have distinct personality characteristics. Like the slender tail in "The Long Tail" theory, these features can vary wildly.

(3) *Identify the truth through analysis*

Error message is worse than no message. Because the dissemination of information in the network is very convenient, the harm caused by false information on the network is also greater. Due to the wide range of big data sources and their diversity, it can help to achieve the de-authentication of information to a certain extent. At present, people are trying to use big data to identify false information.

2.2 Big Data Analysis

As the data in the network increases, the amount of data increases from terabytes to petabytes, and the analysis needs shift from regular analysis to deep analytics [4]. It can be seen in Fig. 2 that people are not satisfied with the analysis and detection of existing data, but more expect to have more analysis and prediction of future trends. These analysis operations include complex statistical analysis such as moving average analysis, data association analysis, regression analysis, and market blue analysis, which we call deep analytics. At the same time, due to the increase in data volume, the database cost has risen, and the hardware platform has also shifted from a high-end server to a large-scale cluster platform composed of low-end and medium-end hardware.

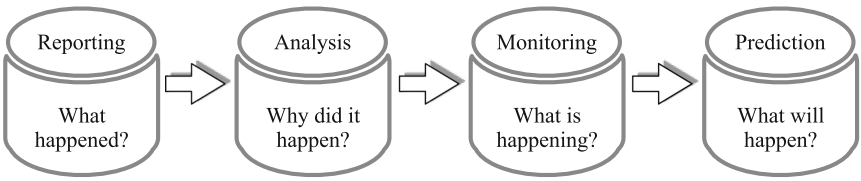

Fig. 2. Trands in data analysis.

Data analysis is the core process of big data applications. According to different levels, it can be roughly divided into three categories: computing architecture, query and index, and data analysis and processing.

In terms of computing architecture, MapReduce is currently a widely used big data set computing model and framework. In order to meet some analysis requirements of high task completion time, its performance was optimized in literature [12]. Literature [3] proposed a data flow analysis solution based on MapReduce architecture, MARISSA, which enables it to support real-time analysis tasks. Literature [2] also proposes a TiMR framework based on MapReduce for real-time stream processing for applications with high real-time requirements such as advertising push.

In terms of query and index, because big data contains a large amount of unstructured or semi-structured data, the query and indexing technology of traditional relational databases is limited, and NoSQL class database technology gets more attention.

In terms of data analysis and processing, the main technologies involved include semantic analysis and data mining. Due to the diversified nature of data in a big data environment, it is difficult to unify terms and then mine information when performing semantic analysis on data. Literature [7] studies the heterogeneity of semantic ontology in semantic analysis. Traditional data mining technology is mainly aimed at structured data, so it is urgent to study unstructured or semi-structured data mining technology. Literature [6] proposed a mining technology for image files, while literature [5] proposed a retrieval and mining technology for large-scale text files.

3 Traffic Analysis Based on Hadoop

3.1 Hadoop Architecture

Hadoop is a software framework based on JAVA for distributed intensive data processing and data analysis. It is largely based on MapReduce technology, but at the same time it is not just a distributed file system for storage, but a framework for executing distributed applications on large clusters of general purpose computing devices.

In Fig. 3 depicts the various layers of the ecosystem, in addition to the core Hadoop distributed file system (HDFS) and MapReduce programming framework, it also includes closely linked HBase database clusters and ZooKeeper clusters [1]. HDFS is a master/slave architecture that performs CRUD (Create, Read, Updata, and Delete) operations on files through directory paths, providing high-reliability underlying storage support for the entire ecosystem. MapReduce adopts the idea of "divide and conquer" to distribute the operation of large-scale data sets to the sub-nodes under the management of a master node, and then the final results are obtained by integrating the intermediate nodes of each subnode, thus providing better computing power for the system. HBase is located in the structured storage layer, and the Zookeeper cluster provides stable services and failover mechanism for HBase.

Hadoop was originally used to deal with single applications such as search, but with the advent of the big data era, Hadoop should be able to adapt to more applications. For different types of applications, the MapReduce parallel

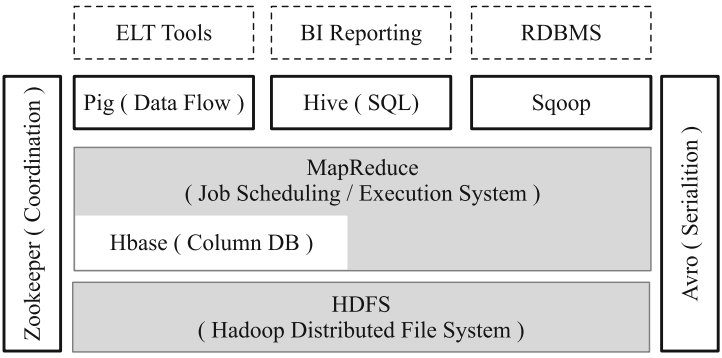

Fig. 3. Hadoop framework.

computing framework needs to optimize its deficiencies. For example, in order for MapReduce to process more tasks at the same time, the job scheduling algorithm needs to be optimized. HBase is an open source database based on Hadoop. It has problems such as slow response speed and single point of failure. It needs to optimize performance in order to provide high performance, high reliability and real-time literacy. As the bottom layer of Hadoop storage, HDFS needs to quickly access files of different sizes and enhance its security performance. Balancing performance, efficiency, and usability from a holistic perspective can further enhance Hadoop's capabilities.

3.2 Hadoop Analyzes the Feasibility of Mobile Traffic Data

With the development of technology, we can get more and more data, and all kinds of companies that can get data want to get more data, which brings data storage problems. There is no doubt that there is a lot of information in the data, and its research value and commercial value are endless. Therefore, how to analyze massive data efficiently and obtain the required information from the data becomes another problem. The collected data has various formats. With the growth of data volume, the traditional database storage method can no longer meet the storage requirements of data. Users gradually feel that the database is bulky, unable to analyze data flexibly, or even cannot store the data volume equal to the data collection volume achieved.

Hadoop can deal with these problems well. It consists of the underlying distributed file system and the upper layer MapReduce distributed computing framework. The distributed file system is highly scalable, can store a variety of data, and uses data redundancy to ensure data reliability and improve computational efficiency. In addition, it supports multiple computing frameworks and provides interfaces that allow users to flexibly extend their applications. And there are hundreds of configuration parameters to provide users with a variety of resource allocation options, all types of users can configure and manage Hadoop according to their actual needs.

Hadoop was originally designed to run on a cluster of inexpensive PCs, with hardware errors and machine failures as normal, so Hadoop doesn't need very expensive hardware devices to support it. Hadoop itself is an open source framework that can be freely modified, and an active public community brings together users and enthusiasts around the world to discover problems in the Hadoop and provide solutions. The existing versions of Hadoop are enough to give users a good experience, and it is very convenient to customize according to their own needs. It does not require a high level of developer skills, and has a lot of learning materials and low learning costs. In addition, although maintenance and management are very big problems for large clusters, because Hadoop is an open source system, maintenance costs are relatively low.

3.3 Hadoop Application in Traffic Analysis

In the field of traffic monitoring and analysis, literature [11] proposed a flow analysis algorithm based on MapReduce, which can analyze the target port of flow records statistically. In this paper, the computational efficiency of the algorithm is compared with "flow-tools", a mainstream flow data processing tool. The results show that the mapreduce-based flow analysis algorithm can save 72% of the computation time. It is also verified that Hadoop can cope well with single point of failure. On this basis, the Yeonhee Lee in order to solve the DDoS attack detection technology can tolerate response time facing the challenge of mass data processing flow, based on graphs is a HTTP GET flood attack detection algorithm, put forward a real-time framework, the use of distributed cluster power to detect DDoS attacks [9]. And puts forward an extensible framework based on Hadoop and transport processing, and designed a new binary input frame, set as high speed calculation and message solution, efficient storage framework includes a variety of graphs algorithm analysis of message, can analyze mass message [8]. In addition to this, and further puts forward a scalable traffic monitoring system based on Hadoop, can from IP, TCP, HTTP and NetFlow perspective TB level of Internet traffic, for different network layer, the system can use graphs of the distributed algorithm to efficiently deal with and analyze network traffic, experiments show that as the growth of the number of cluster nodes, the analysis of the system than CoralReef and TIPE's Pcap more efficient [10]. The main components of the whole system includes a can receive real-time trace binary format, Netflow, IP, TCP and HTTP analysis algorithm of graphs, and a simple query system based on the Hive. At the analysis and presentation layer of the system, the author integrated the previous studies and provided TCP re-traditional count, five-tuple flow statistics and DDoS analysis.

3.4 Inadequacies of the Hadoop Platform

Hadoop has some inherent flaws and problems with unsuitable environments due to its design goals, architecture, and distributed features:

(1) *Small document problem*

HDFS was originally developed for streaming large files. To ensure data redundancy and flexibility, when HDFS stores data, the file is divided into files by 64M per block by default. If you need to store a large number of files smaller than 64M, there will be many problems. Even if the file is smaller than 64M, HDFS will still treat it as a 64M block for storage, which will waste a lot of storage space. The NameNode will record the location of each file. A large number of small files will occupy a large amount of storage space of the NameNode, which is easy to cause a single point bottleneck. Processing many of these small files can also take more time and resources than processing large files.

(2) *Real-time processing access*

Hadoop is not suitable for online data processing that requires low latency data. For example, the update of the stock system requires everyone to know in time that there is no delay. Similarly, implementing a real-time/flow-based processing model is not a strength of Hadoop because it does not support data that continues to arrive at operations that require immediate processing.

(3) *Unable to modify the file at will*

Hadoop supports storage for multiple writes per write, and is very inefficient if you add or modify files.

(4) *Configuration and optimization issues*

Hadoop has more than 190 configuration parameters. How to configure it reasonably in different production environments, and make full use of resources to make the system run efficiently is a very important issue. Some mechanisms of Hadoop itself can not be applied to all environments, and some unreasonable mechanisms are improved, which is another optimization direction.

4 Hadoop Performance Optimization

4.1 Hadoop Job Scheduling Algorithm

In Fig. 4, the TaskScheduler is a component of the JobTracker, and the relationship between the function and the call is between them. The interaction between Client, JobTracker and TaskTracker is through network RPC. Then we will analyze the general principle of the scheduler:

① Client submits a job to JobTracker via submitJob() function.
② TJobTracker notifies TaskScheduler to call its internal function initJob() to initialize the job and create some internal data structures.
③ The TaskTracker reports its resources to the JobTracker via a heartbeat, such as how many free map slots and reduce slots.
④ If JobTracker finds that the first TaskTracker has free resources, the JobTracker will call TaskScheduler's assignTasks() function, returning some task list to the first TaskTracker. At this point, the TaskTracker will execute the task assigned by the scheduler.

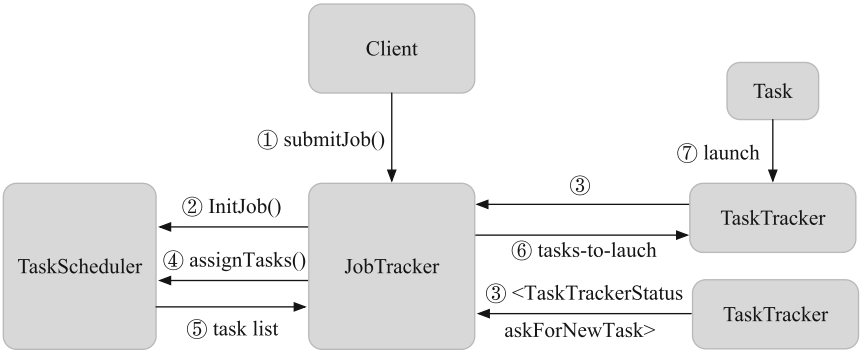

Fig. 4. Job scheduling algorithm.

4.2 HDFS Small File Processing Capability

In response to the handling of the small files mentioned above, Hadoop itself provides three solutions, namely archive file technology, serial file technology and merged file technology, all of which require users to write their own programs, and all of them are insufficient. Therefore, it has not been widely adopted (Fig. 5).

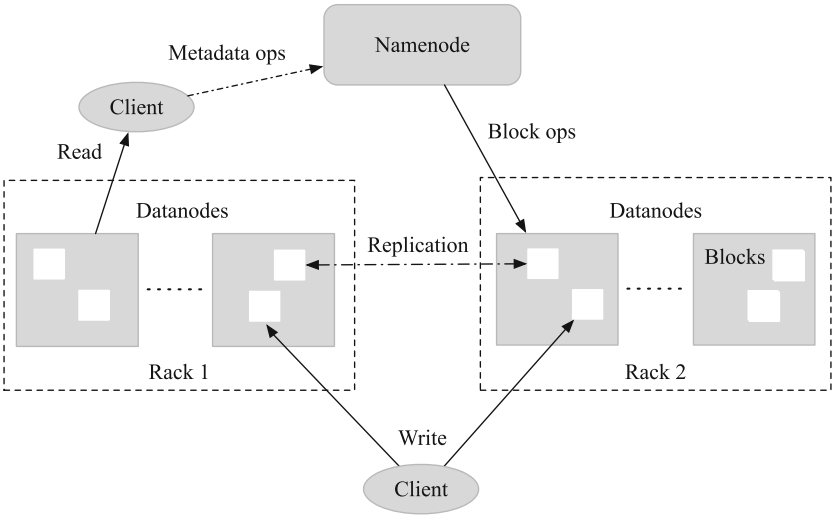

Fig. 5. HDFS architecture.

5 Conclusion

As an open source framework for cloud computing technology, Hadoop has helped many enterprises solve the problem of storage and processing of big data.

Using Hadoop in the field of traffic detection and analysis can solve the problem of storage and processing of massive data traffic. Existing researches use the Hadoop framework for message storage and processing, showing excellent processing efficiency. The above application scenarios show that for different environments, we need to build a Hadoop-based system framework that meets the actual needs. In our future work, the following aspects will be seriously considered.

(1) The current improvements to the MapReduce programming model are generally limited to a specific aspect, and a platform with shared memory is proposed. Therefore, the implementation and optimization of the MapReduce model for parallel computing environments such as distributed mobile platforms will also be an important research direction.
(2) Existing Hadoop plating algorithms are more about fairness, but real-world calculations often require higher efficiency. Combining the resources owned by the system with the current load state, it is still an important research direction to propose a more fair and efficient scheduling algorithm.
(3) Currently HDFS and HBase can support both structured and unstructured data, while fast-generated big data imposes higher real-time requirements on the underlying access platform. Therefore, it is necessary to design an access platform that supports high-efficiency, low-latency, and supports complex types of data.

References

1. Apache Zookeeper. http://zookeeper.apache.org/
2. Chandramouli, B., Goldstein, J., Duan, S.: Temporal analytics on big data for web advertising. In: 2012 IEEE 28th International Conference on Data Engineering, pp. 90–101. IEEE (2012)
3. Dede, E., et al.: MARISSA: MapReduce implementation for streaming science applications. In: 2012 IEEE 8th International Conference on E-Science (e-Science), pp. 1–8. IEEE (2012)
4. Falsafi, B., et al.: Deep analytics (2011)
5. Gubanov, M., Pyayt, A.: MEDREADFAST: a structural information retrieval engine for big clinical text. In: 2012 IEEE 13th International Conference on Information Reuse and Integration (IRI), pp. 371–376. IEEE (2012)
6. Kang, U., Chau, D.H., Faloutsos, C.: PEGASUS: mining billion-scale graphs in the cloud. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5341–5344. IEEE (2012)
7. Ketata, I., Mokadem, R., Morvan, F.: Biomedical resource discovery considering semantic heterogeneity in data grid environments. In: Hruschka, E.R., Watada, J., do Carmo Nicoletti, M. (eds.) INTECH 2011. CCIS, vol. 165, pp. 12–24. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22247-4_2
8. Lee, Y., Kang, W., Lee, Y.: A hadoop-based packet trace processing tool. In: Domingo-Pascual, J., Shavitt, Y., Uhlig, S. (eds.) TMA 2011. LNCS, vol. 6613, pp. 51–63. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20305-3_5

9. Lee, Y., Lee, Y.: Detecting DDoS attacks with Hadoop. In: Proceedings of the ACM CoNEXT Student Workshop, p. 7. ACM (2011)
10. Lee, Y., Lee, Y.: Toward scalable internet traffic measurement and analysis with Hadoop. ACM SIGCOMM Comput. Commun. Rev. **43**(1), 5–13 (2013)
11. Lee, Y., Kang, W., Son, H.: An internet traffic analysis method with MapReduce. In: Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP, pp. 357–361. IEEE (2010)
12. Verma, A., Cherkasova, L., Kumar, V.S., Campbell, R.H.: Deadline-based workload management for MapReduce environments: pieces of the performance puzzle. In: 2012 IEEE Network Operations and Management Symposium (NOMS), pp. 900–905. IEEE (2012)