Acta Technica 62 (2017), No. 7A, 537–544 © 2017 Institute of Thermomechanics CAS, v.v.i.

Design of computer network user behavior forensics analysis system based on system log

Yongdong Pan[1]

Abstract. In view of problems on scalability and difficult parallel programming during parallel processing of massive logs in the traditional distributed model, a web mass search log analysis mechanism based on Hive is proposed. HQL language, Hadoop distributed file system HDFS and MapReduce programming mode are used to analyze and deal with the massive search log for analysis and research on the user search behavior. The analysis results of hot topics, user clicks and URL rankings, and query sessions in user search behavior have some guiding significance for algorithm ranking and system optimization of search engines.

Key words. User management, System log, Search engine, User behavior, Forensic analysis.

1. Introduction

Under the background of rapid development and rapid popularization of Internet, the expansion rate of network information shows exponential growth. As the main tool for current network information retrieval, search engine has become an important way for people to access Internet resources. By recording and analyzing the user's behavior characteristics, performance of the search engine is improved through understanding on user's intent and interest, and a more common search engine technology capable of providing more personalized information services is provided for users.

In general, log record of search engines and main behavior information of interaction between uses and search engines is the main way and carrier to research and analyze true network users' behaviors. Later on, this analysis has been adopted in industrial circle and academic field. For example, in the mid 90s, Cockburn [1] et al. conducted a research analysis on browsing behaviors of users. In 1998, Silverstein

[1] Jinling Institute of Technology, School of Software Engineering, Nanjing, Jiangsu, China, 211169

[2] et al. conducted a large-scale analysis on user logs of commercial search engines.

However, in general, large-scale search engine logs are massive files, while divide-and-conquer thought is applied to processing of super-large scale data files in traditional way, i.e. multithreading and multitask is used for decomposition under the distributed environment. However, it causes relatively high programming difficulties and weak scalability. The distributed architecture based on hadoop adopted in the Thesis is able to deal with massive data files well.

2. Introduction to relevant technology

2.1. Hadoop distributed system

Hadoop is a distributed system architecture of the Apache foundation, where users can develop distributed program without understanding distributed lower-level details. It can make full use of the advantages of clusters to efficiently handle large amounts of data, for which it is an ideal solution for large-scale data processing.

The core of Hadoop consists of two parts: HDFS and MapReduce. HDFS provides a stable file system using Master/Slave architecture [3] with a management node (NameNode) to provide metadata service and to take charge of the entire file management system and N data nodes (DataNode) to take charge of storage and positioning data blocks; while MapReduce is the next parallel programming mode of hadoop which can be used for massive data processing [4]. It abstracts distributed computations into two set operations—Map and Reduce, and uses the key value pair (key/value) method to carry out parallel computation of massive data.

2.2. HIVE data warehouse

Hive [5], a mechanism that can store, query and analyze large scale data in HDFS, is a data warehouse infrastructure based on Hadoop,. It can be used for massive data extraction, transformation and loading ETL [6]. Hive defines a simple similar SQL query language (HQL) and generates a series of MapReduce tasks for data processing after analysis and conversion of languages and provides users with table query characteristics and distributed storage & calculation characteristics which are similar with traditional RDBMS partly. The Hive architecture diagram is shown in Fig.1:

(1) User interface. Hive mainly has three user interfaces–CLI, Client and WUI, in which command line interface (CLI) is the most commonly used one. Client is the client of Hive, and the user is connected to Hive Server. It needs to point out where the Hive Server is located and that Hive Server is started at that node. WUI is accessing Hive through a browser.

(2) Metadata storage. Metadata in Hive includes table name, table column, table partition, partition attributes, table attributes and directories where data of tables is located, etc. Hive stores the metadata in the database.

(3) Interpreters, compilers, and optimizer complete HQL query languages, from lexical analysis, syntax analysis, compilation, optimization, and generation of query

plans. The query plan generated is stored in HDFS and called and executed by MapReduce later.

(4) Data of Hive is stored in the Hadoop file system HDFS, and most of the queries are completed by MapReduce.

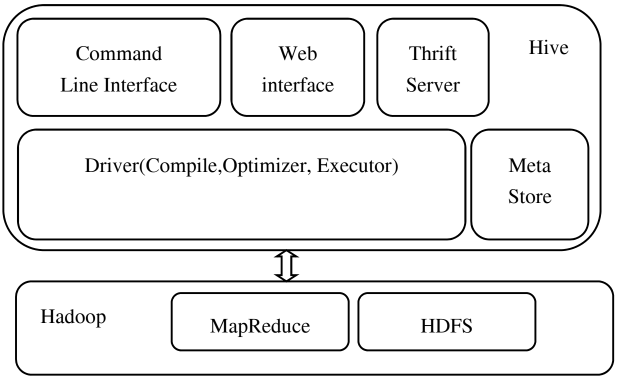

Fig. 1. Hive architecture diagram

2.3. User behavior analysis

User behavior analysis [7] refers to principles of users accessing websites through statistics and analysis on relevant data under the condition that basic data of website traffic is obtained. As for search engines, the current mainstream is about retrieval with keyword query as the carrier; therefore, query length, query number, query types submitted by users determines information contents and amount of information passed from users to search engines.

3. Processing model design of massive logs based on Hive

3.1. Data set and data format

Log files of search engine generally record user UserID (Cookie value when users use the browser to access the search engine), query keywords, ranking of URL clicked, sequence number of user clicking and URL clicked. As for a record "8561366108033201 [Reasons for Wenchuan Earthquake] 32 www.big38.net, the corresponding UserID is "8561366108033201"; the query word is "Reasons for Wenchuan Earthquake"; it ranks third in results of URL clicked; the user is the second one to click with URL clicked as "www.big38.net".

3.2. Systematic architecture design

The architecture diagram designed by the system is shown in Fig. 2. Specific implementation is as follows:

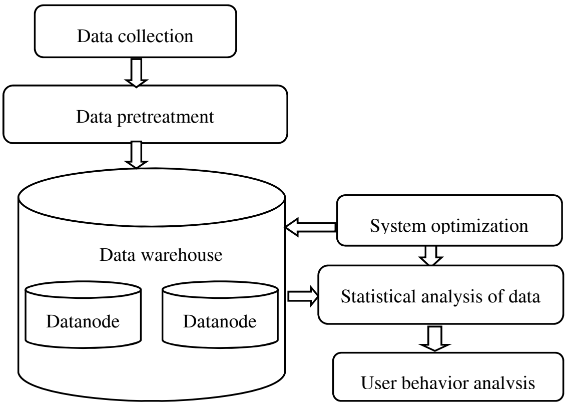

Fig. 2. System architecture diagram

(1) Data acquisition. The data used in this experiment is data set of a web query log based on clicking data in June 2008 provided by Sogou labs (www.sogou.com/labs). There are three types of data available: sample data (877KB), one day data (148MB), and one month complete data (4.23GB).

(2) Data preprocessing. Through research and analysis of the source data, it is necessary to preprocess the source data in the following three aspects: log dividing, data replicate removal and keyword Chinese words segmentation. First, for a single log with 4.23G capacity, the scale is relatively large, at the same time, hdoop performance has to be considered, for which a log needs to be divided into many appropriate small log files; the second task is to remove replicate. A large number of duplicate records are contained in the data set, the same data searched with keywords by the same user needs to be removed; the third task is Chinese words segmentation. Due to the fact that keywords input by a user may be sentences or long phrases, it is necessary to segment words and use shorter keywords to express the same meaning. Chinese words segmentation software httpcws based on HTTP agreement is used in the Thesis for open source. The structured data obtained after treatment are mapped into database tables and imported into the Hive data warehouse. The data tables are partitioned, stored and managed in the distributed file system HDFS. For example, log files are imported into HIVE data warehouse HQL: LOAD DATA LOCAL PATH '/path/20080601_log.txt' OVERWRITE into TABLE sogou_log PARTITION (ds= '2008-06-01');

(3) After steps listed in (1) and (2), data query analysis can be carried out on data warehouse through HQL. The Hive driver will transmit the received HQL statement to the compiler which will parse HQL languages into MapReduce program of hadoop for submission of tasks on hadoop cluster, on which tasks will be implemented.

(4) Finally, Hadoop cluster returns execution results to analysis module of user behavior characteristics. For example, enter the query results into the local directory of HQL: INSERT OVERWRITE DIRECTORY'/tmp/result'SELECT keywords FROM sogou_log a WHERE a.ds='<DATE>';

(5) Carry out adjustment and optimization based on actual conditions of the

system operation with emphasis paid on performance of hadoop cluster and the multi-table join query of hive.

4. Analysis on experimental results and system optimization

4 PCs are used in the Thesis to build Hadoop distributed clusters with a PC equipped with Hive as master node (Master), and the other three PCs as slave nodes (slave1-slave3). Configuration of each PC is as follows: hardware environment: Intel (R) Pentium 4 CPU 3.0GHz, 2G memory, 300G hard disk, 100Mbps Internet access. Software environment: OpenSuse Linux 11, JDK1.6.0_27, hadoop0.20.203, Hive0.7.0. There are 51537393 records in the monthly data set used in the experiment. After analysis and process of the system, three user behavior characteristics—user query topic, user clicks and URL ranking, and query session analysis are obtained.

4.1. Ranking list of query topics

In the search process, interaction between users and search engines is carried out through input of topics or keywords, so it is highly effective to understand users' interests by analyzing user's query topics. According to statistical analysis results after data processing (sorted according to the amount of accesses), there are 4685253 keywords where keywords ranking top 100 account for 62.48% of total of all keywords, different from 70% measured by Yu Huijia [8], which can be explained by word segmentation of keywords through analysis. But it shows that there are many repeated queries. If we can improve query quality, improving quality of overall retrieval, cache mechanism and dynamic index mechanism can be considered to be introduced and established.

4.2. User clicks and URL ranking

Relationship between user click sequence and URL rankings can directly reflects quality of a search engine. When the user submits a query, the search engine may return many results, of course, the user won't browse all the results returned and will only click URL closed to their query target, which requires the ranking algorithm of the search engine to put results meeting user query needs as far as possible. The results of the statistical analysis on the experimental data set are shown in Fig. 3, it can be learnt that that the top 10 URLs clicked by users account for 83.46% of all clicks.

4.3. Analysis on query session

Query session refers to the user submitting a series of queries over a period of time with a fixed search intent. The previous two user behavior characteristics are analyzed for a single query word, while the analysis of the query session can better understand and respond to the user's query intention. According to results

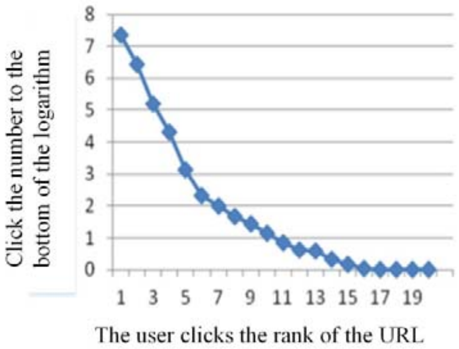

Fig. 3. Relation between user clicks and url rank

of statistical analysis on the experimental data set, it is found that query session is used by 37.45% of users through UserID. When the user fails to find the right result, it may continue by adding, deleting, or modifying the query word. If there is no satisfactory result, the user will even change the browser to carry out query or give up the query.

4.4. System optimization

Some system performance optimization process are mainly carried out through the following four aspects: (1) setting of Map and Reduce task number in hadoop cluster, because the measured efficiency for different data set is not the same; (2) introduction of combiner to reduce data copy of Map task to Reduce tasks; (3) partition clipping used for data tables in HIVE with files satisfying partition conditions read only; (4) at the time of multi-table join query, reducer will cache records of all table except the last table in join sequence, and the last table will be used to serialized into the file system, therefore the table with the largest data amount will be placed at the last position, thereby reducing the amount of memory used. The time consuming before and after optimization is shown in Table 1.

Table 1. Time consuming before and after optimization

	Total time consuming before optimization (s)	Total time consuming after optimization (s)	Efficiency promotion
(877kb) Sample data (877kb)	22,349	20,874	6.6
(148Mb) One-day data (148Mb)	35,482	32,914	7.2
(4.23Gb) One-month data (4.23Gb)	84,793	77,595	8.5

5. Conclusion

User behavior analysis is an important tool for studying information retrieval and performance evaluation, while search logs reflect use trajectory of the user. Massive data processing model based on Hive is proposed in the Thesis and it is applied to process log files of the search engine. Besides, relevant research on users' search behaviors is carried out. Moreover, the analysis results have relatively good guidance significance for retrieval sorting algorithm and personalized search of the search engine and also solves the bottleneck for massive log data parallel computing in the traditional method to a certain extent.

References

[1] W. S. Pan, S. Z. Chen, Z. Y. Feng: *Investigating the Collaborative Intention and Semantic Structure among Co-occurring Tags using Graph Theory.* 2012 International Enterprise Distributed Object Computing Conference, IEEE, Beijing, (2012), 190–195.

[2] J. W. Chan, Y. Y. Zhang, and K. E. Uhrich: *Amphiphilic Macromolecule Self-Assembled Monolayers Suppress Smooth Muscle Cell Proliferation*, Bioconjugate Chemistry, *26* (2015), No. 7, 1359–1369.

[3] Y. Y. Zhang, E. Mintzer, and K. E. Uhrich: *Synthesis and Characterization of PEGylated Bolaamphiphiles with Enhanced Retention in Liposomes*, Journal of Colloid and Interface Science, *482* (2016), 19–26.

[4] J. J. Faig, A. Moretti, L. B. Joseph, Y. Y. Zhang, M. J. Nova, K. Smith, and K. E. Uhrich: *Biodegradable Kojic Acid-Based Polymers: Controlled Delivery of Bioactives for Melanogenesis Inhibition*, Biomacromolecules, *18* (2017), No. 2, 363–373.

[5] Z. Lv, A. Halawani, S. Feng, H. Li, & S. U. Réhman: *Multimodal hand and foot gesture interaction for handheld devices.* ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), *11* (2014), No. 1s, 10.

[6] H. Jing: *Research and implementation of computer network user behavior forensics analysis system based on system log*[J]. Agro Food Industry Hi Tech, *28* (2017), No. 1, 458–461.

[7] B. Zhao: *Research and implementation of analysis system for computer network user behavior based on system log*[J]. Agro Food Industry Hi Tech, *28* (2017), No. 1, 783–786.

[8] C. Shen, Z. Cai, R. A. Maxion, et al.: *On User Interaction Behavior as Evidence for Computer Forensic Analysis*[C]// International Workshop on Digital Watermarking. Springer, Berlin, Heidelberg, (2013), 221–231.

[9] X. Y. Zhong: *A Method of Network Forensics Analysis Based on Frequent Sequence Mining*[J]. Applied Mechanics & Materials, *50-51* (2011), 578–582.

[10] X. Y. Zhong: *Design of Network Forensics Based on Apriori Algorithm*[J]. Computer Technology & Development (2011).

[11] J. Luo, W. Xu: *The application research of electronic evidence system based on analysis of user correlative behavior*[C]// Advanced Research and Technology in Industry Applications. IEEE, (2014), 718–720.

[12] G. Zeng: *Research on Forensics Method Based on Log on SaaS*[J]. Advanced Materials Research, *989-994* (2014), 4432–4436.

[13] S. K. Chavhan, S. M. Nirkhi, Dharaskar R. V.: *Visualization of Data for Host-Based Anomalous Behavior Detection in Computer Forensics Analysis Using Self Organizing Map*[J]. (2013), 4.

[14] N. K. Singh, D. S. Tomar, B. N. Roy: *An Approach to Understand the End User Behavior through Log Analysis*[J]. International Journal of Computer Applications, *5* (2011), No. 11, 27–34.

Received May 7, 2017