Analysis of CDN Website Logs Based on Hadoop

Qing Song [1], Yujun Wen[2], Junpeng Gong[2]

1. New Media Institute, Communication University of China, Beijing, 100024, China

2. School of Computer Science, Communication University of China, Beijing, 100024, China

Keywords: log analysis; mass data; Hadoop; MapReduce

Abstract. This paper designs a framework of CDN website log system based on Hadoop and a set of algorithm on the basis of user action mode excavation to analyze and process logs from searching engines. The monitor and regulation of colonies can be realized in platform monitoring modules. Under the guideline of data excavation process, this paper adopts Hadoop, an analysis tool for mass data as the experiment platform. The MapReduce reflection/excavation programming model is used. Simple and applicable HIVE from SQL and Hbase mass data pool are used to process mass logs. The writer conducts a detailed analysis on user searching action from such perspectives as topics, hits, URL order and conversational analysis to optimize platform performance and compare the system before and after the optimization. Experiment data is shown in this paper to explain that the log platform here is quite stable and efficient.

Introduction

With the development of Internet and increase of online users[1, 2], the picture data is greatly soaring. Sometimes, enterprise pictures can reach TB or even several hundred TB. Usually, DFS (Distributed File System) is used to process those pictures. Hadoop is a recent DFS to process mass data, featuring in reliability, large capacity, simple deployment and maintainability. Hadoop is a widely used distributed –memory and distributed computation framework, applicable to large-scale distributed computation. This tool has been attracted more and more attention and widely applied to advertisement computation, log analysis, webpage searching as well as data excavation. The core technology of Hadoop includes HDFS (Hadoop Distributed File System) and Map/Reduce (Distributed Computation Framework). In HDFS, a document is divided into several document modules in the same size and saved in different nodes in colonies, which is applicable to save mass logs. Map/Reduce is a distributed programming model provided by Hadoop to process large-scale colony mass data processing. By this model, programming codes tend to be written more conveniently to process mass logs.

Design of Mass Advertisement Log Analysis System

The General Desin of System Framework. The general system chart is shown in Picture 1. The framework is introduced from top as following:

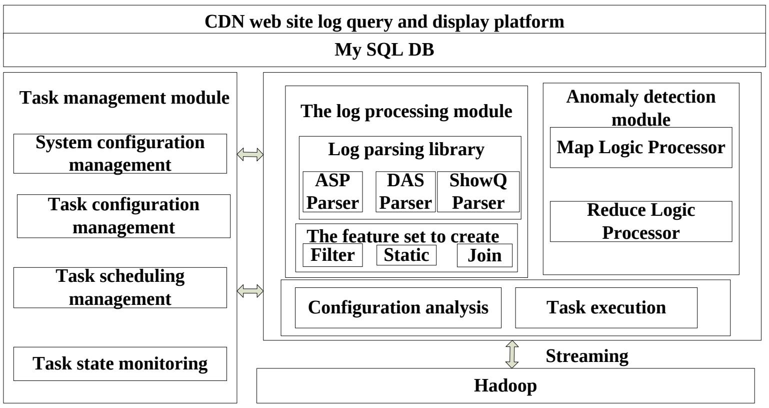
(1) Log Parsing Library: each parsing task, at first, is analyzed by the data adderes from configurative documents. Then the nodes are analyzed and the next step is to use background script. Finally, the data is transffered to regulated log nodes. This library incluedes ASP Parser, DAS Parser and ShowQ Parser.

(2) Task State Monitoring: is the log analysis excavtion module. Starting from different observation points in the system, this module excavtes log data from log parsing library. Different observation points are corresponded to different analysis models.

(3) Tak Management Module: regulates different Hadoop colony tasks including system configuration management, task configuration management and task scheduling management so as to guarantee timely execution of daily tasks in the system.

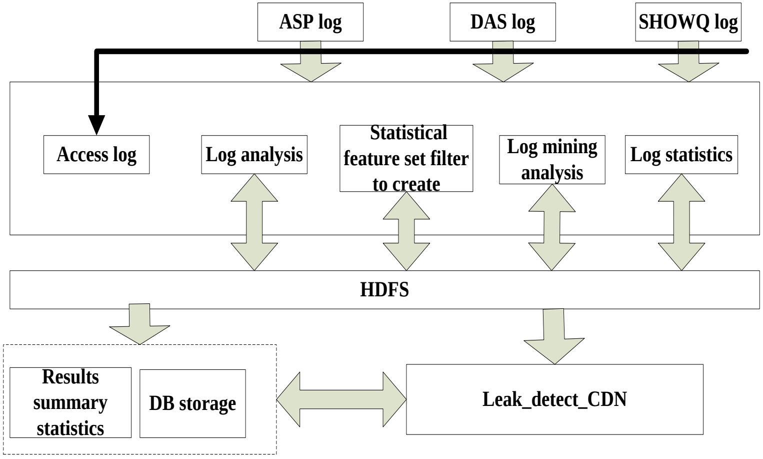
(4) Anomaly Detection Module: is based on CakePHP framework. The module shows data saved in MySQL via webpages and shows each result in a direct way. The shown content includes daily analysis result, general tendency chart and data from each observation points on year-on-year basis

and chain relative ratio.



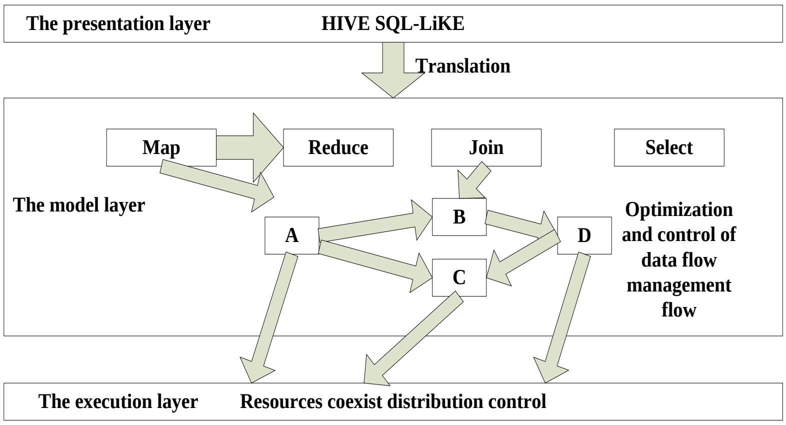
Picture 1 The General System Chart

The Design of Genral System Data Process. Firstly, the system obtains original log data from above and then dilivers data to the log processing module for filter. The analysis excavation module analyzes log data in accordance with various regulations and strategies and conducts multi-dimensional statistics to results. All the data is saved in HDFS of Hadoop and executed by MapReduce. After all the MapReduce tasks finished, the script of local servers download the analysis results from HDFS. Then the results are collected for statistics and saved. Finally, Leak_detec Web shows the results. Detailed data process can be seen in Picture 2.



Picture 2 The Data Flow Design Chart of Log Analysis System

The Design of Mass Log Analysis Model. The computation system model chart of mass searching engine logs is shown in Picture 3. Mass log data is saved in Hbase via the pre-processing module and the data-saving module. Then, under the guideline of this chart, the data is excavated. SOL data such as Hive searches sentences, the system translates it into Hadoop MapReduce for execution and the integration/regulation mode process mass data. Meanwhile, the modle service also has functions realted to data flow oprtimization and task flow management. Tasks after disintegration are directly distributed to execution level and the final processing flow is finished by resource distribution and concurrency control.



Picture 3 Log Computation Model Chart

Data Process of Excavation Logs Based on User Action

Processing Model of User Action. The Hadoop colony advantage should be fully taken, in particular the advantage of high-speed MapReduce computation and reliable save of HDFS[5, 6]. Based on the hierarchical design, HDFS is saved in the bottom level by HDFS. MapReduce computes and uses the bottom level through interface in the higher level. Apriori computation is used to scan and save mass data, which consumes a lot of time and space. Thus, it is the bottleneck of Apriori. Though many parallel plans exist, increasing scale of data lead traditional plans to become inefficient due to large demand of resources and telecommunication consumption [7].

According to hierarchy effects of user action, the action models can be divided into three levels, namely as access, activity and conversation. URL access action can be described by following units. The information is not related to action semantic information.

(1)User$_{id}$ is applied for identifying only on Web user ID and Request ID user is appliedfor identifying the only URL request from the user. Time refers to trial lecture of present URL request. Delay refers to browsing time in the requested webpage. Method refers to HTTP method adopted by URL request. The access of request is signed by URL StateSet is a two-tuples, composed of variables and variable values.

(User$_{id}$, Request ID,Time, Delay, Method,URL.,StateSet) (1)

(2)ActivityID is applied to sign the only code of some activity. Present activity name is represented by ActivityName while the representative of partial semantic information variable is StateSet such as abstract, title, keywords, hyperlinks, etc. The activity level can be described by following units:

(Use$_{rid}$, Activity IQ ActivityNane, T ime, Delay, StateSet) (2)

(3)Conversation can be described by following units. Among them, SessionID is the only sign of present conversation. Activities represent ID colonies of each activity contained by activity order in the conversation. The conversation level can be described in following units:

(User$_{id}$, SessionID, Time, Delay, StateSet, Activities) (3)

Excavation of User Frequent Action Sequence Pattern. The user action sequence pattern excavation is based on the model of user action sequence model and excavation demands, adopting Web excavation among targeted users to observe frequent, common and potential action sequence laws.

Prefix: If sequence a = <a_l,a_2,...,a_n> and sequence b=<b_l,b_2,..._,b_n>(m<n) are matched: ①At that moment, b_i = a_i. ② There is m^m. ③ If items of b_m are removed from a_m, the left items can cofigurate new sequence colony if combined with b_m. And those items must be arranged after b_m.

Suffix: If sequence b = <a_l,a_2,...,a_m> is the prefix of sequence a = <a_l,a_2,...,a_n>,it is named sequence r=<am",am+I, ...an> is the suffix of b compared with a.

Projection Database: If sequence a is a sequence model of data colony D, the sequence with a prefix or subsequence with a the longest suffix compared with a are projection sequences, namely as D|a.

Algorithm Description:

Step 1: Scan a projection database and find a serial item to formulate a sequence pattern.

Step 2: Based on the sequence patter in (1), the projection database that is corresponded to different prefix.

Step 3: Make recursion consistently in corresponding projection database until the sequence patter can not be produced.

Fci is excavated in frequent sequence patterns in different user groups, and the frequent sequence pattern is shown as follow:

$$FS_{ground}= (groupu_{id}, \{(as1, spt1)...\{as_t,spt_t)...\{aS_n, spt_n)\} ,spt_{min}]) \qquad （4）$$

Among it, gruopid is the only sign of the user group. There are L groups in total. spt_i is the support item ($1<i<n$) of this frequent action sequence and the smallest support of spt_{min} sequence pattern excavation.

The cluster technology can be divided into M category according to similarity with analysis objects or similar object cluster. M adopts sequence cluster algorithm to cluster action sequence pattern by changes on timeline and similarity computation.

Experiment Simulation and Verification

The Hadoop distributed cluster in this paper is composed of 4 PCs, one for master with Hive and the other three for slavel-slave3. Each PC is configured: Hardware: Intel (R) Pentium 4 CPU 3.0GHz, 2G RAM, 320G hard disk, 100Mbps internet access. Software: vmware workstation 8.0.4, OpenSuse Linux 11.0, JDKl.6.0 27, hadoop0.20.203, HiveO.7.0. For searching logs, not only can we represent searching with results as feature colony, but also use researching to represent results. Formula can be adopted to show similarity among results:

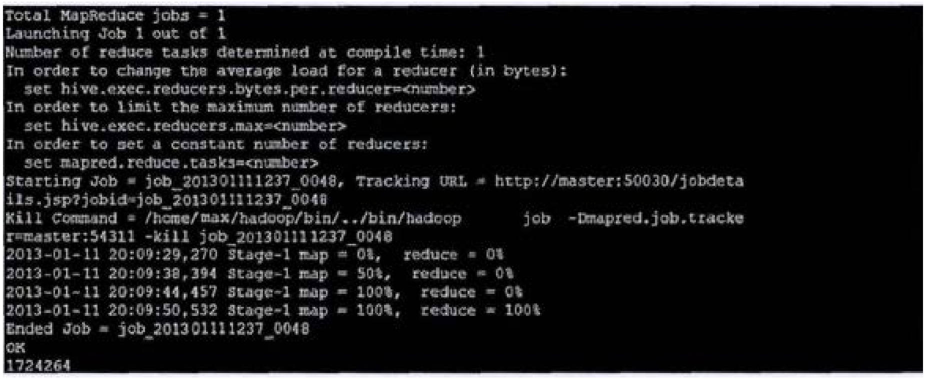
$$S(d_i,d_j)=\frac{visited(d_i,d_j)}{visited(d_i)+visited(d_j)-visited(d_i,d_j)} \qquad （5）$$

For the researching result d, the study hits the checking machine Qd. The frequency of words is tf, which means if we make a special research q, we can calculate the relevance of q and d according to word frequency. In a word, it means we can use word frequency information generated by clicking to estimate the relevance between document q and d.

The platform in this paper conduct analysis and process upon searching engine log data colony provided by sougo lab and obtains three features of user action, namely as topic, click and URL rank.

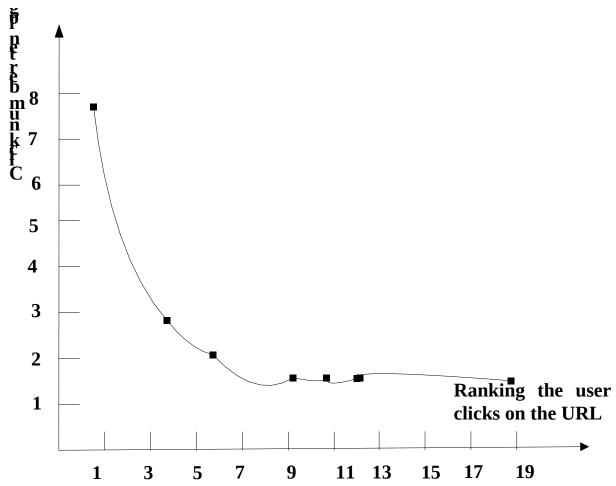
The results of data processing and analysis are ordered by visit. There are 1,724,264 modules and around 4,683,237 key words. The first 100 key words account for 62.48% of the total amount as shown in Picture 4.

The analysis results are shown in Picture 5. It can be seen that URL from top 10 visits account for 83.46% of the total. From the statistics, the URL is mainly focused on the first 10 of returning results, which means the searching engine sequence algorithm needs to put results that reflected user demands in the first page as much as possible.



```
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_201301111237_0048, Tracking URL = http://master:50030/jobdeta
ils.jsp?jobid=job_201301111237_0048
Kill Command = /home/max/hadoop/bin/../bin/hadoop      job  -Dmapred.job.tracke
r=master:54311 -kill job_201301111237_0048
2013-01-11 20:09:29,270 Stage-1 map = 0%,  reduce = 0%
2013-01-11 20:09:38,394 Stage-1 map = 50%,  reduce = 0%
2013-01-11 20:09:44,457 Stage-1 map = 100%,  reduce = 0%
2013-01-11 20:09:50,532 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_201301111237_0048
OK
1724264
```

Picture 4 Numbers of Log in One Day



Picture 5URL Ranks of User Clicks

This paper adopts scale of three data from searching logs. The data is respectively sample data (0.877Mb), daily data (148Mb) and monthly data (4.23 GB). Upon this platform, this paper tests the efficiency of searching topics in the data processing. Figure q shows needed time of different samples. The results show that when the log amount is relatively small, the advantage is not obvious compared with single-click processing and even the processing speed of the latter on is higher. While the amount is increasing, the processing speed of cluster data is prominently high.

Figure 1 Data Processing Time

	1 Node(s)	2 Nodes(s)	3 Nodes(s)	4 Nodes(s)
Sample	14. 489	18.657	20. 815	22. 349
Daily	76.389	58.126	48.397	35.482
Monthly	213.582	156.943	129.76 丨	84. 793

Conclusions

This paper adopts Hadoop framework and MapReduce programming model to conduct reflection/regulation computing mode on mass data. SOL database language, HIVE and HBase are combined together to analyze pattern process of data excavation. The platform can obtain processing results without parallel processing and distributed experiment. This paper sets up a platform for experiment, analyzes topics, clicks, URL order and user conversation. The analysis method mainly features in extension and maintainability, which provides a way for CDN website log analysis and processing.

Acknowledgment

The work was financially supported by project of the National Key Technology R &D Program (2012BAH15B01-1、2014BAK10B01-01、2014BAK10B01-02).

References

[1] Wang Runhua. Researches on Distributed Log Analysis Based on Hadoop Colony [J].Science & Technology Information,2007(15):60-69

[2] Cheng Miao. Web Log Excavation Based on Hadoop [J]. Computer Engineering.2011,06(11):37

[3] Zeng Li. Comparative Expierenment of Hadoop Colony and PC Data Processing Time [J]. Information Science, 2009,19(2): 55-56

[4] Rong Xiang, Li Lingjuan. Frequent Item Cluster Excavation Based on MapReduce [J].Academic Paper of Xi’an Institute of Posts & Telecommunications, 2011,16(4)：37-39+43.

[5] Tom White(write) , Zeng Dadan, Zhou Aoying (translate).Hadoop Authority Book [M].Beijing:

Tsinghua Unibersity Press, 2010.

[6] Cheng Miao, Chen Huaping. Web Log Excavation Based on Hadoop [J]. Computer Engineering, 2011, 37(11): 37-39.

[7] Cheng Ying, Zhang Yunyong, Xu Lei. Research on Mass Data Analysis Based on Hadoop and Relational Database. [J]. Telecommunication Science , 2010, 11: 47-50..

[8] ZhuZhu. Research and Application of Mass Data Processing Model Based on Hadoop [J]. Computer Application Technology, 2008.

[9] Duo Xuesong, Zhang Jing, Gao Qiang. Mass Data Management System Based on Hadoop[J]. Microcomputer Information, 2010(26):202-204..

[10] Cha Li. Mass Data Computing Technology Based on Hadoop [J]. Scientific Research Infromatiozation Technology and Application, 2012, 3(6):26-33.