PROCEEDINGS OF SCIENCE

MapReduce based Parallel Data Processing for Drug-drug Interaction Prediction

Faran Wei[12, 2] **,Beiji Zou**[1, 2], **Kejuan Yue**[1, 2], **Min Zeng**[1, 2], **Xiao Li**[1, 2]

[1] *School of Information Science and Engineering Central South University*
Changsha, 410083, China
[2] *Mobile Health Ministry of Education-China Mobile Joint Laboratory Central South University*
Changsha, 410083, China
E-mail: franwee@163.com

Lei Wang [1, 3]

[1] *School of Information Science and Engineering Central South University*
Changsha, 410083, China
[3] *Xiangya Hospital Central South University,*
Changsha, 410008, China
E-mail: wanglei@csu.edu.cn

Prediction of drug-drug interactions (DDIs) can prevent unexpected adverse drug events (ADEs). ADEs can damage people's health and bring about economic losses to the society. Existing methods usually adopt the dataset of the adverse drug reports (ADRs) to build the DDI prediction statistical model; however, the ADRs dataset contains billions of records and the U. S. government releases new ADRs quarterly so that the data volume keeps growing. It is time consuming to clean these data and extract useful features from ADRs, which delay the development of effective DDIs prediction models. In this paper, a parallel processing framework based on MapReduce model is proposed. The MapReduce model is utilized to extract names and adverse reactions of drugs and count their frequencies based on ADRs, which can improve the efficiency of data cleaning and feature extraction of the Food and Drug Administration (FDA)'s adverse drug reports. The parallelization of the statistical screening and processing method of ADRs are implemented on the Hadoop cluster. Experimental results show the processing of FDA's adverse drug reports can be achieved accurately by this method and the speed of processing can be improved effectively by using the proposed parallel computing framework.

ISCC2015
18-19, December, 2015
Guangzhou, China

[1]Spaker：Faran Wei

[2]This research was conducted with the support of Hunan Provincial Natural Science Foundation of China (No.09JJ6102) and Research Project of Education Department of Hunan Province, China. (No.13C143).

PoS(ISCC2015)014

1.Introduction

Adverse drug event (ADE) is an important problem for patients because these events represent the medication-related patient harm [1]. The US Food and Drug Administration (FDA) started a convenient adverse event reporting system in 1998 and has accumulated a large amount of data in terms of adverse drug reactions [2]. With the increasing of ADEs, how to process this information has become one of the problems to be solved urgently at present. Especially after the 1960s, it is difficult for regular computers to deal with the rapid increase of drug adverse event reports and the data dimension. Traditional ADEs monitoring is completed by screening adverse drug event reports manually, most of which are screened by a single computer [3]. It is costly and time-consuming, which hinder the analysis efficiency of drug side effects. In the big data era, as a framework for analyzing and processing large data Hadoop emerged as the times required [4]. Currently, the study of medical big data based on Hadoop platform has just commenced. Applying Hadoop cluster for distributed processing of the large high-dimensional data can improve the data storage capability, reliability and computing speed [5]. At the same time, it can help doctors to improve work efficiency, guide clinical practice, which is of very great significance on people's livelihood.

Hadoop is a distributed system infrastructure, developed by the Apache Foundation. It mainly consists of Hadoop Distributed File System (HFDS) and MapReduce. Among them, HDFS is a file system that provides scalable and reliable data storage, which was designed to span large clusters of commodity servers [6-8]. MapReduce is a software framework for easily developing applications to process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters [9-10].

This paper mainly studies how to use MapReduce model to preprocess the FDA data [11]. Traditional monitoring of adverse drug events (ADE) use manual filter ADE reports. The method of using standalone Oracle or SQLServer databases and Excel processing the data is not only of high cost, but also wastes a lot of time. In this paper, we use MapReduce programming model to write parallel programs to extract names and adverse drug reactions of drugs [12]. Then we merge them and count the frequency of drugs and their adverse drug reaction. All the outputs of the Reduce task are exactly the pretreatment results of FDA's drug side effects. The focus of this paper is to achieve the parallelization of processing with MapReduce, improve the efficiency of data cleaning and feature extraction of the ADR on the Hadoop cluster, and deal with the daily growing medical data. Experimental results show the processing of FDA's adverse drug reports can be achieved accurately by this method and the speed of processing can be improved effectively by using the proposed parallel computing framework.

2.FDA data prerocessing

2.1FDA's Data Sources

The data sources refer to 20,976,732 reports of adverse drug reactions published by adverse drug reaction reporting systems between January 2004 and April 2014 [13]. The contents of the reports include the patient's basic information, the basic drug information, and adverse reaction information, patient's outcomes, indications and drug information. At present, the amount of data is still rising at a rate of about several millions per year. Figure 1 shows the

statistics for every year since 2004 [14], we can see, FDA data has reached a million level. In the past five years, the amount of FDA's data has doubled. With the increasing amount of data, it is necessary for us to use the Hadoop to deal the large data.

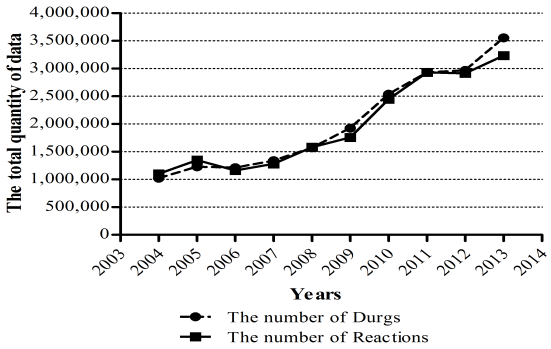

Figure 1: Growth Rate of FDA's data

2.2The FDA's data pre-processing procedure

Our experiments use real data provided by FDA on its official website. These data are standard structured data . We have processed the data by screening, merging, and statistical computing in turn. Figure 2 shows the flowchart of data processing.

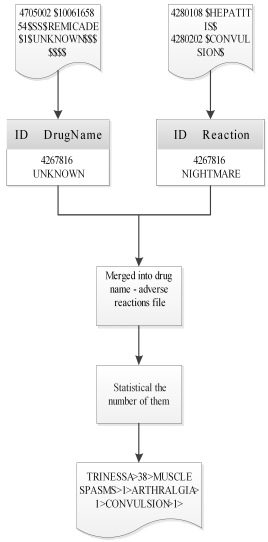

Figure 2: Flowchart of FDA Data Preprocessing

we calculate the frequency of ADR. Define as the frequency of the i'th drug that produce the j'th reaction, and R as the total number of the i'th drug's reports [15]. Thus:

$$F_{ij} = 1/R_i * \sum_{(k=1)}^{R} 1\left(AE_j \in R_{ik}\right) \tag{2.1}$$

3.Parallel Processing descriptions

3.1Extraction of drug name and adverse reaction name

The extraction of drug name and drug reaction shares similar MapReduce programming mode. According to the Programming mechanism of MapReduce, we set the ID number as the key and drug name as the Value. At the same time, the Sum is added on the Reduce side. By setting Sum, a single drug or multiple drugs can be chosen as output. For the adverse drug

reactions resulting from the data, we also use the form of (key, value) and achieve the merger of a variety names of ADR with the same ID number.

3.2Join section

The job of the join part realizes the merger between the drug and the reaction of the drug. Step 1 and 2 introduced the realization of the Join part [16].

Step1. Map of DrugJoin <key,value>.

a) Distinguish data sources, read the file line-by-line, and split.

b)Set id as key.According to different sources, the value takes different forms. When the data source is from DrugName, tag it.

For the Reduce-Function, the input is the data obtained from the Map of each host, which have set tags as to the drug Name. The Reduce-Function is shown in *Step 2*.

Step 2. Redcue of DrugJoin<key,V>.

c)Base on the tag split the values and put them into the different Vectors. Then make the two vectors to do cartesian product.

d)Output<key, values>.

3.3Statistical Calculation

The output of the join section contains drug name and adverse drug reactions. On the basis of statistical analysis, we can get the times of drug and adverse reactions caused by this drug. Step 3 and 4 introduce the realization of the Statistical calculation part.

Step 3. Map of DrugStatistical<key,value>.

a)Read the file and split it into SplitFile,then set SplitFile [1] as key.

b)Split the file of the splitFile[2], and set it to value.

For the Reduce-Function, the input is the data obtained from the Map. The Reduce-Function is shown in Step 4.

Step 4. Reduce of DrugStatistical<key,V>.

c)Define HashMap<String,Integer>.

d)Traverse the value to add up all the side effects and resulting number and write into HashMap.

e)Obtain the drug name as key, the side effects and resulting number as value.

f)Output <key value>.

4.Experiment design and result analysis

Two sets of experiments are carried out to verify the results. One is used to verify the effectiveness of our method for the data preprocessing of ADRs. Another proves that the parallel processing really improved the efficiency of computing, meanwhile it demonstrates the advantage of MapReduce.

4.1Validation of the Methods' Effectiveness

Structured Query Language (SQL), a database query and programming language, is used to store and retrieve data inquire, update and manage the database system. We can take advantage of SQL database to import raw data into it and get an accurate result by SQL. Due to large amount of data and retrieval time, we only extract data of recent years for some common

drugs. According to the same datasets, we process them with our method and SQL Server respectively. The results show that our method is completely consistent with the results obtained by SQL. Meanwhile, an extraction test is taken randomly for special drugs and compared with the data retrieved by SQL database. Hadoop statistical results are the same as SQL query results (Table 1), which can prove the effectiveness of our method.

	Drug Name	**Reaction1**	**Reaction 2**	**Reaction 3**	**Reaction 4**
	Tri-sprintec	***Dysmenorrhoea***	***Pregnan cy on oral contraceptive***	***Metrorrhagia***	***Mood swings***
SQL	30	2	4	3	2
Hadoop	30	2	4	3	2
The frequency F	100%	6.667%	13.333%	10%	6.667%
	Travoprost	***Conjunctival oedema***	***Pruritus***	***Ocular hyperaemia***	***Iritis***
SQL	26	1	4	2	1
Hadoop	26	1	4	2	1
The frequency F	100%	3.846%	15.385%	7.692%	3.846%
	Trisporal	***Gastroenteritis***	***Tinnitus***	***Ear discomfort***	***Pancreatitis acute***
SQL	14	3	1	1	2
Hadoop	14	3	1	1	2
The frequency F	100%	21.429%	7.143%	7.143%	14.286%

Table 1: Analysis result

4.2Verification of the Performance Improvement

Speedup and Sizeup are defined to measure parallel computing performance. This experiment was run on Hadoop platform that can be carried out large-scale data parallel computing. The cluster we used is built by Dell workstation, considering the optimal problem of data quantity and the number of clusters. We adopted three nodes for performance verification. Each server is configured to CPU: E3-1240, 4core, frequency 3300MHz, memory capacity 32G. Software environment is Hadoop version 1.2.1. The data of experiment is described in Table 2.

Data Type	**Records**	**File size**
Standard data	41,396,434	1312.3M
Two times the standard	82,792,868	2658.7M
Four times the standard	165,585,736	5200M

Table 2: Test data

Contrast test:

In order to verify the improvement of our method in efficiency, we increase the nodes and the amount of data to test the effectiveness of our proposed method. One is expanding the amount of data exponentially in a single node (Table 3), another is adding nodes and expanding the amount of data (Table 4). Among them, the column name in the table is the program name for extracting the drug; reaction is the adverse drug reaction procedures for the drug extraction; Join is the combined program of the drug name and the name of adverse drug reaction; Sum is the result of statistical procedures.

Name	**The amount of data**	**Times(s)**	**Name**	**The amount of data**	**Times(s)**	**Name**	**The amount of data**	**Times(s)**
Name	1	94	***Name***	2	170	***Name***	4	324
Reaction	1	96	***Reaction***	2	173	***Reaction***	4	342
Join	1	129	***Join***	2	149	***Join***	4	173

Table 3: Execution time(single node)

Name	**The**	**Times(s)**	**Name**	**The**	**Times(s)**	**Name**	**The**	**Times(s)**

	amount of data			amount of data			amount of data	
Name	1	54	***Name***	2	83	***Name***	4	154
Reaction	1	53	***Reaction***	2	95	***Reaction***	4	186
Join	1	116	***Join***	2	124	***Join***	4	144
Sum	1	58	***Sum***	2	148	***Sum***	4	276

Table 4: Execution time(three nodes)

In Table 3 and Table 4, we've found that when the amount of data increases, the time consumption obviously changes in the part of Name, Reaction, Sum program; but only a few changes in the Join part. As the data is expanded by way of copying data, although the amount of data rises, it produces the same ID number. Before the drugs and side effects are merged by the same ID number, it has been merged by the Name and Reaction procedure; therefore, in the Join, the times of the distributed processing do not increase substantially and the time consumption in the Join part has a little change with increase of the data. In order to show the effectiveness of parallel computing more clearly, the data above has been processed and measured by speedup and sizeup.

Among them, Speedup is defined as follows:

$$Speedup\ （p）=T_1/T_p(p=1,2...) \tag{4.1}$$

Keep the fixed processing task quantity unchanged, and increase the number of child nodes P successively (such as increase from 1 note to 4 notes). Among them, T_1 is the time consumption for a single node to perform the task; T_p is the time consumption for P nodes to perform the task. Speedup should coincide with the straight line y = x ideally. Since the consumption of inter-node communication gradually increased with the increasing of the node data, it is difficult to achieve the deal situation. Figure 3 shows that the experimental results of the Speedup almost achieve the linear results. With the increase of the amount of tasks, it is near to the ideal value and Speedup performance will be expected to further improve with more large-scale data processing. The results of 4 times the benchmark are better than 1 time the benchmark and twice the benchmark. As the task quantity is little of 1 time the benchmark and twice the benchmark, the other added nodes do not come into effect.

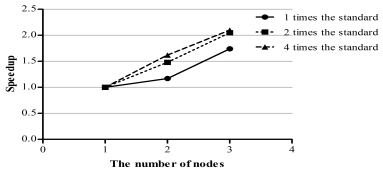

Figure 3: Scalability Evaluation

The Sizeup defined as follows:

$$Sizeup(D,p)=T_M/T_N \tag{4.2}$$

Keep the node number unchanged and increase the volume of each task. T_m is the time to perform the task D of p times the benchmark; T_n is the time required to execute the benchmark task D. The growth rate of Sizeup results will be lower and lower, as an increasing convex function. It indicates that Sizeup performance is great; the curve of Figure 4 shows this characteristic.

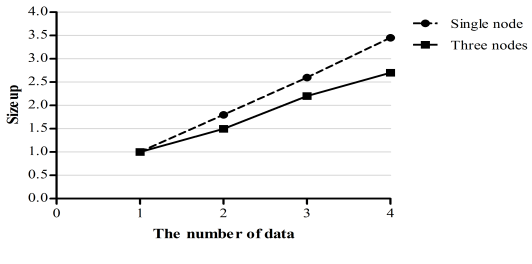

Figure 4: Quantity Evaluation

From Figure 4, we can find that the cluster does not show advantage when the amount of data is less and even limited by the cluster start up and the network communication. There is a little difference between single node and the three nodes on time consumption, but the time of cluster spending would be significantly reduced with the increase of amount of data through further experiments. The time of single machine processing cost would increase exponentially. This method proves the advantage of using cluster for processing as well as the feasibility of the parallel computing.

5.Conclusion

Under the background of big medical data, the increasing numbers of the adverse drug reaction reports are blocking the analysis of ADR. Although the data preprocessing is an important part of data analysis and mining, there has been little focus on the aspect of efficiency and data collection in previously published work on FDA data preprocessing. And Most of the methods are single machine processing and manual extraction. With the growth of FDA data, this will become a challenge.

Hadoop, as the big data processing technology, will become an inevitable choice for data cleaning by Hadoop platform. In this paper, a FDA data preprocessing scheme based on Hadoop-MapReduce model is proposed. According to the characteristics of the Mapreduce model, the processing is divided into several parts, each of which can be executed concurrently. Compared with the use of single machine processing, the experimental results show that the proposed scheme can improve the efficiency. The method is a feasible solution to the process of FDA data in the future. We believe that the model will play its role more often with the growth of FDA data.

References

[1] Dr Desireé L. Kunac, Julia Kennedy, Nicola Austin, David Reith. *Incidence, preventability, and impact of Adverse Drug Events (ADEs) and potential ADEs in hospitalized children in New Zealand: a prospective observational cohort study*[J]. Paediatric Drugs. 11(2):153-160(2009).

[2] Thomas J.Moore, Michael R. Cohen, Curt D. Furberg. *Serious Adverse Drug Events Reported to the Food and Drug Administration,1998-2005*[J]. Archives of Internal Medicine. 167(16):1752-1759(2007).

[3] Jason Lazarou, Bruce H. Pomeranz, Paul N. Corey. *Incidence of adverse drug reactions in hospitalized patients: a meta-analysis of prospective studies*[J]. Jama the Journal of the American Medical Association. 279(1):1200-1205(1998) .

[4] P. Vignesh Raja, E. Sivasankar. *Modern Framework for Distributed Healthcare Data Analytics Based on Hadoop*[M]. Information and Communication Technology, Springer Berlin Heidelberg. 8407:348-355(2014).

[5] Yifan Sun, Yi Xiong, Qian Xu, and Dongqing Wei. *A Hadoop-Based Method to Predict Potential Effective Drug Combination*[J]. BioMed Research International. (7): 196858-196858(2014).

[6] Ronald C Taylor, *An Overview of the Hadoop/MapReduce/HBase Framework and its Current Applications in Bioinformatics*[J]. BMC Bioinformatics. 11(6):3395-3407(2010).

[7] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, Robert Chansler. *The hadoop distributed file system*[C]. Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on. IEEE, 2010:1-10.

[8] S Ghemawat, H Gobioff, ST Leung. *The Google File System*[C]. ACM SIGOPS operating systems review. 37(5):29-43(2003).

[9] Jeffrey Dean, Sanjay Ghemawat. *MapReduce: Simplified Data Processing on Large Clusters*[J]. In Proceedings of Operating Systems Design and Implementation (OSDI). 51(1):107-113(2004).

[10] Bingsheng He, Wenbin Fang, Naga K, Govindaraju, Qiong Luo, Tuyong Wang. *Mars: A MapReduce Framework on Graphics Processors*[J]. PACT'2008. 2008:260-269(2008).

[11] Avrilia Floratou, Jignesh Manubhai Patel, Eugene Jon Shekita, Sandeep Tata. *Column-Oriented Storage Techniques for MapReduce*[J]. PVLDB. 4(7):419–429(2011).

[12] Ralf Lämme. *Google's MapReduce programming model—revisited*[J].Science of Computer Programming. 70(1): 1–30(2008).

[13] Ali Ayad, Hartzema Abraham G. *Assessing the Association Between Omalizumab and Arteriothrombotic Events Through Spontaneous Adverse Event Reporting*[J]. J Asthma Allergy. 5:1-9(2012).

[14] Mohamed A Omar, James P Wilson. *FDA Adverse Event Reports on Statin-Associated Rhabdomyolysis. Annals of Pharmacotherapy.* 36(2): 288-295(2002).

[15] Elisabetta Poluzzi, Emanuel Raschi, Ugo Moretti BiolSc, Fabrizio De Ponti MD. *Drug-induced Torsadesde Pointes: Data Mining of the Public Version of the FDA Adverse Event Reporting System (AERS)*[J]. Pharmacoepidemiology and Drug Safety.18(6):512–518(2009).

[16] Dawei Jiang, Anthony K.H. Tung, Gang Chen. *Map-Join-Reduce: Toward Scalable and Efficient Data Analysis on Large Clusters*[J]. IEEE Transactions on Knowledge and Data Engineering. 23(9):1299-1311(2011).