International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Backup Anomaly Identification with R and Hadoop

Ravindra Phule[1], Madhav Ingle[2]

[1] JSPM'S JSCOE, Pune India, Savitribai Phule Pune University

Abstract: ***In recent years "big data" has become something of a buzzword in business, computer science, information studies, information systems, statistics, and many other fields. As technology continues to advance, we constantly generate an ever-increasing amount of data. This growth does not differentiate between individuals and businesses, private or public sectors, institutions of learning and commercial entities. It is nigh universal and therefore warrants further study. Increasingly larger scale applications are generating an unprecedented amount of data. In order to exploit data mining techniques on collected backup job metadata, we integrate a big data analytics platform with the existing enterprise backup architecture. We build a scalable data mining platform to store, process, and perform advanced data mining techniques on the overall data set. We leverage open source tools to reduce the overall cost while preserving flexibility.***

Keywords: Bigdata; Big data analytic; Hadoop; data analytics; knn; RHadoop, framework.

1. Introduction

This paper focuses on scalable big-data systems, which include a set of tools and mechanisms to load, extract, and improve disparate data while leveraging the massively parallel processing power to perform complex transformations and analysis. Owing to the uniqueness of big-data, designing a scalable big-data system faces a series of technical challenges, including:

- First, due to the variety of disparate data sources and the sheer volume, it is difficult to collect and integrate data with scalability from distributed locations. For instance, more than 175 million tweets containing text, image, video, social relationship are generated by millions of accounts distributed globally [1].
- Second, big data systems need to store and manage the gathered massive and heterogeneous datasets, while provide function and performance guarantee, in terms of fast retrieval, scalability, and privacy protection. For example, Facebook needs to store, access, and analyze over 30 pertabytes of user generate data [1].
- Third, big data analytics must effectively mine massive datasets at different levels in realtime or near realtime - including modelling, visualization, prediction, and optimization - such that inherent promises can be revealed to improve decision making and acquire further advantages.

These technological challenges demand an overhauling re-examination of the current data management systems, ranging from their architectural principle to the implementation details. Indeed, many leading industry companies have discarded the transitional solutions to embrace the emerging big data platforms.

However, traditional data management and analysis systems, mainly based on relational database management system (RDBMS), are inadequate in tackling the aforementioned list of big-data challenges. Specifically, the mismatch between the traditional RDBMS and the emerging big-data paradigm falls into the following two aspects, including:

- From the perspective of data structure, RDBMSs can only support structured data, but offer little support for semi-structured or unstructured data.
- From the perspective of scalability, RDBMSs scale up with expensive hardware and cannot scale out with commodity hardware in parallel, which is unsuitable to cope with the ever growing data volume.

To address these challenges, the research community and industry have proposed various solutions for big data systems in an ac-hoc manner. Cloud computing can be deployed as the infrastructure layer for big data systems to meet certain infrastructure requirements, such as cost-effectiveness, elasticity, and the ability to scale up or down. Distributed file [2] and NoSQL [3] databases are suitable for persistent storage and the management of massive scheme free datasets. MapReduce [4], a programming framework, has achieved great success in processing group-aggregation tasks, such as website ranking. Hadoop [5] integrates data storage, data processing, system management, and other modules to form a powerful system-level solution, which is becoming the mainstay in handling big data challenges. We can construct various big data applications based on these innovative technologies and platforms. In light of the proliferation of big-data technologies, a systematic framework should be in order to capture the fast evolution of big-data research and development efforts and put the development in different frontiers in perspective.

2. Related Work

The big data era arrives with high volume of heterogeneous data (Nature Editorial 2008; Mervis J. 2012; Labrinidis and Jagadish 2012). Every day, [6] 2.5 quintillion bytes of data are generated and 90 percent of the data in the world today were generated within the past two years . Our capability for data generation has never been so powerful and enormous ever since the invention of the Information Technology in the early 19th century. On October 4, 2012, the first presidential debate in between President Barack Obama and Governor Mitt Romney triggered more than 10 million tweets within just two hours (Twitter Blog 2012). Among all these tweets, the specific moments that generated the most discussions

actually revealed the public interests, such as the discussions about Medicare and vouchers. Such online discussions provide a new means to sense the public interests and generate feedback in real-time, and are mostly appealing compared to generic media, such as radio or TV broadcasting. Another example is Flickr, a public picture sharing site, which received 1.8 million photos per day, on average, from February to March 2012 [7]. Assume the size of each photo is 2 megabytes (MB), this resulted in 3.6 terabytes (TB) storage every single day. As "a picture is worth a thousand words", the billions of pictures on Flicker are a treasure tank for us to explore the human society, social events, public affairs, disasters etc., only if we have the power to harness the enormous amount of data. The above examples demonstrate the rise of Big Data applications where data collection has grown tremendously and is beyond the ability of commonly used software tools to collect, manage, and process within a "tolerable elapsed time".

The most fundamental obstacle for the Big Data applications is to explore the large volumes of data and extract useful information or knowledge for future actions (Rajaraman and Ullman, 2011). In many situations, the knowledge extraction process has to be very efficient and close to real-time because storing all observed data is nearly infeasible. For example, the Square Kilometer Array in Radio Astronomy consists of 1,000 to 1,500 15- meter dishes in a central 5km area. It provides hundred times more sensitive vision than any existing radio telescopes, answering fundamental questions about the Universe. However, with a forty gigabytes (GB)/second data volume, the data generated from the SKA is exceptionally large. Although researchers have confirmed that interesting patterns, such as transient radio anomalies can be discovered from the SKA data, existing methods are incapable of handling this Big Data. As a result, the unprecedented data volumes require an effective data analysis and prediction platform to achieve fast response and real-time classification for such Big Data.

3. Motivation

The strengths of R lie in its ability to analyze data using a rich library of packages but fall short when it comes to working on very large datasets. The strength of Hadoop on the other hand is to store and process very large amounts of data in the TB and even PB range. Such vast datasets cannot be processed in memory as the RAM of each machine cannot hold such large datasets. The options would be to run analysis on limited chunks also known as sampling or to correspond the analytical power of R with the storage and processing power of Hadoop and you arrive at an ideal solution.

4. Problem Statement

How to analysis Big Data using the current state-of-the-art technology in agreed upon time and budget? We introduce research of leverage big data analytics in the enterprise IT management domain. In specific, work investigates the backup management solution design from a scalable data mining perspective. This platform integrating big data techniques with enterprise backup analytics architecture is introduced.

5. Big Data Analytics Platform

Figure 1: Big data analytics platform

In order to analyze the massive backup job data, which is also constantly expanding with new backup job information arrivals from operational data centres, we build a scalable data mining platform to store, process, and perform advanced data mining techniques on the overall data set, as illustrated in Figure 1. We leverage open source tools to reduce the overall cost while preserving flexibility. For backup job metadata storage, we leverage Cassandra NoSQL database due to its salient features of scalability and no central point of failure, among others [8]. The backup job metadata are replicated from the database server in a background process for data analytics purpose. The proof-of-concept Cassandra cluster consists of 3 nodes with Ubentu 12.04 and each node has 4 CPU cores, 4GB memory, and 100GB direct-attached disks, which emulates a small cluster using low cost commodity hardware. Note that our nodes are virtual machines and the allocated resources can be adjusted to experiment the impact of heterogeneous cluster hardware configurations on the performance of parallel data mining algorithms, which is another research focus of ours and beyond the scope of this paper. To facilitate parallel data mining for the massive data set, we utilize Hadoop as the backend batch processing parallel computation platform, which is configured on the same set of nodes with Cassandra to facilitate data locality. In addition, we leverage the RHadoop framework [9] on top of our Hadoop and Cassandra cluster. RHadoop utilizes Hadoop streaming capability where each mapper and reducer function can be written in native R language instead of Java. Therefore, the rich capability of R in terms of statistic computing can be leveraged. In our scenario, we will store tens of millions time series data in Cassandra using customized schema, and the data mining tasks (for all time series data) are distributed to all work nodes of the Hadoop cluster in parallel. The platform can scale easily by simply adding commodity nodes with Cassandra, Hadoop client, and RHadoop packages installed. Our parallel data mining platform integrates with the backup analytics frontend tightly and provides graphical visualization to provide additional insights to the backup administrator via web-based management consoles.

6. Backup Anomaly Identification using knn

Prior to implementing the K-Nearest Neighbor technique lets see how to handle the input and output for the implementation. Since we are using the MapReduce paradigm we must make sure that the input is form of a <key,value> pair[10].

The Map routine performs the function of calculating the distance of each data point with the classes and lists it out. The Reduce routine then chooses the first 'k' Neighbors in increasing order of distances and conducts a majority vote. After which it sets the data point's label as the label of the class with the majority vote count.

Now, we need to organize the input and output directories.
To do this let us name the directory that holds the data vectors as vectors and the training and testing data as trainFile and testFile.

Having organized our input/output directories and training and testing data ready, we can apply the k-Nearest Neighbor technique in a distributed environment by following the algorithms discussed below to design the Map and the Reduce functions for the k-Nearest Neighbor Technique.

A. Algorithm for the Map Function

Algorithm 1 Mapper design for k-NN

0: **procedure** K-NN MAPDESIGN
0: Create list to maintain data points in the testing data-set
0: $testList = new\ testList$
0: Load file containing testing data-set
0: $load\ testFile$
0: Update list with data points from file
0: $testList <= testFile$
0: Open file containing training data set
0: $OPEN\ trainFile$
0: Load training data points one at a time and compute distance with every testing data point
0: $distance\ (trainData, testData)$
0: Write the distance of test data points from all the training data points with their respective class labels in ascending order of distances
0: $testFile <= testData(dist, label)$
0: Call Reducer
0: **end procedure**=0

B. Algorithm for the Reduce function

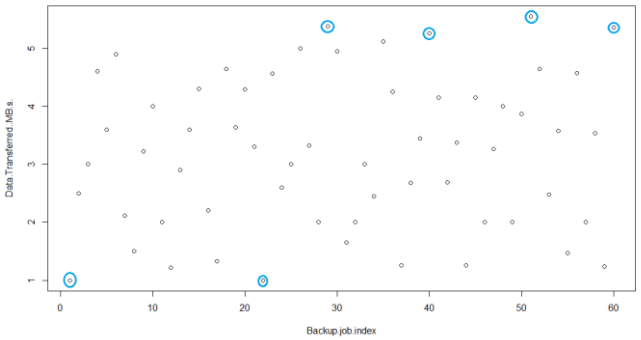
Algorithm 2 Reducer design for k-NN

0: **procedure** K-NN REDUCEDESIGN
0: Load the value of 'k'
0: Load testFile
0: $OPEN\ testFile$
0: Load test data points one at a time
0: $READ\ testDataPoint$
0: Initialize counters for all class labels
0: $SET\ counters\ to\ ZERO$
0: Look through top 'k' distance for the respective test data point and increment the corresponding class label counter
0: $for\ i = 0\ to\ k$
0: $COUNTER_i++$
0: Assign the class label with the highest count for the testDataPoint in question
0: $testDataPoint = classLabel(COUNTER_{max})$
0: Update output file with classified test data point
0: $outFile = outFile + testDataPoint$
0: **end procedure**=0

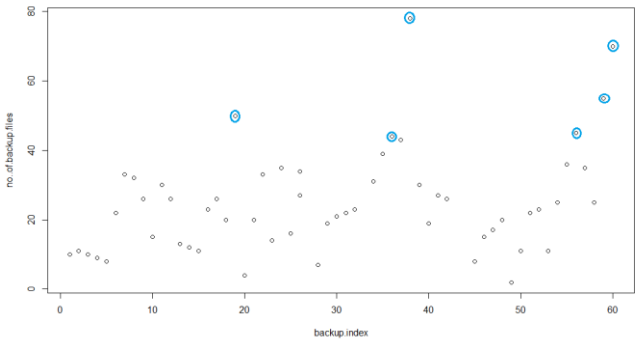
C. Algorithm to Implement the Map and Reduce Functions

Algorithm 3 Implementing kNN Function

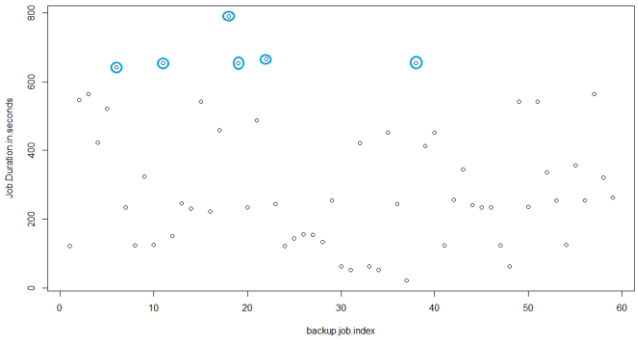
0: **procedure** KNN FUNCTION
0: Read the value of 'k'
0: $SET\ 'k'$
0: Set paths for training and testing data directories
0: $SET\ trainFile$
0: $SET\ testFile$
0: Create new JOB
0: SET MAPPER to map class defined
0: SET REDUCER to reduce class define
0: Set paths for output directory
0: SUBMIT JOB
0: **end procedure**=0



(a)



(b)



(c)

Figure 2: Top 6 outliers of (a) data transferred (b) number of backup files, and (c) job duration during for a backup client.

References

[1] J. Kelly. (2013). *Taming Big Data* [Online]. Available: http://wikibon.org/blog/taming-big-data/

[2] J. H. Howard *et al.*, ``Scale and performance in a distributed _le system,"*ACM Trans. Comput. Syst.*, vol. 6, no. 1, pp. 51_81, 1988.

[3] R. Cattell, ``Scalable SQL and NoSQL data stores," *SIGMOD Rec.*, vol. 39, no. 4, pp. 12_27, 2011.

[4] J. Dean and S. Ghemawat, ``Mapreduce: Simpli_ed data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107_113, 2008.

[5] T. White, *Hadoop: The De_nitive Guide*. Sebastopol, CA, USA: O'Reilly Media, 2012.

[6] IBM 2012, What is big data: Bring big data to the enterprise, http://www-01.ibm.com/software/data/bigdata/, IBM.

[7] Michel F. 2012, How many photos are uploaded to Flickr? http://www.flickr.com/photos/franckmichel/6855169886/.

[8] T. Rabl, M. Sadoghi, H.-A. Jacobsen, S. Gomez-Villamor, V. Muntes- Mulero, and S. Mankovskii, "Solving big data challenges for enterprise application performance management," *VLDB*, vol. 5, 2012.

[9] RevolutionAnalyltics, https://github.com/RevolutionAnalytics/RHadoop/wiki.

[10] P Anchalia, Kaushik Roy The k-Nearest Neighbor Algorithm Using MapReduce Paradigm