BDAP: A Big Data Analysis Platform Based On Spark

Zhixiang WANG 1,a, Yao Bu 1, Demeng BAI 2, Bin WU 1,b and Jiafeng QIN 2

1 Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications Beijing, Beijing, 100876;
2 State Grid ShanDong Electric Power Research Institute, Jinan, 250000

Email: aWangzhixiang513@gmail.com, bwubin@bupt.edu.cn

Abstract. Big data analysis platform consists of a serious of application and technologies for gathering, cleaning, storing, and analysing, which contribute to manage large-scale information, get analysis result and make decision precisely. However, traditional data analysis platforms meet new challenges as both continued growth in both scale and complexity in real data, existing data analysis platforms rarely have comprehensive functions for users. In response to these challenges, the BDAP (Big Data Analysis Platform) is proposed. BDAP supports large-scale data storage, parallel ETL process, data mining, statistical analysis, text mining and power grid data processing which are based on Spark. This demo introduces two application scenarios, transformer power prediction and video classification are presented. Experimental results show that BDAP is efficient and effective for data mining.

1. Introduction

Data analysis platform consists of a serious of application programs and technologies for gathering, cleaning, storing, analysing and providing access to data, which contribute to manage large-scale information, get analysis result and make decision precisely. Big data analysis platform research has attracted more and more attention, the application of Hadoop and other big data processing technology are introduced in paper [1]. From the perspective of open source tools, both MapReduce (based on mahout project) and Spark (based on the MLlib) are focused on the realization of the algorithm implemented parallel. However, they are all single algorithm package, not integrated as one.

Current big data analysis platforms are each have different emphasis. Datix [2] implements an extensible network analysis platform, which proposes intelligent pre-partition storage scheme to lessen the time for pre-processing such as filtering and querying. Tsinghua cloud set up Kloud [3] platform to achieve the table auxiliary index design, which improve the table process efficiency. Big-Provision [4] provides a platform for comparing the accuracy and performance of the same algorithm in different frameworks. Relatively enterprise projects have PDMiner [5] and BC-PDM [6]. These integrations of some basic data mining pre-processing and algorithms, but these are not comprehensive enough, and the correlation between components is too strong to modify easily.

In response to these challenges, the BDAP is proposed which based on Spark. BDAP provides not only multiple data pre-processing and data mining algorithms, but also individual business-specific components. Platform framework also has been significantly improved, the combination of OSGI and Tomcat can build low coupling and scalable workflow engine easily.

Technical background of BDAP includes Hadoop [7] and Spark [8]. HDFS (Hadoop Distributed File System) is a distributed file system based on Hadoop framework. Hadoop includes a variety of
tools including HBase [9], Pig, Hive, and Sqoop, which cover distributed databases, distributed data warehouses, and data transfer. Apache Hadoop YARN [10] is a new MapReduce [11] framework refactored by Hadoop, which can be called as new MapReduce. Alluxio is a high performance, fault-tolerant, memory-based open source distributed storage system which is compatible with features of Hadoop MapReduce and Apache Spark. Workflow engine [12] can determine the information transmission routes by the role of the division of labour and different conditions, content levels and other core solutions. Workflow engine includes the process of node management, flow management, process management and other important features. JBPM, Activiti and Shark are mature workflow engines. Storm is a more mature real-time processing framework now, which always coordinates with Kafka.

The rest of this paper is organized as follows: The architecture and the features of BDAP are described in Section 2. Section 3 present the experimental results. Section 4 introduces the application scenarios of this platform. Finally, we conclude and present directions of future work in Section 5.

2. Platform Overview
The overview architecture of BDAP is presented in Figure 1. The platform consist of four main layers and feature of each layer is described as followed:

- **Cloud Platform Layer** provides cloud computing environment and data storage.
- **Algorithm Layer** provides the core data processing components of this platform, including ETL, data statistical analysis and some commonly used data mining algorithms.
- **Service Layer** provides operational functions such as operating flow engine, scheduling engine, progress monitoring, system management, data visualization analysis and so on.
- **Interface Layer** is based on the web application of the user terminal; it provides good experience for the user to set the data processing flow or the other service. As shown in Figure 2.

And BDAP has also these following features:

2.1. Fully Functions
BDAP can handle various data including structured, semi-structured and unstructured text data. Compared with MLlib, as shown in Table 1, it's clear that BDAP basically covers functions of MLlib.
Table 1. Comparison of algorithms for MLib and BDAP

|                      | MLib                                                                 | BDAP                                                                 |
|----------------------|----------------------------------------------------------------------|----------------------------------------------------------------------|
| Classification       | Text Bayesian/ decision tree/ neural network/ SVM/ linear regression/ logistic regression/ C45/CART/ CHAIID/neural network/ automatic classifier | Text Bayesian/naive Bayes/SVM/ linear regression/logistic regression/C45/CART/ CHAIID/neural network/automatic classifier |
| algorithm            |                                                                      |                                                                      |
| Association rules    | FpGrowth                                                             | Apriori/FpGrowth/Time series association rules/ Association rules based on information entropy |
|                      |                                                                      |                                                                      |

2.2. Low Coupling
In BDAP, functions are highly modular. Import, pre-processing operation, various algorithms and results-presentation are all integrated into independent components.

Workflow and scheduling flow is one of the features of BDAP. Workflow is manually set up by users, it is connected by some processing components and directional arrows, which are dragged by the user to the "canvas". As shown in Figure 3, the operation of the workflow is: Import Metadata → Data Type Checking → Field Type Conversion → where → SVM → Roc.

![Figure 3. Workflow presentation](image)

Workflow engine is the core module of the system batch processing, mainly because of the OSGI. OSGI will be integrated with Tomcat to call the service, BDAP use OSGI to achieve service management. OSGI-based service is a scalable and flexible plug-in architecture, every algorithm component is independent and has good scalability and low coupling.

2.3. Visualization
Visualization of the platform can improve the user experience, users can participate in changing parameters, processes, monitoring and results presentation.

![Figure 4. Results](image)  
![Figure 5. Monitoring](image)
Results Presentation: The output components enable the visualization of the results presentation. As shown in Figure 4, BDAP currently supported output results are text output, pie chart, histogram, line graph, Roc curve, table and decision tree.

Monitoring: A workflow starts running with a monitoring window, including job name, start time, end time, duration time and job status. The performance of algorithms is intuitively shown as Figure 5.

3. Performance
Experimental environment introduction: The Spark cluster consists of 31 slave nodes and 1 master node. The server is dell R720. CPU is Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz. Each machine has 2 processors; 64G running memory. The operating system is 64-bit centos6.5. JDK version is jdk1.7. Parallel computing framework is hadoop 2.6.0 and Spark 1.5.1.

3.1. ETL and Hive
In order to test the large-scale data, this paper uses the randomly generated 25-dimensional data sets sized as 100M, 1G, 2G, 5G respectively.

The SQL statements required in OperGenColumn component are as followed:
- alter table t1 add columns (c string); update t1 set column26 = column1 + column2;

The SQL statements required in the Attribute exchange component are as followed:
- insert into newt1 select column2, column1, column3, column4… column25, column26, from t1;

It is obvious that even a simple function of the components requires a number of SQL statements. Hive maps structured data into a table and it is SQL statement that completes the steps. So more complex functions need more SQL statements, it takes time to convert SQL statements into MapReduce, which explains why ETL is superior to Hive in Figure 6.

![Figure 6. ETL and Hive performance comparison.](image1)

3.2. Algorithm Performance
MLlib is a commonly used machine learning algorithm library. MLlib and BDAP have been compared in aspect of function comprehension. Now, we compare the performance of the algorithm and take the BP algorithm in BDAP as an example. This algorithm adjusts the learning rate dynamically, increases the momentum factor, uses the small batch gradient descent method, trades the cross entropy cost function. The BP algorithm is compared with the multi-level perception classifier (MLPC) in MLlib.

In order to test large-scale data, this paper use data replication to generate the data based on data set covtype. The data scale is half million, 1 million, 2 million, 5 million and 10 million respectively. Result is shown in Figure 7.
Figure 7. Running time comparison between BP and MLPC

Figure 8. Speedup ratio of Spark-BIRCH in different nucleus and different data line

Table 2. Compared With The Experimental Results.

|       | a9a  | Ijcnn1 | skin_nonskin | Webspam | covtype |
|-------|------|--------|--------------|---------|---------|
| Time  | 46   | 30     | 90.94        | 98      | 0.773   | 79.30   | 0.602   | 71.695  |
| Acc   | 0.841| 0.904  | 0.983        | 0.999   | 0.991   | 0.992   | 0.960   |

In addition, BDAP provides a PMDP-SVM [13] (Partial-Model Dividing Parallel Support Vector Machine) model, which combines the characteristics of data distribution. PMDP-SVM combined with the Spark parallel framework, the clustering algorithm is introduced to support parallel training, and proposed a local model partitioning algorithm based on sample distribution. In this part, we evaluate the SVM on the platform by experiments, mainly focus on the performance of the training time and the accuracy of prediction. The experimental results in Table 2 show that the classification performance of PMDP-SVM is close to global optimal solution of LIBSVM, the prediction accuracy is higher than the Spark framework SVMWithMiniBatchSGD provided by MLlib tool.

In general, the speedup ratio of large data sets is proportional to the number of CPU cores. With the increase in the number of CPU cores, the speedup ration also increases. In the case of a certain number of CPU cores, more data points lead to higher speedup ratio. With the extension of data, both of the value of speedup ratio and trend of increase is more obvious. It is shown in Figure 8.

4. Application Scenarios

4.1. Power Grid Data
The EMS data is an important type of various power grid data, relating to the efficiency of power grid running. The changes in each value of EMS data never appear in one kind of value, and is often associated with changing trend of other kinds of value. Mining potential internal relationship between the EMS data has great significance in improving the data processing.

Based on BDAP, we use the improved algorithms and design a workflow to analyse the association rule of the EMS data, including EMS data pre-processing, calculation EMS data threshold, association rules analysis and etc. Figure 9 shows the workflow and we get the number of rules form the dataset through the workflow is far more than the algorithm based on clustering as shown in Figure 10.
4.2. Video Classification

In this part, we use the SVM classification to classify and predict. We use data set KTH to train and test, KTH includes 6 types of human actions: walking, jogging, running, boxing, waving and clapping. 25 subjects respectively in 4 scenarios, 6 types of video recorded, S1 is a small range of outdoor sports scenes, S2 is a large range of outdoor with certain scale transformation scenes, S3 outdoor scenes wearing different clothes, S4 is the indoor scenes.

5. Conclusions And Future Work

In this paper, a big data analysis platform (BDAP) based on Spark and Hadoop is proposed. In this platform, functions are comprehensive. The coupling degree between the components is low. Some operations such as adjusting parameters, establishing processes, monitoring, and results presentation are all visible. BDAP set up concepts of workflow and scheduling flow, workflow can save and delete. In performance, the performance of the algorithm of BDAP is better than that of Hive and MLlib. Furthermore, we analyse 2 application Scenarios, one is on power grid data, the second is on video classification.

In future work, we plan to optimize the authority management in order to let the different users can operate different data. Data exchange progress should be real-time monitored. We will set up more analytical applications around the Storm, such as real-time fault prediction analysis, real-time video mining, real-time text processing, etc.

6. Acknowledgment

This work is supported in part by the National High-tech R&D Program (863 Program) of China (2015AA050204) and State Grid Science and Technology Project (520626170011).

References

[1] Jie Song, Zongzhe Sun and Keming Mao et al. 2017. Journal of software. MapReduce big data processing platform and the research progress in algorithm. 28(3). 514-543
[2] Sarlis D, Papailiou N, Konstantinou I, et al. 2015. Datix: a system for scalable network analytics. *Acm Sigcomm Computer Communication Review*. 45(5). 21-28

[3] An Zhuo. 2012. *Research and implementation of large data analysis platform based on P2P scalable architecture* (Beijing: Tsinghua University)

[4] Li H, Lu K, Meng S. 2015. Bigprovision: a provisioning framework for big data analytics. *IEEE Network*. 29(5). 50-56.

[5] Qing He, Fuzhen Zhuang and Li Zeng et al. 2014. PDMiner: parallel distributed data mining tool platform based on Cloud Computing. *Chinese science*. 44(7). 871-885

[6] Yu L, Zheng J, Shen W C, et al. 2012. BC-PDM: data mining, social network analysis and text mining system based on cloud computing. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1496-1499.

[7] Mckenna A, Hanna M, Banks E, et al. 2010. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research*. 20(9). 1297-303.

[8] Chatzimilioudis G, Costa C, Zeinalipouryazti D, et al. 2016. Distributed In-Memory Processing of All k Nearest Neighbor Queries. *IEEE Transactions on Knowledge & Data Engineering*. 28(4) 925-938.

[9] George, L. 2011. *HBase: The Definitive Guide*. (Sebastopol: O'Reilly Media, Inc).

[10] Vavilapalli V K, Murthy A C, Douglas C, et al. 2013. Apache Hadoop YARN: yet another resource negotiator. *Symposium on Cloud Computing. ACM*. 5. 1-16.

[11] Dean J, Ghemawat S. 2004. MapReduce: Simplified Data Processing on Large Clusters. *Conference on Symposium on Operating Systems Design & Implementation*. 10. 107-113.

[12] Jain A, Ong S P, Chen W, et al. 2015. FireWorks: a dynamic workflow system designed for high-throughput applications. *Concurrency and Computation: Practice and Experience*. 27(17). 5037-5059

[13] Chang Liu, Bin Wu and Yi Yang. et.al. 2016. Multiple Submodels Parallel Support Vector Machine on Spark. *IEEE International Conference on Big Data*. 945-950.