A Real-Time Traffic Flow Prediction System for National Freeways Based on the Spark Streaming Technique

Jichiang Tsai, *Member, IEEE,* Tien-Yu Chang, Yu-Hsiang Fang, and En-Shuo Chang
Department of Electrical Engineering, National Chung Hsing University,
Taichung 402, Taiwan, ROC

***Abstract*-- In this paper, we discuss how to use the Hadoop parallel computing cluster architecture along with Spark's high-speed computing to analyze national road traffic databases for real-time estimation of future traffic information. Hadoop can process data in parallel and is inexpensive to build. It can also use its cloud data processing system to quickly upload collected data to the Hadoop Distributed File System (HDFS). On the other hand, in addition to enjoying the most suitable platform architecture, Spark also provides Spark Streaming to receive files and information in real time. This allows immediate calculations to make the forecasting system quickly aware of the traffic flow minutes ago. Hence, more accurate forecasting information can be acquired. In conclusion, our real-time forecasting system for national road traffic flows will use the HDFS cooperated with the Spark Streaming and Spark MLlib Decision Tree operations. Furthermore, the parameters of machine learning will be adjusted by the difference between the original data and the estimated one to improve the efficiency and accuracy of the overall system.**

I. INTRODUCTION

Hadoop [1] is the most representative and practical application architecture for Big Data. Through the Hadoop platform for processing and analysis, we can understand the behavior of users, such as which sites are regularly visited, which applications are the most frequently downloaded and used, and where to find answers for various questions. The above services have gradually emerged through big data analysis. On the other hand, in terms of computing, Spark's in-memory and real-time streaming computing have gradually become the core data processing infrastructure in the data center. It does not just employ the traditional slow and default computing mode. In addition to high-performance, high-security, and high-reliability features, there are also many other features like graphics, parallel processing, machine learning, elastic expansion, heterogeneous resource integration and global cache acceleration for Spark to meet the needs of real-time computing of contemporary applications.

In this paper, we integrate some applications related to Hadoop and Spark to build a common platform for using open databases to provide users with some interesting functions. In particular, our platform will give drivers more accurate information about the traffic volume at each time point on the freeway. Hence, based on such information, we can achieve a smoother traffic flow. These functions are not provided by the Traffic Data Collection System (TDCS) [2] in the Taiwan Area National Freeway Bureau, MOTC R.O.C.

This work was supported by the Ministry of Science and Technology, Taiwan, ROC, under Grant MOST 106-2221-E-005-021-.

Seeing that Hadoop can process data in parallel and its construction cost is very cheap, we exploit the Hadoop cloud data processing system to build the core part of our platform. Moreover, after the data is uploaded to the Hadoop cloud system, i.e., HDFS [3], Spark Streaming and MLlib operations are performed accordingly. More importantly, the error of each predictive data is provided to adjust the parameters of machine learning so that the whole system can be tuned to predict the traffic information more accurately.

II. PRELIMINARY

A. *Hadoop*

Hadoop, a distributed computing open source framework from Apache open source organization, has been used on many large websites, such as Amazon, Facebook and Yahoo. The core design in the Hadoop framework is MapReduce [4] and HDFS. HDFS is an acronym for Hadoop Distributed File System, which provides the underlying support for distributed computing storage.

B. Spark on Yarn

Yarn [5] is an external cluster manager used by Spark to schedule Spark jobs and allocate resources. In Hadoop 2.x, the resource management system and MapReduce are two separate components. MapReduce manages data processing and Yarn manages resource allocation [6].

In Yarn ResourceManager (RM), Yarn receives a job request and evaluates if any resources are available, and then decides to allocate resources for the job. Yarn is usually installed on the same node as HDFS, so it allows Spark to quickly access HDFS data while running Spark on Yarn. Spark on Yarn also offers user capabilities for sorting, isolating, and prioritizing Spark independent workloads.

C. System Structure

In this work, the Hadoop clustering platform is constructed with several virtual machines to build the HDFS distributed database as the system storage system. The data for storage are the average traveling speed (TDCS_M05A) for each vehicle type between stations in the Traffic Data Collection System (TDCS) in the Taiwan Area National Freeway Bureau, MOTC, R.O.C. After using the streaming read architecture of Spark Streaming to convert archives, adding new feature values and normalizing the data, the data is stored in HDFS. The collected real-time data are used to predict the trips, speed and other information of the vehicles on the national freeway through the Decision Tree Regression [7] of Spark MLlib machine learning. In the regression analysis, RMSE (Root Mean Square Error) [8]

is used to calculate the average error between the predicted result and the actual result, and to evaluate whether the current machine learning method is set up with the best combination of parameters.

Hadoop Multi Node Cluster planning is shown in Fig. 1, which is composed of multiple computers.

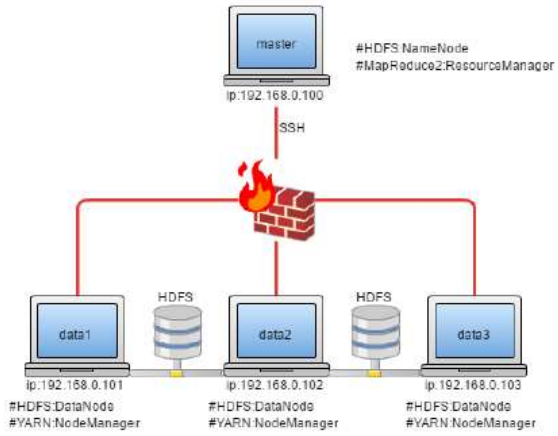

Fig. 1 System Structure.

III. SIMULATION RESULTS

In this system, we use one Master and three Slaves to form a Cluster. Master uses Yarn to manage the three Slaves. The roles of Master are Resource Manager, NameNode and SecondNameNode. Slave's job is to distribute and store data for DataNodes. The system work flow is show in Fig. 2.

The Traffic Data Collection System (TDCS) contains a lot of information, including dates, time points, vehicle types, average speeds, vehicle speeds, and so on. Each piece of information is very valuable and important to the machine learning that follows. In order to make the machine learning more accurate, we need more features and values, and also fine-tune all the information into several items, some of which becomes the most important information on a node. Through the information provided by the TDCS, the directions of vehicles, the national freeway numbers, the station numbers, etc. can be extracted from TDCS for real-time national freeway traffic forecast. Particularly, Spark MLlib's Decision Tree Model is used in our work for data training. Moreover, we add Regression to improve the Root Mean Square Error (RMSE) calculation to find out the best parameters so as to provide the most accurate forecast.

After the training process begins, we start to import the data and establish the data needed for the evaluation. We randomly divide the data into three parts of 80%, 10% and 10%, respectively. When we use a decision tree regression algorithm to perform data training, a decision tree with features and labels is built as a training tree. Finally, after having the best parameters up to now and establishing the best model, the forecast data is imported and we starts to conduct new data training. Finally, 20 pieces of information are randomly taken out as the test data. Then, the forecast is made and the prediction error value is calculated to judge the degree of correctness. Hence, the final forecast result will employ 20 random numbers to predict the data.

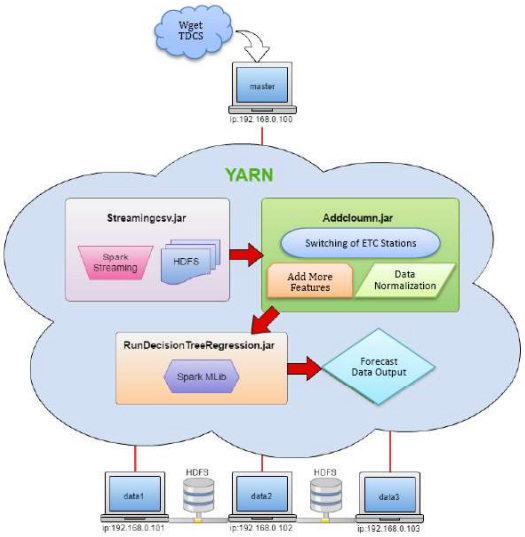

Fig. 2 System Flow.

IV. CONCLUSIONS

Our system is mainly constructed by the use of Spark Streaming and Spark MLlib. Also, the entire system needs to execute three programs sequentially, all of which are developed through the high-end Scala language to significantly highlight the convenience and high-speed computation of Spark. In the Spark MLlib part, we use the Decision Tree Model and the Regression method for data analysis and prediction. Particularly, even with the increase of data volume in the future, we only need to perform data training again to establish a realistic assessment model so that the overall system can still have higher usability and accuracy. As for the database, the TDCS has only been established for a short period of time. Hence, after the traffic information is dramatically increased, our system can accomplish a much better forecast. In the future, we hope to incorporate weather information or more information of local factors into the field of feature values, allowing the decision tree model to have sufficient features to predict the national freeway traffic much more accurately.

REFERENCES

[1] Apache Hadoop Wikipedia, https://zh.wikipedia.org/wiki/Apache_ Hadoop, Jan. 2018.

[2] Traffic Data Collection System (TDCS) at the Taiwan Area National Freeway Bureau, MOTC, http://tisvcloud.freeway.gov.tw/, Jan. 2018.

[3] What is HDFS?, https://www.ibm.com/analytics/us/en/technology/ hadoop/hdfs/, Jan. 2018.

[4] MapReduce research on warehousing of big data, http://ieeexplore.ieee.org/document/7973634/, Jan. 2018.

[5] Running Spark Applications on YARN, https://www.cloudera.com/documentation/enterprise/5-4-x/topics/cdh_ig_running_spark_on_yarn.html, Jan. 2018.

[6] Putting Spark to Use: Fast In-Memory Computing for Your Big Data Applications, https://blog.cloudera.com/blog/2013/11/putting-spark-to-use-fast-in-memory-computing-for-your-big-data-applications/l, Jan. 2018.

[7] Decision Tree Regression, http://scikit- learn.org/stable/auto_examples/tree /plot_tree_regression.html #sphx-glr-auto-examples-tree-plot-tree-regression-py, Jan. 2018.

[8] Root-mean-square deviation From Wikipedia, https://en.wikipedia.org/wiki/Root-mean-square_deviation, Jan. 2018.