Spark기반의 농업 빅데이터 분석 플랫폼 설계

뉘엔 신 녹, 뉘엔 반 퀴엣, 김경백
전자컴퓨터공학부
전남대학교

Design of Spark based Agricultural Big Data Analysis Platform

Ngoc Nguyen-Sinh, Quyet Nguyen-Van, Kyungbaek Kim
Dept of Electronics and Computer Engineering,
Chonnam National University
e-mail : sinhngoc.nguyen@gmail.com, quyetict@utehy.edu.vn, kyungbaekkim@jnu.ac.kr

요 약

Nowadays agriculture plays a crucial role in the economic growth of every country in the world. To improve the quality and productivity, farmers and ranchers need more information related to their products such as weather forecast, farming guide and market demand and so on. This information comes from various sources such as remote sensor, forecasting system, market analytics report. Previous statistic methods are not suitable to analyze such a large amount of heterogeneous data and provide an accurate result. In this paper, we propose a new design of Spark based big data analysis platform in order to enhance the performance of analyzing more accurate prediction of corps.

1. Introduction

There are some factors from the environment concerning to growing of plant such as temperature, moisture, rain, windy. The sudden changing of climate effects to the productivity of crop; therefore, we must adjust appropriately the cultivation for the those changing. For example, when having a changing of forecasting we will adjust the water discharge of the pump automatically for the good growing of potato. To produce the prediction, we must analyze the data in history of the cultivation, which updated by the manager after each crop. This data had accumulated in several years ago as Big data, which challenges for traditional agriculture platform.

Recently, there are several platforms for big data analyzing and processing in agriculture [1, 2]. In which, they use MapReduce [4] techniques in Hadoop to analyze the agricultural data. Hadoop is a good big data platform with MapReduce techniques, it also supports for parallel computation. In reality, analyzing the historical data with MapReduce spends a lot of time to fetch data from hard disk. This is disadvantage of Hadoop in execution iterative application. In this paper, we propose a new design that will accumulate the data from many sources such as sensor, forecasting system, open government data system and process this data to indicate a desire result. That is a Spark based big data analysis platform in order to enhance the performance of analyzing more accurate prediction of corps.

The remainder of this paper is as follow: in Section 2 we describe proposed system architecture include Data Collection, Data Management, and Data Analyzing. We present the case study of the problem above in Section 3 and finally Section 4 we will make a conclusion for overall article.

2. Proposed System

We propose a system that will accumulate various data from the remote sensor of farms such as temperature, humidity, water consumption, the data from forecasting system such as temperature, sunshine, moisture, windy, rainy and so on. Moreover, other data source such as open government data, restaurant, and supermarket. After that, we will analyze the data in history to suggest association rules. A good rule will be selected based on input information for water discharge. In this paper, we focus on the combination and analyzing data in history by RDD operation [5] on Spark Apache with APRIORI algorithm. At the currently, Hadoop Apache had already in Big data

processing but the problem in Hadoop framework is that the speed of read and write operation are too slow. It means that each phase in MapReduce must read the input data from hard disk and write output data to hard disk when it finish. It spends a lot of time for data fetching on hard disk. Another framework to resolve this problem is Spark Apache, which improves the speed of read and write operation data in each phase of MapReduce. Instead of executing the operation on hard disk, it use in-memory approach to speed up the performance.

The overall concept of the proposed system is illustrated in Figure 1. The proposed system consists of three main components: Data Collection, Data Management and Data Analyzing.

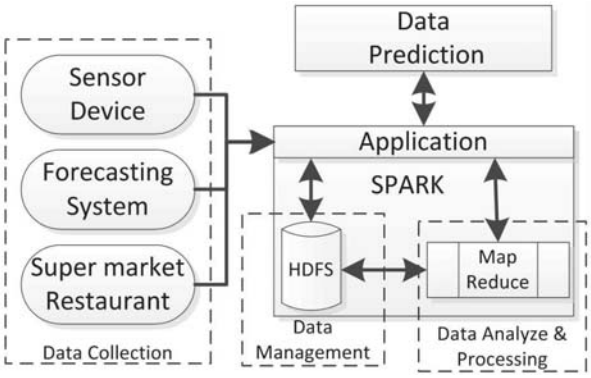

Figure 1 : Agriculture Platform

A. DATA COLLECTION COMPONENT

The first component is Data Collection which collects the data from many sources as the Figure 2.

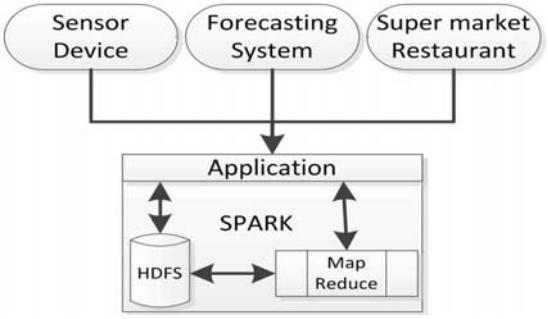

Figure 2: Data Collection Agent

In a farm, many sensors are deployed to monitor the temperature, humidity, and water consumption. A large amount of unstructured data will be collected from these sensors and sent to our system everyday. Another data resource is forecasting system, which gathers the environment information such as temperature, humidity and sunlight. The changing of climate affects significantly to the crop and productivity. Hence, this information is important to select the good rule for water discharge after analyzing data in history. We also acquire the information from market and restaurant for the price, consumption of product. It helps us on demand computation for the product.

B. DATA MANAGEMENT

There are many data sources in our system, each of them giving a different type of data. All information is unstructured and raw. We need to organize it into a structured data set before putting to HDFS storage [6]. We need to summary them into a general dataset before putting into HDFS storage. We use HDFS for storage purposes. Hadoop Distributed File System is a distributed file system designed to run on commodity hardware. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. It provides high throughput access to application data and suitable for application that have large data sets. Also, it is reliable storage system with many large files across machine in a large cluster.

C. DATA ANALYZATION

The traditional analysis method is appropriate with small and concentrates of data. To process a large amount of data sets in daily and historical, big data analytics is a good choose in this case. Big data analytics is the process of examining large data sets containing a variety of data types to uncover hidden patterns, unknown correlations, market trends, customer preferences and other useful business information. Apache Spark integrates with MapReduce are as the representative of non-relational data analysis techniques and suitable for large-scale parallel processing. Apache Spark is a fast and general-purpose cluster computing system like Hadoop. The outstanding of Apache Spark is an in-memory data processing.

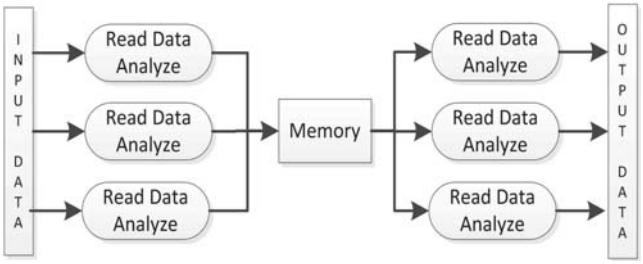

Figure 3: Data Analyzing and Processing

Figure 3 shows the flow of data analyzing, data processing and giving the output after predicting. We use MapReduce technique to divide jobs into the small task. It will be managed and assigned into many machines which take part into this system. Especially, our system runs on Apache Spark which uses RDD operation. In this, we will classify our data into many types, which data is most frequently and the other is less frequently. Many data concerning to history and sensor data is most frequently data. It will be loaded into the memory by RDD to speed up application. The output of Read Data/Analyze phase will be stored in memory. The input data of next phase has already in memory too. So that it will save the time to read/write data from hard disk instead of memory. This technique will speed up the application to 10 times with traditional MapReduce techniques.

3. Case study of big data analyzing in agriculture platform to adjust the water discharge

Assume that, we have a plan for potato, with four stages of growing, each stage includes many days. When having the changing of forecasting, the system will analyze and process data from history to give the best rule to adjust the water discharge at a time for the good growing of potato.

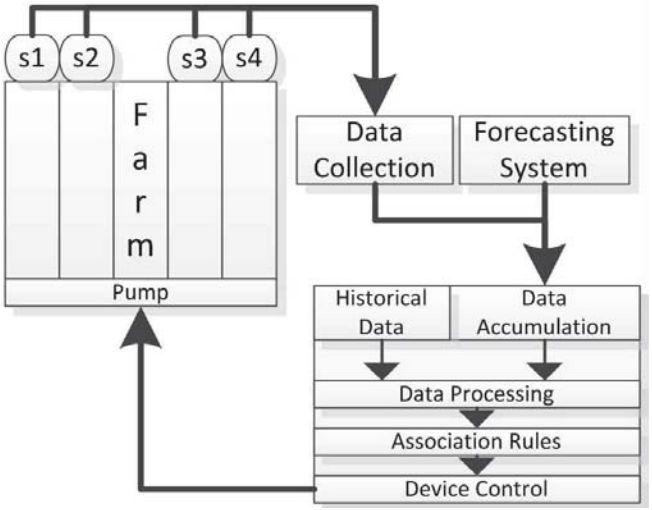

Figure 4 : Proposed System

Figure 4 shows the overview of the proposed system. In which, Data Accumulation will gather data from forecasting system and sensor. Data Processing will process historical data to give association rules and select the best rule based on output of Data Accumulation. Device Control will read the rule and make an instruction to adjust the pump.

To resolve this problem, we need to get the information from data sources and analyze data of history to find the association rules for water discharge. Table 1 shows the structure of data in history with any stages of growing, each stage includes many days. In this table, we save the selected value of each factor which is good for the growth of potato on each day. "Result" field indicate the good or bad value of growth, which updated by the manager each crop.

There are any different methods to analyze and process big data. In this paper, we use APRIORI algorithm to find the association rules. In this algorithm, it is based on the frequent of items and proposes the associated rules by the most frequently of candidates. Also minsup is defined as minimum support threshold. If we define high minsup, we may not find any associated rules. In the other, if we define low minsup, we will have more association rules. So that we must run this algorithm many times with different minsup to find a good rule.

There are several platforms for big data analyzing and processing in agriculture [1, 2]. Most of them use Hadoop framework. Hadoop is a good big data framework with MapReduce techniques, it also supports for parallel computation. The disadvantage of Hadoop is spending most of time for fetching data from hard disk. It must read data from hard disk as input and write data to hard disk as output. So that it is not good in iterative application.

Table 1: Example of Statistic Data History

Stage	day	temper (°C)	sun (%)	moisture (%)	rain (ml)	windy (km/h)	result	water (ml/m²)
Stage 1	1	25	69	66	100	25	G	500
	2	27	60	67	120	35	G	600
	…	…	…	…	…	…	…	…
	15	25	68	65	90	50	G	650
Stage 2	16	23	50	63	80	60	G	700
	…	…	…	…	…	…	…	…
	45	25	52	65	105	30	B	800
Stage 3	46	27	58	67	120	15	G	900
	…	…	…	…	…	…	…	…
	75	30	42	70	160	55	B	1000
Stage 4	76	26	48	66	140	60	G	1200
	…	…	…	…	…	…	…	…
	90	27	68	67	120	40	G	1250

Apache Spark provides an in-memory method which keeps the frequent data in memory to reduce the time for data fetching from hard disk called RDD operation. So it is better than Hadoop Apache in iterative application execution.

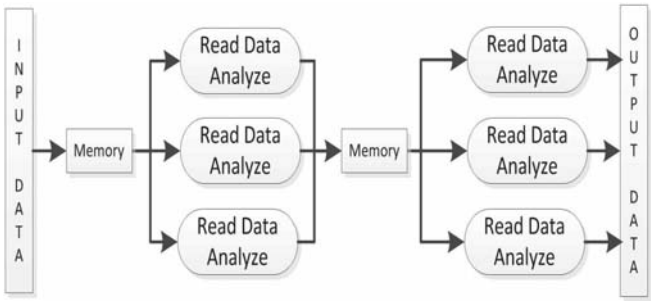

Figure 4: RDD Operation

Figure 4 shows in-memory method in our system. Frequent data has already fetched into memory as input of each phase. And the output after finishing task has stored into memory. This outstanding function makes application to run faster than another one with traditional MapReduce technique.

4. Conclusion

We propose a new design of Spark based big data analysis platform in order to enhance the performance of analyzing more accurate prediction of corps. Our system accumulates the data from many sources such as sensor, forecasting system, open government data system and process this data to indicate a desire result.

Through the case study of big data analyzing in agriculture platform to adjust the water discharge, we show the viability of our design. In the future work, we plan to deeply consider the algorithms for analyzing and processing data in agriculture to provide the solutions for increasing the productivity of the crop.

Acknowledgements

This work was carried out with the support of "Cooperative Research Program for Agriculture Science and Technology Development (Project No. PJ01182302)" Rural Development Administration, Republic of Korea. This work was supported by the National Research Foundation of Korea Grant funded by the korean Government(NRF-2014R1A1A1007734).

References

[1] Dad, Roopam, et al. "Analysis of Agriculture Commodity Prices using MapReduce Model." Analysis 4.4 (2015).

[2] Ramya, M. G., Chetan Balaji, and L. Girish. "Environment Change Prediction to Adapt Climate-Smart Agriculture Using Big Data Analytics." Environment4.5 (2015).

[3] Zaharia, Matei, et al. "Spark: Cluster Computing with Working Sets." HotCloud 10 (2010): 10-10.

[4] Dean, Jeffrey, and Sanjay Ghemawat. "MapReduce: simplified data processing on large clusters." Communications of the ACM 51.1 (2008): 107-113.

[5] Zaharia, Matei, et al. "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing." Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. USENIX Association, 2012.

[6] Shvachko, Konstantin, et al. "The hadoop distributed file system." Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on. IEEE, 2010.