하둡과 플럼 기반 농산품 가격 수집 시스템 설계와 구현

반퀴엣뉘엔, 신녹뉘엔, 둑티엡부, 김경백
전자컴퓨터공학부
전남대학교

Design and Implementation of a Crawling Agriculture Product Prices System based on Hadoop and Flume

Van-Quyet Nguyen, Sinh Ngoc Nguyen, Duc Tiep Vu, Kyungbaek Kim
Dept of Electronics and Computer Engineering,
Chonnam National University
e-mail : quyetict@utehy.edu.vn, sinhngoc.nguyen@gmail.com, ductiep91@gmail.com, kyungbaekkim@jnu.ac.kr

요 약

Massive data in agriculture are generated in every minute through multiple kinds of devices and services such as sensors, social networks, and agricultural web markets. It leads to the challenges of big data problem including data collection, data storage, and data analysis. Although some systems have been proposed for collecting data of agriculture products from the Internet, they are restricted either in the type of data (e.g., only historical data), the type of storage (e.g., only one local disk on a single computer), or the size of data they can handle. In this paper, we design and implement a system in distributed mode for crawling the big data of agriculture product prices from the Internet. Our system consists of two modules: one module is to crawl the product prices data in real-time manner based on Flume framework and another module is to crawl historical product prices data based on Hadoop framework. We used Jsoup API to extract the product information from the web, then stored the data in Hadoop Distributed File System (HDFS). Our system provides huge data which can be used for predicting the trend of agricultural products market price and delivering valuable and helpful market information for farmers and agribusinesses.

1. Introduction

Big data plays an important role in modern agriculture development. It has been a key driver of the progress made in precision agriculture, whereby farmers and agribusinesses are using the resources at their disposal in the most efficient way possible to get maximum yields. However, the massive data in agriculture are generated in every minute through multiple kinds of devices and services such as sensors, social networks, and agricultural web markets. The problem here is that how to collect these data from the many sources, and translate it into useful information to improve business processes and solve problems at scale and speed. Therefore, building a system for crawling the big data in agriculture is a particularly evident for agriculture big data analysis platform.

There are several techniques and tools for extracting content from the web as shown in [1][2]. The most popular web page analyzing tools is Jsoup [3] based on Java. Jie Wang et al. [4] has been designed and implemented a agricultural products big data platform based on Jsoup, in which the data was extracted from the URL of agricultural websites. However, their system only supported to deploy on a single computer that encounters of the challenges of big data problem.

Besides, extracting data from the web has a limitation of the number HTTP requests. It often takes a few seconds per request to obtain available resources. Meanwhile, all most agriculture product prices websites provide the data as the form of a table with many pages that need a lot of HTTP requests to crawl these data. For example, in Seobu market website [5], the agriculture product prices data are generated in around six hundred pages per day. Therein, the data are updated continuously in every a few minutes. Therefore, the crawling system needs to support of extracting data in parallel and real-time fashion to achieve the high performance as well as the availability of data.

In this paper, we present a design and

implementation of a system in distributed mode for crawling the big data of agriculture product prices from the web. Our system consists of two modules: one module is to crawl historical product prices data based on Hadoop framework [6] and another module is to crawl the product prices data in real-time manner based on Flume framework [7]. Hadoop has been the most popular framework that provides a parallel computation model MapReduce [8] and Hadoop Distributed File System (HDFS) [9] for solving the big data problems. While Flume is a framework for collecting, aggregating, and transferring huge amounts of data from multiple sources into central data store such as HDFS. Our implementation used Jsoup API to extract the product price data from agriculture market websites, then stored the data in (HDFS). Our system provides huge data that can be used for predicting of agricultural products market price trends and other issues to get valuable market information available to farmers and agribusinesses.

2. Background

In this section, we describe an overview of Hadoop and Flume framework that are used for big data crawling in this paper. We also present about Jsoup API that is use to extract the information from the web.

2.1 Hadoop/MapReduce

Apache Hadoop is a framework that allows for the distributed processing of large data sets across clusters of computers [6]. The current Hadoop version consists of four main components: (1) HDFS (Hadoop Distributed File System) that provides high-throughput access to application data, (2) YARN (Yet Another Resource Negotiator) that is a framework for job scheduling and cluster resource management, (3) MapReduce is a YARN-based system for parallel processing of huge datasets, and (4) Hadoop Utilities that provides common utilities to support the other Hadoop modules.

MapReduce is a programming model that supports to run programs in parallel on large distributed system. This model uses a map function that processes a key/value to generate a set of intermediate key/value pair and a reduce function that gathers all values with the same intermediate key to process and returns the results.

2.2 Apache Flume

Flume is a top-level project at the Apache Software Foundation. While it can function as a general-purpose event queue manager, in the context of Hadoop it is most often used as a log aggregator, collecting log data from many diverse sources and moving them to a data store such as HDFS.

Flume is made up of five main components as follows: (1) Event is a singular unit of data that is transported by Flume (typically a single log entry); (2) Source is the entity which data enters into Flume, and it is responsible to listen and consume events (data) coming from client (e.g. logs databases) and forwards them to one or more channels; (3) Sink is the entity that delivers the data to the destination; (4) channel is the conduit between the Source and the Sink; and (5) agent is a collection of sources, sinks and channels in which a Flume source receives events delivered to it from external source, then Flume channel is a passive store (in-memory, file, and so on) that keeps the event until it is consumed by a Flume sink.

2.3 Jsoup API

Jsoup is a Java library for working with real-world HTML. It provides a convenient API for extracting and manipulating data, using the best of DOM, CSS, and jquery-like methods. To extracting the data from the web, we first provide relative URL to Jsoup for making a connection and parsing a HTML document. After parsing a document, and finding some elements, we can get the data inside those elements through attributes and methods.

3. Design and Implementation of the Crawling Agriculture Product Prices System

Our goal is to build a system for collecting of agriculture product prices in both two cases: real-time data and historical data

3.1 Design of Real-Time Data Crawling Module

This module is designed to crawl the data generated by other systems or services. To simplify the crawling process, we use Flume's powerful streaming capabilities for efficiently collecting and moving large amounts of data into HDFS.

There are many kind of data sources can be handled by Flume such as logs, sensor data, and social media, in this paper, we use text files as the logs data source of product price. This types of data can be landed in Hadoop for future analysis using big data techniques such as MapReduce programing model.

Figure 1 shows our design for collecting data using Flume. In this model, we specify type of source is Exec that runs with a given command (e.g., tail -F [file]) and the output data is continuously produced. Here, the memory is used as a Flume Channel, in which the records (events) are stored in an in-memory. Note that,

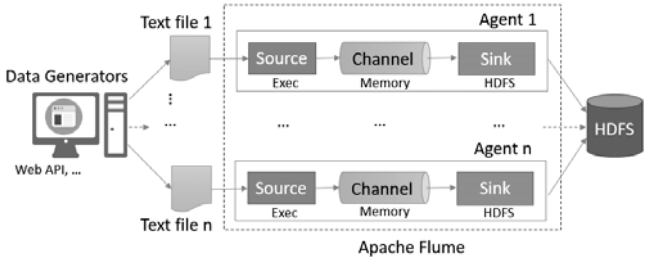

Figure 1. The model of real-time collecting data using Apache Flume

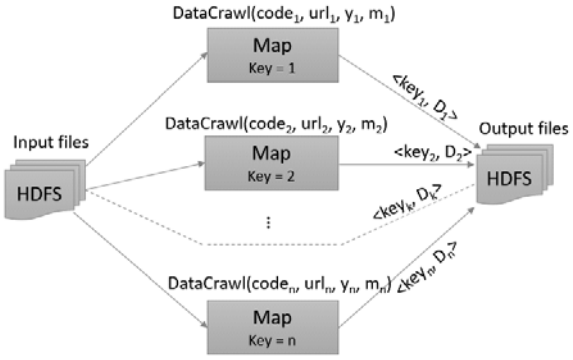

Figure 2. The model for historical data collecting using Hadoop

the number of events in channel is configurable in Flume. For purpose of big data analysis, we chose sink type is HDFS that writes events into the HDFS supporting text file format.

3.2 Design of Historical Data Crawling Module

The historical data that we mention in this paper related to agriculture product price that provide by agriculture web markets, e.i., Seobu Market [5]. For big data analysis, more data is better than more sophisticated modeling. Thus, we design this module for collecting the data from many web sites in the Internet. For each site, we crawl the data from a few years past to current. We observe that, in the most of Korea agriculture web markets, the data structure of displaying product price is similar and the amount of data per year is approximately two millions of records. Therefore, we use Hadoop framework with MapReduce programming model to build this module.

Event though MapReduce provides us two functions map and reduce to handle big data as we mentioned in previous section, for this proposed, we skip reduce function because it is unnecessary in this work that just collects data and saves it to HDFS. Moreover, skipping reduce takes advantage in saving time because of shuffle and sort phases are ignored. The model for historical data collecting using Hadoop is as shown in Figure 2.

In Figure 2, each map function calls DataCrawl procedure that receives the input arguments from the content of a input file in HDFS, then uses Jsoup API to connect and extract data from the web corresponds to input url, code, year, and month. It means that each map function is used to collect the product price data in a agriculture web market at a specific year and month (more details see in the next section).

3.3 Implementation

In this section, we present implementation of data crawling model using MapReduce and Jsoup API to extract data from the web as shown in Algorithm 1.

Algorithm 1 Procedure ***DataCrawl*** is used to crawl the data from the web

Require: A *code* and a *url* of the market site, *year* and *month* as the time we want extract the data
Ensure: The data as list of records that contains product price information

```
1: dayInMonth ← getNumOfDays(year, month);
2: lstRecord ← ∅ ;
3: for (day = 1; day ≤ dayInMonth; day + +) do
4:     boolean hasNext ← true;
5:     int page ← 1;
6:     while (hasNext) do
7:         Document d ← Jsoup.connect(url+"?p=page&y=
   year&m=month&d=day").get();
8:         Element table ← d.getElementById(tableid);
9:         Elements lstTr ← table.getElementByTag("tr");
10:        if (lstTr.size() > 0) then
11:            for (i = 0; i ≤ lstTr.size(); i + +) do
12:                Elements tds =
   lsTr.get(i).getElementsByTag("td");
13:                Record r ← new Record();
14:                for (int j = 0; j < tds.size(); j++) do
15:                    r ← r.add(tds.get(j).text());
16:                end for
17:                lstRecord.add(r);
18:            end for
19:            page ← page + 1;
20:        else hasNext ← false;
21:        end if
22:    end while
23: end for
24: return lstRecord;
```

Procedure *DataCrawl* is implemented to crawl the data of from the web. Each mapper will call this procedure to collect the data for one month corresponding its input data from the input file. This procedure using Jsoup API to connect to a web page (line 7), then extract the data based on HTML tags and attributes (lines 8-12). Finally, it constructs a list of records that contains agriculture product price information.

4. Experimental Evaluation

Our experiments were run on the distributed system, in which, Hadoop environment is deployed on five

machines: one machine for master node, and four others for compute nodes. Each compute node has 4 CPUs and 8GB of RAM. All algorithms are implemented in Java.

For real-time data collecting, we deployed Flume on master node, in which, one agent is configured to collect the data that are generated by Seobu Market. We implemented an application to simulate a service that provides product price data in real-time manner, the data are collected from the Seobu Market site and generated to the text file in every two minutes. The Flume agent monitor this file and transfer the data to HDFS in our system as shown in Figure 3.

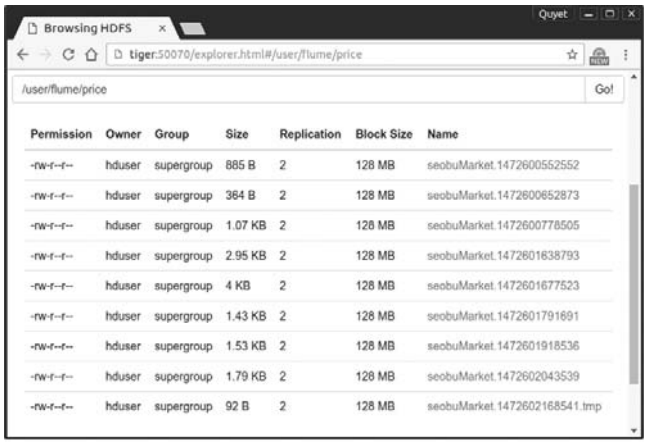

Figure 3. The data stored in HDFS collecting by Flume

Table 1. The result of data collecting from the web using Hadoop/MapReduce

Data source	Number of Items	Size (MB)	Time (minutes)
Seobu Market site	760.801	59	788
Eomgung Makert site	677.866	58	
Total	1.438.667	117	

For historical data collecting, we experimented with crawling data that generated by two sites Seo Market [5] and Eomgung Market [10] in 6 months from January 2016 to June 2016. The result is as shown in Table 1. The total time for collecting these data is 788 minutes.

5. Conclusion

We presented a design and implementation of a system for crawling the big data of agriculture product prices from the Internet. In our system, Apache Flume is used to crawl the real-time data, which simplifies collecting from many data sources and transfers data to HDFS. While Hadoop framework is used to collect the historical data as big data, in which only *map* function is utilized in combination with Jsoup API to speed aggregate data. In the future work, we will focus on development of big data platform for analysis the data which are collected by our proposed in this paper.

Acknowledgements

This work was supported by the National Research Foundation of Korea Grant funded by the korean Government(NRF-2014R1A1A1007734). This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2016-R2718-16-0011) supervised by the IITP(Institute for Information & communications Technology Promotion).

References

[1] Ferrara, Emilio, et al. "Web data extraction, applications and techniques: a survey." Knowledge-based systems 70 (2014): 301-323.

[2] Geng, Hua, Qiang Gao, and Jingui Pan. "Extracting content for news web pages based on DOM." IJCSNS International Journal of Computer Science and Network Security 7.2 (2007): 124-129.

[3] Jonathan Hedley. “Jsoup: Java HTML Parser”, https://jsoup.org/

[4] Wang, Jie, et al. "The crawling and analysis of agricultural products big data based on Jsoup." Fuzzy Systems and Knowledge Discovery (FSKD), 2015 12th International Conference on. IEEE, 2015.

[5] Seobu Market, “http://seobu-market.gwangju.go.kr/”

[6] Hadoop, Apache. "Hadoop." http://hadoop.apache.org (2009).

[7] Apache Flume, “https://flume.apache.org/.”

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

[9] Borthakur, Dhruba. "HDFS architecture guide." HADOOP APACHE PROJECT http://hadoop. apache. org/common/docs/current/hdfs design. pdf (2008): 39.

[10] Eomgung Market, “http://eomgung-market.busan.kr/egmarket_busan_go_kr/realtime_iframe.asp”