http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (17): 4184-4187, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.4184.4187

Study for Efficient Integration and Sharing Architecture for Agriculture Data Resources

[1]Fang Wei, [2]Si Hai-Ping and [2]Wei Xiu-Ran
[1]Institute of Crop Science, Chinese Academy of Agricultural Sciences, Beijing, 100081, China
[2]Colleges of Information and Management Science, Henan Agricultural University, Henan 450008, China

Abstract: Different data integration and sharing architecture applied for different data management scenarios, therefore, universal data integration and sharing architecture structure is impossible. Furthermore, each data integration structure has different resource consumption because of their unique execution complexity. Agriculture data resources need architecture to select appropriate data storage structure and to adapt it with changing data management requirements. In this paper we propose efficient integration and sharing architecture based on Hadoop for agriculture data resources that supports customization and adaptation with changing data management needs. Data integration and sharing architecture are provided based on different contents, themes and users. The study will provide an efficient usage for agriculture big data.

Key words: Agriculture, data integration, data query, hadoop

INTRODUCTION

In recent years, with the increase of the amount of data, the traditional RDBMS has been cannot meet the demand for large data analysis especially in the field of agriculture resource data. Although many manufacturers have claimed to have the characteristics of a database column, the vast majority not really equipped to handle large data capacity. The integration and sharing of agriculture resources data plays an important role for government to master information of crop species, distribution and quantity and to optimize and utilize efficiently crop agriculture resources (Cao and Wei, 2008).

Agriculture resources data are integrated through resources investigation projects especially in the field of crop germplasm materials. Many projects related to crops germplasm resources investigation have been organized by Chinese government since the 1950s and a great quantity underlying data of germplasm resources were accumulated and made remarkable achievements in resource investigation works (Evans-Pughe, 2006). However, there have some outstanding issues in the process of crop germplasm resources data storage and sharing effectively, such as:

- The problems of scatted data information, data redundancy and data expression inconformity are caused by the different database technology that used in agriculture data projects. The data existing in different repositories are independent and incomplete which lead to information islands and hinder the development of crops germplasm resources study
- The lack of effective germplasm resources data integration and sharing mechanism, which makes data resources information decentralized and information sharing incomplete (Gao *et al*., 2011)

In this study, Hadoop-virtual-panel based big data architecture is introduced firstly and then with the architecture blueprint for agriculture data resource (germplasm resources data) solution is proposed in order to share the data better.

HADOOP TECHNICAL CHARACTERISTICS AND APPLICATION

Hadoop is an open-source software framework that supports data-intensive distributed applications, licensed under the Apache v2 license. It enables applications to work with thousands of computational independent computers and petabytes of data. Hadoop was derived from Google's MapReduce and Google File System (GFS) papers. The entire Apache Hadoop "platform" is now commonly considered to consist of the Hadoop kernel, MapReduce and HDFS, as well as a number of related projects-including Apache Hive, Apache Hbase and others.

Hadoop is a new data management system that brings together the traditional world of unstructured or non-relational data storage with the processing power of compute grids. While it borrows heavily from the design patterns of MPP database, Hadoop differentiates in a few critical areas. First, it is designed for low cost of byte economics. Hadoop runs on just about any hardware and

Corresponding Author: Fang Wei, Institute of Crop Science, Chinese Academy of Agricultural Sciences, Beijing, 100081, China

is extremely forgiving to heterogeneous configurations or sporadic failures. Second, Hadoop is incredibly scalable. In its first version Hadoop was able to scale to several thousand nodes and in the current version tests are ongoing to reach over ten thousand nodes. Third, Hadoop is extremely flexible with regards to the type of data that can be stored and processed. Hadoop can accept any kind of data in any format and has a rich set of APIs for reading and writing any format of data.

Today Hadoop is used to tackle several different challenges across a variety of industries. The most common application is to speed up ETL. For example, in financial services, instead of pulling data from many sources systems for every transformation, source systems push data to HDFS where ETL engines process the data and store the results. These ETL flows can be written in Pig or Hive or using commercial solutions such as Informatica, Pentaho, Pervasive and others. The results can be further analyzed in Hadoop or published to traditional reporting and analytics tools. Using Hadoop to store and process structured data has be shown to reduce costs by a factor of ten and speed up processing by four times. Beyond traditional ETL, Hadoop is also used to gather telemetry data from both internal system such as application and web logs as well as remote systems on the network and globally. This detailed sensor data gives companies such as telecommunications and mobile carriers the ability to model and predict quality problems in their networks and devices and to take action proactively. Hadoop has also become a centralized data hub for everything from experimental analytics on cross organizational data sets to a platform for predictive modeling, recommendation engines and fraud, risk and intrusion modeling. These applications are deployed widely in production today and offer just a glimpse of what is possible with all of organizations data is collected together and made available to help drive the business.

HADOOP-BASED AGRICULTURAL DATA MANAGEMENT ARCHITECTURE

In today's world, data is money. However, if those companies cannot use data they have and they're not analyzing it to find those hidden gems, the data is worthless. Currently, only in the field of crop germplasm resources information data, China has built over 200 kinds of crops (under 78 families, 256 genera, 810 kinds or subspecies), 410,000 pieces of germplasm information, 24 million data item value, 4000 megabytes of Chinese crop information resource database that including 1.3 million records. Efficient data management model, storage and sharing solution are the basis for the effective use of these big data resources. Although there is no standard definition of the term of big data, Hadoop is the standard for processing big data in practical applications such as IBM, Oracle, SAP, or even Microsoft.

Hadoop data management model: One of the most challenging tasks when getting started with Hadoop and building a big data solution is figuring out how to take the tools you have and put them together. The Hadoop ecosystem encompasses about a dozen different open-source projects. Combined with agricultural data characteristics, data management model for agriculture data resources was designed three pieces: data ingestion, data storage and data analysis. The flow of these systems is shown as Fig. 1.

Data ingestion model is the connection between the data source and the storage location where the data will reside, while at rest. A data analysis system is used to process the data and produce actionable insights. Translating into a relational architecture, we can replace the generic terms as Fig. 2.

The basic architecture of ingestion, storage and processing that can be mapped onto the Hadoop ecosystem. Canonical Hadoop architecture is shown as Fig. 3.

By pulling in other ecosystem projects, much more complicated systems can be built. However, this is a very common Hadoop architecture and can be used to bootstrap a foray into the field of agriculture related big data.

In this study, SQL queries are used to solve data management. In the Hadoop ecosystem, the Hive project provides a query interface which can be used to query

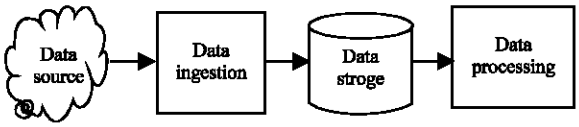

Fig. 1: Data management models and data flows in Hadoop

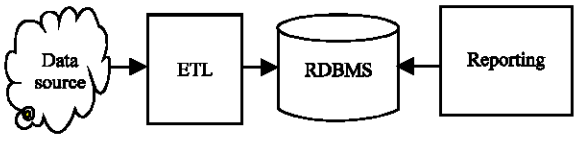

Fig. 2: Core of many relational data management

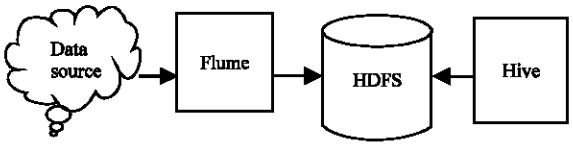

Fig. 3: Canonical hadoop architecture

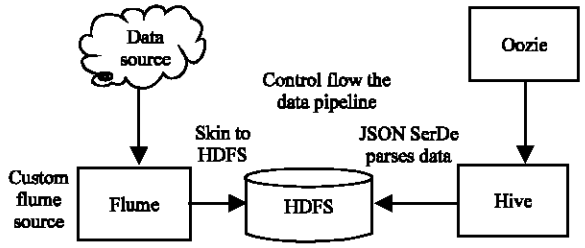

Fig. 4: Control flow of the data pipeline in the hadoop model

agriculture data that resides in HDFS. The query languagelooks very similar to SQL, but allows us to easily model complex types, so we can easily query the type of data we have. The diagram above shows a high-level view of how some of the CDH components can be pieced together to build the data pipeline in order to get agriculture data into Hive.

Gathering and querying complex data with hive and flume: It will be much simpler to use components within CDH to automatically move the files from the API to HDFS, without our manual intervention. Apache Flume is a data ingestion system that is configured by defining endpoints in a data flowcalled sources and sinks. In Flume, each individual piece of data (agri-data, in our case) is called an event; sources produce events and send the events through a channel, which connects the source to the sink. The sink then writes the events out to a predefined location. Flume supports some standard data sources, such as syslog or net cat. For this use case, we'll need to design accustom that accesses the Agriculture Streaming API and sends the agri-data through a channel to a sink that writes to HDFS files.

Before we can query the data, we need to ensure that the Hive table can properly interpret the JSON data. By default, Hive expects that input files use a delimited row format, but our agriculture data is in a JSON format, which will not work with the defaults. This is actually one of Hive's biggest strengths. Hive allows us to flexibly define and redefine, how the data is represented on disk. The schema is only really enforced when we read the data and we can use the Hive SerDeinterface to specify how to interpret what we've loaded. SerDe stands for Serializer and Deserializer, which are interfaces that tell Hive how it should translate the data into something that Hive can process. In particular, the Deserializer interface is used when we read data off of disk and converts the data into objects that Hive knows how to manipulate. We can write a custom SerDe that reads the JSON data in and translates the objects for Hive. Once that's put into place, we can start querying. The JSON SerDe code can be found here. The SerDe will take an agri-data in JSON form and translate the JSON entities into results in:

```
queryable columns:
SELECT created_at, entities, text, user
FROM agricultures
WHERE user.screen_name='ParvezJugon'
 AND agri_data _status.user.screen_name='ScottOstby';
```

An end-to-end system is managed to put together which gathers data from the Agriculture Streaming API, sends the agri_data to files on HDFS through Flume and uses Oozie to periodically load the files into Hive, where we can query the raw JSON data, through the use of a Hive SerDe.

The collected data was about half a GB of JSON data, like the agri above. The data has some structure, but certain fields may or may not exist. The agri_data_status field, for example, will only be present if the agri was a agri_data. Additionally, some of the fields may be arbitrarily complex. The hashtags field is an array of all the hashtags present in the agri_data, but most RDBMS's do not support arrays as a column type. This semi-structured quality of the data makes the data very difficult to query in a traditional RDBMS. Hive can handle this data much more gracefully. The query below will find usernames and the number from data we have:

```
SELECT
 t. agri_data _screen_name,
 sum (agri_data AS total_agri_data
 count (*) AS agri_count
FROM (SELECT
 agri_data _status.user.screen_name as agri_data_screen_name,
 agri_data _status.text,
 max(agri_data _count) as reagri_data
 FROM agri_data
 GROUP BY agri_data _status.user.screen_name,
 agri_data _status.text) t
GROUP BY t. agri_data _screen_name
ORDER BY total_ agri_data DESC
LIMIT 10;
```

From these results, we can see whose agri_data are getting heard by the widest audience and also determine whether these people are communicating on a regular basis or not. We can use this information to more carefully target our messaging, in order to get them talking about our products, which, in turn, will get other people talking about our products.

In this Hadoop_based architecture, we take some of the components of CDH and combine them to create an end-to-end data management system. This same architecture could be used for a variety of applications designed to look at Agriculture data, such as identifying spam accounts, or identifying clusters of keywords. Taking the system even further, the more general

.architecture can be used across numerous applications. By plugging in different Flume sources and Hive SerDes, this application can be customized for many other applications, like analyzing web logs, to give an example. Grab the code and give it a shot yourself.

DATA INTEGRATION FOR AGRICULTURE DATA BASED HADOOP ARCHITECTURE

Data integration is the key to efficiently utilize and share agriculture data resources (An *et al*., 2013). In this section, data integration method based on hahoop architecture is designed based on the analysis among the traditional data integration solutions and technical features.

ESB_based data integration method: ESB is used as a service container implementation approach and XML service interface is used to access various heterogeneous data sources. Extensible style sheet Language transformations (XSLT) is used for data mapping between heterogeneous data. ESB provides functions of heterogeneous data extraction, data conversion, data exchange standards, patterns and exchange formats. By calling the different services, the corresponding functional block are executed. When external data need to be integrated, the first step is to connect to the database system and select different elements needed from the data schema and bind them to component fields within an application. The different elements that comprise the data architecture are identified. Data binding is represented by the arrows between the components. As shown in the Fig. 3, data bindings need to be set up between properties of UI controls and properties of a data set component; and between a resolver and a connector component.

In the Fig. 5, the binding layer provides a way to map data elements to properties of data components, the connectivity layer gives connector components that let user connect to an external data source to send and receive data in need, the management layer provides components that enable intelligent supervision of common data operations, such as editing, sorting, filtering, aggregation and translation of changes and the resolution layer shows resolver components that can translate changed data into a format that is consumable by an external data source.

Data conversion of heterogeneous database system into the uniform data format is the main process of data integration. When user requests data investigation or exchange service, data extract service serve for the function of data extract from heterogeneous database system, the data converted according to data investigation standards return to user. The specific data extract and exchange process model is shown in Fig. 5.

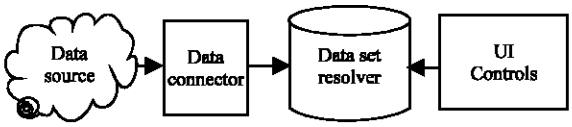

Fig. 5: Data investigation data flow and different layer offer various functions

XML data service interface is the data interactive hub between data source and data mapping and conversion service. Data conversion service servers for data conversion of data element name, date type, data value etc. XSLT processor convert the origin data into target data according to the mapping rules of extract data. The specific mapping and conversion steps are as follows (Salem *et al*., 2013).

XSLT (XSL Transformations) (Chen *et al*., 2011) is a declarative, XML-based language used for the transformation of XML documents into other XML documents. XSLT processors is an XML template engine (or XML template processor) and a specialized template processor for XML input and output, working in an XML template system context.

CONCLUSION

In this study, the Hadoop based agriculture data integration architecture data are established and then unified heterogeneous system data exchange and integration scheme is designed based on ESB integration method. With the multi-level and perspectives, scientific data show ways, decision support services can be provided for resource management and scientific researches.

REFERENCES

An, J.F., F.M. Lu, H. Duan and Q.T. Zeng, 2013. A middleware for metadata management oriented distributed data sharing. Adv. Mater. Res., 756-759: 940-945.

Cao, Y.S. and F. Wei, 2008. Establishment and application of national crop germplasm resources infrastructure in China. Biodiversity Sci., 18: 454-460.

Chen, G., Y.G. Wu, J. Liu and G.W.M. Yang, 2011. Optimization of sub-query processing in distributed data integration systems. J. Network Comput. Appl., 34: 1035-1042.

Evans-Pughe, C., 2006. Share and share alike [data sharing]. Eng. Technol., 1: 30-35.

Gao, F., J.S. He and S.N. Ma, 2011. Privacy preserving in data sharing applications. J. Southeast Univ. (Natural Sci. Edn.), 41: 233-236.

Salem, R., O. Boussaid and J. Darmont, 2013. Active XML-based Web data integration. Inform. Syst. Frontiers, 15: 371-398.