DESIGN AND CONSTRUCTION OF THE DATA WAREHOUSE BASED ON HADOOP ECOSYSTEM AT HLS-II*

Y. Song, X. Chen, C. Li, K. Xuan, J. Wang, G. Liu†, NSRL, USTC, Hefei, Anhui 230029, China

Abstract

A data warehouse based on Hadoop ecosystem is designed and constructed for Hefei Light Source II (HLS-II). The ETL program based on Spark migrates data to HDFS from RDB Channel Archiver and the EPICS Archiver Appliance continuously and store them in Parquet format. The distributed data analysis engine based on Impala greatly improves the performance of data retrieval and reduces the response time of queries. In this paper, we will describe our efforts and experience to use various open sources software and tools to effectively manage the big data. We will also report the plans on this data warehouse in the future.

INTRODUCTION

The Hefei Light Source (HLS) at National Synchrotron Radiation Laboratory is the first dedicated synchrotron radiation facility in China, which provides radiations from IR to vacuum ultraviole (VUV) for various user programs. HLS was upgraded from 2010 to 2015 to improve its performance. The upgraded light source is renamed as Hefei Light Source II (HLS-II) [1]. It is comprised of an 800MeV linac, an 800MeV storage ring and a transport line connecting the linac and the storage ring. The HLS-II was fully open to users in January 2016. The control system of HLS-II is a distributed system based on Experimental Physics and Industrial Control System (EPICS) [2].

With the continuous upgrade, a series of data archiving tools has been used in HLS-II supported by the EPICS community, such as Channel Archiver, RDB Channel Archiver and the EPICS Archiver Appliance. As more and more data have been stored, the exciting data archiving tools can not meet our requirements of data query and processing. In order to deal with these problems, we are designing and constructing the data warehouse based on Apache Hadoop ecosystem. In addition, our laboratory is conducting pre-research on Hefei Advanced Light Source (HALS) project [3]. Compared to HLS-II, HALS is a larger and more complex synchrotron facility. The work described in this paper also provides technical verification for the future construction of HALS.

This paper will describe our efforts and experience to use various open sources software and tools to manage the big data effectively. We will also report the plans on the data warehouse in the future.

* Work supported by National Natural Science Foundation of China (No.11375186)

† Corresponding author, gfliu@ustc.edu.cn

HADOOP ECOSYSTEM AND DATA WAREHOUSE

Hadoop Ecosystem

Apache Hadoop is a collection of open-source software utilities that facilitate using a network of many computers to slove problem involving massive amounts of data and computation. It provides a software framework for distributed storage and processing of big data [4]. As shown in Figure 1, Hadoop core components include HDFS, Yarn and MapReduce, as well as some open source projects based on Hadoop, including Spark and Impala, which provide necessary support for the whole life cycle of big data processing. Our warehouse system mainly uses the following components:

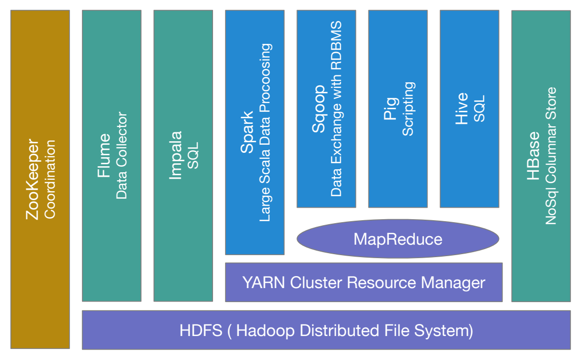


Figure 1: Hadoop Ecosystem.

HDFS HDFS is a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster.

Spark Spark has the advantages of Hadoop MapReduce, but unlike MapReduce, its intermediate output can be stored in memory, instead of the HDFS. Tests have shown that Spark is more efficient than MapReduce for the same operation. Spark is widely used in the field of ETL (Extract-Transform-Load) and machine learning [5].

Impala Impala is an open source new query system that can query PB-level big data stored in Hadoop's HDFS and HBase. Compared to MapReduce, Impala provides data query function more efficient and convenient.

Sqoop Sqoop is command-line interface application for transferring data between relational databases and Hadoop.

Hadoop Configuration

Because the Hadoop ecosystem contains a variety of software, it is difficult to handle version compatibility and configuration files, and is also inconvenient to manage and monitor the state of the entire Hadoop cluster. Therefore, Cloudera Distribution Hadoop (CDH) is used as the tool for cluster construction. CDH is an open-source Apache Hadoop distribution provided by Cloudera Inc. and is the most complete, tested, and widely deployed distribution of Apache Hadoop [6].

We built the test system with 5 nodes on the VMware vSphere virtualization platform based on CDH. Figure 2 shows the management of web pages provides by CDH. Through this web page, the cluster administrators can simply complete the management work, such as adding or deleting services, adding computer nodes, and viewing the status of the cluster, without paying attention to the details of the configuration file.

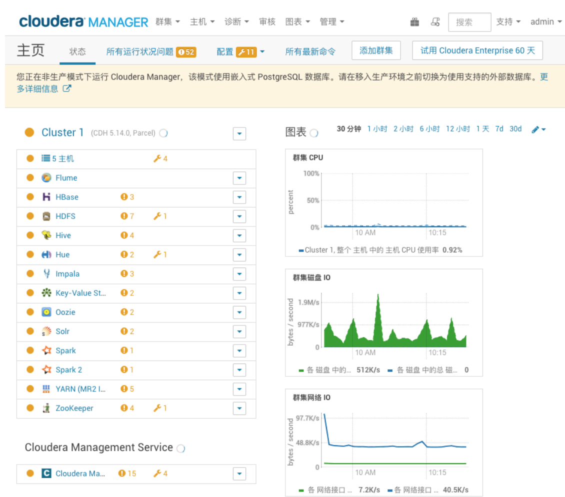


Figure 2: Web pages of cloudera manager provided by CDH.

SOFTWARE ARCHITECTURE

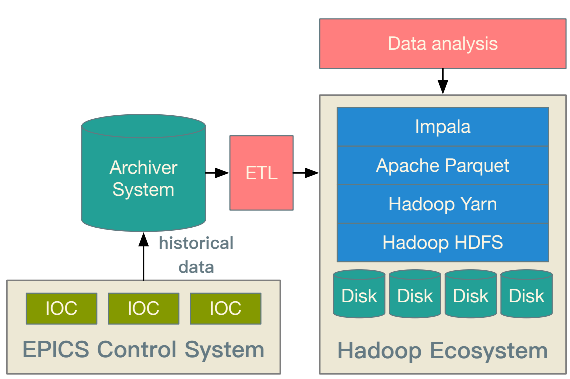


Figure 3: Software structure of the entire data warehouse.

Figure 3 shows the software structure of the entire data warehouse. The whole system is mainly divided into three parts: the control system based on EPICS, the big data platform based on Hadoop Ecosystem and the archiver & ETL program. The Records in EPICS IOC hold the real-time state of the controlled devices. Instead of redeveloping data acquisition and archiving software, we are still using data archiving tools already available in the EPICS community, such as RDB Channel Archiver and Archiver Appliance. These data archiving tools collect historical data and store them in its own storage system in a custom format. ETL program is developed to migrate data from archiving tools to HDFS and store them in Parquet format. Impala provides a convenient data analysis interface for scientists and engineers.

IMPLEMENTATION OF DATA WAREHOUSE

Archiving Tools at HLS-II

At HLS-II, there are two data archiving systems running at the same time: Archiver Appliance and RDB Channel Archiver respectively. The Archiver Appliance collects data from IOC and stores them on the hard disk in Protocal Buffer (PB) format [7]. The Archive Engine of RDB Channel Archiver is also used for collecting data from IOCs of control system and storing them in Oracle database [8]. Both types of data from different data sources need to be migrated to HDFS.

ETL Program

Data migration refers to transferring the original data stored in the archive system to the HDFS file system, storing them in Parquet format, and performing data cleaning and calculation as needed during the transfer process. Usually this process is called Extract-Transform-Load (ETL). The ETL program is the core part of connecting archiving tools and the Hadoop Ecosystem.

The whole process of data migration is shown in Figure 4. Spark's APIs are easy to use and have good support for accessing various databases. Therefore, Spark is very efficient in big data processing and is used to implement the ETL process. The process of data migration is divided into two steps.

In the first step, the original data in the archiving tools are exported to the distributed file system HDFS. For the Archiver Appliance, it provides an HTTP interface for querying data, so a Python program is developed to get data from the Archiver Appliance via HTTP in CSV format and upload the CSV files to the HDFS file system. This process is performed periodically in batch mode. For the RDB Channel Archiver, Sqoop is used to migrate the data from Oracle database directly to the HDFS system.

In the second step, The ETL program written based on Spark is triggered after the data are uploaded. It completes the ETL process, and finally export the Parquet format file into the data warehouse with a predefined data model. In this process, the ETL program also performs a series of data cleaning tasks, including the elimination of abnormal data points and the alignment of sampling time.

In addition, the workflow management program is written to ensure the stable operation of the ETL process, including the workflow task scheduling, metadata acquisition, logging

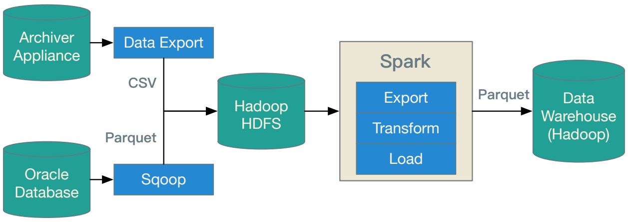
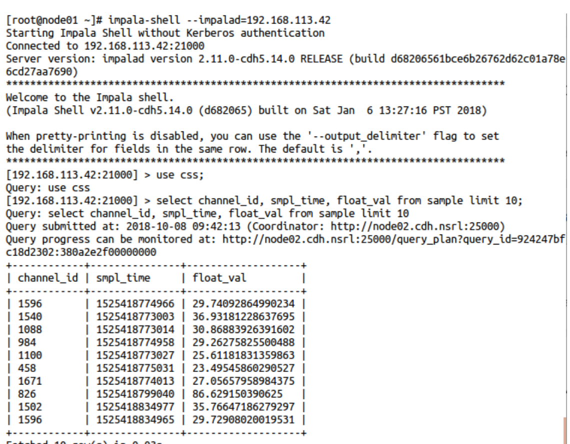


Figure 4: Data migration process.

and notification function. The open source community has released an library Apache Airflow for building complex batch workflows. This system use Python language to develop tthe workflow management program based on Apache Airflow to control the whole workflow.

Data Analysis

After the data are stored in HDFS , the users can conveniently use Impala to perform data query and analysis in the Parquet format. Figure 5 shows the screenshot of the query data through the Impala command line. The user queries the first 10 records of the sample table in the css database through SQL statement and the result is returned within 0.03s.



```
[root@node01 ~]# impala-shell --impalad=192.168.113.42
Starting Impala Shell without Kerberos authentication
Connected to 192.168.113.42:21000
Server version: impalad version 2.11.0-cdh5.14.0 RELEASE (build d68206561bce6b26762d62c01a78e
6cd27aa7690)
***********************************************************************************
Welcome to the Impala shell.
(Impala Shell v2.11.0-cdh5.14.0 (d682065) built on Sat Jan  6 13:27:16 PST 2018)

When pretty-printing is disabled, you can use the '--output_delimiter' flag to set
the delimiter for fields in the same row. The default is ','.
***********************************************************************************
[192.168.113.42:21000] > use css;
Query: use css
[192.168.113.42:21000] > select channel_id, smpl_time, float_val from sample limit 10;
Query: select channel_id, smpl_time, float_val from sample limit 10
Query submitted at: 2018-10-08 09:42:13 (Coordinator: http://node02.cdh.nsrl:25000)
Query progress can be monitored at: http://node02.cdh.nsrl:25000/query_plan?query_id=924247bf
c18d2302:380a2e2f00000000
+------------+---------------+------------------+
| channel_id | smpl_time     | float_val        |
+------------+---------------+------------------+
| 1596       | 1525418774966 | 29.74092864990234 |
| 1540       | 1525418773003 | 36.93181228637695 |
| 1088       | 1525418773014 | 30.86883926391602 |
| 984        | 1525418774958 | 29.26275825500488 |
| 1100       | 1525418773027 | 25.61181831359863 |
| 458        | 1525418775031 | 23.49545860290527 |
| 1671       | 1525418774013 | 27.05657958984375 |
| 826        | 1525418799040 | 86.629150390625  |
| 1502       | 1525418834977 | 35.76647186279297 |
| 1596       | 1525418834965 | 29.72908020019531 |
+------------+---------------+------------------+
Fetched 10 row(s) in 0.03s
```

Figure 5: Query data through the Impala command line.

CURRENT PROGRESS AND NEXT PLAN

At present, we have built the test system on the VMware vSphere virtualization platform and developed the ETL program. In the next step, we will deploy the program to the server in the production environment, and carry out performance testing.

In the future, we plan to complete the following tasks:

- Provide data reporting functions. We will develop a series of programs for calculating data reports based on Impala, such as operating status statistics and integral current calculations. With the efficient computing power of the Hadoop cluster and the optimization of the SQL statement, our goal is to control the response time of the report calculation task within a few seconds.
- Provide more convenient data query entry. Currently, users can only query data by executing SQL statements on the Impala command line. In the future, we plan to provide a web-based data query interface and develop the web applications to facilitate scientists and engineers to query data stored in the data warehouse.
- Carry out research on data mining and Artificial intelligence (AI) algorithm applications. As Spark's machine learning library, MLlib consists of some common learning algorithms and tools, including classification, regression, recommendation, etc.. We plan to use the spark MLlib to mine the characteristics of the data stored in the data warehouse, carry out research on AI algorithm, and improve the intelligence of the entire control system.

CONCLUSION

A data warehouse based on Hadoop ecosystem is designed and constructed for Hefei Light Source II (HLS-II). The ETL program based on Spark and Sqoop migrates data to HDFS from RDB Channel Archiver and the EPICS Archiver Appliance continuously and store them in Parquet format. Such a technology plan combines various open sources tools under Hadoop framework and the archiving tools in EPICS community. It provides a sulotion to solve the big data problem for the large-scale scientific facility.

REFERENCES

[1] J. Y. Li, G. Huang, W. Wei, *et al.*, "Operational Status of HLS-II", in *Proc. IPAC'16*, Busan, Korea, May 2016, pp. 4155–4158. doi:10.18429/JACoW-IPAC2016-THPOY028

[2] C. Li, J. Wang, K. Xuan, *et al.*, "Event-driven timing system based on MRF cPCI hardware for HLS-II", *Nuclear Science and Techniques*, vol. 26, pp. 060401, 2015. doi:10.13538/j.1001-8042/nst.26.060401

[3] L. Wang, Z. Bai, N. Hu, *et al.*, "Hefei Advanced Light Source: A Future Soft X-Ray Diffraction-Limited Storage Ring at NSRL", in *Proc. IPAC'18*, Vancouver, BC, Canada, Apr./May 2018, pp. 4598–4600. doi:10.18429/JACoW-IPAC2018-THPMK120

[4] Hadoop home page, http://hadoop.apache.org/.

[5] Spark home page, https://spark.apache.org/.

[6] Cloudera CDH home page, https://www.cloudera.com/products/open-source/apache-hadoop/key-cdh-components.html

[7] Y. F. Song, C. Li, K. Xuan, *et al.*, "Automatic data archiving and visualization at HLS-II", *Nuclear Science and Techniques*, vol. 29(9), pp. 129, 2018. doi:10.1007/s41365-018-0461-6

[8] J. Qiang, G. Liu and Jie Qiang, "The data archive system and preliminary applications of Hefei light source based on Oracle", *Nuclear Techniques*, vol. 36, pp. 1–5, 2013.