



Big data compression processing and verification based on Hive for smart substation

Zhijian QU, Ge CHEN (✉)



Abstract The capacity and the scale of smart substation are expanding constantly, with the characteristics of information digitization and automation, leading to a quantitative trend of data. Aiming at the existing processing shortages in the big data processing, the query and analysis of smart substation, a data compression processing method is proposed for analyzing smart substation and Hive. Experimental results show that the compression ratio and query time of RCFile storage format are better than those of TextFile and SequenceFile. The query efficiency is improved for data compressed by Deflate, Gzip and Lzo compression formats. The results verify the correctness of adjacent speedup defined as the index of cluster efficiency. Results also prove that the method has a significant theoretical and practical value for big data processing of smart substation.

Keywords Hive, Smart substation, Lossless compression

1 Introduction

Smart substation acts as an important foundation and pillar of strong smart grid, which has characteristics of information digitization, networking communication platform and information sharing standardization [1, 2], and

completes some functions of system monitoring, controlling and protection, etc. A trend of huge amount of data with characteristics of large scale, complex types, and wide area distribution produced by smart substation makes the traditional relational database more and more difficult to adapt to the requirements of large scale data processing from power enterprises [3, 4]. Presently, big data storage and processing are mostly based on large scale servers with relational database management systems, which need huge investment and have a shortage of low utilization ratio and poor scalability. Therefore, the design of power data center using traditional system is far from the requirements of big data storage, analysis and processing. Thus, how to process and analyze massive data produced by smart substation effectively becomes a great challenge. It is urgent to research on effective storage technology for big data [5].

Data warehouse using Hive is an infrastructure built on top of Hadoop cloud computing framework, with good scalability and fault tolerance [6, 7], which can integrate with lossless compression algorithms, such as BZip2, Deflate, Gzip and Lzo. Its underlying operations can be transformed into MapReduce parallel tasks [8–10], and its application interface uses HQL language, which provides the ability of quick development. Hive is different from the relational database. It has no special data formats, but it has three kinds of storage formats, including TextFile, SequenceFile and RCFile. Hive is designed towards the query and analysis of massive data, which can be used to build a data warehouse for processing big data of smart substation.

Considering the characteristics of big data of smart substation and Hive, a data compression processing method based on Hive is proposed to solve the mentioned problems. Experimental results show that it has a significant theoretical and practical value for processing big data of smart substation.

CrossCheck date: 3 February 2015

Received: 27 June 2014 / Accepted: 25 December 2014 / Published online: 8 August 2015

© The Author(s) 2015. This article is published with open access at Springerlink.com

Z. QU, G. CHEN, School of Electrical & Electronic Engineering, East China Jiaotong University, Nanchang 330013, China

(✉) e-mail: chenge880601@163.com



2 Hive and storage formats

2.1 Processing flow of Hive

Hive is introduced firstly in order to study the smart substation based on Hive. Hive is an open source data warehouse project with an extension based on Hadoop cloud computing platform published by apache software foundation, thus it supports a wide of data types, various kinds of structured and unstructured data with complex and heterogeneous storage formats [11]. Combined with the traditional structured query syntax, Hive itself defines Hive query language (HQL), through the analysis of HQL syntax by the driver. HQL tasks are transformed into MapReduce parallel tasks, thus they can take full advantage of the high performance and scalability of the cloud computing and realize complex processing for the big data. MapReduce parallel processing flow of Hive is shown in Fig. 1.

Hadoop distributed file system (HDFS) is the file management foundation of reading or writing data based on Hive. The unified management of distributed data is carried out by Namenode, Datanodes and client applications. Data processing flow based on Hive is shown in Fig. 2.

Namenode acts as the management master of HDFS. Datanodes are responsible for the data blocks storage in HDFS, and reporting their status to Namenode with the heartbeat response periodically. If the Namenode does not obtain heartbeats from a Datanode, it will modify the configuration for Datanodes' directory, and determine whether the Datanode appears fault. If so, it will not get the data operation request, then the client will read the same blocks from another Datanode, and the client applications access to the data in a streaming way in the HDFS system. Hive provides the applications with command line interface (CLI), client interface (Client) and web user interface (WUI). Attributes such as table name, column, and partition of Hive are stored in metadata database.

Reading request of data is sent to Namenode by the client process, and then the client reads the data in an FSInput streaming way, according to the distribution of data blocks stored in different Datanodes. Writing request of data is sent to Namenode by the client process, and then the client writes data in an FSOutput streaming way to different Datanodes specified by Namenode.

2.2 Compression storage formats of Hive

- 1) TextFile acts as the default storage format, which can be combined with the different lossless compression algorithms, as well as be detected and decompressed automatically by Hive.
- 2) SequenceFile is a kind of binary file which the Hadoop provides, the data will be serialized in files in the form of

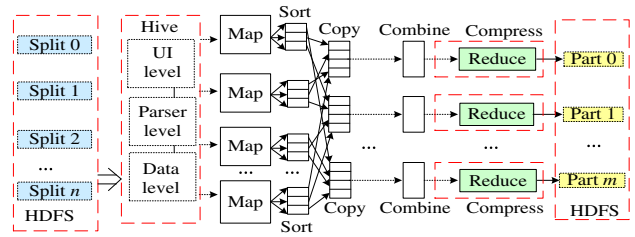


Fig. 1 MapReduce parallel processing flow of Hive

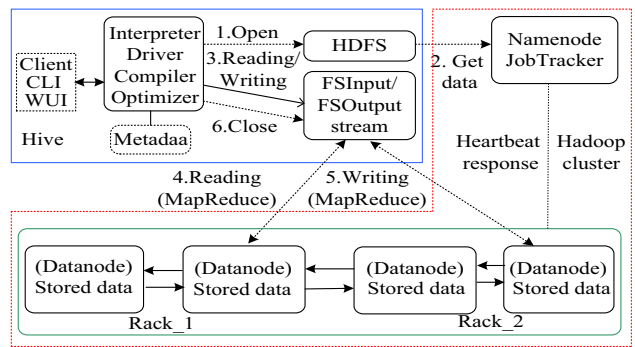


Fig. 2 Data processing flow based on Hive

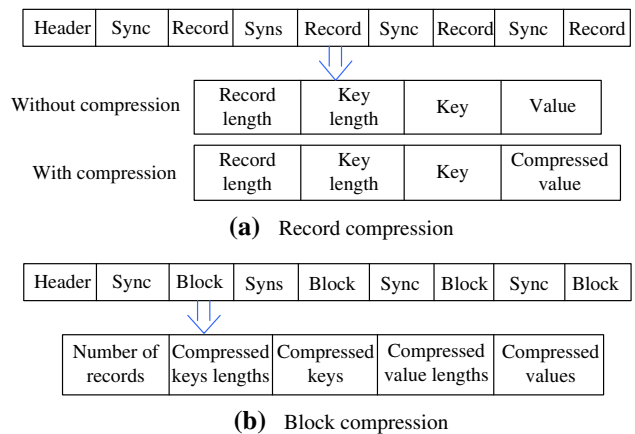


Fig. 3 SequenceFile format and its compression ways

- 3) RCFile is a special column oriented storage format, which skips the unrelated columns in query process. In fact, it does not really skip unwanted columns to jump to the target columns, but scan the stored metadata header of each row group to complete the above function. RCFile and its compression way are shown in Fig. 4.

This section introduces the principle of data processing and storage formats of Hive, which lays a theoretical foundation for the following sections.

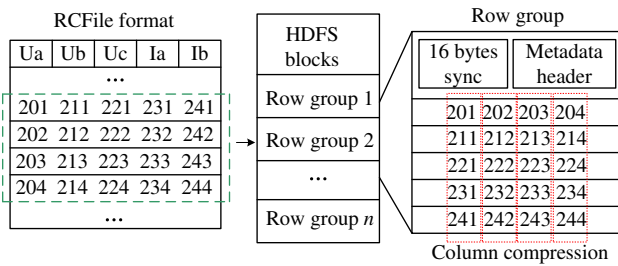


Fig. 4 RCFile format and its compression way

Table 1 Main monitoring data of smart substation

Device	Monitoring data
Transformer	Gas discharge, minim moisture content
Capacitor	Capacitive current, dielectric loss, unbalanced three-phase voltage
GIS	Partial discharge, gas pressure
Smart switcher	Switcher action times, the closing coil current, voltage

3 Applications of substation based on Hive

Smart grid acts as the future development direction of the power grid, which includes power generation, transmission, distribution, conversion and dispatching, etc. Undoubtedly, smart substation is one of the most important links in the power grid [12–14], which is mainly composed of primary intelligent electronic device (IED) and secondary networking equipments. Monitoring and control systems play an important role in completing the ordinary operation of smart substation. Some main monitoring data of smart substation is listed in Table 1.

In order to deal with the big data problems of smart substation, the applications of Hive can be integrated into the system of the smart substation, which is divided into three layers (processing layer, bay layer and substation layer). The applications of substation layer are based on bay layer and processing layer, include SCADA monitoring system and some other management systems.

The monitoring system and management system are integrated with Hive, not only can complete functions of automatic monitoring, automatic control, auxiliary decision and information sharing, but also can complete functions of big data mining and multidimensional data analysis, etc. The structure of smart substation system based on Hive is shown in Fig. 5.

The data processing flow of smart substation based on Hive can be logically divided into data source layer, computing layer, control layer and application layer. SCADA, data mining, auxiliary decision and multidimensional data analysis and other functions can be realized by

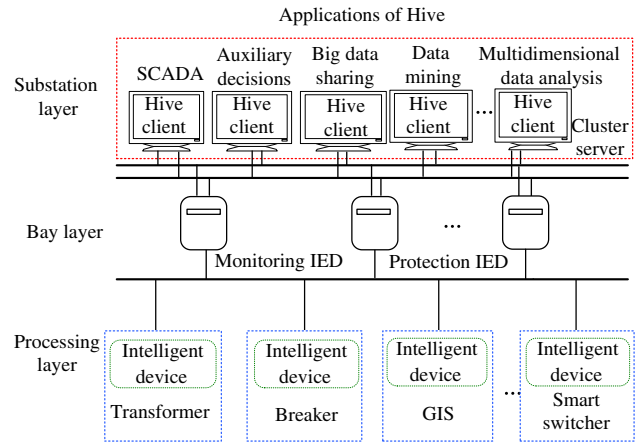


Fig. 5 Structure of smart substation system based on Hive

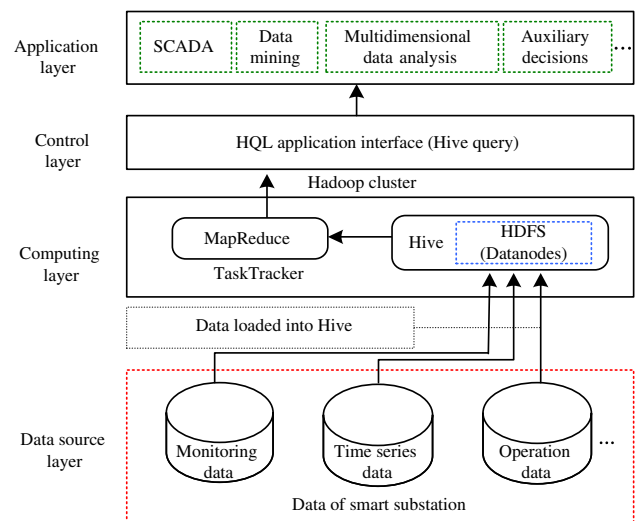


Fig. 6 Four logical layers of data processing flow based on Hive

using HQL interfaces. Four logical layers of data processing flow in the smart substation based on Hive are shown in Fig. 6.

4 Analysis of results

Firstly, cloud computing cluster is built on Hadoop platform constructed in Ubuntu11.10 system, composed of a Namenode (Master) and three Datanodes (Data 1, Data 2 and Data 3). Hive data warehouse infrastructure is built on top of Hadoop. Distributed cloud computing cluster of Hive is shown in Fig. 7.

Secondly, load the massive substation data into Hive data warehouse. Take 15 monitoring simulation values of substation as an example, to study the data compression and storage.

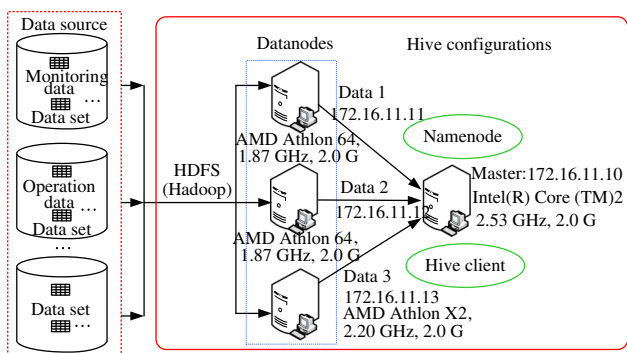


Fig. 7 Distributed cloud computing cluster of Hive

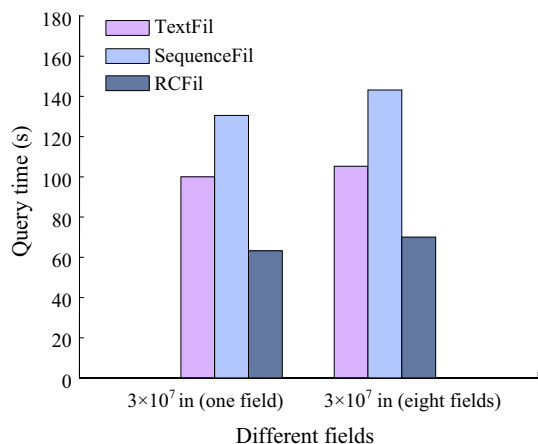


Fig. 8 Query time in three formats

4.1 Comparison of query time

The first experiment is carried out on three kinds of storage formats to study the query efficiency. Thirty million monitoring data records are stored in three kinds of storage formats, respectively. The query time of one field and eight fields is shown in Fig. 8.

As shown in Fig. 8, comparing with the query time of one field and eight fields in three kinds of storage formats, the query time of RCFil is relatively less, the query time of TextFile is middle, while query time of SequenceFile is relatively more.

4.2 Lossless compression

Hive supports Bzip2, Deflate, Gzip and Lzo compression type. In order to verify the query efficiency after compression, the second experiment is carried out under condition of five million monitoring records, testing three kinds of storage formats, i.e., TextFile, SequenceFile (compressed in block way) and RCFil by using four kinds of lossless compression (BZip2, Gzip, Deflate and Lzo) [15–17], respectively.

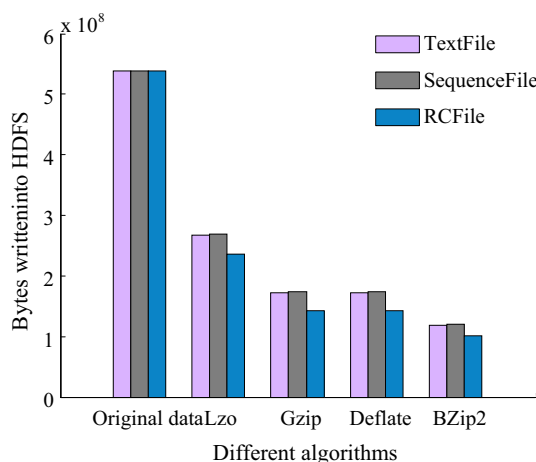


Fig. 9 Lossless compression ratios based on Hive

The lossless compression ratios processed by different kinds of algorithms on three kinds of storage formats based on Hive are shown in Fig. 9.

As shown in Fig. 9, the BZip2 compression ratio is higher than those of the other three kinds of lossless compression algorithms. In condition of RCFil storage format, the compression ratio of RCFil reaches about 81.3 %, approximately 3.5 % higher than those of TextFile and SequenceFile. The lossless compression ratios of Deflate and Gzip algorithms reach about 73.4 %, while the Lzo compression ratio reaches about 56.8 %.

Query time with and without data compressions on three kinds of storage formats is shown in Fig. 10 (select V001 from table_name where Num = Num_max; select V001,...,V008 from table_name where Num = Num_max).

Experimental results show that query time of BZip2 algorithm is relatively higher, and the efficiency is reduced by data compression.

Query time after Deflate, Gzip and Lzo become less than that without compression, which improves the query efficiency, at the same time, saving the storage capacity.

Although the BZip2 compression does not improve the query efficiency, when data stored in RCFil storage format, the query time of BZip2 almost equals to the efficiency without compression. It is showed that the RCFil improves the query efficiency to some extent.

Based on the above experimental results, big data of smart substation can be stored into Hive after compression according to actual demands.

4.3 Efficiency analysis of cluster

In Hive cluster system with p processors, if the parallel degree i satisfy $i \leq p$ ($i = 1, 2, \dots, n$), without considering the parallel overhead, the adjacent speedup in the system can be defined simply as follows:



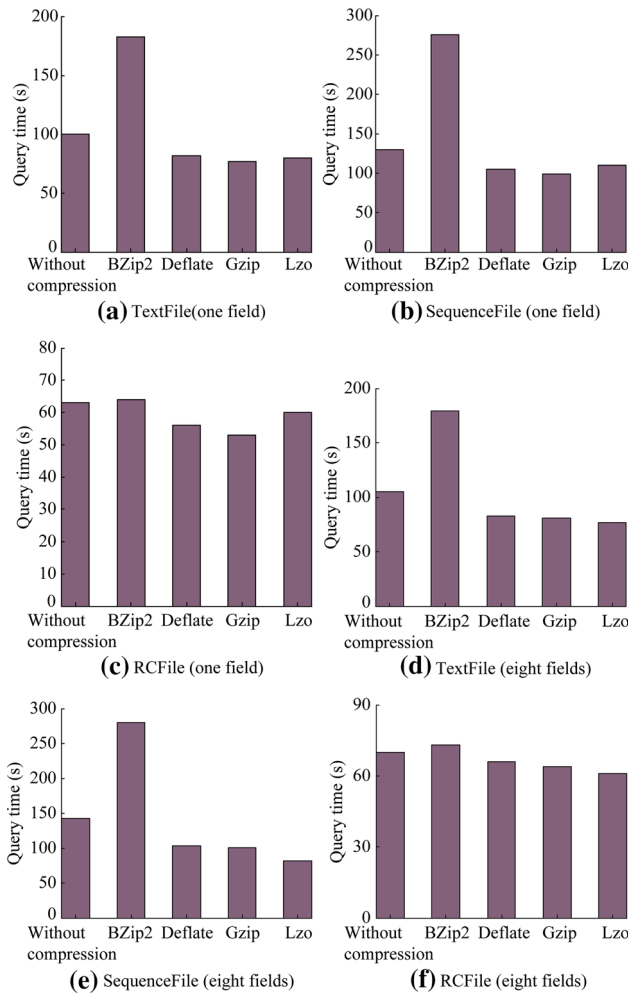


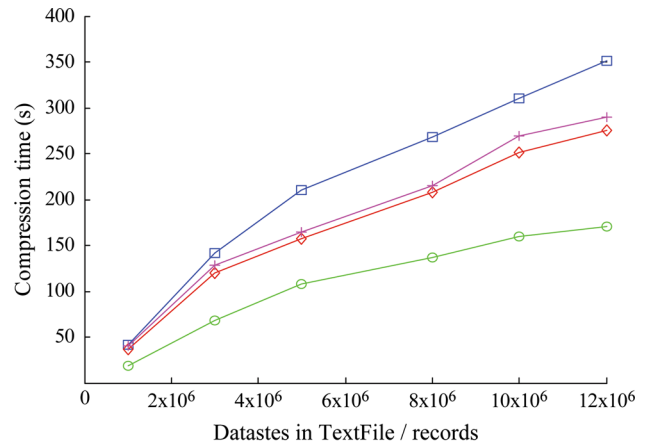
Fig. 10 Comparison of query time

$$S_{(m,n)}(p) = \frac{X_n/T(X_n)}{X_m/T(X_m)} = \frac{X_n T(X_m)}{X_m T(X_n)} \quad (1)$$

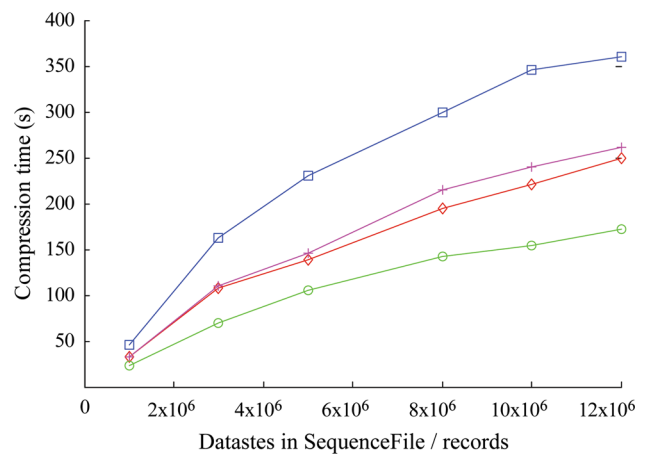
where X_m and X_n are the work loads; $T(X_m)$ and $T(X_n)$ are the parallel running time.

Considering the parallel overhead, the adjacent speedup can be further described as:

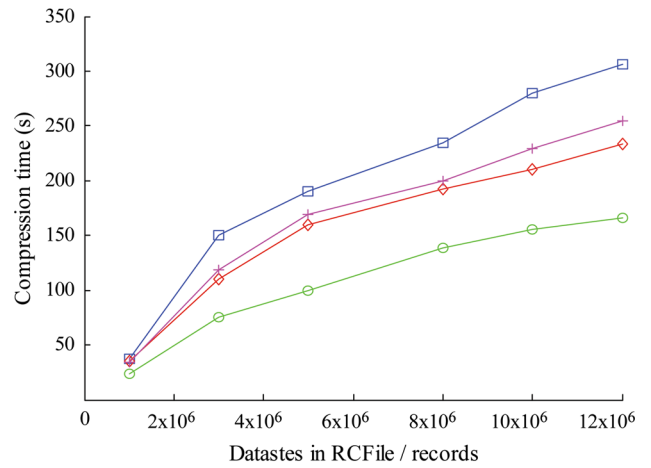
$$S'_{(m,n)}(p) = \frac{X_n/(T(X_n) + O(X_n))}{X_m/(T(X_m) + O(X_m))} = \frac{\sum_{j=1}^m X_{m,j}/V_{m,j} + O(X_m)}{\sum_{i=1}^n X_{n,i}/V_{n,i} + O(X_n)} \frac{X_n}{X_m} = \frac{X_m \sum_{j=1}^m f_{m,j}/f_{m,j} V_{m,j} + O(X_m)}{X_n \sum_{i=1}^n f_{n,i}/f_{n,i} V_{n,i} + O(X_n)} \frac{X_n}{X_m} = \frac{E_n}{E_m} \quad (2)$$



(a) Compression time of TextFile



(b) Compression time of SequenceFile



(c) Compression time of RCFile

—□— Bzip2; —◇— Deflate; —+— Gzip; —○— Lzo

Fig. 11 Compression consuming time on three storage formats

Table 2 $S'_{(m,n)}(p)$ and $C_{(m,n)}(p)$ of cloud cluster

Format	CA	$S'_{(1,3)}$	$S'_{(3,5)}$	$S'_{(5,8)}$	$S'_{(8,10)}$	$S'_{(10,12)}$	$C_{(1,3)}$	$C_{(3,5)}$	$C_{(5,8)}$	$C_{(8,10)}$	$C_{(10,12)}$
TF	BZip2	0.90	1.12	1.25	1.09	1.06	-0.06	0.18	0.42	0.32	0.3
	Deflate	0.91	1.27	1.21	1.04	1.09	-0.05	0.41	0.35	0.14	0.48
	Gzip	0.95	1.29	1.23	1.00	1.12	-0.02	0.44	0.38	0	0.57
	Lzo	0.83	1.06	1.25	1.08	1.12	-0.09	0.09	0.42	0.28	0.65
SF	BZip2	0.85	1.18	1.23	1.09	1.15	-0.08	0.26	0.38	0.34	0.77
	Deflate	0.93	1.29	1.14	1.09	1.06	-0.04	0.44	0.23	0.41	0.3
	Gzip	0.91	1.25	1.07	1.12	1.09	-0.05	0.38	0.14	0.48	0.52
	Lzo	1.01	1.09	1.18	1.15	1.07	-0.07	0.15	0.49	0.61	0.37
RCF	BZip2	0.77	1.32	1.29	1.05	1.10	-0.12	0.47	0.49	0.2	0.49
	Deflate	0.95	1.15	1.33	1.14	1.08	-0.02	0.22	0.56	0.57	0.38
	Gzip	0.86	1.18	1.35	1.09	1.09	-0.07	0.26	0.59	0.37	0.39
	Lzo	0.96	1.25	1.15	1.11	1.12	-0.02	0.38	0.25	0.46	0.64

For a system which parallel degree is i , $X_{n,i} = f_{n,i}X_n$, $X_{m,j} = f_{m,j}X_m$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$; $f_{n,i}$ and $f_{m,j}$ are the work load coefficients; $V_{n,i}$ and $V_{m,j}$ are running speed; $O(X_n)$ and $O(X_m)$ are parallel overhead time;

$$E_n = X_n X_m \sum_{j=1}^m f_{m,j} / V_{m,j} + X_n O(X_m);$$

$$E_m = X_m X_n \sum_{i=1}^n f_{n,i} / V_{n,i} + X_m O(X_n).$$

Parallel computing should be executed as $\lceil i/p \rceil$ times, the computing should be grouped by p to complete the computation of parallel degree i , when i is larger than p , at his time the adjacent speedup is described as:

$$S'_{(m,n)}(p) = \frac{X_m \sum_{j=1}^m \lceil j/p \rceil f_{m,j} / V_{m,j} + O(X_m)}{X_n \sum_{i=1}^n \lceil i/p \rceil f_{n,i} / V_{n,i} + O(X_n)} \frac{X_n}{X_m} = \frac{E'_n}{E'_m} \quad (3)$$

where the value of $\lceil i/p \rceil$ is the minimum integer not less than i/p . Parallel overhead $O(x)$ which is a complicated function related with software and hardware and application including interactive, communicational and parallel overhead.

In fact, many factors impact on the parallel efficiency, therefore, the relative efficiency increment caused by the relative amount increment of data can be used to reflect the performance of the cluster comprehensively. Hence, the following mathematical formula can be obtained:

$$C_{(m,n)}(p) = \left(\lim_{\Delta x \rightarrow 0} \frac{E_n - E_m}{E_m} \bigg/ \frac{x_n - x_m}{x_m} \right) = \lim_{\Delta x \rightarrow 0} \left(\frac{\Delta E / E}{\Delta x / x} \right) = \lim_{\Delta p \rightarrow 0} \frac{\Delta E / \Delta x}{E / x} = x \frac{dE}{dx} \bigg/ E = x \frac{d \ln E}{dx} \quad (4)$$

where variation $C_{(m,n)}(p)$ is a complex function which reflects the capability of running programs in parallel

processing system, related with the workload X , the serial bottleneck, the load coefficient and some other factors.

Operations of HQL tasks are transformed into MapReduce parallel tasks, so the third experiment is carried out in order to test the parallel compression consuming time in three kinds of storage formats of Hive, by using BZip2, Deflate, Gzip, and Lzo four kinds of lossless compression algorithms, respectively.

Take one million, three million, five million, eight million, ten million, and twelve million monitoring data records as the data research object, record the parallel compression consuming time in different number of data records, then the curve of compression time is draw, as shown in Fig. 11.

It is can be seen from Fig. 11 that the curve presents a convex trend, that is to say, the compression time of the more records is less than that of the less records. In order to quantitatively analyze the curve of the compression time in Fig. 11, $S'_{(m,n)}(p)$ and $C_{(m,n)}(p)$ are calculated with (2), (3) and (4). $S'_{(m,n)}(p)$ and $C_{(m,n)}(p)$ are shown in Table 2.

As shown in Table 2, when data records exceed three million, $S'_{(3,5)}$, $S'_{(5,8)}$, $S'_{(8,10)}$, and $S'_{(10,12)}$ are not less than one, which means that the processed data size in a unit of time increases, compression efficiency improves to some extent, as curves shown in Fig. 11 that, with data records increases, Hadoop cluster has a better compression executing efficiency, the compression efficiency increases to some extent.

$C_{(m,n)}(p)$ shows that different compression algorithms on different storage formats can provide detail information.

5 Conclusions

- 1) Storage format experiments verify that the query time of RCFile for big data is relatively less than that of



TextFile and SequenceFile, and so big data of smart substation can be stored with RCFile format because of its better time response.

- 2) Lossless compression experiments verify that big data of smart substation can be stored into Hive after compression, and query efficiency of data compressed by Lzo is higher than that by Gzip, Deflate and BZip2, while BZip2 compression ratio of data is relatively higher.
- 3) Parallel compression experiments verify that with the data records increase in a certain range, the cluster has a better parallel processing efficiency, and $S'_{(m,n)}(p)$ and $C_{(m,n)}(p)$ of cloud cluster further prove that big data processing of smart substation based on Hive is feasible.

Acknowledgments This work is supported by National Natural Science Foundation of China (No. 51267005) and Jiangxi Province University Visiting Scholar Special Funds for Young Teacher Development Plan (No. G201415, No. GJJ13350).

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- [1] Shafiullah GM, Oo AMT, Shawkat Ali ABM et al (2013) Smart grid for a sustainable future. *Smart Grid Renew Energy* 4(1):23–34
- [2] Chen JL, Huang C, Zeng ZX et al (2012) Smart grid oriented smart substation characteristics analysis. In: Proceedings of the 2012 IEEE conference on innovative smart grid technologies—Asia (ISGT Asia'12), Tianjin, China, 21–24 May 2012, 4 pp
- [3] Lü HL, Wang FY, Yan AM et al (2012) Design of cloud data warehouse and its application in smart grid. In: Proceedings of the international conference on automatic control and artificial intelligence (ACAI'12), Xiamen, China, 3–5 Mar 2012, pp 849–852
- [4] Thusoo A, Sarma JS, Jain N et al (2012) Hive—a petabyte scale data warehouse using Hadoop. In: Proceedings of the IEEE 26th international conference on data engineering (ICDE'12), Long Beach, CA, USA, 1–6 Mar 2012, pp 996–1005
- [5] Chuang CC, Chiu YS, Chen ZH et al (2013) A compression algorithm for fluctuant data in smart grid database systems. In: Proceedings of the data compression conference (DCC'13), Snowbird, UT, USA, 20–22 Mar 2013, 485 pp
- [6] Kaur R, Goyal M (2013) A survey on the different text data compression techniques. *Int J Adv Res Comput Eng Technol* 2(2):711–714
- [7] Kim HM, Lee JJ, Shin MC et al (2009) A multi-functional platform for implementing intelligent and ubiquitous functions of smart substations under SCADA. *Inf Syst Front* 11(5):523–528
- [8] White T (2010) *Hadoop: the definitive guide*, 2nd edn. O'Reilly, Sebastopol, pp 366–405
- [9] Wang DW, Xiao L (2012) Storage and query of condition monitoring data in smart grid based on Hadoop. In: Proceedings of the 4th international conference on computational and information sciences (ICIS'12), Chongqing, China, 17–19 Aug 2012, pp 377–380
- [10] Padhy RP (2012) Big data processing with Hadoop-MapReduce in cloud systems. *Int J Cloud Comput Serv Sci* 2(1):16–27
- [11] Korat VG, Deshmukh AP, Pamu KS (2012) Introduction to Hadoop distributed file system. *Int J Eng Innov Res* 1(2):172–178
- [12] Song Y, Li JR (2012) Analysis of the life cycle cost and intelligent investment benefit of smart substation. In: Proceedings of the 2012 IEEE conference on innovative smart grid technologies—Asia (ISGT Asia'12), Tianjin, China, 21–24 May 2012, 5 pp
- [13] Su YC, Wang XM (2010) Research of data acquisition method on smart substation. In: Proceedings of the 2010 international conference on power system technology (POWERCON'10), Hangzhou, China, 24–28 Oct 2010, 4 pp
- [14] Li HW (2012) Research on technologies of intelligent equipment in smart substation. In: Proceedings of the 2012 IEEE conference on innovative smart grid technologies—Asia (ISGT Asia'12), Tianjin, China, 21–24 May 2012, 5 pp
- [15] Kane J, Yang Q (2012) Compression speed enhancements to LZ0 for multi-core systems. In: Proceedings of the IEEE 24th international symposium on computer architecture and high performance computing (SBAC-PAD'12), New York, NY, USA, 24–26 Oct 2012, pp 108–115
- [16] Patel RA, Zhang Y, Mak J et al (2012) Parallel lossless data compression on the GPU. In: Proceedings of the Innovative parallel computing conference (InPar'12), San Jose, CA, USA, 13–14 May 2012, 9 pp
- [17] Yazdanpanah A, Hashemi MR (2011) A simple lossless preprocessing algorithm for hardware implementation of deflate data compression. In: Proceedings of the 19th Iranian conference on electrical engineering (ICEE'11), Tehran, Iran, 17–19 May 2011, 5 pp

Zhijian QU received the M.S. degree in School of Electrical & Electronics Engineering, East China Jiaotong University of China, Nanchang in 2004 and Ph.D degree in School of Electrical Engineering, Beijing Jiaotong University of China, Beijing in 2012. His recent research includes smart grid information network and big large data sets information system, and intelligent monitoring system.

Ge CHEN is currently a postgraduate in School of Electrical Engineering, East China Jiaotong University. His research interests include intelligent dispatching and information system, Hadoop and Hive.