Visualization Method Based on Cloud Computing for Real Estate Information

Mingyuan Yu
College of Computer Science and Technology
Zhejiang University of Technology
Hangzhou, China
E-mail: yu_mingyuan@163.com

Donghui Yu, Lei Ye, Xiwei Liu
College of Computer Science and Technology
Zhejiang University of Technology
Hangzhou, China
E-mail: donghui_hz@qq.com,
yelei@zjut.edu.cn,
lxwcc_2011@163.com

***Abstract*—In order to accelerate the computing speed and display efficiency of large-scale information visualization, this paper proposes a visualization method based on cloud computing for real estate information. The core idea is combining cloud computing and information visualization, making the data processing stage of information visualization on the cloud computing platform, so as to enhance the ability of parallel processing of big data and accelerate the computing speed and display efficiency. To verify the validity of this idea, we design and implement a visualization prototype system based on cloud computing. This system uses *Hive* to get the query results, uses *MapReduce* to process data in the background, and returns the result data to the server, and then the server generates relevant results by using information visualization technologies and back to the Web.**

Keywords-Cloud Computing; Information Visualization; Real Estate Information; MapReduce; Hive.

I. INTRODUCTION

Cloud computing has already become a hotspot of research and application on information services. Since Google's cloud computing model emerging in 2006, more and more companies participate in cloud computing, and cloud computing has become the mainstream model of new network service applications. Amazon, IBM, and other large companies have proposed their own cloud platforms, which can be regarded as a service. Such as Amazon's Elastic Compute Cloud (EC2) [1], users can build their own private cloud [8] platforms on it. The open-source and free framework--*Hadoop*--has been the most widely used, and some small companies use *Hadoop* to build their own private cloud entirely. However, the private cloud also faces challenges. Although it is easy to set up, ensures the data security, and users can utilize all the resources in the private cloud, it requires too many physical resources. The resources which the cloud obtains are their own cloud equipment resources. It costs too much. The public cloud [8] is the future mainstream framework of cloud computing development.

Compared with grid computing and distributed computing, cloud computing has many advantages. First, it cost less, which is the most prominent advantage. Second, it is supported by virtual machines, so that it can handle some things easily which is difficult in grid. Third, it can execute mirror deployment, which makes us to deal with heterogeneous process expediently. In addition, cloud computing emphasizes services, which is more suitable for commercial operation.

Information visualization [9] is also a popular emerging field of visualization. Information visualization combines several theories and methods such as scientific visualization, human-computer interaction, data mining, image, graphics, cognitive science, and gradually develops [10]. The traditional scientific visualization generally pays attention to three-dimensional visualization, including areas such as medicine, biology and architecture. While Information Visualization attaches importance to display information to users by any diagrams, so that users can get the information conveniently. A good interactive technology can show the information and data batter, and facilitate the user's operation. Therefore, human-computer interaction [11] in information visualization is particularly important. With the increasing of the information and data, the traditional Web applications cannot store all the information data in cache, and data processing efficiency is obviously insufficient to address these issues. We use parallel processing to solve this problem in the cloud. Users submit the demands to the system on the Web. Then process data in the cloud and return the results to the Web. Although this method increases the additional overhead of interaction time, it is faster than the traditional method. Besides, it helps solve the problem of information and data processing of big data.

In the field of information visualization, it generally has the following six data types: multi-dimension (including 1D, 2D), time, space, hierarchical (tree) structure, the network (figure) structure, text and so on. For different data types, you need to use different visualization techniques. In this paper, we mainly use the hierarchical structure--*TreeMap*.

TreeMap is a very space-saving way to display hierarchical data [15]. The display space is divided into a series of rectangles, and each rectangle is a data item. If one data item contains other data in the hierarchy, the rectangle is subdivided into smaller rectangles. In the past decade, *TreeMap* is widely used in the field of information visualization.

In this paper, we use *Hadoop* to build our private cloud platform, and design a visualization prototype system for real estate information. When users login our prototype system in

 ISBN: 978-1-61208-215-8

the Web, they can apply for the released service, and use the released data to apply for a new service. They can also upload the relevant data and then apply for a new service. After users submitting the demands to the system, the system will analyze data in the background, and return the results to the Web. Only have a browser, can users access the system and get results.

Section II describes the related work of cloud computing and information visualization, including using cloud computing to solve current information services problems, information visualization progress at home and abroad, and the application progress of combining cloud computing and engineering visualization. In Section III，we describe the basic idea and system framework of the real estate information visualization model based on cloud. And we implement this model and do some experiments in Section IV. The last section draws our conclusions and future research directions.

II. RELATED WORK

Cloud computing is a hotspot in the current field of information services. Zhou et al. analyzed the cloud services [12], and exampled the three mainstream services, Data as a Service (DaaS), Network as a Service (NaaS) and Identity Policy Management as a Service (IPMaaS). Chen et al. applied cloud service to computer-aided design [13], and designed intelligent house. Liu et al. applied cloud computing to the elasticity public VPN service model [14], which was different from the traditional VPN. This model would pre-assess the resource consumption and dynamically adjust.

Hoang et al. proposed an elastic cloud storage system—ecStore [17], which supported automatic data partitioning and replication, load balancing, efficient range query and transaction access. Kossmann et al. proposed a modular cloud storage system—Cloudy [18], which provided a highly flexible architecture for distributed data storage, and manipulated various workloads. Based on a common data model, Cloudy could be customized to meet different application needs. Because of more and more applications and their data placed on mobile devices, independent storage of mobile systems had become a key issue. Yuan et al. proposed a wireless Network File System—RFS [19], which realized a device-aware cache management, data security of customers and privacy protection.

Nowadays, the space-time data used in the information visualization are becoming larger and reaches to the scale of TB or PB level, and the combination of cloud computing and information visualization is the general trend. In 2010, Ma et al. published Multi-GPU volume rendering using *MapReduce* [3]. Then he transplanted this system to the Web, and proposed the interface design for future cloud-based visualization services [4]. Many scholars applied cloud computing to geographic information services [5][6][16], which were appropriate to solve the problem that large GIS data processing at one computer was slowly and could not meet user demands. As we know, the data of Google Earth is large and updated in real time, which requires us to explore other ways to meet this demand. Google Earth is built upon the Google distributed systems based on cloud computing.

Combination of the design concept of above-mentioned areas, we have designed and implemented a house information visualization system for the computational speed and display efficiency of large-scale information visualization, which combines cloud computing and information visualization. We use *MapReduce* to process data in the background, generate results through *TreeMap*, and return the final results to the Web.

III. VISUALIZATION MODEL BASED ON CLOUD COMPUTING FOR REAL ESTATE INFORMATION

A. Basic Idea

The basic idea of this model is using *MapReduce* to analyze and process real estate data in the background, and using *TreeMap* to display results on the Web.

After login the system, users can see the released services and released data in the menu--*Released Services*. If the released services meeting users' needs, user can apply for the released services in the menu--*Applying Services*. If the released services cannot meet users' needs, the user can choose to apply for a new service. Users submit the detailed description of the service and the data service uses, and wait for the administrator to open this new service.

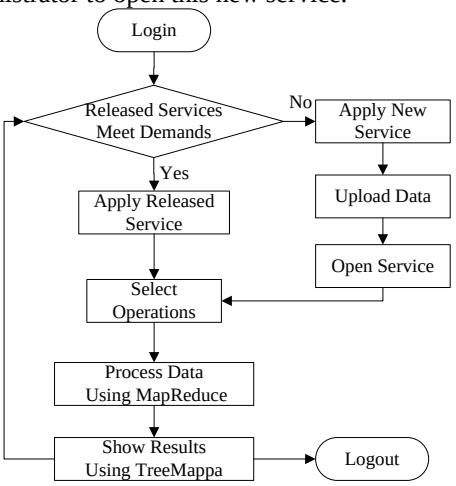


Figure 1. System flowchart

After applying for the service, users will see their own services in the Applying Services menu, such as Average House Prices. Click on this option enter Selecting Operation interface which provides the selections as city, year, showing effect etc. After submitting the selection results, users can see the relevant results on the Web. The system flowchart is shown in Figure 1.

This system is divided into three steps. Firstly, upload the data to HDFS. Secondly, use *MapReduce* to process data. Finally, display the results on the Web.

1) Data uploading

Uploading data also can be divided into two steps--Uploading data to the Tomcat server and then uploading to HDFS.

Released data format is shown in TABLE I.

 ISBN: 978-1-61208-215-8

TABLE I. DATA FORMAT

Column Names	Data Types
House	String
City	String
District	String
Type	String
Year	Int
Month	Int
HousePriceperQuaremeter	Float
BuildingArea	Float
HousePrice	Float
RecordCount	Int

If users want to apply for released services, the data users upload must meet the above format. If users apply for a new service, the data format can be any form of text files. The administrator will open the related services in accordance with the uploaded data.

2) Data processing

The uploaded data are divided into several independent blocks, and parallel processed by *Map* function. According to the different demands of users, specify different fields as *Key* and *Value* of *Map* functions. For example, if users want to see the average house prices of every city, specify *House* and *City* as *Key*, *HousePriceperQuaremeter* as *Value*. If users want to see the average prices of every city and every year, specify *House*, *City* and *Year* as *Key*, *HousePriceperQuaremeter* as *Value*. *MapReduce* framework would sort the *Map* output, and transmit to *Reduce*. *Reduce* function returns different results according to different demands.

If users want to see the average prices of every city, *Map* and *Reduce* function is as follows.

Map function
Input: LongWritable key, Text value, Context context
Output: Null
change Value to String;
change String to StringTokenizer;
for(i=0;hasMoreTokenizer;i++)
  specify nextToken as Key;
  specify nextToken as Value;
write context (Key, Value);

Reduce function
Input: Text key, Iterable<IntWritable> values,
        Context context
Output: Null
sum = 0, num = 0;
for (IntWritable value : values)
num++;
sum += get value;
set result (sum/num);
write context (key, result);

3) Results displaying

Users do not concern about the background how to deal with the data. What they care about is whether the results meeting user demands or not, intuitive and easy to understand, so that users can see the desired data directly from the results. In our system, we display the results by *TreeMap*. At this stage, rewrite *TreeMap* configuration file and data file according to the different demands, and then display the results on the Web, as shown in Figure 4.

B. Model Architecture

In Figure 2, the cloud cluster structures above several servers. First, we build a Hadoop distributed file system (HDFS), consists of a namenode and several datanodes. Above HDFS, we build a distributed database (Hbase) for data management. Users can login the system through various devices and apply for services.

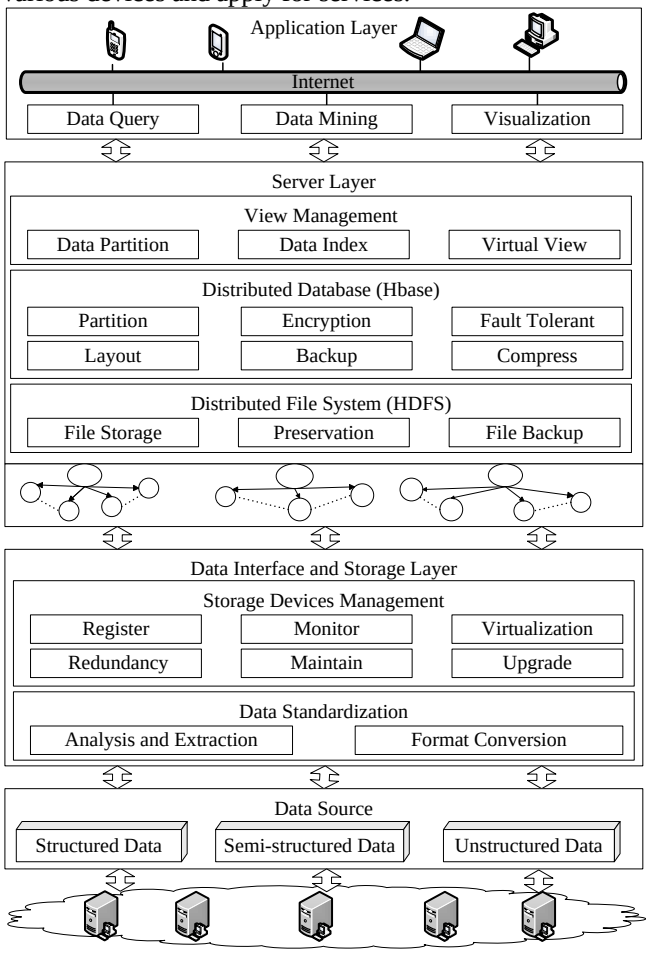


Figure 2. Cloud platform architecture

1) Data Source (DS)

The data source is the data center that provides a wide range of structured, semi-structured or unstructured data.

Commonly used data source includes, observational data—the practical observational and survey data, including field survey data and record data of observation stations; analyze data—using chemistry, physics and other scientific methods analysis of the survey data; graphic data—graphic data of medical, aerospace and other industry demand; survey data—various types of survey reports, social survey data, and so on. At present, Chinese huge number of data source produce the amount of data which achieve TB or PB level, and have different structures, the dynamic changes and updated in real time.

 ISBN: 978-1-61208-215-8

2) Data Interface and Storage Layer (DISL)

DISL completes the standardization of data acquisition provided by the data source and management of storage devices.

Due to the various structures of the data provided by the data source, the data have to be standardized analyzed, extracted and format converted in DISL and data, and convert to a standard format for storage devices storage. The huge number of storage devices in the cloud storage distribute in different regions, connecting with each other through the WAN, Internet or Fibre Channel (FC) network. There is a unified management system of storage devices in DISL, which manages logic virtualization of storage devices, registers storage devices, virtualizes storage devices, manages multi-link redundancy, and monitors, maintains, upgrades storage devices.

3) Data Server layer (DSL)

In the DSL, we build a distributed file system using *Hadoop*, which uses master/slave architecture, consisting of a namenode and several datanodes. The namenode is a central server responsible for managing the file system and client access to files. The datanode is distributed in the cluster responsible for managing the storage of its own node. Internally, a file is divided into one or more blocks, and each block is stored in one datanode. The namenode is responsible for file system operations, such as open, close, rename files and directories, and decide the block mapping to the specific datanode. Under the command of the namenode, datanode creates, deletes, and copies blocks.

Above *HDFS*, we build a distributed database (Hbase) for data management. Through data partition and data layout technology, data can be stored in *HDFS*. And then through data encryption, fault tolerance, compression, backup technologies and measures, we ensure that data will not lose in the cloud, and the cloud is safe and stable.

In addition, this layer can do data partition, create data index and virtual view in order to efficient query.

4) Data Application layer (DAL)

In the DAL, we can design efficient parallel data query optimization algorithms, data mining and analysis algorithms to reduce the query response time. Through mobile phones, PDAs, PCs and others, users login our system and achieve data access, data analysis services.

IV. PROTOTYPE SYSTEM AND EFFICIENCY ANALYSIS

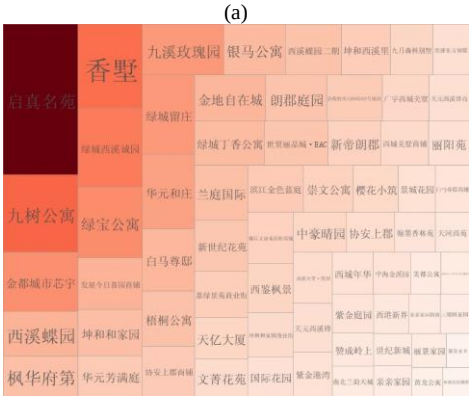
A. System Implementation

The prototype system shown in Figure 3 provides several services, such as *Released Services*, *Applying Services*, *Being Nodes* and others. *Released Services* displays the released services and released data. If users want to apply for the released services, they can apply for the services directly in *Applying Services*. If the released services cannot meet user needs, users can choose to apply for a new service and upload related data, and the administrator will open the service in time.

Figure 3. Prototype system

After the services opened which users apply for, users can see the specific services in *Released Services*, such as *Average House Prices*. If users select *City='Hangzhou'*, they can get average house prices in Hangzhou, as shown in Figure 4.

(a)



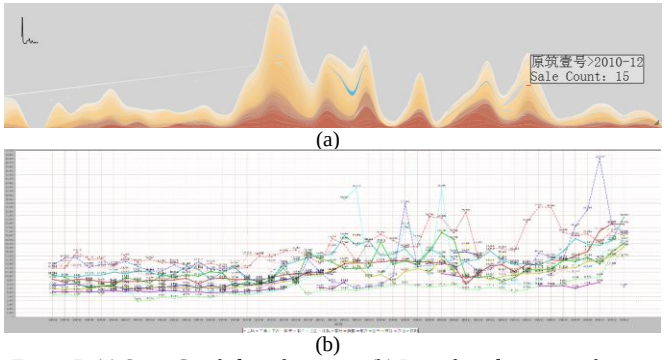
(b)

Figure 4. (a)Average house prices in Hangzhou;
(b)Average house prices of Xihu district in Hangzhou

In the *Treemap* diagram shown in Figure 4, both the size of the rectangle and the color depth stand for the house average prices. The deeper the color, larger rectangle, means higher house prices.

In addition, users can also select other visualization methods like *StringGraph* and *Line Chart* to see results, shown in Figure 5.

 ISBN: 978-1-61208-215-8



(a)

(b)

Figure 5. (a) StringGraph for sale counts; (b) Line chart for average house prices of every district in Hangzhou

B. *Efficiency Analysis*

Hardware environment: one Dell PowerEdge R410 server (Intel Xeon quad-core 5600 processors, 4G memory, 160G hard disk), seven Dell Optiplex 780 PCs (Intel Core 2 Duo E7500 processors, 4G memory, 160G hard drive) .

Software environment: Ubuntu10.10, JDK1.6.0_24, Hadoop-0.20.2, Hbase-0.20.6, Zookeeper-3.3.3.

TABLE II. MAPREDUCE COSTS

Data Size	Map (s)	Reduce (s)	MapReduce (s)
64M	38	36	74
1G	40	38	78
8G	41	42	83
32G	43	45	88
128G	49	56	105

Firstly, use the server as the namenode and virtual machines created on other PCs as datanodes to build HDFS. Secondly, build Hbase above HDFS. TABLE II shows the average time that this cluster process data using *MapReduce*.

Seen from TABLE II, the cost using *MapReduce* computational framework to deal with 64MB of data and 128GB of data is little different. Therefore, using *MapReduce* to deal with big data has obvious advantages.

We also do the experiment on traditional single-server--PowerEdge R410 server (Intel Xeon quad-core 5600 processors, 4G memory, 160G hard disk). Figure 6 shows the contrast about the costs of above two experiments.

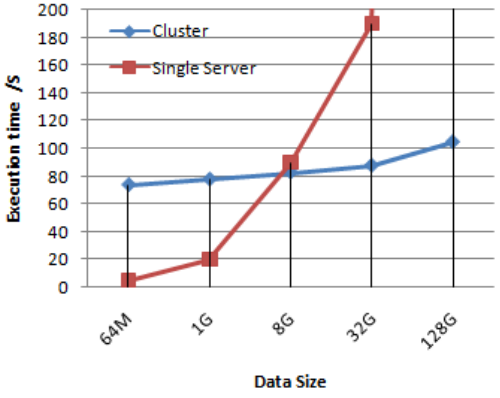


Figure 6. The execution time of cluster and single server

Seen from Figure 6, when the data size is about 8GB, the execution time of cluster and single server is nearly. For small-size data (less than 8GB), the traditional single-server service model has advantages. When the data is larger than 8GB, the advantages using *MapReduce* computational framework to parallel process are very obvious, while single server costs too much time. Therefore, the cloud computing model is more suitable for big data processing.

In addition, in accordance with the format of TABLE I, we use Hive to create partition table of real estate information. The HiveQL is as follows.

CREATE TABLE HouseInfo_Partition (House String, District String, Type String,Year Int, Month Int, HousePriceperQuarem-eter Float, BuildArea Float, HousePrice Float, RecordCount Int) PARTITIONED BY (City String);

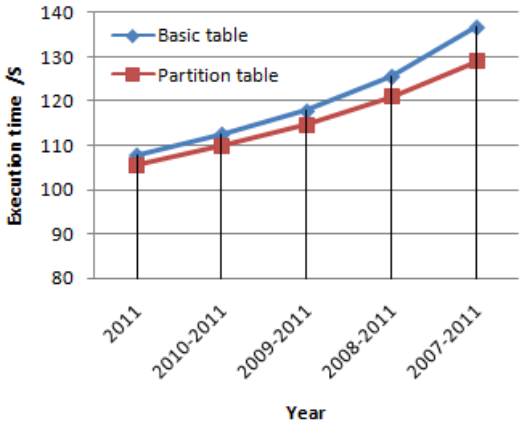


Figure 7. The query execution time of partition table and basic table

User's operation can be directly transformed to Hive query. In Hive, a partition of the table corresponds to a directory. Which partition table we create partitioned by *City*. Some queries need not all data in the table. For example, when we query the prices of Hangzhou real estates, we do not have to search for redundant information of other cities. Seen from Figure 7, the query execution time of partition table is slightly less than basic table.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a visualization method based on cloud computing for real estate information. The core idea is the combination of cloud computing and information visualization, making the data processing stage of information visualization on cloud computing platform, to speed up the calculation efficiency. Use *MapReduce* to process data and use *TreeMap* to show the results.

In addition, we design and implement a prototype system for real estate information visualization. Users can apply for the relevant real estate information visualization services on our system. The experimental analysis shows that the advantages using *MapReduce* computational framework to parallel process big data are very obvious, which provides a new approach for massive information visualization.

This system also has some unsatisfactory areas. For example, human-computer interaction is not convenient, users' waiting time is too long and the system also has some limitations. Therefore, in future, it is necessary to consider the improvement of human-computer interaction, but also consider the results effect and user-friendly operation.

 ISBN: 978-1-61208-215-8

VI. ACKNOWLEDGMENT

This work was partly supported by the National Natural Foundation of China (NSFC) project under grant No. 60703002, Major Program of Zhejiang Provincial Natural Foundation under grant No.Z1090630, the Natural Science Foundation of Zhejiang Province (No.Y1110768, Y110950), and Dr start-up fund research of Zhejiang University of technology (No: 119001229).

REFERENCES

[1] G. Turcu, I. Foster, and S. Nestorov. "Reshaping text data for efficient processing on Amazon EC2." Scientific Programming, 2011, 19(2-3), pp. 133-145.

[2] J. Dean, and S. Ghemawat. "Map/Reduce: Simplied Data Processing on Large Clusters." Communications of the ACM, 2004, 50(1), pp. 107-113.

[3] Jeff A. Stuart, C.K. Chen, K.L. Ma, and John D. Owens. "Multi-gpu volume rendering using mapreduce." 1st International Workshop on MapReduce and its Applications, 2010.

[4] T. Yuzuru, C.K. Chen, M. Stephane, and K.L. Ma. "An Interface Design for Future Cloud-based Visualization Services." 2nd IEEE International Conference on Cloud Computing Technology and Science, 2010, pp. 609-613.

[5] X.Q. Yang, and Y.J. Deng. "Exploration of Cloud Computing Technologies for Geographic Information Services." 2010 18th International Conference on Geoinformatics, 2010, pp. 1-5.

[6] C.W. Yang, M. Goodchild, Q.Y Huang, D. Nebert, R. Raskin, Y. Xu, M. Bambacus, and D. Fay. "Spatial cloud computing: how can the geospatial sciences use and help shape cloud computing." International Journal of Digital Earth, 2011, 4(4), pp. 305-329.

[7] W.G. Teng, W.H. Wen, and Y.C. Liu. "From experience to expertise: digesting cumulative information for informational web search." Journal of information science and engineering, 2012, 28(1), pp. 161-176.

[8] H. Brian. "Cloud computing." Communications of the ACM, 2008, 51 (7), pp. 9-11.

[9] Nahum D. Gershon, and Stephen G. Eick. "Information Visualization." ACM Interaction, 1998, 5(2), pp. 9-15.

[10] M. Kanae, Y. Masato, and S. Hideki. "A Proposal of Framework for Information Visualization in Developing of Web Application." 2011 IEEE/IPSJ 11th Informational Symposium on Applications and the Internet(SAINT), 2011, pp. 457-462.

[11] V. Rantanen, T. Vanhala, and O. Tuisku. "A Wearable, Wireless Gaze Tracker with Integrated Selection Command Source for Human-Computer Interaction." IEEE Transactions on Information Technology in Biomedicine. 2011, 15(5), pp. 795-801.

[12] M.Q. Zhou, R. Zhang, D.D. Zeng, and W.N. Qian. "Services in the Cloud Computing Era: A Survey." 2010 4th International Universal Communication Sysposium, 2010, 10, pp. 40-46.

[13] S.Y. Chen and Y.F. Chang. "he cmputer-aded dsign sftware for sart hme dvice based on coud cmputing service." 010 Second WRI World Congress on Software Engineering, 2010, 7, pp. 273-278.

[14] Q. Liu and W.Q. Gu. "An Elastic Public VPN Service Model Based on Cloud Computing." 2011 IEEE 2nd International Conference on Software Engineering and Service Science(ICSESS), 2011, pp. 290-294.

[15] A.B. Vigas, W. Martin, V.H. Frank, K. Jesse, and M. Matt. "Many eyes: A site for visualization at internet scale." In Proceedings of Infovis, 2007, 13(6), pp. 1121-1128.

[16] K. Andrews and M. Lessacher. "Liquid Diagrams: Information Visualisation Gadgets." Information Visualisation(IV), 2010 14th International Conference, 2010, pp. 104-109.

[17] H.T. Vo, C. Chen, and B.C. Ooi. "Towards Elastic Transactional Cloud Storage with Range Query Support." Proceedings of the VLDB Endowment, 2010, 3(1-2), pp. 506-514.

[18] D. Kossmann, T. Kraska, S. Loesing, S. Merkli, R. Mittal, and F. Pfaffhauser. "Cloudy: A Modular Cloud Storage System." Proceedings of the VLDB Endowment, 2010, 3(1-2), pp. 1533-1536.

[19] Y. Dong, H. Zhu, J.Z. Peng, F. Wang, M.P. Mesnier, D.W. Wang, and S.C. Chan. "RFS: A Network File System for Mobile Devices and the Cloud." ACM SIGOPS Operating Systems Review, 2011,45(1), pp. 101-111.

[20] A. Thusoo, S.J. Sen, N. Jain, Z.Shao, P. Chakka, N. Zhang, S. Antony, H. Liu, and R. Murthy. "Hive – A Petabyte Scale Data Warehouse Using Hadoop". 26th IEEE International Conference on Data Engineering (ICDE 20-10), 2010, pp. 996-1005.

 ISBN: 978-1-61208-215-8