International Journal of u- and e- Service, Science and Technology
Vol.9, No. 6 (2016), pp.399-408
http://dx.doi.org/10.14257/ijunesst.2016.9.6.36

Online Integrated Development Environment for MapReduce Programming

Zhiqiang Ma, Shuangtao Yang, Zhida Shi and Rui Yan

School of Information Engineering, Inner Mongolia University of Technology, Hohhot, 010080, China
mzq_bim@163.com

Abstract

Though MapReduce programming model simplifies the development of parallel program, ordinary users have difficulties in setting up the development environment for MapReduce. The online integrated development environment for MapReduce programming can solve this problem, thus users need not build the environment themselves, only need to focus on the logical design of the parallel program. During the software construction, the problem of independent space setting and naming conflict of the file in the multi-user environment, and the problem of online compiling, execution and instant feedback message to client are solved. The software has been deployed and tested in Hadoop cluster, and can meet users' basic requirements for the development of MapReduce.

Keywords *: Parallel Programming; Distributed Cluster; Online IDE; Hadoop; MapReduce*

1. Introduction

MapReduce, Google file system and big table, which are proposed by the Google Corporation, have become the core technologies of cloud computing [1-3]. MapReduce is a programming model and a set of algorithms for distributed storage and distributed processing of very large data sets on computer clusters. Distributed cluster is the foundation of MapReduce，which provides an environment for computing and communication. MapReduce programming model simplifies the development process of parallel program, so developers just need to concentrate on the logic design and implementation of application. At present, Hadoop provided by the Aparche Foundation implements an open-source MapReduce framework, thus MapReduce programming model is widely used.

MapReduce programming model reduces the difficulty of parallel program development, but a developer has to deploy and configure Hadoop on his own computer. So, there are several difficulties in a development : (1) Hadoop framework is complex, so it is easy to make mistakes for junior developers in deployment; (2) There are great differences between Hadoop versions, thus developers will need to spend more time and energy on deployment when they update Hadoop; (3) There is not an integrated development environment for MapReduce programming，meanwhile the version of MapReduce plug-ins for IDE is even farther behind Hadoop, so the application of MapReduce programming is limited; (4) A junior developer lacks the hardware resources for running Hadoop; he can only deploy Hadoop on single PC to simulate a computer cluster, so the performance of Hadoop can not reflect the advantages of the cluster.

In this paper, we propose an online integrated development environment for MapReduce programming, online IDEMP for short. It supports multiple developers and

ISSN: 2005-4246 IJUNESST

multiple applications; it provides an environment for compiling, building and running MapReduce programs; and it supports to upload and download the data sets of application. So it is flexible enough for developers to configure MapReduce version and data sets.

2. Related Work

General integrated development environment includes four parts: a code editor, a compiler, debugging tools and runtime environment. Its working model has desktop model and online model [4-6]. Desktop model has been widely used to develop desktop applications and server programs by programmer, such as Microsoft visual studio and open source Eclipse. The other model, which adopts Browser/Server design pattern, is online integrated development environment and also a developing tool. We use online model to build IDEMP, so compiling, debugging and running program are working at the server, programmers use the client to code, give commands and receive output messages. Thus programmers need to register the accounts before developing. When they use online environment, they need to open the web page without deploying runtime environment. So development environment of client requires almost nothing to hardware, only PC. For example CodeRun [7] tool based on JavaScipt supports code highlighting, debugging and deployment for C# language; Cloud9IDE [8] tool based on NodeJS includes built-in VIM and supports code highlighting for JS, HTML and CSS ; C-Eclipse[9] tool , whose core function is web services, includes code editing, code promoting, debugging and running for Java language.

3. Software Architecture

Online IDEMP adopts three layers software architecture, so it has been improved greatly in independence between functions, reuse of module, expandability and maintenance [10-11]. The software architecture of online IDEMP is shown in Figure1, which consists of presentation layer, business logical layer and data access layer.

Presentation layer is user interface, which contains a number of significant enhancements such as: list of project, engineering tools, code tools, control panel, code workspace, and error message. It provides development and management operations, such as project management, code management, and project deployment and so on.

Business logic layer is software core, which contains a number of significant enhancements such as: user management, project management, code editing management, project deployment, and project debugging. It provides users authentication, code compiling and debugging, and project deployment functions.

Data access layer contains a number of significant enhancements such as: file management, package management, and data set management; it mainly provides file operations, including the file basic operation and the database operations.

When a developer uses online IDEMP, the commands are passed from presentation layer to business logical layer, and they will run on business logical layer. In this progress data are from data access layer.

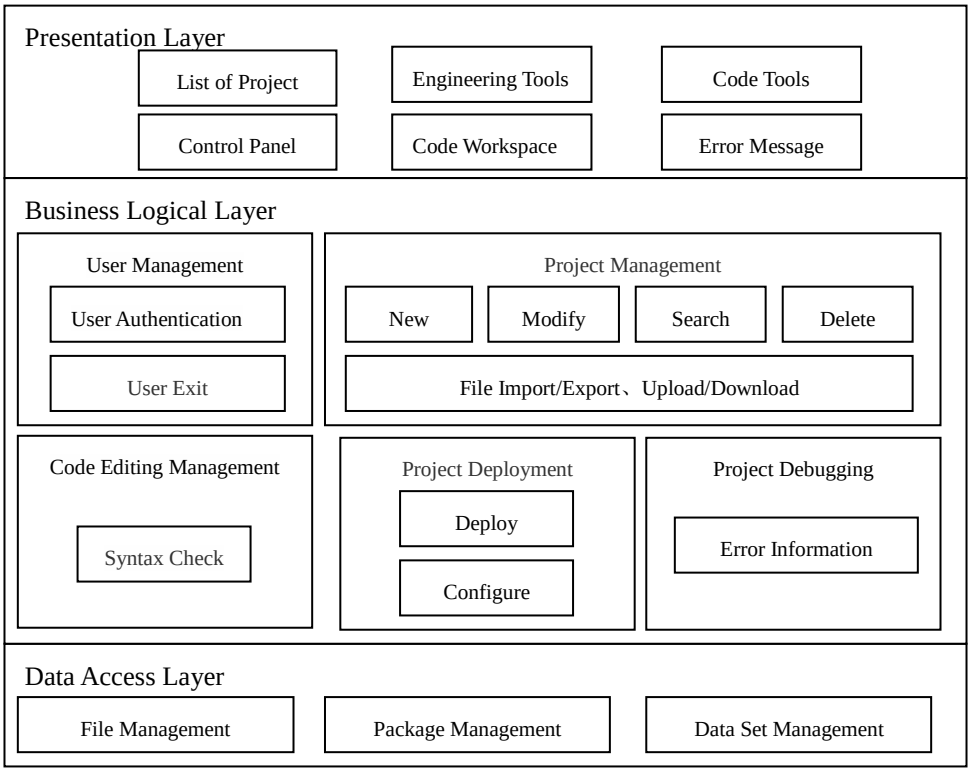

Figure 1. Software Architecture

Online IDEMP is deployed on the server, which includes Hadoop cluster, web server, database server and file server. When it goes, user management, project management, task control and information feedback run on the web server. Files of project are stored in file server; user information, file path information and configuration information are stored in database server; executable programs and data sets are stored in Hadoop cluster, and jar programs run on Hadoop cluster.

4. File System Architecture

In online IDEMP, developer's workspace includes configuration files, project source files, executable files, Jar files and data set files *etc.*. Since file management is pretty complicated, a key challenge is how to avoid name conflict and storage independence. We design a file system to solve the above problem. The file system is deployed on web server, file server, Hadoop cluster and data server, which is shown as Figure2.

Online IDEMP is deployed on web server which sends commands to other servers with JDBC and SSH protocols. All tables of online IDEMP which are stored on database server aim to reduce the complexity of implementation. Project files and JDK are stored on file server which realize the independent storage and project compiling. The jar file of compiled project is stored on the master node of Hadoop, and data sets and the output file are stored as HDFS file type. User program runs as MapReduce model, and it outputs results on Hadoop cluster.

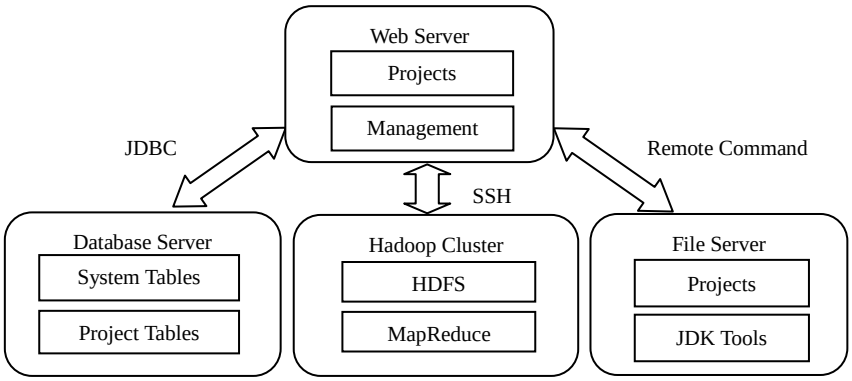

Figure 2. File System Architecture

4.1 Directory Hierarchy

Since online IDEMP is a multi-user system, the file system must support more than one project which contains several files and directories, and support file operations which include creating file, reading file, modifying file and deleting file according to name. We design the tree structure (or hierarchy of files and directories) to achieve the above requirements, as shown in Figure 3. Directory hierarchy consists of four significant enhancements such as: root directory, system table, user table, and project table.

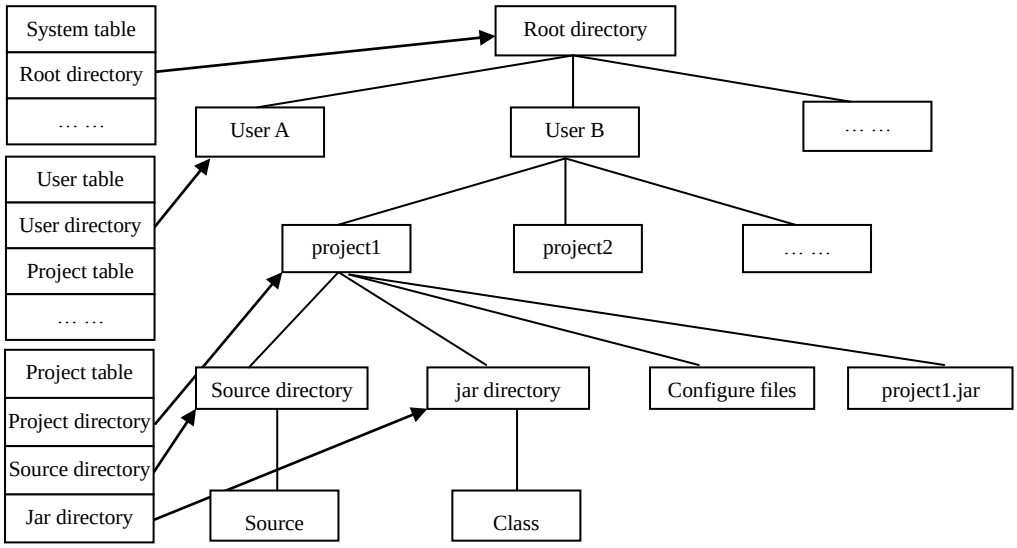

Figure 3. Directory Hierarchy

The root directory is the starting point of the file system. When online IDEMP initializes, the root directory is specified by the administrator. The root directory is the only one in online IDEMP. Home directory refers to user's workspace. When a developer registers an account on online IDEMP, file system creates a home directory. The directory's name is always the same as the developer's login identifier, which most developers refer to their login and which the file system calls userid. When a developer log in, the system places them in their home directory. Project directory refers to each project's workspace. When a developer creates a project, project's directory will be created automatically in developer's home directory. The project directory's name is the same as the project's name, or the project directory's name is named by a developer. Source directory, Jar directory and configuration files are also created automatically by

online system. The jar directory is designed for the hidden directory, where the compiled class files are saved. If the project is compiled, a named "project.jar" file will be established in project directory.

A developer's files are placed on file server, and system table, user table and project table are placed on the database server. System table has the attribute of the root directory and the attribute of user table name. User table has userid, home directory and project table name. Project table has some attributes, such as project directory, source directory, and jar directory and so on.

4.2. Configuration of HDFS Directory

Online IDEMP needs to upload the compiled "project.jar" and data set to Hadoop cluster before they run. Because MapReduce runtime environment will be used by multiple developers at the same time, we design the directory structure in order to solve the naming conflicts of file. The configuration of HDFS directory is shown in Figure4.

The project files are stored in the namenode of Hadoop, the directory is called "execjar". The "execjar" directory locates below the "/home/USERID" directory of NameNode, and USERID is an inside identifier when a developer log in online IDEMP. The corresponding data sets are uploaded and stored below the "/dataset/USERID/" directory on the HDFS system. When a developer debugs and runs the program, he can create different data sets as required. On the contrary, the system administrator provides a default data set for testing program. The result files will be output to the "output" directory of the HDFS system.

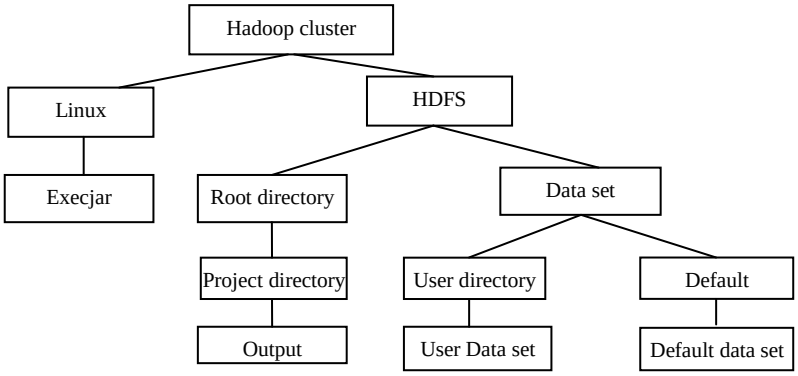

Figure 4. Configuration of HDFS Directory

5. Runtime of MapReduce

5.1. ConsoleTask Class

While the jar program is running on the server, if a client does not get feedback messages from the server in time, then developers think that there is something wrong with their program. Due to the need for a "dynamic execution to obtain information" performance, a real-time data communication model to implement "real-time output" messages can be given, and output messages are stored in the ConsoleTask object, as shown in Figure5. ConsoleTask class depends on SSHRunner class, ShellRunner class, JarTools class, HadoopHelper class, and ConsoleRunner interface. They are in com.imut.szd.commom package.

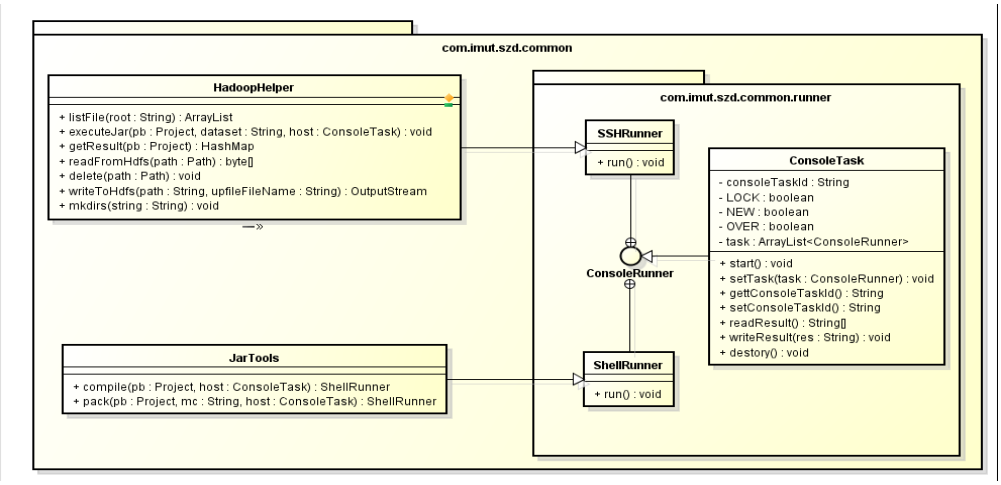

Figure 5. Class Diagram of ConsoleTask

When two or more threads in the system share the same ConsoleTask object, there will be a critical section problem. So, a variable of type Boolean LOCK in the ConsoleTask class is defined. If the data is read, then the LOCK is true; or else the LOCK is false. When the LOCK is accessed, the Synchronized operation of java is used. In addition to the Lock, there are two attributes of OVER and NEW in the ConsoleTask class. If the OVER is true, which illustrates the end of object, the object will be removed. When the NEW is true, the new data in pool arrives and is not read. ConsoleTask also supports multiple task execution sequence. When a task is completed, if there are still tasks in the task list, the next task will continue to go until the task list is null, the OVER of ConsoleTask object will be marked with true.

In the system implementation, the ConsoleTask object is stored in a Session object. Because the Web server maintains and updates a Session object, the client will immediately receive a new message.

5.2. Compiling Procedure

When a developer performs a compile command, online IDEMP compiles the program files with specified version of MapReduce and builds the executable jar file.

Compile command：javac –D java.ext.dirs = [BuildPath] -d [outputpath] [filelist].

BuildPath locate jar files of MapReduce which is configured by the administrator. Outpath is the jar directory in Figure3.

Build a .jar file command：jar cvfe [objectfile] [mainclass] [filelist].

Compiling procedure:

Step1: submit the compile project;

Step2: locate the project path from project table;

Step3: format the compile command;

Step4: instantiate a ConsoleTaske object;

Step5: execute the compile command with Runtime class;

Step6: If the preceding command succeeds then goto Step7; else goto Step8;

Step7: build “projectname.jar” file, goto Step9;

Step8: feedback the error message to the client;

Step9: end.

5.3. Running Procedure

When a developer performs a running command, online IDEMP executes the executable jar file of project.

Executive command: $HADOOP_HOME/bin/hadoop jar [project] [input] [output].

Input is a directory which locates the data set. Output is a directory which stores the result files.

Running procedure:

Step1: upload the project jar file to /home/user/execjar on Hadoop cluster with SSH;

Step2: store the jar file;

Step3: format the executive command;

Step4: instantiate a ConsoleTaske object;

Step5: execute the executive command with SSH;

Step6: display the execution results at client;

Step7: end.

6. Experiment and Evaluation

Online IDEMP consists of four main components: user management, file management, project management and data set management. In order to test the performance of online IDEMP, it is built and deployed on the cloud computing platform which is composed of 15 servers and 13.62T disk array. Online IDEMP is running on 9 virtual machines. Web application server, file server and database server are deployed on the same virtual machine, and Hadoop cluster which includes 1 Master and 7 Slaves is deployed on 8 virtual machines. Table 1 describes the specific configuration of test environment.

Table 1. Configuration of Test Environment

Server	Machine configuration	Number	Software
Web server File server Database server	CPU：4 Core AMD Opteron(tm) Memory：4G,Hard disk：150G Network card：Intel(R)PRO/1000	1	Centos 6.5，JDK1.6.0，Tomcat 7.0.55，MySql 5.5.16
Hadoop cluster		8	Centos6.5，JDK1.6.0，Hadoop-2.2.0，SSH2（JSch 0.1.51）

(1) Software testing pressure

We do an experiment which tests online IDEMP with LoadRunner11 to evaluate the response time and the number of users. The changing process of added users and ended users is shown in Figure6-1. The purple line expresses the changing process of the number of users who have logged in system. The yellow line expresses the changing process of the number of users who have logged out. Figure6-2 shows the changing process of response time and the number of users. The conclusion is the increase of response time with the increase of the number of user, thus the max number of users is 99 in this experiment and the response time is 10ns.

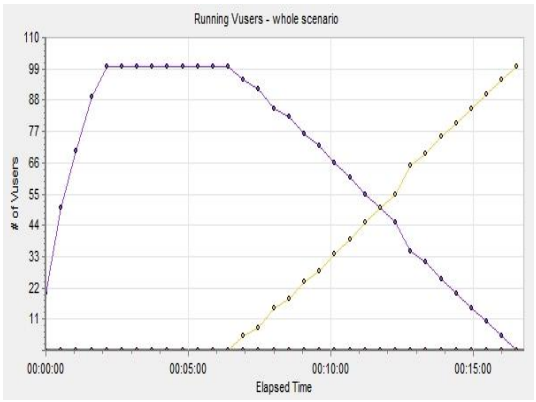

Figure 6-1. Process of Number of Users Change

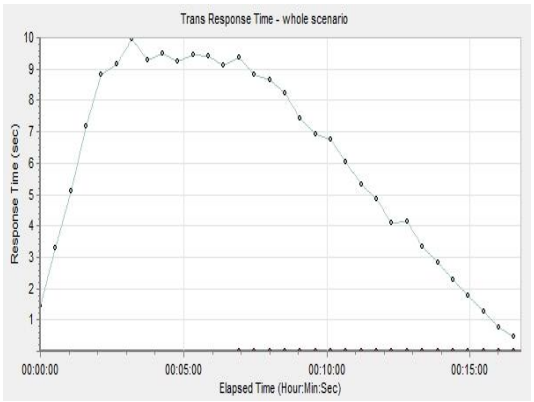

Figure6-2. Process of Response Time Change

Figure 6. Software Testing Pressure

(2) Cluster performance testing

The WordCount is a program of Hadoop that counts words from a file, thus the program which counts words from a data file of 752MB is a baseline task in next experiments. Three experiments which include 10, 20 and 50 parallel tasks are respectively tested on Hadoop cluster, so the test results are shown in Table2.

Table 2. Result of Parallel Task Execution

Execution time（s） / Number of concurrent	Maximal value	Minimum value	Average value
1	152	152	152
10	215	163	194
20	503	114	332
50	981	151	662

7. Conclusion

This paper introduces the current development of MapReduce programming and online IDE, thus we propose to build IDEMP based on Web. The software architecture uses three layers software architecture to improve the reuse of module, expandability and maintenance. Directory hierarchy and configuration of runtime environment solve the problem of naming conflicts. When online IDEMP is implemented, ConsoleTask is a core class which provides synchronization operation and data feedback. Online IDEMP is deployed on the cloud computing platform and is tested, thus it can be used as a tool for MapReduce programming. Since Online IDEMP has good scalability, while a user is not satisfied with the performance, he can increase the number of virtual machine to improve the performance.

Online IDEMP also has some insufficiencies: it only supports Java currently; it does not support collaborative development and breakpoint debugging *etc.*. We will focus on the above problems in the future.

Acknowledgements

Funding project: Inner Mongolia Autonomous Region Natural Sciences Foundation project (2014MS0608). Inner Mongolia Autonomous Region college science research project (NJZY12052). Inner Mongolian university of Technology key Fund (ZD201118).

References

[1] S. Ghemawat, H.Gobioff and S.-T. Leung, "The Google File System", [EB/OL]. http://labs.google.com/papers/gfs.html, **(2014)** November 10.
[2] S. Ghemawat, H. Gobioff and S. T. Leung, "The Google file system", [C]//ACM SIGOPS Operating Systems Review. ACM, vol. 37, no. 5, **(2003)**, pp. 29-43.
[3] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters ",[EB/OL, http://labs.google.com/papers/mapreduce.html, **(2014)** November 10.
[4] T. Gunarathne, T. L. Wu and J. Qiu, "Map Reduce in the Clouds for Science", [C]//Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on. IEEE, **(2010)**, pp. 565-572.
[5] F. Chang, J. Dean and S. Ghemawat, "Bigtable:A Distributed Storage System for Structured Data", [EB/OL], http://labs.google.com/papers/bigtable.html, **(2014)** November 10.
[6] R. Buyya, C. S. Yeo, S. Venugopal, "Cloud computing and emerging It platforms: Vision, hype, and reality for delivering computing as the 5th utility", [J]. Future Generation Computer System, vol. 25, no. 6, **(2009)**, pp. 599-616.
[7] M. Goldman, G. Little and R. C. Miller, "Real-time collaborative coding in a Web IDE", [C]. Proceedings of the 24th annual ACM symposium on User interface software and technology. ACM, **(2011)**, pp. 155-164
[8] M. Nordio, H. Mstler and C. A. Furia, "Collaborative software development on the Web", [EB/OL]. http://arxiv.org/abs/1105.0768.
[9] A. V. Deursen and A. Mesbah, "Cornelissen B. Adinada: a knowledgeable, browser-based IDE", [C]. Proceedings of the 32nd ACM.IEEE International Conference on Software Engineering, vol 2, ACM, **(2010)**, pp. 203-206.
[10] S. Babu, "Towards automatic optimization of MapReduce programs", [C]. In: Kansal A, ed. Proc. of the ACM Symp. on Cloud Computing(SoCC). New York: ACM Press, **(2010)**, pp. 137-142.
[11] T. Condie, N. Conway, P. Alvaro and J. M. Hellerstein, "Online aggregation and continuous query support in MapReduce", [C]. In: Elmagarmid AK, Agrawal D, eds. Proc. of the SIGMOD. Indianapolis: ACM Press, **(2010)**, pp. 1115-1118.

Authors

Zhiqiang Ma (1972-), male (Hui nationality), Inner Mongolia Hohhot people. He received a B.S. degree in computer application technology from Hohai University in China in 1995. He worked in Inner Mongolia University of Technology. He received a M.S. degree in computer application technology from Beijing Information Science & Technology University in 2007. He became associate professor and graduate students advisor in 2010. His research interests include the big data, cloud computation, and machine learning.

Shuangtao Yang (1990-), male (Han nationality), Henan Zhoukou people. He received a B.S. degree in Inner Mongolia University of Technology in China in 2013.He is a graduate student in Inner Mongolia University of Technology. His research interests include the deep learning, cloud computation.

Zhida Shi (1992-) male (Mongolian nationality), Inner Mongolia people. He received a B.S. degree in Inner Mongolia University of Technology in China in 2013. His research interests cloud computation.

Rui Yan (1988-), male (Hannationality), Inner Mongolia Erdos people. He received a B.S. degree in Inner Mongolia University of Technologyin China in 2012.He is a graduate student in Inner Mongolia University of Technology. His research interests include the speech recognition, data mining.