

OPC:An Effective Memory Computing Solution of Big Data For PC

Zhi Yang¹, Yunshan Tang¹, Chunping Zhang¹, Zhiyao Hu¹, Meng Wu¹, Feng Lin¹

¹State Grid Electric Power Science Research Institute ,Nanjing, China
yangzhi1@sgepri.sgcc.com.cn

Abstract— The Big Data computing is one of hot spots of the internet of things and cloud computing. The memory computing is one of key factors of improving performance. Many companies and institutions provide technology and produce. But expensive cost and processing unstructured data are urgent to solve. To deal with the problem, this paper provides an effective memory computing solution with Objectification Parallel Computing (OPC) for PC. In the solution, the unstructured data and structured can be formatted into object. Then the objects are distributed stored in the computer memories and parallel compute to complete tasks. The OPC is applied to the Electric Asset Quality Supervision Manage System (EAQSMS) of State Grid of China, the result shows that with PCs the system is efficiently available, reliable, and flexible expandable.

Keywords—component; Big Data, Distributed Memory Computing, Parallel Computing, Objectification Parallel Computing, Architecture of Objectification Parallel Computing

I. INTRODUCTION

The Big Data[1,2,3,4] computing is one of hot spots of the internet of things and cloud computing , whose research contents are acquisition, management, processing, show, and so on of massive data. In the processing link, the memory computing [5] is one of key factors of improving performance. Many companies and institutions provide technology and produce. But expensive cost and processing unstructured data are urgent to solve.

Currently, in memory computing, using powerful computing, system can satisfy real-time demand and so on. The memory database computing research is the mainstream in Big Data. There are many research results about the question for example [6, 7, 8, and 9]. Database can deal with structured data conveniently. With the help of plugins, the semi-structured data could also be dealt with, for example [6, 7]. The principle is that the semi-structured data is formatted into structured data. Using special structure, the database manager system can read the data. But the unstructured data In addition, SAP HANA [12] provides SAP HANA Database to solve real-time in Big Data. The SAP HANA database is designed from the ground up around the idea that memory is available in abundance, considering that roughly 18 billion gigabytes or 18 exabytes are the theoretical limits of memory capacity for 64-bit systems, and that I/O access to the hard disk is not a constraint. Instead of optimizing I/O hard disk access, SAP HANA optimizes memory access between the CPU cache and main memory. However, the HANA database needs to be fixed in specified device which is expensive.

Although the above methods can improve performance in Big Data, special equipments, for example all-in-one machine [8, 9], are used to deploy software. These equipments are expensive whose price is over millions. So the cost is so high that the application is not wide. In addition, by means of database technology, the semi-structured data could be managed. But the unstructured data cannot be managed. To address the aforementioned problems, using common PC, we provide the Objectification Parallel Computing (OPC) for unstructured and structured data. In this solution, with objectification, the data is formatted into objects. Then the objects are distributed stored in computer memories. Finally, we can complete tasks with the data objects of cluster. The solution is applied to the EAQSMS, which can meet the demands and achieve real-time computing.

This paper is organized as follows. Section 2 states the improved parallel task scheduling. The architecture, meaning, solution of Objectification Parallel Computing are proposed and analyzed in section 3. In addition, the executing procedure of task is introduced in detail. Section 4 verifies this system solution in the EAQSMS. The last section draws to a conclusion.

II. IMPROVED PARALLEL TASK SCHEDULING

A. distributed memory computing

At present, the memory capacity is far less than hard disk capacity. So in order to deal with Big Data, many computer memories compose one cluster to complete a task. The Big Data are distributed loaded in different computer according to some principle. In this case, the distributed memory, as a whole, works to carry out user task efficiently.

In memory computing, SAP HANA [11] is one of typical technologies and produces. The announcement of the SAP HANA® platform[11,13] has created a lot of buzz in the IT and business world. As new business demands challenge the status quo, the scale is larger, expectations are greater, and the stakes are higher. An in-memory approach is the only way to tackle a real-time-data future that includes new data types such as social media monitoring and Web-automated sensors and meter readings.

B. parallel computing

The Apache™ Hadoop® [14] project develops open-source software for reliable, scalable, distributed computing. The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It

is designed to scale up from single servers to thousands of machines, each offering local computation and storage. In hadoop, MapReduce [15, 16] is a distributed programming framework proposed by Google. The framework showed in Figure 1 is similar to the architecture presented in [16, 17], that both use Map and Reduce.

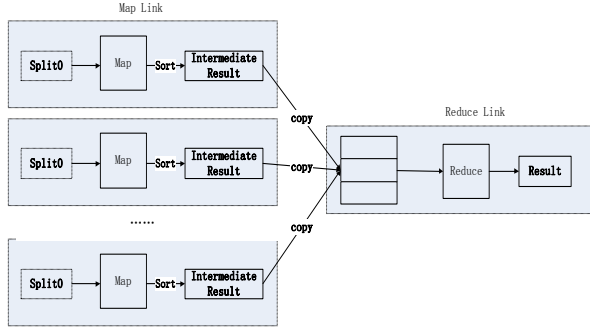


Figure 1. MapReduce Framework.

The core technology of Hadoop is the HDFS [18] (Hadoop's Distributed File System), MapReduce, HBase [19] (Hadoop's database).

From the figure 1, we know that with MapReduce, the task can be split into subtask and the distributed parallel computing can be realized easily.

C. Improved Parallel Task Scheduling

Because that in MapReduce framework and distributed memory computing, there is one node which manages all the tasks, the load of node is heavy. So we improve the task scheduling algorithm. The principle is that the task is managed not by one node, but by all the nodes in the cluster. According to the stored data, all the nodes can be correlated on the logic. The node can receive task and find out whether there are other nodes to complete the task. If yes, the other nodes can complete subtask and send the result to parent node. For example, the nodes of cluster logic relation are shown in figure 2.

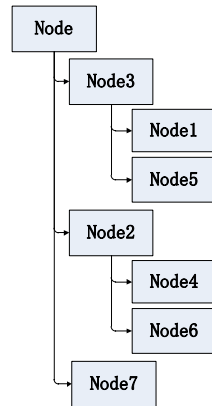


Figure 2. The node logic relation of cluster.

From the figure, the node3 is parent node and its sub-nodes are node1 and node5. The node2 is parent node and its sub-nodes are node4 and node6.

The procedure of task scheduling is shown as following.

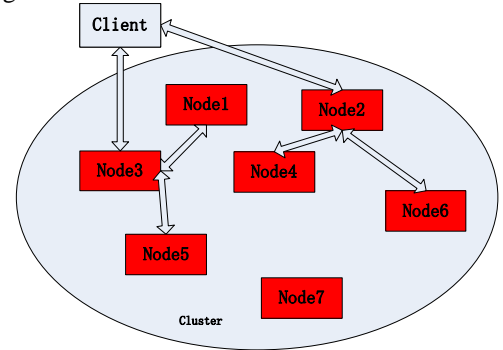


Figure 3. The procedure of task scheduling.

From the figure, the node 3 receives the task and sends subtask to the node1 and node5. The two nodes complete the subtask and return results to node3. Finally, the node3 returns the result to client. In this case, the task load of cluster is light. With reference to the thought, we provide OPC solution. In OPC, firstly, the task is split into subtask. Then the subtask can be done parallel. Finally, the result is return to the client. Due in Big Data, disk I/O bottleneck is one of keys of impacting computational efficiency. The reading and writing of computer memory are faster than disk, so the memory is used to store data. In this way, it can satisfy the real-time demand.

III. OBJECTIFICATION PARALLEL COMPUTING

Reference to the thought of hadoop and realization, fusing objectification thought and using speed advantage of memory computing, we provide Objectification Parallel Computing, whose aim is to improve efficiency of real-time computing in Big Data. In OPC, the data is formatted into object. Then, the objects are distributed stored in the computer memory of cluster. Finally, the objects can be used to execute tasks efficiently.

A. The meaning of OPC

There are three meanings in OPC as follows:

- ◆ Objectification: In order to convenient for operation and management, the objectification thought is introduced. Using it, the unstructured or unorganized data could be formatted into structured object.
- ◆ Memory Computing: The memory computing is faster than disk computing. It is obvious in Big Data. With the advantage, the objects are stored in memory and used to do tasks.
- ◆ Parallel Computing: Although one PC memory is limited, by means of the thought of hadoop, many PCs compose a cluster in which all the memory can be used to do one large task. In this way, the memory quantity is huge. Every PC memory could parallel complete the part of task. Through parallel computing, the task complete efficiently.

B. The architecture of OPC

The architecture of OPC (OPCA) is showed as follow:

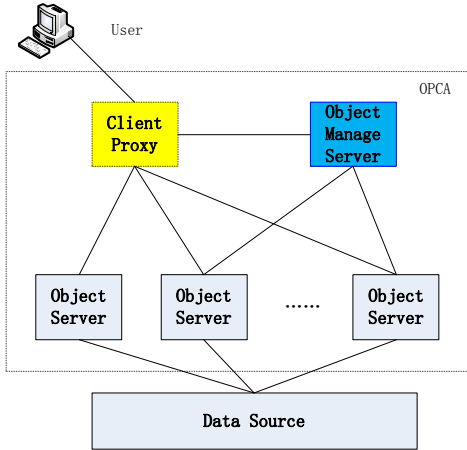


Figure 4. The architecture of OPC.

The OPCA is composed of ClientProxy, ObjectManageServer and ObjectServer.

The functions of each part of OPCA are the following:

- **ClientProxy (CP):** It receives user task. By accessing OMS, the object locations which are used to do the task are obtained. CP gets task result and returns it to user.
- **ObjectManageServer (OMS):** It receives and maintains object index which indicates object location. In addition, it is responsible for data recovery of OS and hot standby.
- **ObjectServer (OS):** From the data source, the data is formatted into object. The objects are stored in the object pool in OS. There are two kinds of objects in the object pool: data object and computing data. OS manages and maintains object pools and objects. The locations are termly sent to OMS.

C. The solution of OPC

There are three steps in the solution of OPC. The first step is that according to some principle, the data is formatted into object. The second step is that referring to memory quantity of every PC, the objects are distributed stored in the memory. The third step is that every OS sends object index to OMS and OMS creates it. After the step, the cluster can execute user task. The procedure is shown as Figure 5.

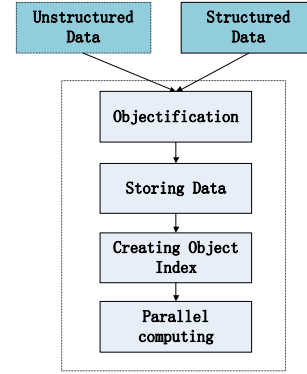


Figure 5. The procedure of OPC.

The detailed procedure is as follows:

The first step: objectification. According to business, the unstructured and structured data formatted into different data objects. There are two kinds of data object. The first kind is single object, in which, there is no relation between objects. The second kind is object set, in which, there is relation between objects. By means of more one kind object, the task can be completed. In this case, one kind object cannot do the task. So the objectification principle is:

- ✧ Business relation
- ✧ Coupling low
- ✧ Computing convenient

Under the principle, as far as possible, we format the data into single object not object set.

The second step: storing object. Because every computer memory quantity is not the same, according to the under principle, the objects are distributed stored to different PC memory. The distributed principle is:

- ✧ Computer quantity of the cluster
- ✧ Computer memory quantity
- ✧ Quantity of single object or object set

Note: all the objects in the object set must be stored in the same computer memory.

The third step: creating object index. Every computer loads object from data source. After OS loads objects, the object index is sent to the OMS. OMS creates object index of the whole cluster.

The forth step: parallel computing. The whole cluster is ready for executing task. After user submits tasks, the computers of the whole cluster can receive and parallel compute them

D. Executing procedure of task

Firstly, User starts a task. The CP receives the task and obtains the object index from OMS and send task to responsive OS. In order to know the object index of subtask, OS gets the object index from OMS. If there are subtasks, they are sent to responsive OS. Otherwise, the task is completed in local host. After subtasks are completed, the results are returned to parent task. Finally, the result is returned to user.

For example, the whole process to complete a task is shown as following.

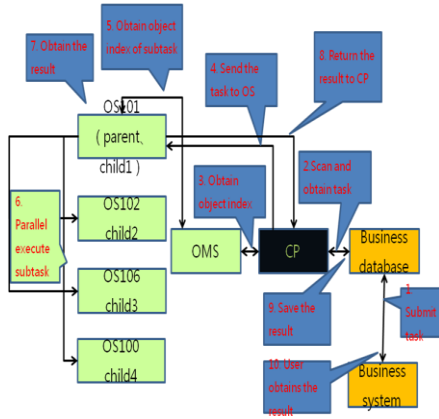


Figure 6. The whole process of task.

In the figure 6, the parent object and child1 object are stored in OS101. The child2 object is stored in OS102. The child3 object is stored in OS106. The child4 object is stored in OS100.

The detailed process is:

- 1 By means of business system, user submits task. The task is stored in the business database.
- 2 CP scans the database and gets the task.
- 3 Task object is created and through object index of OMS, CP obtains OS (101) location.
- 4 CP sends the task object to the OS (101).
- 5 Through object index of OMS, subtask location list is obtained. The list is {OS (101), OS (100), OS (102), and OS (106)}.
- 6 All the subtasks are parallel completed and the results are returned to parent task OS (101).
- 7 OS (101) obtain the result.
- 8 OS (101) sends the result to CP.
- 9 CP stores the result into business database.
- 10 Using the business system, user can obtain the result.

Solution analysis

The detailed analysis is:

- Objectification, parallel computing and memory computing are fused into this solution. In this way, the solution adequately takes the advantage of them and improves computing efficiency obviously.
- The unstructured and structured data can be formatted into object, so the solution can parallel compute with them.
- The data stored location changes from disk to memory, so the computing efficiency is improving greatly. The objects are distributed stored in different computer memory, so in one computer, the object quantity is not huge. Every task can be completed very quickly. The response of whole task is real-time.
- The requirement for computer of the cluster is low, such as common PC. So long as there are CPU, memory, disk, network card in the computer, it can be added to the cluster.
- The computer quantity of cluster is not limited, so we can extend cluster infinitely to store large data.

In this case, the computer efficiency does not reduce obviously. In addition, when there is exception, such as power cutting, the objects are lost. In this case, we need reload the objects. After this, the cluster is working normally and reliably.

IV. EXPERIMENT AND ANALYSIS

The above solution is applied to the Electric Asset Quality Supervision Manage System (EAQSMS) of State Grid of China. Because the EAQSMS is deployed in the head quarters of State Grid of China, the data is stored in the Oracle database which is Big Data. The record quantity is about 100 hundred million. The business logic relation is complicated. In this case, the computing time is about several hours which cannot satisfy real-time demand which is that the computing time is about 10 second.

With the help of open code of hadoop, the OPC system is developed in Eclipse [20] environment with hadoop0.20.2 [21]. There are three softwares in the system: ObjectManageServerSoft (OMSS), ObjectServerSoft (OSS), and ClientProxySoft (CPS). The functions of these softwares are the same as the part of OPCA.

A. Preprocess stage

Four computers, which compose a cluster, are used to verify the solution. In the cluster, the installing softwares are OMSS, OSS, and CPS. Note: there is one OMSS and CPS in the cluster. Because the computer quantity is limited, the part of data is selected to verify our solution. Selected data is register and running data, about 20 million, which is the data of 15 months. The scope is all the provinces and regions of State Grid of China.

The cluster configuration is:

Table1. The configuration of cluster.

No	Computer configuration	Network configuration	Software configuration
1	Double core: 2.8GHz , Memory:8G	Network card:100M, IP: 192.168.42.100	linuxOS ; ObjectManageServer ; ObjectServer
2	Double core: 3.2GHz , Memory:16G	Network card:100M, IP: 192.168.42.106	linuxOS; ObjectServer
3	Double core: 1.6GHz , Memory:4G	Network card:100M, IP: 192.168.42.102	linuxOS ; apador ; ObjectServer
4	Double core: 3.0GHz , Memory:4G	Network card:100M, IP: 192.168.42.101	Oracle database ; windowsXPOS ; ObjectServer

The deployed data is :

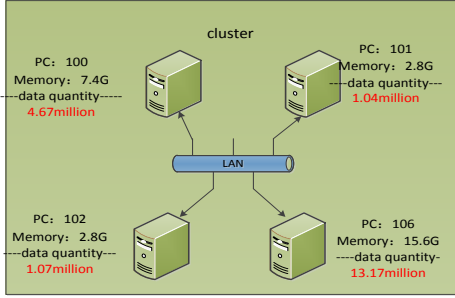


Figure 7.Data deployed in the cluster.

B. Objectification

According to the objectification principle, the business data is formatted into six kinds of object.

Table2. Six kinds of object.

No	object
1	gd_zc_ZYXDSJ
2	gd_zc_ZYYHSJ
3	gd_yx_ZYTDSJ
4	gd_yx_ZYTDSJ_ky
5	gd_yx_zytdyh
6	gd_yx_zytdyh_ky

Because the task is completed with six kinds of object, the six kinds of object compose one object set.

C. Storing object

According to memory distributed principle and practical experience, the formula is:

$$M = \begin{cases} \min(mq \times 80\%, 1.4) & 32os \\ \max(mq \times 80\%, mq - 1.5) & 64os \end{cases} \quad (1)$$

In the formula, mq indicates physical memory of computer. The number unit is GB. M indicates configured memory number. 32os and 64os indicate 32 bit operating system and 64 bit operating system.

Using the formula, the configured memory is obtained shown as follows:

Table3. The result of memory distributed.

No	IP	Memory (G)	JVM (M)
1	192.168.42.100	7.4	5200
2	192.168.42.101	2.8	1350
3	192.168.42.102	2.8	2048
4	192.168.42.106	15.6	14240

D. Creating object index

After all the OSS loads object set and send object index to OMSS, the object index is created shown as follow:

=====1		01	192.168.42.106	2012-1-1	0:00:00
	2	0130		2012-1-1	0:00:00
		3	013032	192.168.42.106	2012-1-1 0:00:00
		3	013035	192.168.42.106	2012-1-1 0:00:00
		3	013034	192.168.42.106	2012-1-1 0:00:00
		3	013037	192.168.42.106	2012-1-1 0:00:00
		3	013030	192.168.42.106	2012-1-1 0:00:00
		3	013033	192.168.42.106	2012-1-1 0:00:00
	2	0140		2012-1-1	0:00:00
		3	014043	192.168.42.100	2012-1-1 0:00:00
		3	014042	192.168.42.100	2012-1-1 0:00:00
		3	014047	192.168.42.100	2012-1-1 0:00:00
		3	014044	192.168.42.100	2012-1-1 0:00:00
		3	014046	192.168.42.100	2012-1-1 0:00:00
		3	014040	192.168.42.100	2012-1-1 0:00:00
		3	014041	192.168.42.100	2012-1-1 0:00:00
	2	0160		2012-1-1	0:00:00
		3	016062	192.168.42.102	2012-1-1 0:00:00
		3	016063	192.168.42.102	2012-1-1 0:00:00
		3	016064	192.168.42.102	2012-1-1 0:00:00
		3	016060	192.168.42.106	2012-1-1 0:00:00
		3	016065	192.168.42.106	2012-1-1 0:00:00
		3	016061	192.168.42.106	2012-1-1 0:00:00
	2	0120		2012-1-1	0:00:00
		3	012022	192.168.42.106	2012-1-1 0:00:00
		3	012023	192.168.42.106	2012-1-1 0:00:00
		3	012024	192.168.42.106	2012-1-1 0:00:00
		3	012020	192.168.42.106	2012-1-1 0:00:00
		3	012021	192.168.42.106	2012-1-1 0:00:00
	2	0110		2012-1-1	0:00:00
		3	011011	192.168.42.106	2012-1-1 0:00:00
		3	011010	192.168.42.106	2012-1-1 0:00:00
		3	011017	192.168.42.102	2012-1-1 0:00:00
		3	011016	192.168.42.106	2012-1-1 0:00:00
		3	011014	192.168.42.106	2012-1-1 0:00:00
		3	011013	192.168.42.106	2012-1-1 0:00:00

Figure8. Object index.

Using the above object index, the computers which execute task can be found.

E. Test result

In the system, many tests are done to verify the availability, reliability, expansibility, and so on of system. The test result is shown as following.

Execution time test result

In cluster, the most complicated task is about 6.8 second shown as the left of figure 9 which is about several hours in Oracle database using stored procedure.

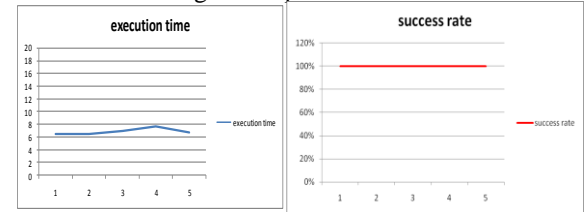


Figure 9. The execution time and success rate of task.

The ordinate indicates the execution time of task. The abscissa indicates the running time.

From the left of figure 9, we can see that the execution time is from 6 to 8 second. The least execution time is 6.2 second and the most execution time is 7.7 second. It suggests that the task can be completed efficiently. It satisfies the requirement and real-time.

Reliability test result

In case of different number of concurrent task, more than 20 thousand tasks are executed. When concurrent task testing, more tasks run uninterruptedly at the same time. The success rate is computed by the formula:

$$sr = \frac{snum}{num} \quad (2)$$

In the formula, sr indicates success rate. snum indicates the task success time. num indicates the total task time.

From the right of figure 9, we can see that in different concurrent task number, the success rate is 100%. In concurrent task number 10, the success rate is 100%. It indicates that the system works well in different concurrent

task number. OPC system is stable and reliable in the case of concurrent task.

Expansibility test result

In case of different computer number of a cluster, the expansibility of system is:

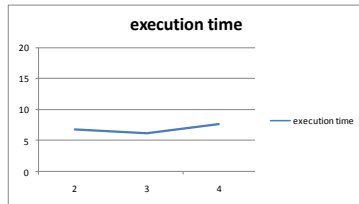


Figure 10. Expansibility of system.

The ordinate indicates the execution time. The abscissa indicates the number of computer number of a cluster.

When expansibility testing, the computer number of cluster changes from 2 to 4. In different computer number, the task is executed and the execution time is 6-8 second. From the figure, we can see that with the increasing of computer number, the execution time of task does not increase.

Analysis

- Although selected data is not huge, the cluster is extendable. As long as there are enough computers to store the objects, the solution is available and reliable.
- Comparing with the result of executing stored procedure of database, the computing result of system is the same. It indicates the system is available. In case of concurrent task, the executing result is right. It indicates the system is reliable.
- The executing time of the most complicated task is from several hours to about 10second. The efficiency is improved very obviously.
- The expansibility of system is flexible. With the increasing of data, it is convenient to increase computer to the cluster. In this case, the computing efficiency does not decrease obviously.

V. CONCLUSION

In case of Big Data, the memory computing is one of key factors of improving performance. But expensive cost and processing unstructured data are urgent to solve. To deal with the problem, this paper provides an effective memory computing solution with Objectification Parallel Computing (OPC) for PC. In the solution, the unstructured data and structured can be formatted into object. Then the objects are distributed stored in the computer memories and parallel compute to complete tasks. The OPC is applied to the Electric Asset Quality Supervision Manage System (EAQSMS) of State Grid of China, the result shows that with PCs the system is efficiently available, reliable, and flexible expansible

Although our solution can solve real-time of Big Data, in our OPC solution, there are some questions to research and solve, for example, data compression, monitoring the cluster, hot standby, and so on. These issues need further research. At present, the technology development changes quickly. We need continuously focus on new technology to solve the issue.

REFERENCES

- [1] Li Guojie, Cheng Xueqi .Research Status and Scientific Thinking of Big Data.Bulletin of Chinese Academy of Sciences,Vol. 06,2012,pp:647-657
- [2] Hunter, P..Journey to the centre of big data.Engineering & Technology,Volume: 8 , Issue: 3 ,2013,pp:56-59
- [3] Garlasu, D. ; Sandulescu, V. ; Halcu, I. ; Neculoiu, G. ; Grigoriu, O. ; Marinescu, M. ; Marinescu, V..A big data implementation based on Grid computing .Roedunet International Conference (RoEduNet), 2013 11th ,pp:1-4
- [4] Wang Shan, Wang Hui Ju,Qin Xiong-Pai,Zhou Xuan.Architecting Big Data: Challenges, Studies and Forecasts.CHINESE JOURNAL OF COMPUTERS,Vol. 34 No. 10 ,pp:1741-1752
- [5] Xi Xiaoyin, He Zhongjiang, Wang Dan.Research on Construction of the Telecom Industry's Data Warehouse System Based on Memory Computing Technology.Telecommunications Science, 2012,vol 2,pp: 18-21
- [6] deng bo.study of secondary development based on Oracle ERP system plugins.university of electronic science and technology of china.engineering master dissertation,2007
- [7] Run-yang Zhong Qing-yun Dai Ke Zhou.Realization of Program Based on Plug-Universal Database-aided Design.Proceedings of the 10th guangdong-Hongkong Mechanical and Electrical Engineering Technology and Application Seminal&Professor Liang-Tianpei commemoration.2008,pp:17-20
- [8] Yuqing Xiong.A Formal Definition of Logic Topology for All-to-One Reduces in Distributed Memory Parallel Computing.Intelligent Human-Machine Systems and Cybernetics, 2009. IHMSC '09. International Conference on,pp: 474 – 476
- [9] Exadata <http://www.oracle.com/cn/products/database/exadata>
- [10] HANA,<http://www.saphana.com/welcome>
- [11] SAP HANA® Platform – Technical Overview,<http://www.saphana.com/docs/DOC-1506>
- [12] Hana,<http://www.saphana.com/community/blogs/blog/2013/06/26/android-apps-on-sap-hana-one>
- [13] Lee, Juchang ; Kwon, Yong Sik ; Farber, Franz ; Muehle, Michael ; Lee, Chulwon ; Bensberg, Christian ; Lee, Joo Yeon ; Lee, Arthur H. ; Lehner, Wolfgang.SAP HANA distributed in-memory database system: Transaction, session, and metadata management.Data Engineering (ICDE), 2013 IEEE 29th International Conference on,pp:1165 - 1173
- [14] hadoop,<http://hadoop.apache.org/>
- [15] DEAN, J., AND GHEMAWAT, S. MapReduce: Simplified data processing on large clusters. In Proc. of the 6th OSDI (Dec. 2004), pp.137.150.

- [16] Jing Zhao ; Ruisheng Zhang ; Zhili Zhao ; Dianwei Chen ; Lujie Hou. Hadoop MapReduce Framework to Implement Molecular Docking of Large-Scale Virtual Screening. Services Computing Conference (APSCC, 2012) ,pp: 350 - 353
- [17] Tom White. hadoop the definitive guide. Tsinghua University press.2010
- [18] Dhruba Borthakur, HDFS Architecture.The Apache Software Foundation, 2008.
- [19] Ankur Khetrapal, Vinay Ganesh, HBase and Hypertable for large scale distributed storage systems. Computer Science, 2006.
- [20] Eclipse, <http://www.eclipse.org/>
- [21] hadoop0.20.2, <http://hadoop.apache.org/releases.html#Download>