International Journal of New Developments in Engineering and Society
ISSN 2522-3488 Vol. 4, Issue 1: 101-110, DOI: 10.25236/IJNDES.040114

Computer Data Processing Model in Big Data Era

Beibei Sun

Zibo Vocational Institute, Zibo Shandong 255000, China

ABSTRACT. *Traditional computer data processing mode cannot operate efficiently, and is inefficient in processing computer data in big data era. Therefore, the research on computer data processing model in big data era is carried out. This paper plans the overall architecture of computer data processing mode. On the basis of the overall architecture, big data processing mode is divided into three modes: offline batch data processing, query data processing and real-time data processing. Detailed design the core functions of the three modes, implement computer data processing in big data environment, complete the design of computer data processing mode in big data era. The experiment data proved its efficiency, the designed computer data processing mode in big data era is more efficient than the traditional data processing mode, The processing efficiency of this mode increased by 45%.*

KEYWORDS: *Big data era, Computers, Data processing, Processing mode*

1. Introduction

In recent years, with the continuous popularization of computer technology and network technology, the emergence of e-commerce, social networking and Internet has changed people's daily life to a great extent, and it also brings large amount of data, people have entered the era of big data[1]. Literally, big data is a huge amount of data, it's not only just include the huge amount of information, but also include the complexity of data information, the diversification of information and the duplication of data. With the rapid and continuous development of virtual data in contemporary society, big data is the inevitable outcome of today's society. Compared with traditional calculation, big data has many advantages. For example, large data has a low cost, high ratio of resource utilization rate, large scale, fast speed and so on. In the era of big data, conventional computer information processing technology has been unable to cope with the demand for massive data processing. Big data brings great challenges to computer information processing technology, people need to deal with big data problems with new perspectives and innovative ideas. The traditional computer data processing mode is difficult to adapt to the reality needs in the era of big data. When it comes to massive amounts of data, this data processing mode is not only difficult to operate normally and effectively,

ISSN 2522-3488 Vol. 4, Issue 1: 101-110, DOI: 10.25236/IJNDES.040114

but also may cause the computer crashed and paralyzed, resulting in more serious risks and losses[2]. Therefore, studying the computer data processing model in the big data era, developing and improving the old problems, has the far-reaching practical significance.

To solve the above problem, studying the computer data processing model in big data era. This paper plans the overall architecture of computer data processing mode. On the basis of the overall architecture, big data processing mode is divided into three modes: offline batch data processing, query data processing and real-time data processing. Detailed design the core functions of the three modes, implement computer data processing in big data environment, complete the design of computer data processing mode in big data era. In order to ensure the effectiveness of the design, design the experiments to test methods. The experiment data proved its efficiency, compared with the traditional data processing mode, the computer data processing mode designed in this paper is more efficient.

2. Design of Computer Data Processing Mode in Big Data Era

2.1 Overall Architecture of Computer Data Processing Mode

To understand the overall framework of big data processing, comparing the software architecture of big data processing mode with the traditional software architecture of single machine processing mode[4]. The comparison diagram is shown in figure 1.

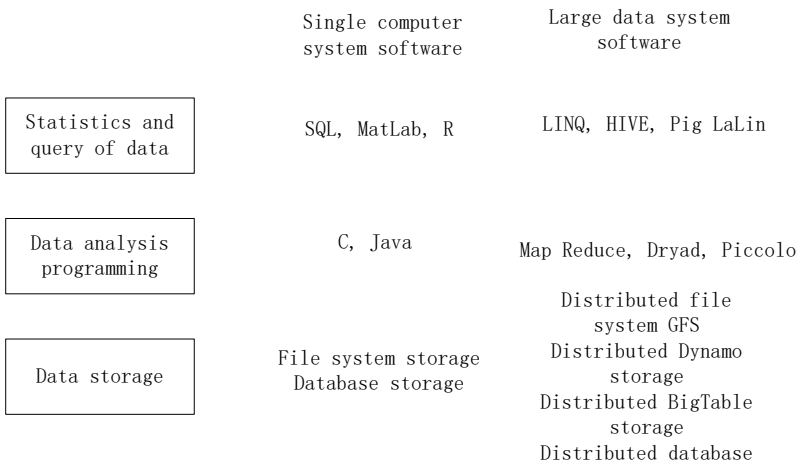

Fig.1 Software Architecture Contrast Diagram

As we can see from Figure 1, the architecture of big data processing is an

ISSN 2522-3488 Vol. 4, Issue 1: 101-110, DOI: 10.25236/IJNDES.040114

extension of single system software architecture. It extend the level of single software architecture to clusters with hundreds or even thousands of nodes[3]. On the basis of the overall architecture of computer data processing mode, the mode of big data processing is divided into three modes according to the time characteristics of data processing: offline batch data processing, query data processing and real-time data processing. The processing time required for these three modes is more than hour level, seconds to minutes, and real-time level. For the above three big data processing modes, there is not much difference in the overall system structure of big data processing. Therefore, the above system structure is used uniformly, and the differences are described in detail.

2.2 Detailed Design of Data Processing Mode

2.2.1 Detailed Design of Offline Batch Data Processing

Based on the overall system structure, detailed designing the offline batch data processing. In offline batch data processing, the core problem is the data storage. The offline storage of data is further divided into the storage of key value pairs and the storage of database model. In local storage, key-value pairs are stored in three main structures: hash table structure, log-structure and sequence table structure. The advantages and disadvantages of these three structures are as follows. For the storage of key value pairs, the most intuitive storage structure is hash table structure. The open addressing method is used to process and calculate the hash table. The formula is as follows:

$$Hi = \left[H\left(key\right) + di \right] \times m \quad (1)$$

$H\left(key\right)$ is the hash function, m is the hash table length, di is the incremental sequence. The data entry operation and data query operation of hash table structure are better than others, but this structure destroys the locality of data storage. When the data is saved, there will be a lot of random read and write operations, greatly reduced disk performance. Therefore, this data structure is suitable for setting up key value pairs in memory. If disk operations are involved, this structure requires some optimization. Another serious problem is that hash tables structure cannot directly support range query operations. Because in essence, the hash table structure is not an orderly structure, it requires special processing to complete the range query[5]. In order to support range queries, a series of sequence table structures are generated. It includes the way of simple sorting array and B tree, B+ tree and so on. The sequence table structure can be used to query the scope well, but the process of data insertion and the process of obtaining are complex, and the disk efficiency is not very high for direct use. The most efficient way to disk is to make full use of the sequential read and write characteristics, to avoid random reading and writing. Therefore, accessing data through logs is the most effective way to exploit disk performance. In addition to hash tables and sorting tables, log storage structure is an important supplement in key value pairs storage, it is mainly used for

ISSN 2522-3488 Vol. 4, Issue 1: 101-110, DOI: 10.25236/IJNDES.040114

real data storage, the former two are used as index structures for log storage.

The core contents of database storage include the data model stored in database and the consistency of database operation[6]. The former defines the concrete format of data storage in the database, and the latter explains how to maintain the correctness of the database under concurrent conditions. The biggest difference between database storage and file storage is that database storage is not completely transparent to users. Users need to set up a data storage model, which is no exception for distributed data in big data processing. Such a storage mode provides dynamically growing data columns, and can store multiple versions of data units. For example, the data model of Big-Table is shown in Figure 2.

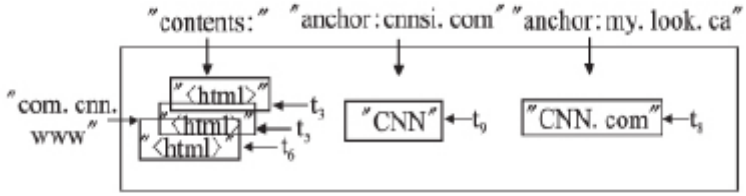

Fig.2 Data Model of Big-Table

Through the above design, offline batch data processing is completed.

2.2.2 Detailed Design of Query Data Processing

In this paper, the Dryad programming model is used to detailed design the query data processing based on the overall system structure. The Dryad programming model is used to translate a query language similar to SQL into an intermediate execution process, and execute in the distributed execution environment and gets the results. The process is shown as Figure 3.

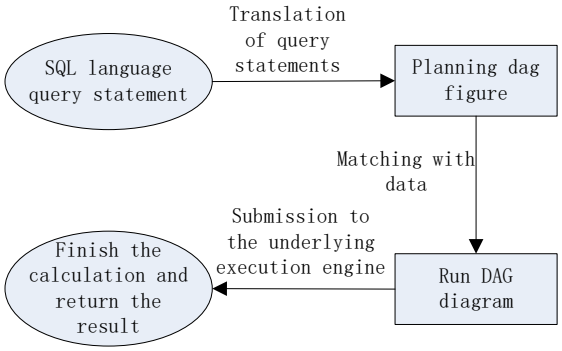

Fig.3 The Execution Flow of Sql Statements in a Distributed Environment

ISSN 2522-3488 Vol. 4, Issue 1: 101-110, DOI: 10.25236/IJNDES.040114

Dryad is built on the Microsoft cluster architecture, and encapsulates a number of advanced languages API, a typical example is DryadLINQ[7]. Dryad constructs the task as the set of points and edges, the whole task is made up of a DAG (Directed Acyclic Graph) diagram. The structure is shown in Figure 4.

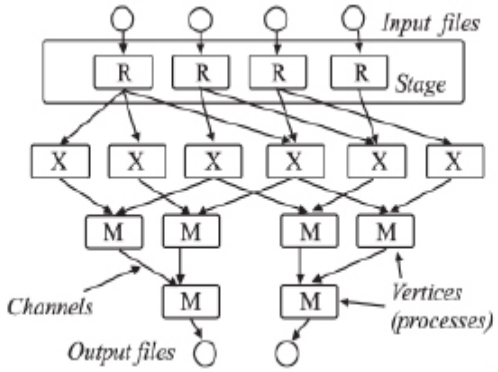

Fig.4 Dryad Program Running Structure

As we can see from Figure 4, a large task will be decomposed into multiple steps, each step is a node in the task DAG diagram. Each node in the task DAG diagram can be programmed by a programmer and then put into a large cluster [8]. The directed edges in DAG diagram are one-way paths for data transmission. Data can be transferred through memory, and can also be transmitted through network channels. The actual situation is decided by actual execution. Because the size of the data needed is huge, the data of the task node is far more than the number of computing nodes. The dispatcher can make full use of this feature to achieve high scalability, and at the same time accomplish the task's fault tolerance by restarting execution.

Through the above design, the query data processing is completed

2.2.3 Detailed Design of Real-Time Data Processing

Based on the overall system structure, detailed designing the real-time data processing. Setting up a suitable programming framework, decoupling the system from application to adapt to the needs of similar application scenarios. The S4 system is used for real-time data processing. Its logical processing node is shown in Figure 5.

ISSN 2522-3488 Vol. 4, Issue 1: 101-110, DOI: 10.25236/IJNDES.040114

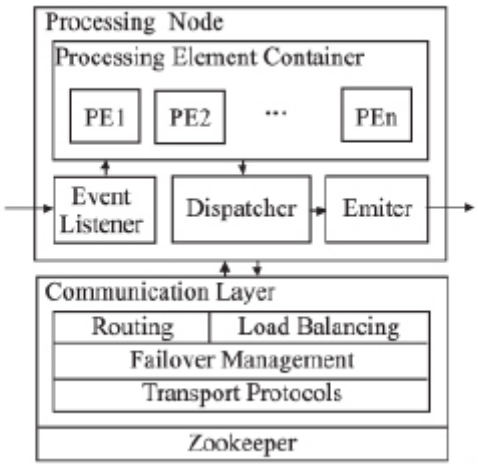

Fig.5 Logical Processing Node of the S4 System

Logical processing nodes are the basic processing units in S4, including processing nodes and communication layers. Processing the user's data in the processing node. The data to be processed is similar to the form of Key Value data, including the type of event, the key values and data contained in the event. This part is called the processing elements in S4. The processing unit encodes the functionality, types of e-vents, key of attributes, value of attributes. All in all, these are the data that need to be processed, and the data will be processed in different processing nodes. And if it needs to be processed after processing, a new processing unit will be generated and processed by the new processing unit. The underlying communication layer performs data unit routing[9].

Through the above design, real-time data processing is completed.

2.3 Implement Computer Data Processing

This paper plans the overall architecture of computer data processing mode. On the basis of the overall architecture, big data processing mode is divided into three modes: offline batch data processing, query data processing and real-time data processing. Detailed design the core functions of the three modes, implement computer data processing in big data environment, complete the design of computer data processing mode in big data era.

3. Simulation Experiment

In order to ensure the effectiveness of computer data processing mode in era of

ISSN 2522-3488 Vol. 4, Issue 1: 101-110, DOI: 10.25236/IJNDES.040114

big data that was designed in this paper, simulation experiments are designed. In the process of the experiment, a large amount of data in the background of a website is taken as the experimental object. Computer data processing for these kinds of complex and numerous computer data. Compare the traditional computer data processing mode and the computer data processing mode in big data era that was designed in this paper, and observe the experimental results. Raw data information goes from data sources to data processing centers through the network. The data processing center uses the computer data processing mode to process the raw data information. The processed data processing results are sent to computer monitors or related clients[10]. The process is shown in Figure 6.

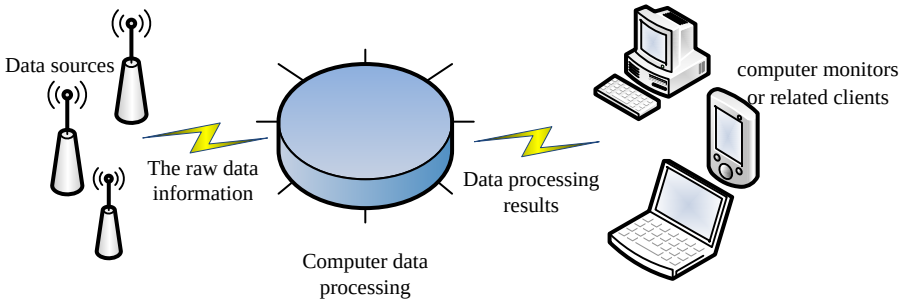

Fig.6 Computer Data Processing

3.1 Data Preparation

In order to ensure the accuracy of the experimental results, the experimental variables are controlled. This paper tests the processing efficiency of computer data processing mode in big data era. Because of the different speed of data transmission in different system environment, the degree of difficulty of the computer data processing mode is different. Therefore, it is necessary to control the other variables except the processing mode to avoid interfering with the accuracy of the experimental results. The experimental variable control setting in this article is shown in Table 1.

Table 1 Setting of Experimental Environment

Function	Technical requirement
Long-range data communication function	GPRS,Ethernet,ASDL and so on
Data input function	Serial digital input channel, Resolution analog input channel, Switch input channel

ISSN 2522-3488 Vol. 4, Issue 1: 101-110, DOI: 10.25236/IJNDES.040114

Display function	Display screen
Power supply	Use 220V AC power, equipped with battery or uninterrupted power supply
Related protocol functions	Defining, importing, receiving and storing. Heat treating.
Hardware environment	Intel P4 2G processor
Ethernet controller	Cygnal CP2200
Interface mode	BNC
Cable requirements	Support 8 wire twisted pair wire Coaxial cable
CPU	Pentium 4 and above

3.2 Comparison Test Results

In the test process, different computer data processing modes are used to process computer data in the case of big data. As the experimental time increases, the amount of information to be processed will be increased. Recording the amount and time of processing required for the computer data processing of the two modes. Calculating its working efficiency and drawing a comparison diagram. The comparison results are shown in figure 7.

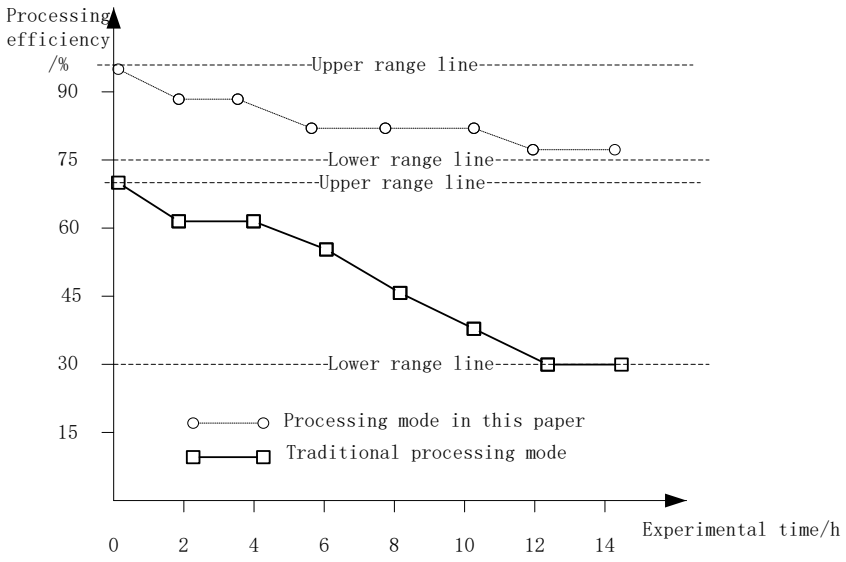

Fig.6 Comparison Results of the Test

ISSN 2522-3488 Vol. 4, Issue 1: 101-110, DOI: 10.25236/IJNDES.040114

As can be seen from figure 7, compared with traditional methods, the computer data processing model in big data era designed in this paper is more efficient in processing the same amount of computer data in the case of big data. In the process of comparative test. Continuous observation of 14h data processing efficiency of design modes in the case of increased data processing quantity. The efficiency of data processing using the design modes of this paper has always been an obvious advantage. Using this design modes for data processing, the data processing efficiency has been maintained between 75% and 95%. Although the increases of the amount of data affects efficiency, the decrease is small and the relative change is stable. When using traditional methods for data processing, the processing efficiency dropped from 73% to 30%. It shows that when the amount of data increases, its effectiveness is extremely unstable, and it is difficult to adapt to the actual needs. It can be proved by the above data, compared with traditional mode, using the mode that was designed in this paper to process computer data in big data era, the processing efficiency can be increased by 45%. Therefore, the computer data processing mode in big data era designed in this paper is very effective.

4. Conclusions

The computer data processing mode designed in this paper can efficiently process large amounts of data in computers, and it is highly effective.

References

[1] CHEN Wen, PU Qingping, ZOU Fangming (2017). Transformation and coping strategies of university students' educational management mode in the era of big data. Jiangsu Higher Education, vol.19, no.1, pp.67-69.
[2] SUN Rui (2017). Virtual Computer Data Storage Space Stability Optimization Simulation. Computer Simulation, vol.234, no.9, pp.345-348.
[3] SONG Jie, SUN Zongzhe, MAO Keming (2017). Research Advance on MapReduce Based Big Data Processing Platforms and Algorithms. Journal of Software, vol.28, no.3, pp.514-543.
[4] WEI Wei, JIANG Dejun, XIONG Jin (2017). Study of the performance of in-memory key-value stores with non-volatile memory. Chinese High Technology Letters, vol.27, no.6, pp.519-529.
[5] QIN Yi, YANG Yun, MIN Yujuan (2018). IPv6 phased routing lookup algorithm combining hash table and multi bit Trie. Minicomputer system, vol.39, no.5, pp.66-71.
[6] GAO Guangjun, SUN Lingjie, LI Zili (2017). Research on Storage and Updating Mechanism of Real Estate Cadastral Database Based On “Multi-standard and Multi-source” Data. Modern Surveying and Mapping, vol.40, no.2, pp.10-12.
[7] KANG Yanli, LI Feng, WANG Lei (2017). Incremental Optimization Method for Periodic Query in Data Warehouse. Journal of Software, vol.28. no.8, pp.2126-2147.

ISSN 2522-3488 Vol. 4, Issue 1: 101-110, DOI: 10.25236/IJNDES.040114

[8] LIANG Yuxuan, QI Xin, HAN Junnan (2017). Research on pre stack depth migration parallel scheduling strategy in virtual computing environment. Journal of Shengli College China University of Petroleum, vol.31, no.2, pp.42-44.
[9] WU Haiqin, WANG Liangmin (2017). Research on optimal support tree for Top-k query in wireless sensor networks based on connected dominating set. Chinese Journal of Electronics, vol.45, no.1, pp.119-127.
[10] ZHOU Biao, LI Qiao, ZHOU Xiaohang (2017). A data processing method for bridge modal parameter identification based on Exploratory Data Analysis. Sichuan Building Science, vol.43, no.2, pp.33-37.