

METRICS FOR DATA-DRIVEN ENERGY EFFICIENCY

A THESIS SUBMITTED TO AUCKLAND UNIVERSITY OF TECHNOLOGY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF COMPUTER AND INFORMATION SCIENCES

June 2020

By

Shuang Song

School of Engineering, Computer and Mathematical Sciences

Abstract

Currently, the world is witnessing a mounting avalanche of data due to the tremendous growth of the Information and Communications Technologies (ICT). This trend is continuing to develop in a quick and diverse manner in the form of big data, which is emerging as one of the most powerful technological drivers to improve productivity and support innovation for humanity. But it also gives a non-negligible contribution to world electricity consumption and carbon dioxide (CO_2) footprint as well as their consequences on climate change, which are urgently calling for energy efficient solutions. Lots of research and development efforts have emphasized on studying energy efficiency metrics, because these metrics are measures and indicators of energy efficiency. Understanding those metrics provides us a better view on how energy efficiency can be achieved at every corner, e.g., process, component, equipment, service, application and network/system level, of an ICT system/network.

From our observation, the energy efficiency metrics in ICT area are conventionally introduced according to the physical-thermodynamic definition, and the measure of the ICT-based output in physical unit is the number of bits of the data sequence. The problem emerges when using physical measurements of data sequence, because it only measures the quantity of bits, and does not necessarily factor in data quality considerations. In other words, the metric is not making any distinction between low and higher quality data sequences. From this basis, it could consequently argue that the data, when measured in physical amount, cannot be added up or compared

because it has different qualities. The data quality ignorance is therefore a fundamental problem in constructing conceptually sound ICT related energy efficiency metrics. This insight led to the new development of data quality-aware energy efficiency metrics for more efficient network/system approaches and mechanisms which can be reconfigured depending on the difference level of data quality.

This thesis selects data processing and storage as an example from the life cycle of big data. It is proposed that before data processing and storage, the value of data quality is calculated and prioritized according to the calculation formula of data quality. First, the concept of data quality classification is proposed, and specific calculation formulas are given from the aspects of data integrity, consistency, and timeliness.

Secondly, on the premise of data priority determination, an energy-saving scheduling algorithm based on data quality (DQ-TSA) and an energy-saving storage algorithm based on data quality (DQ-HSA) are proposed.

Finally, the two algorithms are extended and implemented on the simulation platform Cloudsim, and to verify that whether the concept of pre-graded data quality can help the data center energy efficiency.

Contents

Abstract	ii
Attestation of Authorship	viii
Acknowledgements	ix
1 Introduction	1
1.1 Introduction	1
1.2 Background	2
1.3 Motivation	3
1.4 Contribution	4
1.5 Thesis structure	5
2 Literature Review	7
2.1 Introduction	7
2.2 Big data	8
2.2.1 Definition of big data	8
2.2.2 Research status of big data	10
2.2.3 How to respond to data growth	12
2.3 Energy Efficiency Metrics	15
2.4 Data Quality (DQ)	24
2.5 Simulation tools	28
2.5.1 GreenCloud	28
2.5.2 MDCSim	30
2.5.3 iFogSim	31
2.5.4 WorkflowSim	32
2.5.5 Cloudsim	33
2.6 Summary	35
3 Data-driven energy efficiency indicator model	36
3.1 Introduction	36
3.2 Measurement of data quality	36
3.2.1 Data set integrity calculation	38
3.2.2 Data set consistency calculation	39

3.2.3	Data set Accuracy calculation	40
3.2.4	Data set timeliness calculation	40
3.2.5	Data Quality assessment model	41
3.3	An energy-saving scheduling algorithm based on data quality (DQ-TSA)	42
3.3.1	The introduction and process of cloud computing task scheduling	42
3.3.2	Cloud task scheduling model based on data quality	45
3.3.3	The idea of DQ-TSA	46
3.4	An energy-saving storage algorithm based on data quality(DQ-HSA) .	53
3.4.1	The system model of DQ-HSA	54
3.4.2	Functional module design of DQ-HSA	56
3.5	Summary	62
4	Simulation studies	64
4.1	Introduction	64
4.2	Cloudsim Simulations	64
4.3	Case study 1: DQ-TSA algorithm	66
4.3.1	Energy consumption model	68
4.3.2	The simulation configuration	71
4.3.3	The simulation results	71
4.4	Case study 2: DQ-HSA algorithm	76
4.4.1	The simulation configuration	76
4.4.2	Workload	76
4.4.3	The simulation results	80
5	Conclusion and Future Work	87
5.1	Conclusion	87
5.2	Future Work	88
	References	89

List of Tables


2.1	Data Quality Dimensions	26
3.1	The relevant parameters of DQ-HSA algorithm	61
4.1	Core classes in Cloudsim	65
4.2	Physical Host hardware configuration	71
4.3	Storage device configuration	77
4.4	Workload parameters	79

List of Figures

1.1	A Big Data Life Cycle	4
1.2	Thesis Structure	5
2.1	DIKW modle	8
2.2	The total energy consumption of data centers	17
2.3	Energy Efficiency Metrics for the data centre	18
2.4	GreenCloud architecture diagram	29
2.5	MDCSim	30
2.6	iFogSim	31
2.7	WorkflowSim	33
2.8	CloudSim	34
3.1	Cloud Scheduling/storage based on task priority	37
3.2	Data Quality metric model	41
3.3	Tasks schedule model	44
3.4	DQ-TSA Algorithm Tasks schedule model	46
3.5	The process of DQ-TSA algorithm	52
3.6	Data storage placement in cloud storage environment	55
3.7	The system model of DQ-HSA	57
3.8	Preselection rules for hierarchical storage of data	61
4.1	Data simulation communication process	67
4.2	Compare the energy consumption of different algorithms	73
4.3	Compare the simulation time of different algorithms	73
4.4	Compare the overall SLA violation of different algorithms	75
4.5	Compare the average SLA violation of different algorithms	75
4.6	The system energy consumption of the algorithm under different file sizes	81
4.7	The system energy consumption of the algorithm under different file access frequencies	82
4.8	Time comparison between different file volumes	83
4.9	Time comparison between different file visits	83
4.10	Workload comparison between different file volumes	84
4.11	Workload between comparison between different file visits	85

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a university or other institution of higher learning.



Signature of student

Acknowledgements

This research work was completed as the part of the Master of Computer and Information Sciences (MCIS) course at the School of Computer and Mathematical Sciences (SCMS) in the Faculty of Design and Creative Technologies (DCT) at the Auckland University of Technology (AUT) in New Zealand.

Here, at first, I would like to express my deep and sincere gratitude to my supervisor, Dr William Liu, who provided me many precious opportunities and constant support and acted as a role model to me not only academic but also in life. His rigorous academic attitude always motivates me to continue to move forward. He has taught me so many things, but nothing was more valuable than his passion for knowledge. After I selected the thesis topic, anytime if i need help, Dr Liu gave me a lot of academic suggestions.

Secondly, I would like to thank Professor Jinsong Wu for sharing his professional advice. His work ethic taught me not to settle for mediocrity, which has deeply changed the way that I think. The inspiration for this thesis came from the beginning of 2019. At the recommendation of Dr Liu, I read Professor Wu's article about green big data. These articles published by Professor Wu and his team on big data meeting green challenges in 2016 should be the earliest discussion on the data life cycle as well as the relations of the green concept to the data life cycle. Except the works in 2016 from Professor Wu and his team, it first occurred to methought that in the era of big data there have not been much relevant existing work stalking about the fundamental concerns of green development about big data. Thus this thesis would like to make some further investigations about green and energy saving in big data era.

Then, I would also like to thank administrators in our school for their support and guidance through the MCIS in the past years. Especially Mrs Sharda Mujoo, every time I needed her, she enthusiastically gave me help. Next, I would like to thank my friends and classmates, Na An, Bo Ma, Yu Hang Fu, YuFeng Xiang and Shouming Sun. They gave me many valuable suggestions during the group discussion.

Finally, I want to thank my mother for her care and dedication over the years. Especially, this year of 2020 is a year of chaos caused by the COVID-19 pandemic and it is considered as the most crucial global health calamity of the century and the greatest challenge with massive disruptions that our humankind faced since the 2nd World War.

During this period, my mother was in Hubei, China. Not only did she overcome many problems in life, but she also gave me a strong belief and supported me to complete my studies.

While for me, it is a year started from the restrictions and frustrations but later on, it becomes my year of freedom of innovation and happy (also mixed with pains) growth. I have strived myself and turned my frustrations into new growth and innovation, an appreciation time on hatching ideas to cope with the COVID-19 chaos to my daily life and work.

Chapter 1

Introduction

1.1 Introduction

There are four sections in the first chapter of this thesis. In the first two parts, the background and motivation of this thesis are introduced respectively. Nowadays, people's lives are surrounded by a lot of data. A lot of research has focused on how to deal with large amounts of data and big data. Indicators that determine whether the data itself is valid and effective are rarely studied, and the energy consumption indicators of the electronic devices/methods that collect/process the data are not clearly defined. This thesis will study the correlation between the quality problem of data itself and the energy consumption index and propose to develop new index to measure the energy efficiency of different data, which brings a new measurement perspective and potential for solving the energy efficiency problem in the emerging era of big data. Finally, we will present an overview of the structure of this thesis in section 4.

1.2 Background

Cloud computing, Internet of things, social network and other emerging services promote the data type and scale of human society to increase at an unprecedented speed. Data is transformed from a simple processing object into a basic resource (Manyika et al., 2011). In 2002 there were 5 EB ($1EB = 10^{18}Byte$) of data on all the world's print and film collections, an order of magnitude equivalent to the 37,000 books in the U.S. library of congress. It was estimated that the entire human history could be stored in just 12 exabytes. By 2007 it had reached 24 exabytes, by 2011 it had reached 1.4ZB ($1ZB = 10^{21}Byte$), by 2013 it had reached 4.4ZB, and by 2015 it had generated more data than the previous 5,000 years combined to reach 8.6 ZB (Majidpour & Hasanzadeh, 2020). According to Gantz and Reinsel (2012) that in the past decade, emerging markets have increased their share in the expanding data world from one third to nearly 70%, marking the official arrival of the era of big data (Gantz & Reinsel, 2012). By 2020, International Data Corporation (IDC) predicts, the data will be 44ZB, which some have calculated is 57 times the size of all the sand grains on all the beaches on earth (Majidpour & Hasanzadeh, 2020). Big data contains great value and has important strategic significance for social, economic, scientific research and other aspects, providing people with unprecedented wealth of knowledge for deeper perception, understanding and control of the physical world. Therefore, researches related to big data have attracted widespread attention in various industries around the world. Many developed countries have formulated and launched big data research plans and invested a lot of funds to support big data research.

At the same time, in order to meet the problems of massive data, high concurrency, rapid response and scalability of big data applications, the construction of cloud data center also presents the trend of scale development. The proportion of data centers with more than 100 racks is increasing year by year, reaching 60% in 2016. Energy

consumption become the biggest expenses, data processing center in 2010, the global total data center power consumption is 235.5 billion KWH, accounts for about 1.3% of global electricity consumption in the United States, the proportion of electricity data center is more high, according to the U.S. environmental protection agency in 2011, a report on the data center, according to the United States all of the data center energy consumption accounted for 2% of the total power grid, and it also shows the tendency of doubling every five years. Such rising energy consumption indirectly causes greenhouse gas emissions, global warming and other problems.

1.3 Motivation

With the gradual reduction of non-renewable resources and the worsening of the natural environment in which we live, people have realized the importance of saving energy. Since the energy consumption of cloud data center is huge and the energy utilization efficiency is low, it is natural to improve the effective utilization of resources and reduce the overall energy consumption of cloud data center to become the mainstream direction of data center construction. In fact, in addition to the high energy and resource consumption in the data processing phase, a large amount of resources and energy will be consumed in the whole life cycle of data, such as data generation, collection, transmission and storage (as Figure 1.1).

Especially with the rapid development of Internet of things technology, in order to better provide high-quality services for human life, it is inevitable to use a large number of Internet of things devices to collect data (Wu, Guo, Huang, Liu & Xiang, 2018).

The purpose of this thesis is to discuss the relationship between data quality and energy consumption from the perspective of how to measure data quality. Then evaluate the priority of the data based on the data quality. And, follow the principle that the higher the priority, the better the resources, and match the corresponding processing

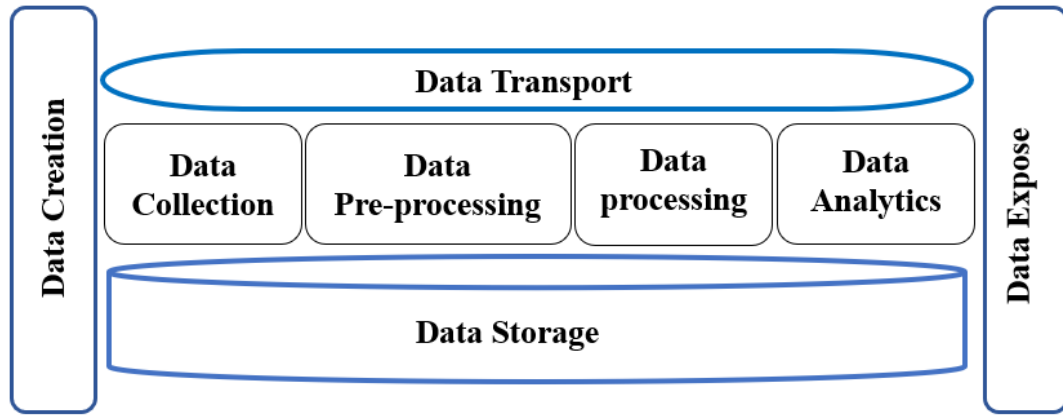


Figure 1.1: A Big Data Life Cycle

methods to achieve the purpose of energy saving in the data center.

1.4 Contribution

This thesis focuses on the relationship between data quality and energy consumption, and solves the problem of data center energy consumption from the perspective of data pre-grading.

First of all, with the continuous development of the information society, information systems are filled with massive, multi-structured, and multi-dimensional data resources. The value of big data has been fully recognized by society. How to tap the value of data has become one of the most concerned applications in various research fields and industries. Whether the data is garbage or treasure, the most important question is whether the data to be analyzed is of high quality. A low-quality data source will not only fail to reflect the value of the data, but may also run counter to the actual situation, but has a side effect. This article summarizes the dimensions of data integrity, timeliness, consistency, and accuracy by studying the literature about data quality in the past, except for the measurement methods of data quality indicators.

Secondly, the existing literature shows that most of the research on reducing energy

consumption in data centers focuses on optimizing the performance of data scheduling algorithms. In this paper, prior to data scheduling or storage, data priority is determined based on the quality of the data. Under this premise, an energy-saving scheduling algorithm (DQ-TSA) based on data quality and an energy-saving storage algorithm based on data quality (DQ-HSA) are proposed. After the experiment on the simulation platform Cloudsim, the results show that the two algorithms based on data priority, compared with traditional algorithms, have better performance in energy consumption and time.

1.5 Thesis structure

The thesis is structured as follows:

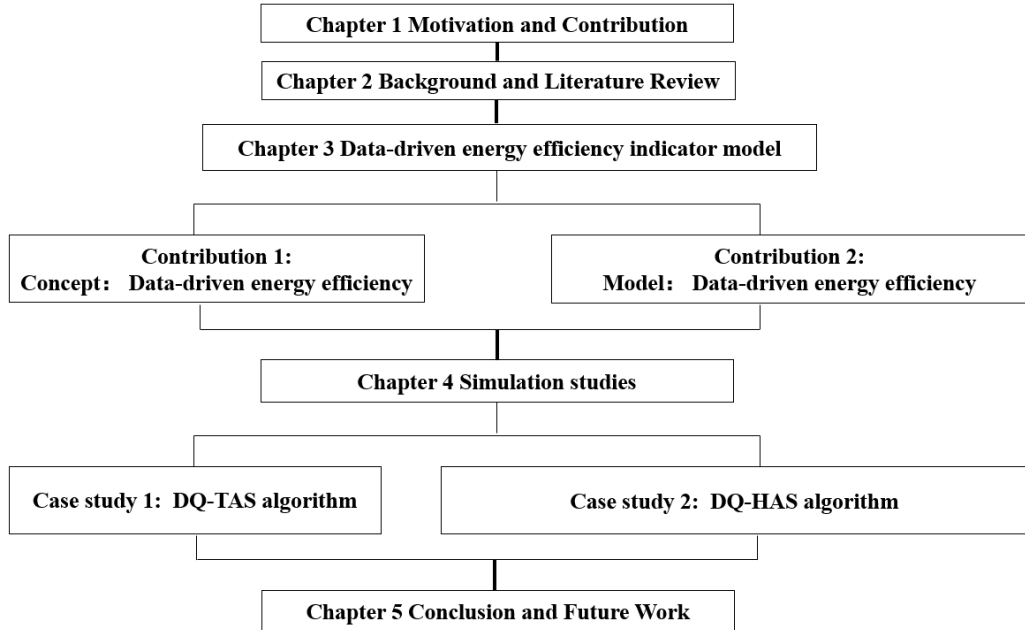


Figure 1.2: Thesis Structure

In chapter 2, the development trend of social big data is introduced in detail, and the relationship between the Internet and energy consumption is researched. In addition, the concept of data quality measurement is proposed based on the existing literature.

In addition, several existing simulation platforms were studied, and simulation tools suitable for this paper were selected.

In Chapters 3 and 4, the simulation tools mentioned in Chapter 2 will be used to implement the algorithm we introduced. In addition, the experimental results and conclusions will be explained in detail with the support of tables and figures. The limitations of the project will also be resolved.

Chapter 2

Literature Review

2.1 Introduction

With the increasing popularity of the Internet and related applications, more and more data are generated and analyzed. With the rapid development of Internet of things technology, the generation of big data from sensors, cameras and other available data sources has created great pressure on existing data devices. From the perspective of the whole life cycle of big data, the generation, transmission, calculation and storage of data will bring energy consumption.

In response to these problems, there have been many studies in previous work that have investigated deeply about how to deal with data growth, the correlation between data (Morley, Widdicks & Hazas, 2018) , energy efficiency and the quality of data (Wu et al., 2018).

2.2 Big data

2.2.1 Definition of big data

Big data is also a kind of data. The difference between it and data is not only the most basic volume, but also other differences. The following Figure 2.1 reveals the relationship between data, information, knowledge and wisdom (Bihl, Young II & Weckman, 2016).

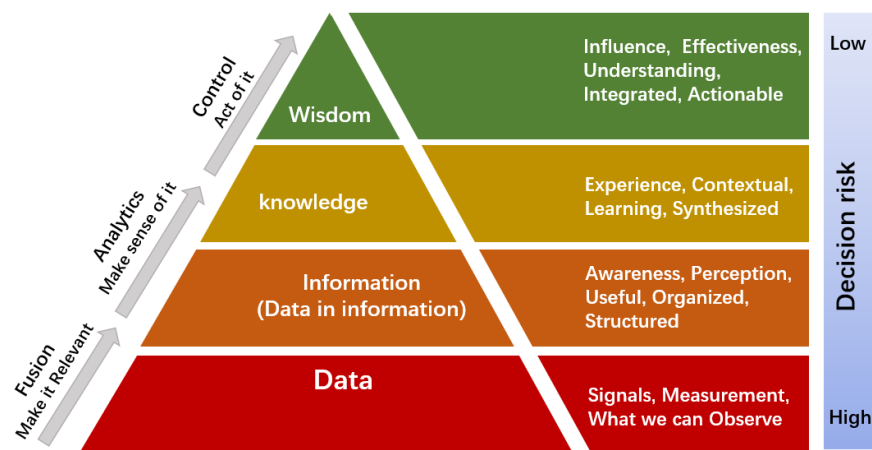


Figure 2.1: DIKW model

Data is the result of facts or observations, the logical induction of objective things, and the raw material used to represent objective things. It is an identifiable, abstract symbol. In general, data is contained and Shared by four storage media: print, film, magnetism, and optics (Lyman, 2003). Therefore, it not only refers to Numbers in a narrow sense, but also refers to the combination of characters, letters, numerical symbols, graphics, images, videos, audio and so on, which have certain meanings. It is also an abstract representation of the attributes, quantities, positions and their mutual relations of objective things. For example, "0, 1, 2...", "Yin, rain, fall, temperature", "students' records, the transport of goods" and so on are data. However, the data itself does not represent any potential meaning until it has been processed. Only when the data needs to be organized and analyzed in a certain way can the significance of the data

be shown, thus evolving into relevant information, knowledge and wisdom (Salzmann, 2000). The DIKW model is a good model to help us understand the relationship between Data, Information, Knowledge, and Wisdom. It also shows us how Data is transformed into Information, Knowledge, and even Wisdom step by step, as Figure 2.1 (Bihl et al., 2016).

Big data itself is an abstract concept. Literally, it means data on a large scale. But only large data quantity huge apparently unable to see what is the difference between this concept and past definition such as "massive data", "very large data". Big data is not yet an accepted definition, Mayer-Schönberger and Cukier (2013) definition of big data is straightforward: big data refers to the use of commonly used software tools to capture, manage, and process data are time consuming more than can tolerate time data sets. The definition of the different basic characteristics from large data, according to these characteristics and inductive trying to gives its definition, in these definitions, the more representative is 3V is defined, which is to think big data to meet the three features: volume, variety and velocity. In addition, a definition of 4V is proposed, that is, an attempt to add a new feature to 3V (Takaishi, Nishiyama, Kato & Miura, 2014). International Data Corporation (IDC) think big data shall also have the value, the value of big data often presents the sparse sexual characteristics, and IBM think big data must have veracity. This veracity is not only reflected in the real reaction to the objective things, but also reflected in the response to the wholeness of the cognitive objects. Microsoft believes that it has the Value of valuable content (Kambatla, Kollias, Kumar & Grama, 2014). Big data reflects the integrity of data records. This completeness not only records a large number of rare events, but also results in the correlation between data that cannot be represented by local data due to the integrity of the data, which will lead to the emergence of new valuable events. Other scholars believe that it has the value of Vitality. Big data records and provides information continuously and comprehensively, so it can meet customers' flexible demands for information content.

These definitions show that big data is a true reflection of the complete set of information in the objective world. The accuracy of big data analysis and processing is very important for all walks of life, and it can bring more valuable information to the society than oil. Therefore, big data is not only huge and complex, but also contains the meaning of "big" in terms of internal capacity and value.

2.2.2 Research status of big data

Toffler and Alvin (1980) first proposed the concept of "big data". In their book "the third wave", they predicted that the third wave of social memory would not only increase in number, but also infuse human memory with life. Big data has been called the third wave of brilliance.

Azevedo and Santos (2008) pointed that knowledge discovery in databases (KDD) was coined by Fayyad, Piatetsky-Shapiro and Smyth in 1989. In addition, Fayyad, Piatetsky-Shapiro and Smyth (1996) proposed that the process of knowledge discovery in the database is to identify novel, effective, potentially useful and understandable patterns from a large number of data sets, and pointed out that knowledge discovery is a high-level process to find such patterns. Many people regard knowledge discovery and data mining as equivalent concepts. Knowledge discovery is a common term in the field of artificial intelligence, while the field of database is called data mining.

The title of data scientist was first mentioned by Natahn Yau in 2009, who believed that a data scientist is a person who can extract data from a large data set and provide something that can be used by non-data experts (Segaran & Hammerbacher, 2009). Davenport and Patil (2012) argues that data scientists need to use a combination of statistics and programming to extract useful information from the vast amount of data they collect to identify factors that have a significant impact on a company's bottom line.

Graham-Rowe et al. (2008) stated that With the discovery of more and more research disciplines, a large amount of data is facing new challenges that need to be solved as soon as possible, for example, when the researchers study the inner workings of the cells, they are now collected a large number of genome sequences, protein sequences, protein structure and function, double molecular interactions, signaling and metabolic pathways, such as adjusting the motif of the scientific research data, even if is the smartest scientist will turn to advanced data mining tool, as a result, the term "big data" began to be widely spread.

Hey, Tansley, Tolle et al. (2009) revealed the fourth paradigm of scientific research, namely data-intensive scientific discovery, which is complementary to experimental science, theoretical deduction and computer simulation. And further probes into the connotation and extension of this new paradigm, including the use of various tools uninterrupted research data, establishing systematic tools and facilities to manage the entire data life cycle, the development based on scientific research data analysis and visualization tools and methods, and further discusses the new paradigm of scientific research, science, education, academic communication, and the long-term impact of scientist groups.

Barabási and Gelman (2010) believes that humans have entered the era of big data and can predict the future. Behind human behaviors lies the "outbreak" of patterns. The daily behavior patterns of humans are "explosive" rather than random. The deep order in human behavior is shaped by an explosion, and big data makes it easier than expected to predict the future. The impact of revealing patterns is comparable to the physics or genetic revolution of the early 20th century.

Mayer-Schönberger and Cukier (2013) proposed the concept of big data thinking, and predicted in advance that the emergence of big data would bring great changes to people's life, work and thinking, and big data brought great changes to the era. In addition, it describes the business changes and management changes brought to

the society under the thinking changes in the era of big data, and holds that the core of big data is prediction. The three changes of big data thinking have caused great repercussions in the society. First, big data is not a random sample but all data. Second, big data is not precise but promiscuous and fuzzy. Third, big data is not causation but correlation. It is believed that big data can change the fuzzy concept in human life into quantifiable index and make people's life change unprecedentedly.

In recent years, the development of big data has always been in the development stage of technology field, which mainly studies big data from two aspects of data processing tools and processing difficulty based on data sources. In other words, big data research is becoming more and more popularized and commercialized from new technologies in previous years. In particular, the emergence of cloud computing mode enables network resources to be configured and accessed on demand. Enterprises only need to invest a small amount of money to enjoy ultra-high-speed computing, super-capacity storage and high-speed network transmission. At the same time, due to the continuous advancement of big data research, data-based artificial intelligence, machine learning, Internet of things and other fields will also make more and more achievements. For example, the concept of big data cloud map can be used to describe the distribution of enterprises in the big data industry, and through the analysis of successful enterprises, people can easily find where the industrial opportunities of big data will emerge and how to find these opportunities (Stephenson, 2018).

2.2.3 How to respond to data growth

Internet digital technology is widely expected to play a critical role in the transition to a more sustainable and energy-efficient future (Atat et al., 2018). For example, interest in smart meters, power grids, and cities. However, the increasing in the number of connected devices, the number and type of services, and the level of data traffic,

processing and storage, which means that the energy used to power the Internet is growing dramatically.

The research conducted by Hazas, Morley, Bates and Friday (2016), use the nature of data traffic growth on the Internet as a research basis to see if such growth will slow or limit. Data growth has been intense over the past decade, and there are predictions that this pattern of sustained growth will continue in the future. Since this phenomenon is closely related to increased electricity consumption, the global impact of this trend on reducing carbon emissions is significant. Hazas et al. (2016) selectively explore the aspects of data growth related to day-to-day practices and how they leverage and generate Internet data and believe that there are some conceivable limitations to this growth. In practice, however, the nature of "Internet use" is changing and emerging forms of growth are out of touch with human activities and time use. For example, in the trend of the Internet of Things being widely used, the potential self-generation cycles for data generation, processing and circulation are largely automated and not limited using human activities.

Widdicks, Bates, Hazas, Friday and Beresford (2017) also believe that the demand for mobile technologies and related services in people's lives is unprecedented and growing. In their study, they conducted quantitative and qualitative surveys of users of eight Android devices and compared them with quantitative surveys of 398 Android devices. How to reduce the data requirements for mobile devices beyond the device itself is studied. These include targeting to watch, listen and use social networks at specific times of the day; make full use of existing capabilities with SMS; help ingesting help in the peak demand; pass the time in a positive and relaxed way through a dedicated lack of mobile technology. Morley et al. (2018) argue that in the Internet, different services require different amounts of energy, and the more data they move, the more power they consume. Therefore, the assumption of research on data demand as a representative of energy demand, conceptualizes several ways to study the processes that

underpin growing data demand, thereby supporting infrastructure energy demand. These include the design and delivery of services, the increasingly nature of data-intensive practices and the more widespread and extensive integration of data-based services throughout society. While many stresses the importance of improving Information and Communication Technologies (ICT) energy efficiency, including standby features, routers, mobile and fixed access networks, and data centres, the idea of limiting data requirements in any form runs counter to the dominant paradigm of digital services and government policy design (Chao, Chen & Wu, 2011). Current government policies in many countries aim not only to extend Internet access to households and citizens who do not already have Internet access, but also to make existing connections faster and faster. Addressing the challenge of data demand growth requires a detailed focus on the trends that underpin data demand. While the statistics clearly show that video traffic is growing rapidly, by investigating when and how people use online viewing services, there have been studies that show how the time spent on data traffic becomes relevant at the national level. At the same time, day-to-day practices, reflected in time usage data and mobile device usage, are important places to understand these changes. This is not to say that access to digital services is the most appropriate intervention point. It is also important to consider how to reorganize the provision of video-related services and the role of policy, institutions and business organizations in these developments. For example, activities such as checking social media have become more data-intensive because videos, including ads, are increasingly embedded in feeds. The shift to UHD streaming media has also increased the often-invisible energy need to watch television and movies over the Internet.

All in all, in the context of climate change targets, there must be better options to deal with Internet-related energy needs, rather than making it a "problem".

2.3 Energy Efficiency Metrics

Cloud computing was born in 2007 and is the further development of parallel computing, distributed computing, grid computing and other technologies. As a new business computing model, it is a fusion product of computer technologies such as distributed computing, virtualization, network storage and load balancing (V. Chang & Gütl, 2010).

Cloud computing can allow users to access services at different locations and operate different terminals, enabling Virtualization technology is used, and the concept of "cloud resource pool" is also involved (Sreenivas, Prathap & Kemal, 2014). The resource requested by the user is obtained from this resource pool, rather than a fixed entity. The application arranged by the user also runs somewhere in the cloud system without knowing the specific location of the application. Only one Networked terminals can obtain stronger computing power through the Internet. For the average user, there is no need to understand the complex mechanisms behind the system, which is the convenience brought by virtualization (Marston, Li, Bandyopadhyay, Zhang & Ghalsasi, 2011). By using measures such as data multi-copy fault tolerance and interchangeable homogeneous computing nodes, cloud computing uses multiple computing nodes in parallel to ensure its reliability. This effective multi-group backup method is analyzed from a practical perspective, rather than using only local ,the computer is more reliable (Gawali, June 2014). But in achieving the aforesaid, a data centre requires huge amount of power needed to process data, to store it so as to fulfil the communication job. Beside that, a simultaneous negative impacts is also thrust upon the environment(Akula & Potluri, 2014) in the form of emission of carbon dioxide (CO_2). One data centre can emit 170 Million Metric Tonnes (MMT) carbon per year which can be estimated to be 670 MMT carbon per year by 2020 due to data centres present worldwide over (Ranky, 2010). The huge energy consumption in data centres results into high operational costs.

Typically data centre utilizes energy as required for 25000 households annually. Hence the need is imperative as green IT vision for reducing CO_2 emissions and enhancing energy efficiency. Performance and energy efficiency metrics serve as backbone to achieve this goal (Vatsal & Agarwal, 2019).

Energy efficiency metric can be used by managers to measure and maintain the implementation of cost saving and CO_2 emissions in data centers (Yu & Lai, 2016).

In general, energy efficiency is the use of less energy to produce the same amount of service or useful output (Wu, Rangan & Zhang, 2016). The formula for calculating energy efficiency in a broad sense is:

$$\text{Energy efficiency} = \frac{\text{Useful output of a process}}{\text{Energy input of a process}}$$

For a study of energy efficiency directions, Patterson (1996) demonstrates the range of energy efficiency indicators that can be used to define how energy efficiency indicators can be used to monitor changes in energy efficiency from the perspectives of kinetics, physical thermodynamics, economic thermodynamics, and economics (Wu, Guo, Li & Zeng, 2016).

1) From a thermodynamic point of view, the output and input of energy is based entirely on measurements derived from thermodynamic science.

2) From the perspective of physical thermodynamics, in the calculation formula of energy consumption, the input is the thermodynamic unit, and the output is the physical unit.

3) From the perspective of economic thermodynamics, the input of the energy efficiency formula is a thermodynamic unit, and the output of service provision is measured at market prices.

4) From the perspective of economics, in the calculation of energy efficiency, energy input and output of service provision are purely based on market currency value.

In addition, the impact of energy quality issues on measuring energy indicators is also considered. They believe that these different sources, different forms of energy, need to be adjusted before any energy efficiency calculations are made (Hidalgo-León et al., 2017).

At present, there are some researches on energy consumption monitoring and management of data center from a global perspective. The energy consumption of the data center is the total energy consumption of various energy-using equipment in the data center, including not only the energy consumption of IT equipment such as servers, but also the energy consumption of auxiliary systems such as air conditioning and power distribution (Hidalgo-León et al., 2018). The IT system composed of hardware such as server, storage and network communication is the most energy-consuming part, accounting for about 50% of the total energy consumption of the data center. The energy consumption of air conditioning system accounts for about 40%, ranking the second in the total energy consumption of data centers. Power distribution systems consume about 10% of the data center's energy. As Figure 2.2.

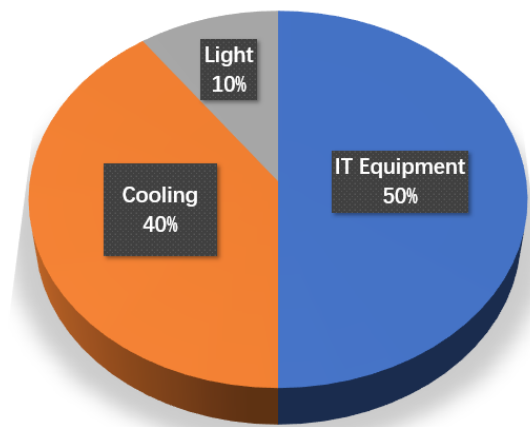


Figure 2.2: The total energy consumption of data centers

There are evaluation metrics model for energy efficiency in data center is shown in Figure 2.3.

To evaluate the efficiency of data centers, ICTs industry experts have come up

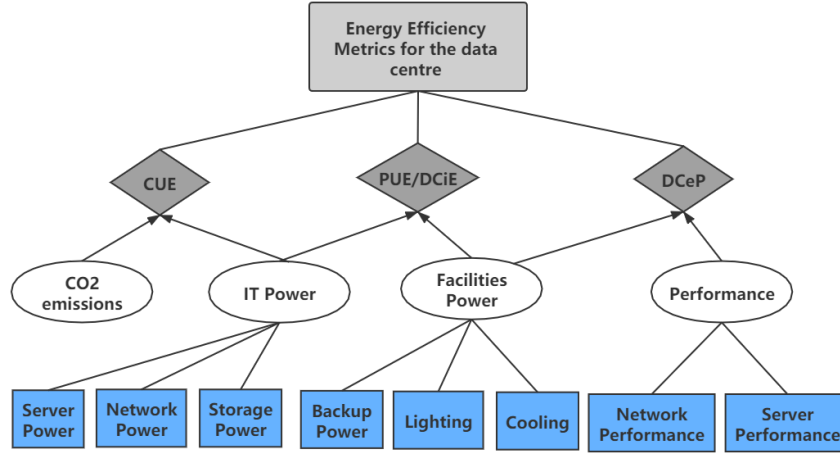


Figure 2.3: Energy Efficiency Metrics for the data centre

with a number of measures, the most influential of which is the Power Usage Efficiency (PUE) proposed by the Green Grid (Jaureguiualzo, 2011). The definition formula of electric energy use efficiency is as follows:

$$PUE = \frac{\text{The total facility power}}{\text{The energy used by IT equipment}}$$

The metric of Data Center Infrastructure Efficiency (DCiE) just the reciprocal of PUE.

$$DCiE = \frac{1}{PUE} = \frac{\text{The energy used by IT equipment}}{\text{The total facility power}}$$

The lower the PUE value, the more efficient the data center. An efficient data center PUE is typically less than 1.6 (AL-Hazemi, Mohammed, Laku & Alanazi, 2019). Although PUE has long been an important indicator for evaluating the energy efficiency of data centers, it fails to consider the degree to which data centers pollute the environment. For example, data centers power ed primarily by new energy sources may have higher PUEs, but they produce less pollution. In order to evaluate the Carbon emission of data centers, green grid put forward the indicator of Carbon Usage Effectiveness (CUE) (Herzog, 2013). It represents the carbon intensity per kilowatt-hour

of electricity used. The definition formula of CUE is as follows:

$$CUE = \frac{\text{The total } CO_2 \text{ emissions of data center}}{\text{The energy used by IT equipment}}$$

Data center energy productivity (DCeP) refers to a measure of the useful work performed by the data center relative to the energy consumed by the work performed by the data center (Sego et al., 2012). The formula of DCeP is as follow:

$$DCeP = \frac{\text{Useful work produced}}{\text{Total energy consumed by the data center}}$$

In order to monitor and manage cloud server more fine-grained, a large number of researches mainly estimate the energy consumption of cloud server (including physical machine and virtual machine) from the software level, mainly including: for physical machine, its energy consumption measurement is mainly divided into three steps: information collection, modeling and estimation. There are two types of energy consumption models: component energy consumption accumulation model and utilization model. For each kind of model, it can be divided into linear model and nonlinear model.

1) Accumulation model of component energy consumption: Resource characteristics are used to represent the energy consumption of corresponding hardware resources, and finally, they are added up to represent the total energy consumption of the server. Ge, Feng and Cameron (2009) gave a general energy consumption formula, which can be expressed as:

$$E_{Server} = E_{CPU} + E_{Mem} + E_{Other} \quad (2.1)$$

The resource characteristics can be resource utilization, Performance Monitor Counter (PMC) and hardware counters. From simple to complex, Roy, Rudra and Verma (2013) first expressed the server energy consumption as the sum of CPU energy consumption

and memory energy consumption. Jain, Molnar and Ramzan (2005) further decomposed CPU and memory into the accumulation of data and instruction features respectively. In addition to CPU and memory, Tudor and Teo (2013) added IO power consumption to the model. Song, Barker and Kerbyson (2013) replaced the previous IO energy consumption with disk and network card energy consumption.

Since energy consumption is the accumulation of power in time, Song et al. (2013) extended the original formula and expressed the total energy consumption by multiplying the average energy consumption of processor, network card and network equipment with their respective running time. Lewis, Ghosh and Tzeng (2008) used hardware counters to model and sum five components separately, including the motherboard, but the final energy consumption model was still linear. Alan, Arslan and Kosar (2014) modeled server energy consumption using a linear combination of CPU, memory, disk, and network card utilization. Lent (2013) further decomposed the resource utilization of each component into its corresponding child component utilization to establish a linear model. Generally speaking, the total power P_{total} of the server can also be expressed as the sum of static power P_{static} and dynamic power $P_{dynamic}$ (Beloglazov, Buyya, Lee & Zomaya, 2011), (F. Chen, Grundy, Yang, Schneider & He, 2013), (Xiao, Hu, Liu, Yan & Qu, 2013).

$$P_{total} = P_{static} + P_{dynamic} \quad (2.2)$$

Among them static power is also called base power which is the active server to maintain the operation must consume the minimum power, dynamic power is the program running using different hardware resources to bring variable power.

Bircher and John (2007) proposed to use the performance counters of the microprocessor to model each sub-component (including CPU, memory, disk, I/O and chipset) separately, and finally combine them together as the energy output of the physical

machine. Here, the energy consumption model for each sub-component is nonlinear.

2) Resource utilization model: only CPU utilization is used to model the total energy consumption of the server. For CPUs with multiple frequencies, the different frequencies can be seen as an indication of utilization. Elnozahy, Kistler and Rajamony (2002) first gave the energy consumption model of the server under different CPU frequencies. The power P_f of the server under frequency f was expressed as:

$$P_f = c_0 + c_1 f^3 \quad (2.3)$$

c_0 and c_1 are model coefficients, which can be obtained by linear regression.

Fan, Weber and Barroso (2007) and Gao, Guan, Qi, Wang and Liu (2013) demonstrated through experiments that the server power and CPU utilization are almost linear, which is widely used in the energy consumption model of data centers, specifically expressed as:

$$P_u = (P_{peak} - P_{idle})u + P_{idle} \quad (2.4)$$

u is resource utilization, while P_{peak} and P_{idle} represent peak power and base power of the server, respectively.

In addition, there are many literatures that use nonlinear methods to model server energy consumption. For example, Tang and Dai (2011) added two server-dependent model parameters to formula (2.4), making the original model more complex. Due to the constant change of CPU utilization over time, Beloglazov, Abawajy and Buyya (2012) presented a power consumption model integrating power over time, in which power is a function related to utilization. H. Li, Casale and Ellahi (2010) proposed a normalized expression of server power and a compound power model based on utilization. V. Gupta, Nathuji and Schwan (2011) also status that to estimate the server energy consumption at different request arrival rates using the queuing theory method. Similarly, literature Tian, Lin and Li (2014) and Yao, Huang, Sharma, Golubchik and Neely (2012) also modeled

server energy consumption based on utilization and request arrival rate. Horvath and Skadron (2008) modeled the server in terms of both CPU frequency and utilization. Relatively speaking, linear model has the advantages of simple, practical and high accuracy.

For virtual machines, there are two main methods of measuring energy consumption: white box method and black box method (Jiang, Lu, Cai, Jiang & Ma, 2013). The white box method collects the resource information inside the virtual machine by inserting the agent program in the virtual machine and makes use of the collected information for modeling. A typical research paper of Y. Li, Wang, Yin and Guan (2012), the model is:

$$P_{server} = P_{static} + \alpha \sum_i^n U_{VM_i}^{CPU} + \beta \sum_i^n U_{VM_i}^{Mem} + \gamma \sum_i^n U_{VM_i}^{IO} + ne \quad (2.5)$$

Each virtual machine can be calculated using the following formula:

$$P_{VM_i} = \alpha U_{VM_i}^{CPU} + \beta U_{VM_i}^{Mem} + \gamma U_{VM_i}^{IO} + e \quad (2.6)$$

($U_{VM_i}^{CPU}$, $U_{VM_i}^{Mem}$ and $U_{VM_i}^{IO}$ respectively represent the utilization of CPU, memory and IO collected inside virtual machine i , and e means the migration constant in the model. In this paper, a piece-wise modeling idea is proposed to divide the data set into three segments according to the high, medium and low resource utilization. However, the definition of high, medium and low is vague and too subjective, and it is unfair to divide the bias in the model evenly to each virtual machine.

The disadvantage of the white box approach is that the resource information obtained inside the virtual machine is not a good reflection of the virtual machine's use of physical hardware resources. To solve this problem, the black box method first uses the collected physical machine resource characteristics to model the server energy consumption. Based on the physical machine model, the energy consumption

of the virtual machine can be calculated by bringing the physical resources occupied by the virtual machine into the model. In other words, the accuracy of virtual machine energy consumption measurement depends on the physical machine energy consumption model.

Kansal, Zhao, Liu, Kothari and Bhattacharya (2010) used the CPU utilization u_{CPU} , Last Level Cache Misses N_{LLCM} , and IO transfer time b_{IO} to model the energy consumption of the physical machine, which is expressed as:

$$P_{Total} = P_{static} + \alpha u_{CPU} + \beta N_{LLCM}(T) + \gamma b_{IO} \quad (2.7)$$

Bohra and Chaudhary (2010) also used a linear model to model the energy consumption of physical machines, expressed as;

$$P_{Total} = \alpha P_{CPU} + \beta P_{Cache} + \gamma P_{DRAM} + \delta \quad (2.8)$$

In order to measure the energy consumption of virtual machines, most studies first model the energy consumption of physical machines. The commonly used linear model mainly includes multiple Regression method, Mantis and Lasso Regression (Economou, Rivoire, Kozyrakis & Ranganathan, 2006), (Rivoire, Ranganathan & Kozyrakis, 2008). The nonlinear models mainly include polynomial with exponential or Lasso (Bircher & John, 2007), gaussian mixture model, and support vector regression (Dhiman, Mihic & Rosing, 2010). McCullough et al. (2011) believed that the more complex the model, the higher the accuracy. In practical scenarios, however, there is a trade-off between model accuracy and practicality, so the linear approach is widely used for its simplicity and high accuracy.

2.4 Data Quality (DQ)

Big data has the characteristics of large volume, wide sources, variety, high frequency and low value density. With the widespread application of big data in practical business, data quality problems gradually emerge. In the United States, corporate losses due to incorrect data exceed \$700 billion annually (L. Chen, He, Yang, Niu & Ren, 2017). The detection rate of enterprise error data is between 1% and 30% (Ghemawat, Gobioff & Leung, 2003). There are many data warehouse projects, and the time spend on Extract-Transform-Load may account for 30% to 80% of the overall development time and budget. Improving data quality is crucial for system construction. On the web, XML is the most common document format, accounting for about 58%, of which only a third are valid, 14% lack legitimacy, and simple errors like mismatching tags and missing tags render the entire XML technology useless for these document processing (Dean & Ghemawat, 2008).

Although the concept of "data quality" seems obvious, in current practice, data quality is not well defined. Our research shows that data quality has many dimensions for data users, such as accuracy, reliability, relevance, and timeliness, and requires clear and unified data quality metrics. However, the fact is that even a relatively obvious dimension, such as accuracy, does not have a strong enough and generic definition to make the technique of measuring data accuracy universally acceptable. Thus, the concept of data quality is relativistic, and what one group of users considers good data may be bad for another. Only when the evaluation method, knowledge reference, data set type, use purpose and strategy of data quality can effectively balance the decision analysis can the data be meaningful and of high quality. In general, the current data quality focuses on the following aspects : (1) from the user's perspective, it focuses on the user's perception of data; (2) Data quality requires overall governance, a sound organizational system, and multi-dimensional evaluation of data quality; (3) the data

quality is a multidimensional concept, a multi-dimensional evaluation criteria, the need to build a complete evaluation methods. In particular, we need to identify the key data quality dimensions, the precise definition and significance of each dimension, determine the evaluation data of each dimension method and calculating each dimension value of data quality.

Since 1980, accuracy has been the most important dimension in evaluating data quality. The connotation of data quality is gradually deepened, and its concepts and methods are also constantly expanded. While accuracy is important, it is not the only measure of data quality. Fox, Levitin and Redman (1994) believes that data quality is also a multidimensional concept because the data itself has multiple sources and multiple structures. In the late 1990s, Redman (1995) put forward the following idea: "Data is of high quality if it is applicable to business operations, decision making, plan execution, etc. Data is suitable for use if it is complete and has the desired data characteristics." (F. Chang et al., 2008).

Wang, Storey and Firth (1995) published a survey on data quality in 1995, in which they recommended the use of dimension sets to describe data quality. Since then, other scholars have conducted in-depth research on the quality dimension. Based on the information system model, Wand and Wang (1996) proposes five dimensions of data quality: accuracy, integrity, consistency, timeliness and reliability. Wang and Strong (1996) for data users who have rich practical experience and can make correct decisions on data, It is concluded that data quality is data that is applicable to a business scenario. And through the in-depth investigation of 179 features of data quality, the quality of 15 common dimensions is determined. They also point out that data quality cannot be isolated from the user and must be closely tied to the user experience in order to be assessed.

In 2003, Kerr, Norris and Stockdale (2007) proposed that data quality solutions and planning strategies within an organization must consider the needs of data users

and allow them to evaluate data quality by meeting their needs. That is, the purpose and users of data should be clear from the very beginning. Redman and Blanton (1997) indicates that the data quality dimension can be divided into three groups, corresponding to the concept of view data and data format and data value respectively. Jarke, Lenzerini, Vassiliou and Vassiliadis (2013) proposes a detailed data quality dimension to guide the design of data warehouse. Bovee, Srivastava and Mak (2003) defines the quality of data in data that is suitable for use, including accessibility, interpret-ability, relevance, and reliability. In order to integrate the WEB information system, Naumann (2003) defines 4 categories of 21 quality dimensions. Through the above research, correctness, completeness and consistency are considered as the basic evaluation dimensions, as shown in the below table 2.1:

Table 2.1: Data Quality Dimensions

Dimension	Definition	Research work
Accuracy	Data accuracy is defined as when the data value stored in the database is identical to the data value in the real world.	Batini and Scanapieco (2016)
	Data accuracy is a modification of the data set.	McGilvray (2008)
	Accuracy is a measure of how well a data value matches the real data.	Redman and Blanton (1997)
	It is the standard of reliability, accuracy and certification of data.	Wang and Strong (1996)
<i>Continued over page</i>		

Table 2.1: Extended version. . . (*continued*)

Completeness	Data accuracy is defined as when the data value stored in the database is identical to the data value in the real world.	Batini and Scanapieco (2016)
	Data accuracy is defined as when the data value stored in the database is identical to the data value in the real world.	Wang and Strong (1996)
Completeness	Data accuracy is defined as when the data value stored in the database is identical to the data value in the real world.	Batini and Scanapieco (2016)
	How helpful the data is to existing work.	Wang and Strong (1996)
	The perspective of the data collection process.	Redman and Blanton (1997)
	The percentage of real data contained in the data warehouse.	Jarke et al. (2013)
	The data contains all the necessary parts of the entity information.	Batini and Scanapieco (2016)
Consistency	The data contains all the necessary parts of the entity information.	Bovee et al. (2003)

Continued over page

Table 2.1: Extended version. . . (*continued*)

	Consistent with the semantic rules defined by the data set.	Wang and Strong (1996)
reduced	Unwanted duplicate measures that exist for a particular field, record, or dataset.	McGilvray (2008)

2.5 Simulation tools

Because different applications may have different standards for resource allocation and deployment requirements, the load, energy consumption and system size of the application and service models on the cloud infrastructure are constantly changing. To reduce the cost of accessing the infrastructure in a cloud computing environment, evaluating and simulating the entire scheduling process before composing, configuring, and deploying the software is a feasible solution. Therefore, tools that can provide adjustable simulation environment come into being, which allow users to test their services for free and repeatable, and adjust performance bottlenecks before deployment, not only reducing the cost and threshold of research and testing, but also reducing the cost and risk of experiments (Bahwaireth, Benkhelifa, Jararweh, Tawalbeh et al., 2016). At present, there are five main cloud computing simulation tools as follows.

2.5.1 GreenCloud

GreenCloud is a package-level network simulator NS2 extension, which complies with the GPL protocol, is used for advanced energy-sensing research on cloud computing data centers in actual Settings, and is an open source cloud environment simulator

(Kliazovich, Bouvry & Khan, 2012). It focuses on the energy perception in the cloud communication process and the fine-grained modeling of the energy consumption of the data center, including the energy consumption of the server, network switch and communication link. It differs from existing emulators in the way it extracts, aggregates, and provides information about the data center.

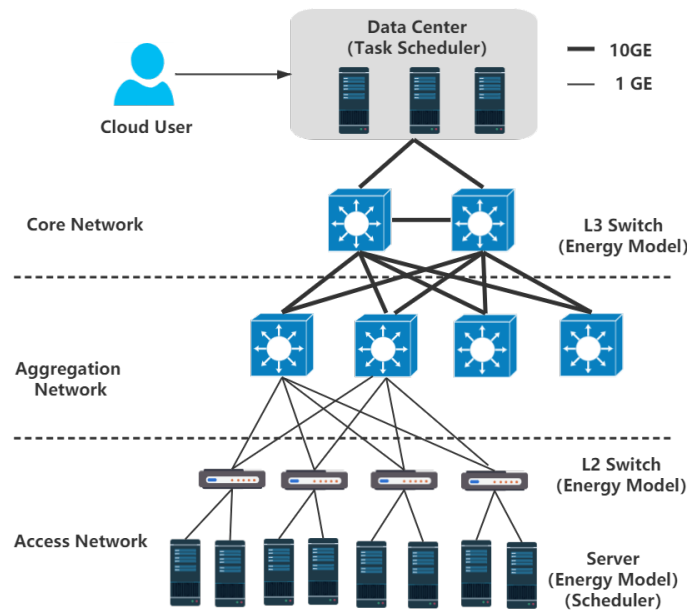


Figure 2.4: GreenCloud architecture diagram

At the bottom of it is the core part of the data center – Servers, whose main function is to perform the assigned tasks. It is a single-core component that presets computing power, memory, storage space, and differentiated task scheduling policies (Frej, Di-chter & Gupta, 2018). The server is connected to the access network layer, which is responsible for data exchange between the server and the secondary switch. The middle part is the core network layer and the converged network layer, which is mainly the connection between the switch and the data center. The outermost layer is encapsulated as a data center, which directly interacts with users. Cloud users can directly access the data center for business-level operations and task scheduling (see Figure 2.4).

2.5.2 MDCSim

MDCSim is a multi-tier data center simulation tool released by Pennsylvania state university. It is designed as a pluggable three-tier architecture that captures all the important design details of the underlying communication paradigm, kernel-level scheduling artifacts, and application-level interactions between the three-tier data centers (Lim, Sharma, Nam, Kim & Das, 2009). The flexibility of the simulator lies in its ability to experiment with different designs in three layers and compare performance and power consumption under real workloads. Although MDCSim can run across platforms and simulate quickly, as a commercial software, it needs to pay for the software license. In the absence of GUI operation interface and effective functions, it cannot give researchers sufficient reasons to buy it. The MDCSim architecture is shown in Figure 2.5.

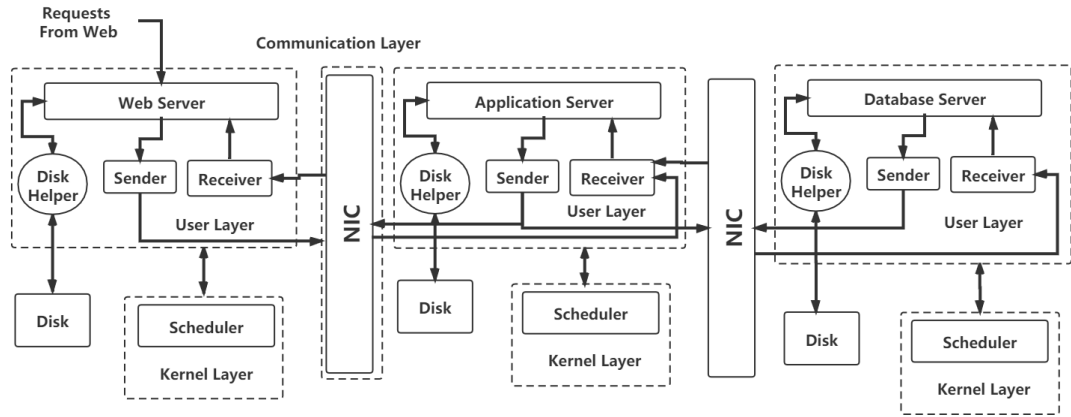


Figure 2.5: Architectural Details of MDCSim

Emulators are divided into communication layer, core layer and user layer. Such an abstract approach enables the emulators to have better flexibility and greater scalability. Different communication modes can be integrated into the communication layer only according to the specification semantics and writing specific protocols.

2.5.3 iFogSim

iFogSim supports the modeling and simulation of Fog Computing environment to evaluate the resource management and scheduling policies across edge and cloud resources in different scenarios (H. Gupta, Vahid Dastjerdi, Ghosh & Buyya, 2017).

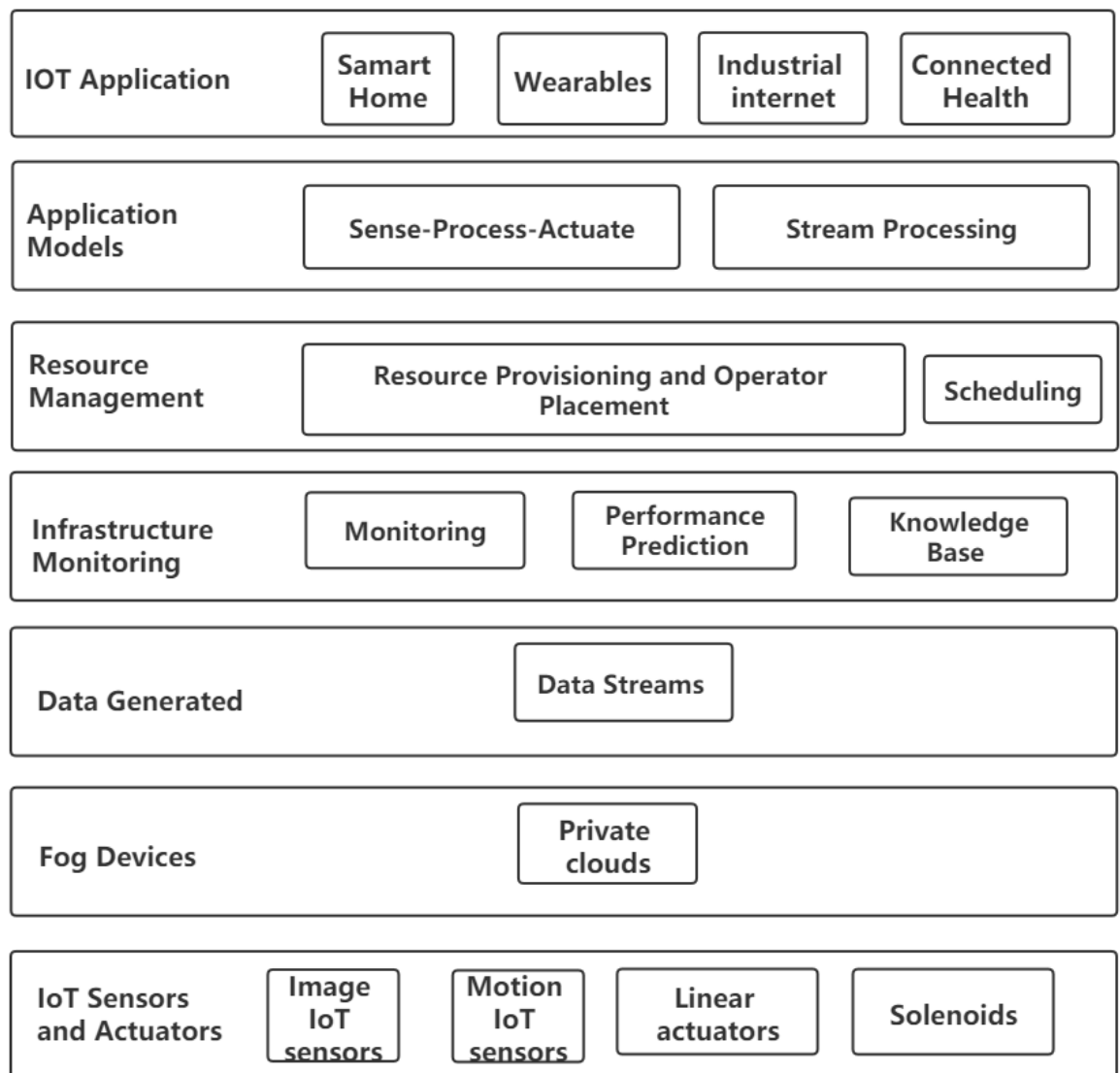


Figure 2.6: The Architecture of iFogSim

iFogSim mainly supports the sense-process-actuate application model. In this model, the sensor publishes the data to the IoT network, the application running on the

fog device receives and processes the data from the sensor, and finally forwards the acquired information to the actuator. The core functionality of iFogSim is done by the CloudSim extension.

$$Fog = Cloud + Internet\ of\ Things\ (IoT)$$

Fog computing is the intermediate layer between the remote cloud and the end user, and addresses the issues of network bandwidth, latency, delay and jitter-all of which can be avoided. It improves the overall performance of the network environment.

The simulator supports the assessment of resource management strategies, focusing on their impact on latency, energy consumption, and network congestion on operating costs. It can simulate the number of edge devices, cloud data centers, and network connections to measure performance. Its system architecture is shown in Figure 2.6.

2.5.4 WorkflowSim

WorkflowSim is a simulator developed by Weiwei Chen et al., university of southern California. In the absence of support for widely used workflow optimization techniques (Deelman et al., 2015), such as task clustering, and in the absence of existing tools that take account of heterogeneous system overhead and failures, CloudSim is extended to accomplish higher-level workflow management tasks (Wangsom, Lavangananda & Bouvry, 2017). The principle is to provide a workflow level simulation based on the existing CloudSim simulation software and introduce a workflow that can better describe more complex big data applications. The Figure 2.7 is the architecture diagram of the WorkflowSim. It models the failure and delay of workflow management system at different levels, then runs it in simulation, and builds a series of popular workflow scheduling algorithms (including HEFT, min-min, max-min) and task clustering algorithm into the system. Run parameters can be obtained and used directly from the

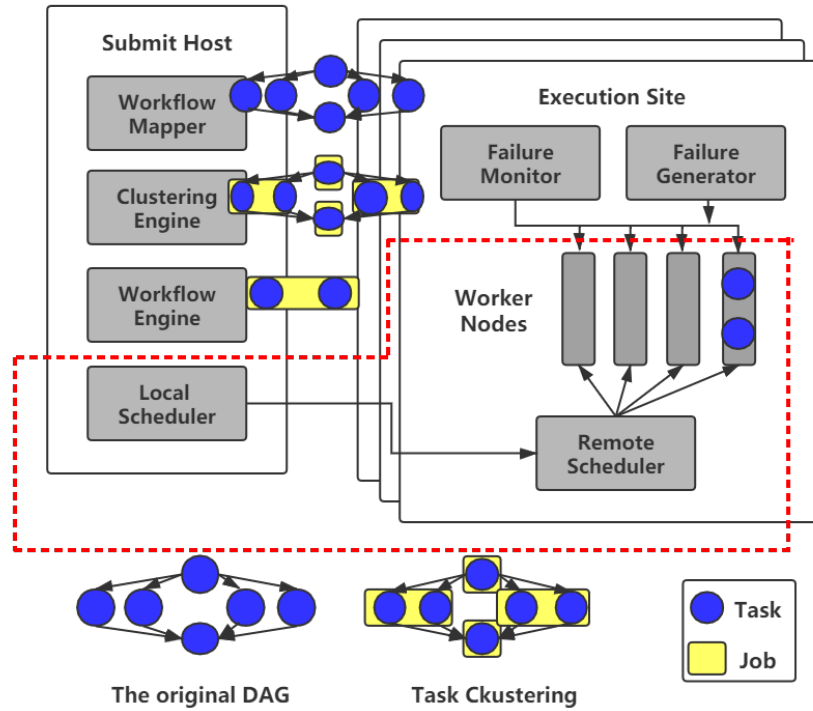


Figure 2.7: The system architecture of WorkflowSim

actual execution of the same type of workflow management system (such as Pegasus) running.

2.5.5 Cloudsim

CloudSim is an open source cloud computing simulator developed by the grid lab Gridbus project team led by professor Rajkumar Buyya from the university of Melbourne (Buyya, Ranjan & Calheiros, 2009). Its primary goal is to quantify and compare the scheduling and allocation strategies of different application and service models on the cloud infrastructure to optimize the allocation of cloud computing resources. CloudSim is extended on the basis of GridSim simulator to support cloud resource management and scheduling simulation. It extends a series of interfaces to provide data-center based virtualization technologies, modeling and simulation of

virtualized cloud environments. As an open source software, CloudSim is written in Java, so it can run cross-platform on Windows and Linux, and users can add their own code according to their own research content, and then recompile the release. The CloudSim architecture is shown in Figure 2.8.

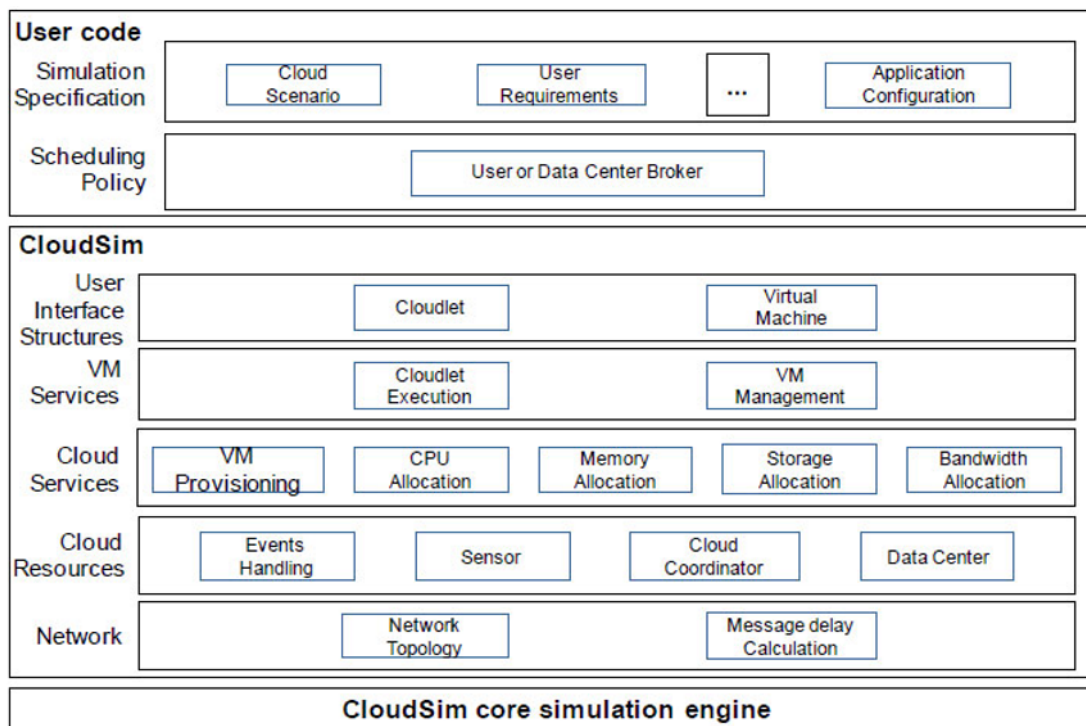


Figure 2.8: CloudSim architecture

The main function of its underlying layer is to handle the communication between entities and components, and the whole emulator is event-driven due to the new event management framework. The second layer is the simulation layer, which supports the modeling and simulation of the cloud-based data center environment, including the dedicated interface management of virtual machine, memory, storage capacity and bandwidth. At the top is the user code layer, which provides interfaces to host entities, applications, number of virtual machines, number of users, application categories, and scheduling policies.

Cloudsim has good portability, support large-scale cloud computing data center

and the joint modeling and simulation of cloud, support for dynamic insert element simulation, and the network center and messaging have good support, in addition to the above advantages, for the user custom task scheduling and host resource allocation strategy, Cloudsim also give the greatest degree of freedom. In recent years, more and more researchers have focused on the secondary development of CloudSim, which also demonstrates its scientific value that cannot be ignored. In conclusion, it is quite appropriate to choose CloudSim as the basic platform for this experiment.

2.6 Summary

In this chapter, we start to study from the definition of data and life cycle, and analyze the existing literature on energy consumption measurement, data quality research, and also the main energy consumption measurement methods of existing data centers. Also made a certain investigation. I found that at this stage, the main way people increase energy consumption lies in how to improve the efficiency of the data center. This efficiency improvement is mainly aimed at optimizing some algorithms in the data processing process. There is no research aimed at grading data in the data preprocessing stage. Therefore, in this article, we will propose a method for applying data grading to cloud computing. Because there are many simulation tools for the cloud computing platform at this stage, we have also done some research on the current simulation platform, and finally selected Cloudsim as the simulation platform for this experiment.

Chapter 3

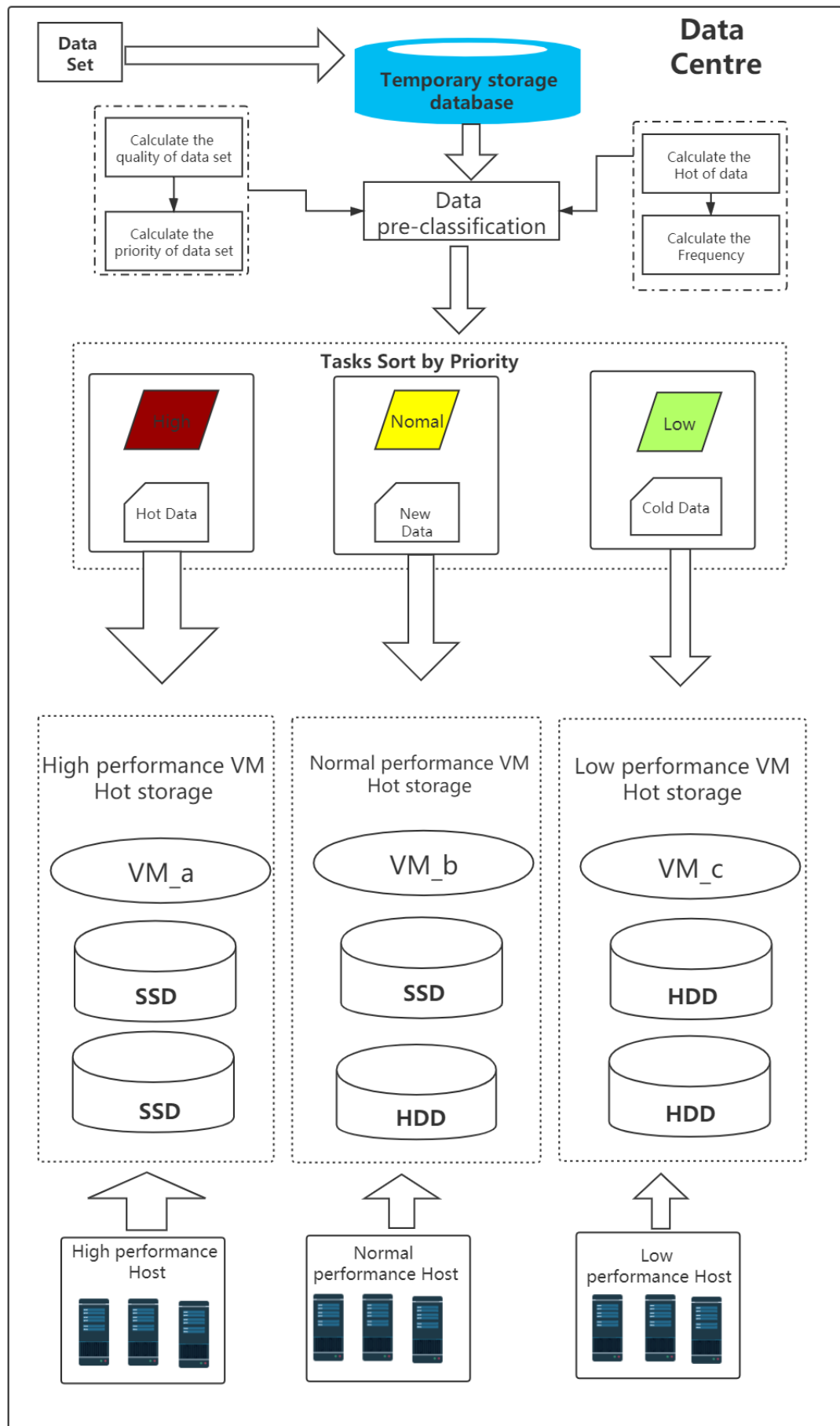
Data-driven energy efficiency indicator model

3.1 Introduction

This chapter mainly introduces the measurement formula of data set quality and the implementation process of applying data quality to cloud computing and cloud storage.

3.2 Measurement of data quality

In all kinds of data quality evaluation methods, relevant definitions are given for the main dimension indexes of data quality, and the important characteristics of each index are emphasized. However, due to different implementation methods, the evaluation results of data quality even for a specific data set are not the same. From the perspective of data quality evaluation index, comprehensive operation needs to be managed through system, process, technology, audit and inspection. In view of the characteristics of large number, high speed, diversity and low value density of data in the big data environment,



this section puts forward a general calculation formula and calculation method for six main data quality indexes. For the elements used in the formula, the unified definition is as follows:

Suppose there are n objects in data set D , which can be defined as:

$$D = \{D_1, D_2, \dots, D_n\}$$

Take any element D_i in D , which has m features and can be defined as:

$$f_i = \{(k_1, v_1), \dots, (k_i, v_i), \dots, (k_m, v_m)\}$$

Where, k represents the feature attribute and v represents the eigenvalue. C_{ij} represents the J th data quality feature of the i th object in the data set. For example, C_{i1} represents the first feature of the i th object, namely the completeness in the following.

Q_i represents the i th data quality feature of the measured data set, Q_1 represents the integrity of the data set.

3.2.1 Data set integrity calculation

Integrity refers to the completeness of the contents described in a given data set relative to the data in a real object set. There are two criteria, one is whether there is a null value and the cause of the null value, the other is whether the correlation between values is clear. We will integrity into two indicators, one is the integrity of the data, the other is the integrity of the object. For the integrity of data, it can be considered that after sampling or discretization of the required data sources, multiple query sampling of several data sources is carried out. Suppose the number of data sources is S and the number of queries is T , then the result set returned by the i TH query of the j TH data source is R_{ij} , then:

$$Q_1 = \frac{1}{T} \sum_{i=1}^T \frac{R_{ij}}{\bigcup_{i=1}^T R_{ij}} \quad (3.1)$$

According to the integrity definition, the complete feature space of object f_i is $f_i = \{k_i, k_3, \dots, k_s\}$, C_{i2} represents the data integrity of the i th object, and Q_2 represents the object integrity of data set D, then:

$$C_{i2} = \frac{\sum_{i=1}^m f_i(v_i)}{m}, Q_2 = \frac{\sum_{i=1}^n C_{i2}}{n} \quad (3.2)$$

3.2.2 Data set consistency calculation

The consistency dimension represents the violation of a semantic rule defined for a set of data items, that is, the degree of consistency between the data sets. In relation theory, the uniform constraint can be divided into two basic categories: inter-relation constraint and inter-relation constraint. Here the first case is called external consistency, represented by Q_3 , and C_{i3} by the external consistency of item I data. For the values of two or more data sources that actually correspond to an object, S_i is used to represent the similarity between the two values, D_{ik} is used to represent the data sources that point to the same objective object with data D_i , and the number i_s represented by nu , then:

$$C_{i3} = \frac{\sum_{i=1}^m \sum_{k=1}^{nu} Si(D_i, D_{ik})}{m * nu}, Q_3 = \frac{\sum_{i=1}^n C_{i3}}{n} \quad (3.3)$$

The second case is called internal consistency, which is expressed by Q_4 and C_{i4} as the internal consistency of item I data. Ra represents the correlation between the two data (in this formula, it is assumed that the two data characteristics are k_i and k_j , respectively, with values of v_i and v_j); D_s represents the logical distance between the data, then:

$$C_{i4} = \frac{\sqrt{\sum_{i=1}^m \sum_{j=1}^m Ra(k_i, k_j) * Ds(v_i, v_j)}}{m}, Q_4 = \frac{\sum_{i=1}^n C_{i4}}{n} \quad (3.4)$$

3.2.3 Data set Accuracy calculation

The accuracy of the data indicates the degree to which the data can accurately reflect the objective things. The smaller the difference between the measurement value of the accuracy and the actual value of the objective things is, the higher the accuracy of the data will be. Accuracy is represented by Q_5 , $f_5()$ is the attribute judgment function, and C_i is the accuracy measurement of the i th data in this set, then:

$$C_{i5} = \frac{\sum_{m=1}^{i=1} f_5(v_i)}{m}, Q_5 = \frac{\sum_{n=1}^{i=1} C_{i5}}{n} \quad (3.5)$$

3.2.4 Data set timeliness calculation

Data timeliness refers to the expectation of time for the accessibility and availability of information, which represents the ability for data to be available when needed. There are h data sources, from T_1 to T_p discrete points in time, the same condition of the query, the query number is q , the result set is $R_{ijk} = \{D_{111}, \dots, D_{kpq}\}$ Let CT be the update function of the obtained results, ET be the earliest time function of the obtained results, $d\{\}$ represent the distance function between vectors (the commonly used Euclidian or Manhattan distance can be used to replace the actual calculation), and Q_6 represent the timeliness measurement of the i th data, then

$$Q_6 = \frac{1}{pq} \sum_{j=1}^p \sum_{k=1}^q d\{CT(R_{ijk}), ET(R_{ijk})\} \quad (3.6)$$

3.2.5 Data Quality assessment model

According to the above data quality evaluation formula, the data quality value can be calculated for the data set, and the priority of the data set can be further calculated.

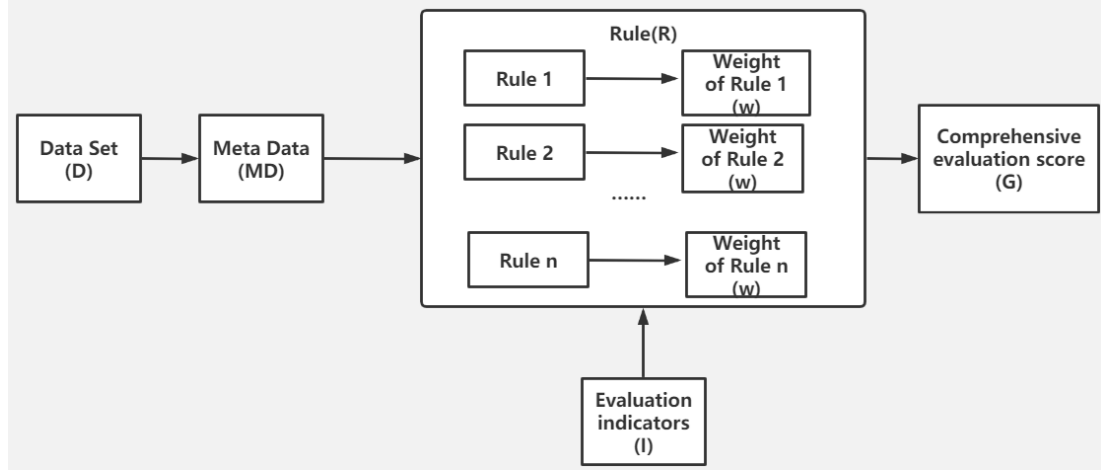


Figure 3.2: Data Quality metric model

As shown in Figure 3.3, the data quality assessment model consists of three parts: data set, rule series and index comprehensive assessment score. The meaning of each node element is as follows:

D: Data set to be evaluated. Data set objects that need to be evaluated.

MD: Metadata, that needs to be evaluated, to extract Metadata to establish the corresponding evaluation Metadata.

R: Rules, specific Rules for each indicator evaluation setting on data set D. The specific calculation formula can be referred to the formula 3.1, 3.2, 3.3, 3.4, 3.5, 3.6.

I: that's Indicators to be evaluated.

W: Weight, the Weight of the rule series, the sum is 1, the Weight value can be customized for the user, in order to explain the proportion of each index in the rule series, the importance of each rule from the user's perspective.

G: Goal, a comprehensive evaluation of the score. The evaluation score reflects the result size of index comprehensive evaluation on data set D after the combination of

each rule series S , that is, the level of data quality.

In the evaluation index system of this thesis, the quality score of the whole data set can be the average score of all indexes. By combining the weight of each rule corresponding to each index, namely the weight of each rule series, the calculation of the data quality evaluation score of each index is obtained as the formula:

$$R_i = \frac{\sum_{j=1}^C w_j * Q_i}{c} \quad (3.7)$$

Where R_i is the evaluation result of the i indicator, C is the total number of evaluation rules, w_j the weight of each rule.

According to the evaluation results of the indicators and the weight of each rule series, the score of the comprehensive evaluation of data quality can be obtained. Formula 3.7 for the score calculation of the comprehensive evaluation is as follows:

$$G = Q = \sum_{i=1}^n R_i \quad (3.8)$$

3.3 An energy-saving scheduling algorithm based on data quality (DQ-TSA)

3.3.1 The introduction and process of cloud computing task scheduling

In the cloud computing environment, a wide range of application groups will submit massive tasks and generate different forms of service demands all the time, so the amount of data that needs to be processed in the cloud computing data is very large. How to assign tasks to appropriate virtual machines, efficiently implement task scheduling and meet the corresponding goals has become an important problem to be

solved in cloud task scheduling, is also a NP-Complete problem (Navimipour & Milani, 2015).

In general, the task set submitted by users will be divided into multiple MAP and Reduce tasks by the MAP and Reduce process of the cloud computing distributed framework. The task model composed of these tasks is the cloud computing task model. Once the number of these sub-tasks reaches a certain width, they will be uniformly submitted for scheduling (Zhang & Zhou, 2017). Task scheduling can be divided into independent task scheduling and workflow task scheduling according to the data interdependence between tasks (Geng, Mao, Xiong & Liu, 2019). Independent task refers to that all tasks scheduled are relatively independent and they will not depend on each other, while workflow task refers to a set of interdependent and mutually constrained tasks. Task scheduling is based on the constraint conditions between the tasks assigned to for its also in the appropriate computing resources in data processing, and through the virtualization technology, data also in each physical HOST will be mapped to a virtual machine, so the user submits in the process of task scheduling of each task according to the appropriate scheduling policy choice only is matched with the virtual machine can complete the whole scheduling process.

Generally, task scheduling can be divided into several steps: task submission, task demand analysis and search for available resources, resource selection, task scheduling and real-time monitoring of computing resources.

Suppose there are hosts P in the cloud data centre, and each host can be mapped to multiple virtual machines (VM) . $VM = \{VM_1, VM_2, \dots, VM_m\}$. And assume that the user has also submitted i tasks to the cloud data as $T = \{T_1, T_2, \dots, T_i\}$.

Cloud computing task scheduling is that when the user submits i tasks, the task analyzer will analyze the corresponding user requirements of the task and submit these requirements to the cloud proxy server. The proxy server can find the appropriate computing resources according to the user's requirements as an alternative resource set.

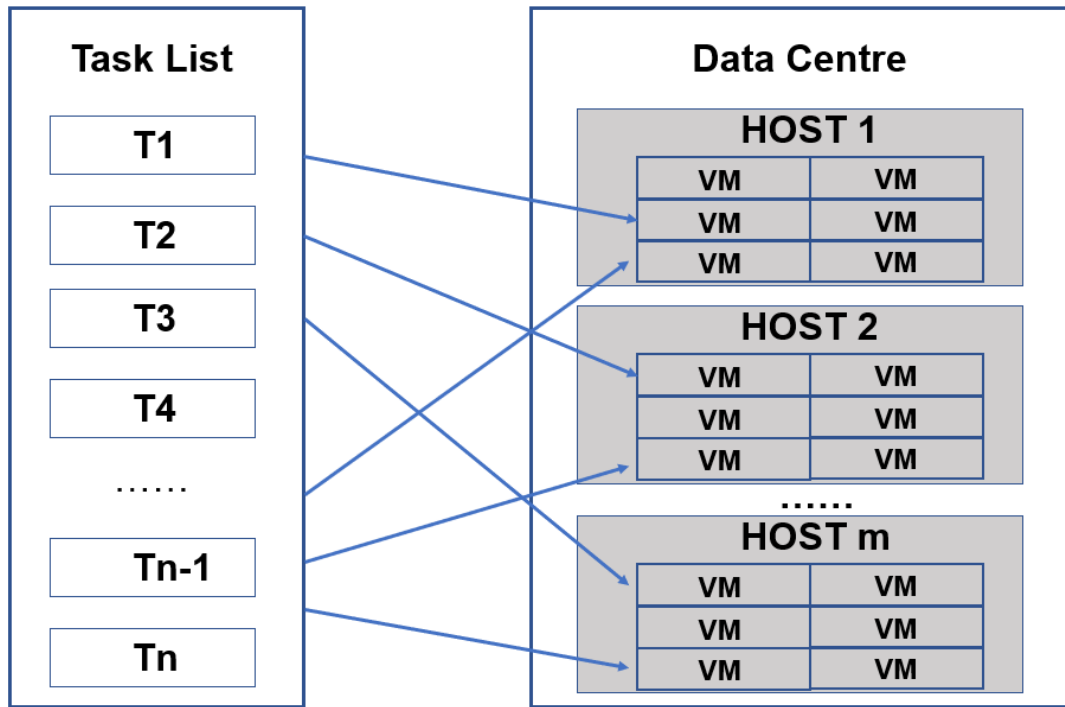


Figure 3.3: Tasks schedule model

These resources can be filtered in accordance with user satisfaction. If the alternative resource set is already occupied by another task or is in a state where it is not available, other computing resources in the alternative resource set can be selected for the task. Once a relatively matched computing resource is selected, the task is assigned to the selected computing resource according to the corresponding scheduling scheme under the condition of satisfying user demand. In addition, the operation of computing resources needs to be monitored in real time during task scheduling. Once the user releases or adds new computing resources, the information of available resources in the proxy server needs to be updated in a timely manner to facilitate the next computing resource allocation.

Where, each task can only be scheduled once, and the number of tasks is more than the number of virtual machines.

3.3.2 Cloud task scheduling model based on data quality

As commercial calculation model, cloud computing must meet the needs of users and experience, the other cloud service providers also to maximize the utilization of computing resources and energy consumption to improve the service benefits, usually a cloud task scheduling algorithm of concerns mainly include four aspects, minimizing the mission completion time, minimize the task execution cost, reliability, and load balancing.

From an energy saving perspective, when assigning a batch of tasks to multiple virtual machines, the task end time depends on the time of the last virtual machine to end. If some virtual machines finish tasks very early and others finish tasks very late, the resources of virtual machines that finish early will be wasted. Therefore, the task completion time of each virtual machine should be as close as possible. For this purpose, naturally, we assign long-running tasks first and short-running tasks later, because it is easier to align the virtual machine's task finish time. In addition, it's not just a matter of task duration, it's important to take full advantage of the performance of the virtual machine at every moment. That is, a virtual machine that always has a 90% CPU utilization and a virtual machine that always has a 60% CPU utilization can make better use of resources and therefore complete tasks faster. Therefore, it is important to pay attention to the final task completion time, task running time and CPU utilization.

In addition, the main purpose of cloud computing is to improve the service satisfaction of users. High-quality data means that the data submitted to these tasks is more user-level and timely than other data, in other words, the value of the data itself is proportional to the quality of the data. Therefore, high quality, high value data/tasks should be prioritized. High quality, high value tasks will be scheduled to better resources, provide the best resource allocation, and ensure the quality of service.

Therefore, in order to simultaneously meet the basic needs of users and providers

of cloud services, this paper consider the tasks at the same time, the execution time and load balancing three properties, design a priority task scheduling algorithm based on data quality (DQ-TSA), virtual fleet column assembly according to priority size real-time scheduling, which in the case of guarantee customer satisfaction to achieve the purpose of energy saving.

3.3.3 The idea of DQ-TSA

In cloud computing, users have different task sizes and types, and the underlying resources of cloud computing are heterogeneous, which will bring unbalanced problems to the system. The performance effect of load balancing will affect the overall performance and resource utilization of cloud computing system.

DQ-TSA algorithm is proposed to improve the load balancing degree of the virtual machine on the basis of guaranteeing the minimum task scheduling span and energy consumption. The algorithm uses the following method to measure the load of the virtual machine: the sum of the execution time of the tasks executing on the virtual machine and the tasks in the waiting state on the virtual machine is the load of the virtual machine. As Figure 3.4.

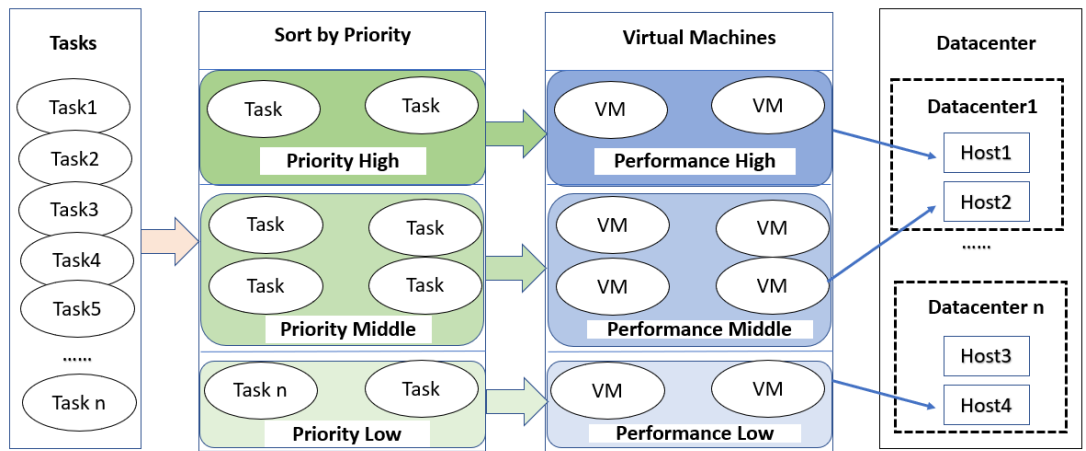


Figure 3.4: DQ-TSA Algorithm Tasks schedule model

DQ-TSA algorithm idea is as follows:

1) Standardize the specified attributes of the data set, and calculate the data quality (as fomular 3.12), task running time^{3.14} and required resources of the task^{3.13}. The task set is sorted with data quality as the first priority(P1). In the case of the same data quality, the task with long running time and more CPU resources will be executed first.

2) Calculate the total service capacity of resources,use formula 3.20, 3.19 and group them into groups, establish the scheduling constraint association between task group and resource group, and schedule high-priority tasks to better resources to provide the best resource configuration and guarantee the quality of service.

3) Calculate various resources respectively and give a standard value $Normal(t, R_i, H_k)$. The resource with the largest standard value is regarded as the main resource of the task. Traversing the task, assigning the task to the virtual machine with the least standard load of the current main resource.

Suppose there is a cloud data center with different types of physical host H , Host set $H = \{H_a | 1 \leq a \leq b\}$, H_a Represents the $H_a = \{ID, Comp, Bw, Io, Sto, EC\}$. This formula represents the multidimensional attributes of the P , which are host ID, host computing power, communication power, Io power, storage power and energy consumption. The host set can be expressed as the matrix below:

$$H_a = \begin{bmatrix} H_1^{ID} & H_1^{Comp} & H_1^{Bw} & H_1^{Io} & H_1^{Sto} & H_1^{EC} \\ H_2^{ID} & H_2^{Comp} & H_2^{Bw} & H_2^{I/O} & H_2^{Sto} & H_2^{EC} \\ & & & \dots & & \\ H_b^{ID} & H_b^{Comp} & H_b^{Bw} & H_b^{Io} & H_b^{Sto} & H_b^{EC} \end{bmatrix} \quad (3.9)$$

The task set T , $T = \{Task_i | 1 \leq i \leq n\}$, $Task_i$ Represents the task i , $Task_i = \{ID, Length, source, DQ, Comp, Bw, I/O, Sto\}$.

This set represents the multidimensional attributes of the task, which are task id,

task length, task source, task quality, task computing power demand, task bandwidth demand, task IO demand and task storage capacity demand respectively. Among them, the calculation method of task quality as formula 3.8. The task set can be expressed as the matrix below:

$$T_i = \begin{bmatrix} Task_1^{ID} & Task_1^{Length} & Task_1^{source} & Task_1^{DQ} & Task_1^{Comp} \\ Task_1^{Bw} & Task_1^{I/O} & Task_1^{Sto} & & \\ Task_2^{ID} & Task_2^{Length} & Task_2^{source} & Task_2^{DQ} & Task_2^{Comp} \\ Task_2^{Bw} & Task_2^{I/O} & Task_2^{Sto} & & \\ & & & & \\ Task_n^{ID} & Task_n^{Length} & Task_n^{source} & Task_n^{DQ} & Task_n^{Comp} \\ Task_n^{Bw} & Task_n^{I/O} & Task_n^{Sto} & & \end{bmatrix} \quad (3.10)$$

Vm set $Vm = \{Vm_j | 1 \leq j \leq m\}$, Vm_i Represents the $Vm_j = \{ID, Comp, Bw, I/O, Sto, EC\}$. This formula represents the multidimensional attributes of the Vm, which are Virtual machine ID, virtual machine computing power, communication power, Io power, storage power, energy consumption. The Vm set can be expressed as the matrix below:

$$Vm_j = \begin{bmatrix} Vm_1^{ID} & Vm_1^{Comp} & Vm_1^{Bw} & Vm_1^{I/O} & Vm_1^{Sto} & Vm_1^{EC} \\ Vm_2^{ID} & Vm_2^{Comp} & Vm_2^{Bw} & Vm_2^{I/O} & Vm_2^{Sto} & Vm_2^{EC} \\ & & & & & \\ Vm_m^{ID} & Vm_m^{Comp} & Vm_m^{Bw} & Vm_m^{I/O} & Vm_m^{Sto} & Vm_m^{EC} \end{bmatrix} \quad (3.11)$$

The first data priority by mathematical formula, Where xy is the weight of each attribute, $T_{quality} = G$

$$P1 = x * T_{source} + y * T_{quality} \quad (3.12)$$

$R_i(H_j)$ represents the total amount of resources R_i owned by host H_j , the resource

standard load of a task is used to measure the resource load of a task. The standard load is calculated as follows:

$$\text{Load}(t, R_i, Vm_j) = \text{time}(t) \text{Normal}(t, R_i, H_k) \quad (3.13)$$

Standard(t, R_i, H_k): The standard value of resource R for the virtual machine assigned by task x to host H_k .

When task i is scheduled to execute on virtual machine j , if there are still executing or waiting tasks in virtual machine j , in the case of the same priority, according to the principle of "first come, first serve", so task i also needs to wait for all tasks in front of it to complete before starting to execute. The completion time of task i on Vm j is the sum of the queuing time of task i on Vm j and the execution time on Vm j .

Task running time :

$$\text{time}(t) = \frac{T^{Length}}{R_{Comp}(Vm)} \quad (3.14)$$

Task completion time :

$$\text{expTime}(t) = \text{time}(t) + \text{wait}(t) \quad (3.15)$$

The standard load on a virtual machine's resources is a measure of how much the virtual machine has to do with that resource, and is the sum of the standard load on that resource for all the tasks that are not executed and are being executed in the virtual machine.

$$\text{Load}(Vm_j, R_i) = \sum_{t \in E(i)} \text{Load}(t, R_i, Vm_j) \quad (3.16)$$

The comprehensive results obtained by data of different properties during addition and subtraction operation cannot correctly reflect the impact of the data on the result, so the data need to be pre-processed and standardized here. The processed set of virtual

machine attributes is represented by a matrix ST :

$$ST(Vm) = \begin{bmatrix} Vm_1^{ID} & Vm_{1.st}^{Comp} & Vm_{1.st}^{Bw} & Vm_{1.st}^{I/O} & Vm_{1.st}^{Sto} & Vm_1^{EC} \\ Vm_2^{ID} & Vm_{2.st}^{Comp} & Vm_{2.st}^{Bw} & Vm_{2.st}^{I/O} & Vm_{2.st}^{Sto} & Vm_2^{EC} \\ & & & \dots & & \\ Vm_m^{ID} & Vm_{m.st}^{Comp} & Vm_{m.st}^{Bw} & Vm_{m.st}^{I/O} & Vm_{m.st}^{Sto} & Vm_m^{EC} \end{bmatrix} \quad (3.17)$$

$Vm_{j.st}^{Comp}$ represents the result after comp attribute data processing of the computational performance of the J th virtual machine, which can be calculated by referring to the following formula. Other virtual machine property values are also processed in this standardized way.

The processed set of Host attributes is represented by a matrix ST :

$$ST(H) = \begin{bmatrix} H_1^{ID} & H_{1.st}^{Comp} & H_{1.st}^{Bw} & H_{1.st}^{I/O} & H_{1.st}^{Sto} & H_{1.st}^{EC} \\ H_2^{ID} & H_{2.st}^{Comp} & H_{2.st}^{Bw} & H_{2.st}^{I/O} & H_{2.st}^{Sto} & H_{2.st}^{EC} \\ & & & \dots & & \\ H_b^{ID} & H_{b.st}^{Comp} & H_{b.st}^{Bw} & H_{b.st}^{I/O} & H_{b.st}^{Sto} & H_{b.st}^{EC} \end{bmatrix} \quad (3.18)$$

The overall performance of the virtual machine resources is calculated as follows:

$$R(Vm)_j = \sqrt{(Vm_{j.st}^{Comp})^2 + (Vm_{j.st}^{Bw})^2 + (Vm_{j.st}^{I/O})^2 + (Vm_{j.st}^{Sto})^2} \quad (3.19)$$

The overall performance of the Host resources is calculated as follows:

$$R(H)_a = \sqrt{(H_{a.st}^{Comp})^2 + (H_{a.st}^{Bw})^2 + (H_{j.st}^{I/O})^2 + (H_{a.st}^{Sto})^2} \quad (3.20)$$

The expected energy consumption of a task($expEC(T_i)$.) refers to the computing resources, bandwidth resources, Io resources and storage resources of Vm_j to be used for the execution of task i on Vm_j . Thus, the power consumption of virtual machines

with stronger service capacity should be higher than that of virtual machines with weaker service capacity.

$$expEC(T_i) = Vm_j^{Comp} * time(t) + Task_j^{Bw} + Task_j^{I/O} + Task_j^{Sto} \quad (3.21)$$

The comprehensive energy efficiency value $TimeEc(T_i)$ refers to the weighted processing result of the execution time and energy consumption of task i on virtual machine j . The smaller the value, the lower the task completion time and energy consumption.

$$TimeEc(T_i) = \frac{expEC(T_i)}{avgEC(T_i)} + \frac{expTime(T_i)}{avgTime(T_i)} \quad (3.22)$$

The pseudo code:

Algorithm 1 DQ-TSA

- 1: Task, resource matrix initialization, T, Vm ,
 - 2: Calculate the priority of the task, P ,
 - 3: Sort the task in order of priority from high to low,
 - 4: Calculate all Host resource service capabilities, $ST(H)$,
 - 5: Calculate all virtual machine resource service capabilities, $ST(Vm)$,
 - 6: Rank virtual machine resources in order of their service capabilities,
 - 7: Initialize the task execution time matrix, $time(T_i)$,
 - 8: Initialize task completion time matrix, $expTime(T_i)$,
 - 9: Initialize expected energy consumption of a task matrix, $expEC(T_i)$,
 - 10: Determine whether the task queue is empty, if not, take one task, T_x ,
 - 11: Calculate the load of the T_x on each virtual machine, $Load(t, R_i, Vm_j)$,
 - 12: Schedule tasks to the virtual machine that corresponds to the minimum load,
 - 13: Update task completion time matrix, $expTime(T_x)$,
 - 14: Update expected energy consumption of a task matrix, $expEC(T)$.
 - 15: return Task list.
-

DQ-TSA algorithm aims to improve the system load balancing degree of task scheduling on the premise of ensuring low energy consumption and short completion time. This algorithm also takes load balancing as an optimization objective when carrying

out intra-group scheduling. Before scheduling, the current load of each resource should be determined to prevent scheduling on the resource from overloading the resource.

The process of DQ-TSA algorithm, 3.5

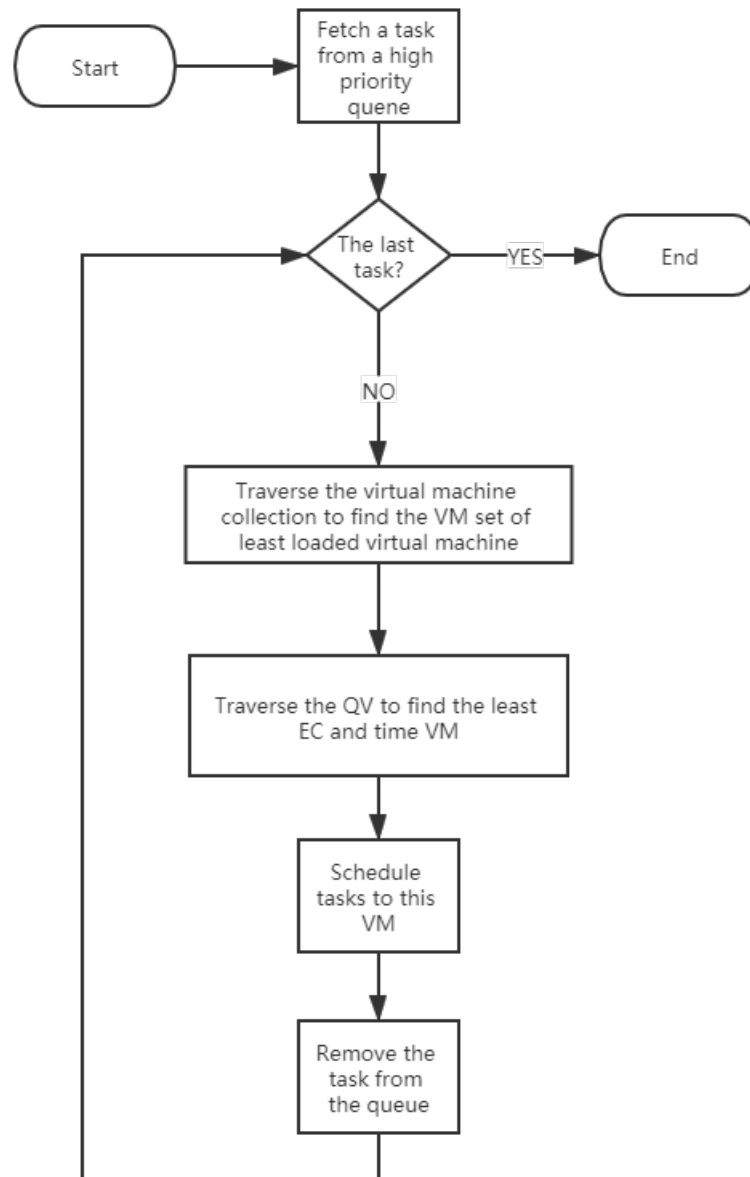


Figure 3.5: The process of DQ-TSA algorithm

3.4 An energy-saving storage algorithm based on data quality(DQ-HSA)

At present, the cloud storage technology research content includes the data security, data access control, data migration, data and direction, such as to improve the system's high availability and storage performance and reducing energy consumption as a basic part of the cloud storage technology system (Hale, 2013). In order to realize the system resource utility maximization, optimized data storage place is direct and effective means.

In the existing researches on cloud storage data optimization storage placement, data access rules are mostly monitored through metadata information of data files, and data storage placement decisions are made according to their respective data storage value evaluation methods, so as to meet some system performance optimization objectives. Other schemes rely on the functional features inherent in the cloud storage system or the newly applied optimization technology to make data optimization storage placement decisions. The optimization schemes proposed in this part of the study have their own performance optimization focus and technical characteristics, and have different degrees of system storage utility improvement, but there are still the following problems.

1) Due to the huge amount of data in the cloud storage system, frequent reliance on metadata to fully monitor data access rules will also consume system resources and bring a large additional cost. At the same time, for the stored data, the factors to evaluate the data quality should also fully consider the data heat evaluation basis and refine the classification of the data.

2) The performance structure of device resources in cloud storage system is complex, and the heterogeneous nature is obvious. The homogeneous data placement and layout method is difficult to apply to the differentiated device configuration.

3) There are many performance optimization indexes of cloud storage system,

which are related to each other and restrict each other. Therefore, the performance optimization direction of cloud storage system needs to be considered on the whole.

Based on the above research and the shortcomings of existing technology, this paper presents a decision scheme of hierarchical cloud storage based on data quality. Firstly, according to the quality of data and the existing data classification method (You, Dong, Zhou, Huang & Jiang, 2015), a method of data classification and hierarchical storage based on data quality (DQ-HSA) is designed to improve the efficiency of hierarchical decision making. At the same time, by considering the storage performance indexes of cloud storage system, a hierarchical decision-making method of multi-objective optimization was designed and realized based on the decomposing multi-objective optimization framework. The model presented in this paper takes into account four system performance metrics that most hierarchical schemes take into account, including average access time, average access delay, average migration cost, and load variation.

3.4.1 The system model of DQ-HSA

Given that there are m different storage devices and n different data files to be stored in the cloud storage cluster, the hierarchical processing process can be summarized as: the storage placement mapping process from the data items in the list of files to the list of devices. See Figure 3.6.

The Figure 3.6 shows the basic data storage placement process in the cloud storage system, so it can be seen that the main processing objects of optimization storage placement decision are data files and storage devices. Therefore, the main work of the system model in this paper can be simply summarized as obtaining data quality and data heat distribution, organizing storage devices, and optimizing data storage placement.

In the actual application process of cloud storage, the data access arrival pattern roughly follows the *Zipf* distribution and has diversified access characteristics (Abad,

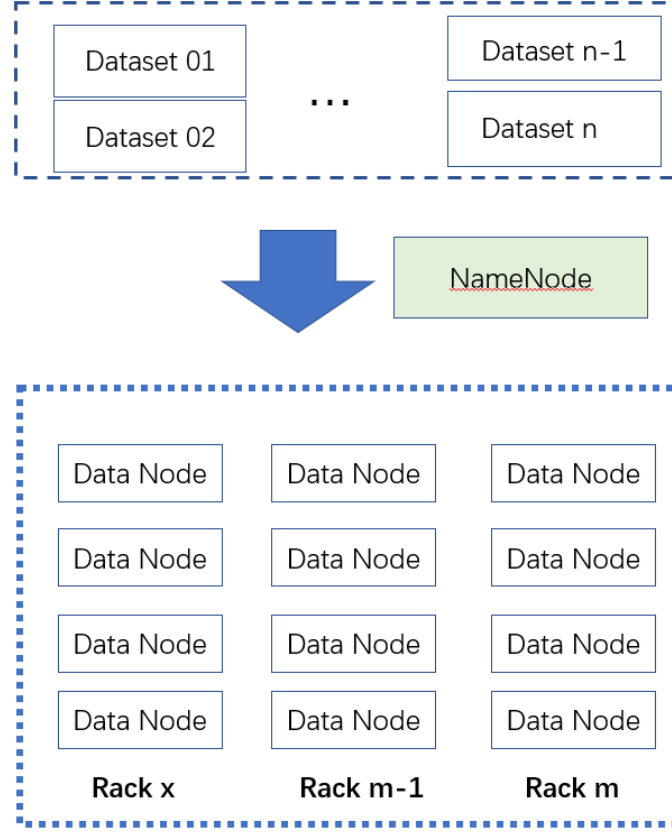


Figure 3.6: Data storage placement in cloud storage environment

Luu et al., 2012). The proposed hierarchical cloud storage scheme should fully consider the access characteristics of data. In addition, during the construction of cloud storage cluster, most cloud storage service providers use mixed and heterogeneous device management schemes including SSD, HDD, NAS, etc. (Hale, 2013), which can provide corresponding data storage services according to the storage requirements of users. Therefore, the proposed hierarchical cloud storage also needs to meet the storage requirements of users and device performance constraints, combined with device hierarchical organization and management, to select the optimal storage target for data.

Combined with the above research and problem analysis, the hierarchical cloud storage model proposed in this paper manages cloud storage resources through the organization of three-level storage classification, so as to store and place data with

different storage values at different levels (Kakoulli & Herodotou, 2017). Data storage quality assessment based on file granularity can comprehensively evaluate data quality and locate hot data in the cloud storage environment with complex data access characteristics (Xie & Sun, 2009). Therefore, this paper conducts a heat assessment on the file data stored in the cluster and carries out a fine-grained classification to pre-configure matching storage targets for each type of data (Zhou, Feng, Tan & Zheng, 2018). Finally, with the average access time, average access delay, average migration cost and load change as the optimization objectives (Long, Zhao & Chen, 2014), the decomposition based multi-objective optimization method (Carvalho, Saldanha, Gomes, Lisboa & Martins, 2012) was applied to make the optimal decision on the placement mapping of data to storage targets. Cloud storage cluster manages heterogeneous storage devices and provides resource-shared data storage services. The application of hierarchical storage scheme can fit the heterogeneous characteristics of cluster devices and optimize the placement of data storage. Therefore, on the basis of Figure 3.6, the hierarchical cloud storage data placement optimization model proposed in this paper is shown in Figure 3.7.

3.4.2 Functional module design of DQ-HSA

Based on the above discussion, the modules of DQ-HSA proposed in this thesis mainly include hierarchical organization, value assessment and data pre-classification, as well as data storage placement decision modules.

1) Hierarchical organization management module: by evaluating the storage performance of devices in the cloud storage cluster, the storage devices are divided into three hierarchical organizational structures: hot storage, basic storage and cold storage.

2) Data pre-classification module based on value assessment: according to the principle of information life cycle, by evaluating the storage value of data in different stages,

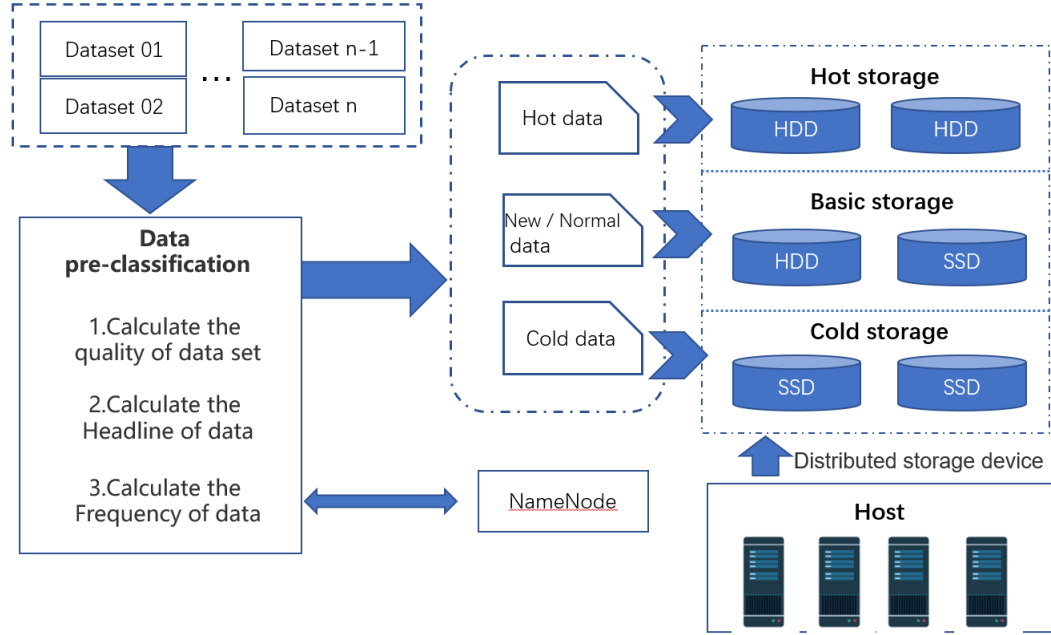


Figure 3.7: The system model of DQ-HSA

the data is pre-processed by classification, and the storage target area of corresponding level is pre-selected.

3) Data storage placement decision module: through the improved multi-objective optimization algorithm, the optimal storage device is selected for each data.

Hierarchical organization management module

For m Storage devices in the cluster, by calculating their Storage performance, the hashed Storage device set is divided into Storage Level (SL) according to a certain ratio, that is, the three-level Storage structure of hot Storage SL1, basic Storage SL2 and cold Storage SL3, as shown in Figure 3.7. Different from the device hierarchical organization management scheme proposed in the existing hierarchical storage method (Kakoulli & Herodotou, 2017), the hierarchical management of heterogeneous mixed cloud storage cluster resources can reduce the implementation difficulty of the scheme to a certain extent and facilitate the expansion of storage function of the system.

As can be seen from Figure 3.7, the three-level storage structure in this paper can

cope with complex data quality classification and data headline performance. Among them, the hot spot storage layer is a set of high-performance storage devices, which is used to store hot spot data with a relatively small amount of data and a high frequency of access, so as to ensure a high Io rate of hot spot data. The basic storage layer has a general performance and a large number of devices, which are used to store a large number of data with a general heat performance, so as to balance the load of the system and improve the throughput of the system. The storage device of cold zone storage layer has low performance, low cost and easy access, and is used to store the data with low value density and long time unaccessed, so as to ensure the availability of data.

Data pre-classification module

In the cloud storage environment, there is a large amount of data storage. Before the final storage optimization and placement decision, it is necessary to conduct feature processing on the data and analyze the distribution of data value to improve the performance optimization ability. Due to the files stored in the cluster, the value of data storage is constantly changing in the information life cycle, and frequent data optimization storage places increase the system consumption.

Therefore, the data pre-processing classification algorithm proposed in this paper only performs data access feature statistics within the stage time ΔT . The stage time ΔT is a dynamically configurable time interval, whose size is shown by the frequency of the system carrying out data access feature statistics, and the usual configuration parameter is 12 or 24 hours. Then calculate the data heat of the data file within the stage time ΔT and use this as the data classification basis.

Data classification and hierarchical storage based on data quality algorithm (DQ-HSA)

This thesis integrates the above hierarchical organization management module with the data pre-classification module, that is, the pre-selection process of the data storage target area, which can be described as an algorithm 2 : DQ-HSA.

Algorithm 2 Data classification and hierarchical storage based on data quality algorithm (DQ-HSA)

```

Input FileList  $F = \{f_1, f_2, \dots, f_n\}$ 
,   StorageList  $S = \{s_1, s_2, \dots, s_m\}$ 
Output  $FSML$  // A mapping collection of pending data files and a list of preselected storage devices
1: for each  $s_j$  in StorageList  $S = \{s_1, s_2, \dots, s_m\}$ 
2:    $PVS_j = \text{Performance Calculation}(S_j)$ 
3:    $Level_j = \text{Storage Tiering}(PVS_j, S_{\text{left}}, S_{\text{right}})$ 
4:    $SL \cup \{s_j, level_j\}$ 
5: end for
6: for each  $f_i$  in FileList  $\{f_1, \dots, f_i, \dots, f_n\}$  do
7:    $TF_i = \text{Temperature Calculation}(f_i)$ 
8:    $DCLlist = \text{Data Classification}(TF_i, \text{nowTime})$ 
9: end for
10: for each  $f_i$  in  $DCLlist$  do  $FSM(f_i) = \text{Selection}(f_i, SL)$ 
11: end for
12:

```

To classify the calculated files, the classification rules are: According to the classification of data quality and data heat value, the files are divided into hot, normal, and cold data. For the newly created file, because the access operation probability may be higher in the next stage, and its current access volume is less, it is not conducive to assess its popularity. Therefore, this algorithm directly divides this part of data into normal data.

The processing steps of this algorithm are as follows

First, calculate the quality of the data set according to the calculation formula of data quality 3.8.

Secondly, according to the number of file accesses during the time ΔT of the statistical stage, the access heat is calculated to evaluate the storage value. In the algorithm DQ-HSA, take the attribute tuple of the file $FV = \{FV_{size}, FV_{User}, FV_{Io}, FV_{LastTime}, FV_{RW}, FV_{TF}\}$. The calculation formula of TF is as follows:

$$TF = \frac{FV_{user} * FV_{Io} * \frac{FV_R}{FV_W}}{FV_{size} * (FV_{LastTime} - nowtime) * C_{TF}} \quad (3.23)$$

$\frac{FV_R}{FV_W}$ is the file read-write ratio, which means that the more file read operations, the higher the heat value, and $nowTime$ is the current time of the system. Based on the file's most recent access time, it is used to assess the frequency of recent file access operations, C_{TF} is the constant for evaluating file heat value.

In addition, the performance of each storage device should be calculated, and the storage device should be divided into three hierarchical structures. Take the performance set as $PV = \{CPU, storage\ capacity, read/write\ rate, I/O, bandwidth, latency, energy\ consumption...\}$. For performance evaluation and calculation. Therefore, for each storage device S_j , the performance calculation method shown in formula 3.24.

$$PV S_j = \sum_{k=1}^N \varepsilon_k PV_k \quad (3.24)$$

Where, PV_k is the performance attribute of the device, and ε_k is the evaluation parameter of the performance attribute. Sort all storage devices according to their performance, and grade them by the storage cabinet value $S_{left}; S_{right}$, divides the original storage device set S into hot storage layer SL1, basic storage layer SL2, and cold storage layer SL3.

Finally, the optimized storage area is selected for the data divided by storage value. The pre-selection rule is to pre-select the underlying storage layer for normal data or new data, and select the storage area for promotion and degradation of old data

according to the change of storage value through threshold comparison. As Figure 3.8

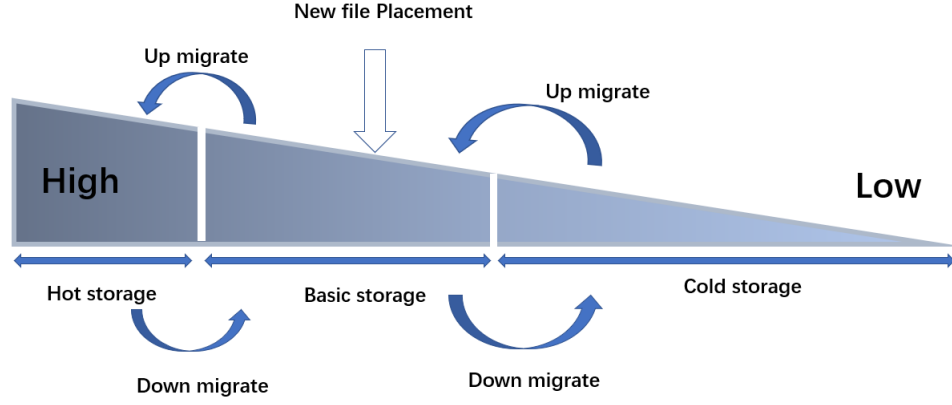


Figure 3.8: Preselection rules for hierarchical storage of data

The relevant parameters of this algorithm are shown in Table 3.1

Table 3.1: The relevant parameters of DQ-HSA algorithm

Parameters	Parameter Description
$\dagger FV$	the attribute tuple of the file $FV = \{FV_{size}, FV_{User}, FV_{I/O}, FV_{LastTime}, FV_{RW}, FV_{Quality}\}$.
$\dagger FV_{size}$	The size of file
$\dagger FV_{User}$	User number
$\dagger FV_{I/O}$	Io rate
$\dagger FV_{LastTime}$	Time of last visit
$\dagger FV_{RW}$	Read/write ratio
$\dagger FV_{Quality}$	The quality of file
$\dagger PV$	the attribute tuple of the file $PV = \{PV_{CPU}, PV_{RAM}, PV_{I/O}, PV_{bw}, PV_{delay}, PV_{EC}\}$.
$\dagger PV_K$	The k th performance attribute of the storage device
$\dagger PV_{S_j}$	The performance evaluation value of storage device S_j

Continued over page

Table 3.1: Extended version. . . (*continued*)

Parameters	Parameter Description
$\dagger \quad \varepsilon_k$	The k th attribute evaluation parameter of the storage device
$\dagger \quad S_{left}, S_{right}$	Thresholds for performance grading of storage devices
$\dagger \quad \Delta T$	Periodic statistical evaluation interval
$\dagger \quad TF$	The heat value of the current phase of the data file
$\dagger \quad FSML$	A mapping collection of pending data files and a list of preselected storage devices

3.5 Summary

In this chapter, we propose the measurement and calculation methods of six important data quality indicators in the big data environment. At the same time, according to the formula of data quality measurement, an energy saving scheduling algorithm based on data quality (DQ-TSA) and an energy saving storage algorithm based on data quality (DQ-HSA) are proposed in the cloud data center.

In the DQ-TSA algorithm, when a user submits a task to the cloud data center, it first calculates the quality level of the data, using the quality of the data, the source, and the length of the task as the adjustment parameters of the task priority. Sort tasks by task priority, and then traverse the task to assign it. The data with high priority will be assigned a high performance Vm and will be executed first. In the process of task execution, the load of Vm will also be concerned, and the task will be assigned to the virtual machine with the least resource load.

The core idea of the DQ-HSA algorithm is similar to the DQ-TSA algorithm, which is to first evaluate the data quality of the stored data, and classify the stored data by data quality and data popularity. And divide the storage device into a three-level

hierarchical organizational structure of hot storage, cold storage and ordinary storage. According to the priority of the data, select the optimal storage device for each data.

In the next chapter, we will conduct a large number of simulation studies to verify and evaluate these two proposed algorithms. The purpose is to verify that these two algorithms are more energy efficient than traditional cloud data center resource scheduling and cloud storage algorithms.

Chapter 4

Simulation studies

4.1 Introduction

In this chapter, we will use the cloudSim as simulation platform to verify the DQ-TSA algorithm and DQ-HSA algorithm proposed in the previous chapter.

In the next section, we'll introduce the running environment of cloudsim and its core.

4.2 Cloudsim Simulations

CloudSim needs to run in a Java runtime environment. The development environment used in this paper is as follows:

Operating System: Windows 10 professional edition 64-bit

System Model: ASUS ROG GX531

Processor: Intel(R) Core(TM) i7-9750 CPU @ 2.60GHz

Memory: 16384MB RAM

Runtime Environment: JRE 1.8.0_131

The configuration process of Cloudsim is very simple. First, to go to the official website (<http://www.cloudbus.org/cloudsim/>) to download the toolkit. After the download is complete, unzip the toolkit, create a new project in it, add a custom name to this project, and add all the packages in the directory to this one. In this way, the configuration work is completed, in this project, you can call the corresponding various classes and interfaces in accordance with the custom cloud computing task scheduling requirements.

The core classes of Cloudsim, as shown in table 4.1.

Table 4.1: Core classes in Cloudsim

Core classes	The main function
† Cloudlet	Used to build the tasks (Cloudlet) submitted by the user to the DatacenterBroker, which allows the user to configure the number of tasks, CloudletLength, and Cloudletid properties.
† Datacenter	It is used to model cloud data as well, to customize the configuration of the property W in the data as well as parameter values such as number of virtual machines, computing energy and memory, and to define virtual machine allocation policies in this class.
† DatacenterBroker	This class is used in the simulation data as a proxy. The relevant CIS are queried through the user's quality of service requirements and the cloud service provider that can meet the requirements is selected. Researchers can design a series of task scheduling policies in this class and evaluate and test them.

Continued over page

Table 4.1: Core classes in Cloudsim. . . (*continued*)

Core classes	The main function
† Host	This class is a host class that defines properties such as the id of the physical resource, the memory size, VmScheduler, and so on. Each data also contains at least one physical resource, and each physical resource can be instantiated into multiple virtual machines.
† Vm	It is used to simulate the virtual machine instance, define the id, MIPS and other basic attributes of the virtual machine, and the virtual machine can be instantiated according to the predefined sharing policy.
† VmScheduler	This class is used to simulate resource sharing for multiple VMS on the same host.
† VmAllocationPolicy	This class is used to define the allocation scheme of mapping host to Vm, that is, to select the available physical hosts in the Cloud Data Center that can meet the configuration requirements of virtual machine CPU, memory, etc.

4.3 Case study 1: DQ-TSA algorithm

The details of the CloudSim simulation process are as follows: first create the data center and CPU, memory, bandwidth and other data resources. The data center entity then sends a registration message to the Cloud Information Service (CIS) for registration, and when the user request arrives, the CIS will select the appropriate one from the list of Cloud Service providers according to the request and provide it to the user. And then, the data center agent queries the CIS to see if there is any available

data, and if so, creates a virtual machine on the data for task scheduling. As Figure 4.1. During the run, resources are repeatedly allocated and recycled at regular intervals. At each interval, the Data center calculates the energy consumption of each host. Finally, when all the cloudlets are complete, we will get the the states of Cloudlet and power consumption.

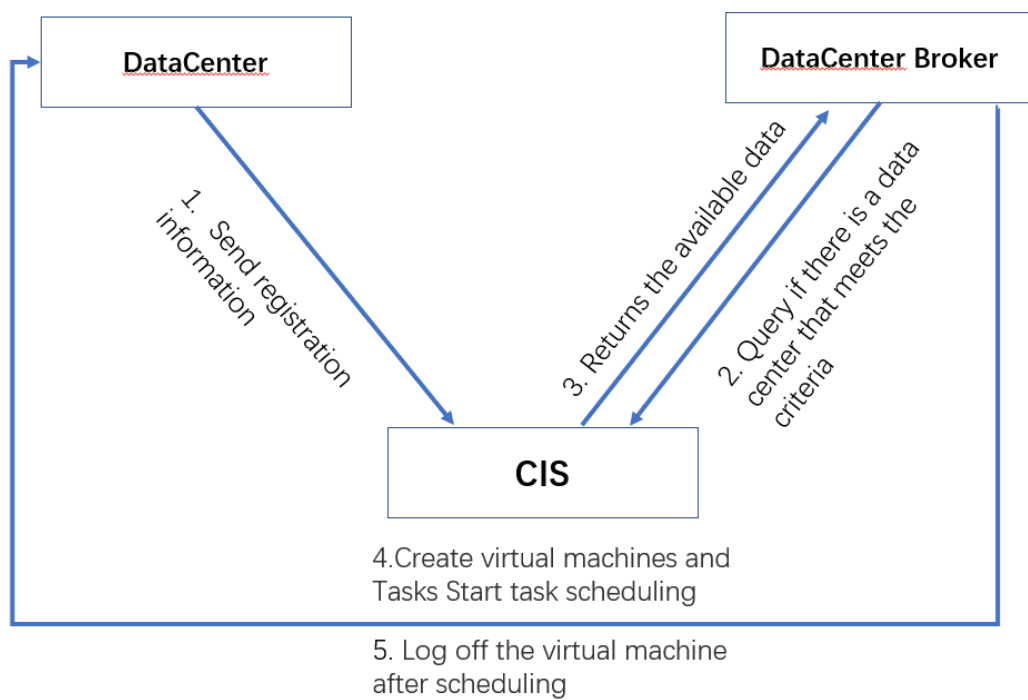


Figure 4.1: Data simulation communication process

Due to the adoption of virtualization technology, the resource scheduling problem of cloud computing data center is different from the traditional distributed computing resource scheduling mode. It is divided into two levels of scheduling. Scheduling tasks to a virtual machine according to a certain strategy is a first-level scheduling, while the deployment of virtual machines on physical servers is a second-level scheduling. The scheduling of virtual machine to physical host is to choose the appropriate host for virtual machine and to adopt certain strategies to solve the problem of insufficient resources when the requirements of users cannot be met. The simulation experiment in

this chapter is oriented to the research of first-level scheduling, that is, task scheduling. Task scheduling is the task scheduling to the virtual machine to run according to certain strategy, at the same time must satisfy some conditions, such as energy, time, SLA violation. The overall goal of cloud computing task scheduling is to schedule the tasks submitted by users to the resources that best meet the needs of users and to improve the overall throughput rate of cloud computing system as much as possible. When dispatching user tasks to virtual machines for execution, the scheduling strategy needs to consider optimal task scheduling span, quality of service (QoS), and for service providers, system resource utilization, system load balance, economic cost and other factors.

4.3.1 Energy consumption model

The overall energy consumption of a cloud computing system can be expressed as the formula 2.2.

In this simulation, the detail of calculate energy consumption in terms of the following formula:

$$E_{Cloud} = E_{Node} + E_{Switch} + E_{Storage} + E_{Other} \quad (4.1)$$

E_{Node} represents the node's energy consumption, E_{Switch} represents the energy consumption of all the switching equipment. $E_{Storage}$ represents the energy consumption of the storage device. E_{Others} represents the energy consumption of other parts, including the fans, the current conversion loss and others. The above formula can be further decomposed, a cloud computing environment with n nodes, m switching equipment and a centralized storage device, its energy consumption can be expressed

as:

$$\begin{aligned}
 E_{Cloud} = & n(E_{CPU} + E_{Ram} + E_{Disk} + E_{Main-board} + E_{NIC}) \\
 & + m(E_{Chassis} + E_{LineCards} + E_{Ports}) + (E_{NASServer}) \\
 & + E_{StorageController} + E_{DiskArray}) + E_{Others}
 \end{aligned} \tag{4.2}$$

Generally, CPU is the most energy consuming part of cloud system. Moreover, for computer-intensive applications, CPU utilization is proportional to cloud system load. For CPU, the power of the idle server is still 70 percent of that of the full-load server, indicating that setting the idle server to sleep mode can reduce the overall energy consumption of the system. In this experiment, the power model of linear rating relationship based on DVFS (dynamic voltage frequency adjustment) technology will be used to calculate the function of CPU. The algorithm idea about DVFS is: in consideration of the insufficient CPU utilization when performing tasks, reduce the CPU power supply voltage and clock frequency to achieve the purpose of reducing CPU performance. This method can not only greatly reduce CPU power consumption, but also ensure service performance.

The power model in this thesis is as follows:

$$P(u) = k * P_{max} + (1 - k) * P_{max} * u$$

P_{max} is the maximum power consumption when the system is fully loaded, k is the proportion of power consumption when the system is idle, and u is the utilization rate of CPU. Generally, load execution in a cloud system is dynamic, therefore, u is usually represented as $u(t)$, so, total energy consumption is E . It can be defined as:

$$E_{CPU} = \int_{t_0}^{t_1} P(u(t))dt$$

Memory power model. We design the following simple memory power model (implemented in the class called `PowerModelRamSimple`), where P denotes power (the unit is Watt), u denotes memory utilization, P_{Max} denotes the power when memory utilization is 100% and r denotes the total host memory. The unit of r is MB and means 1024 MB memory brings about one W energy consumption.

$$P = u * P_{max} / (r / 1024)$$

Suppose there are m physical machines available to support application execution in the cloud environment, denoted as $H = \{H_1, H_2, \dots, H_m\}$. Physical hosts can be located in a single data center or distributed across data centers.

A physical host in a cloud data center is seen as consisting of a large number of resource blocks, that is, multiple instances of virtual machines with the same configuration. Resource requirements for data-intensive computing tasks can be measured by the number of virtual machine instances.

Cloud service providers typically provide multiple types of virtual machine instances for cloud users to choose from. For example: computationally intensive virtual machine instances, memory optimized virtual machine instances, high I/O virtual machine instances, and memory intensive virtual machine instances. Each type of virtual machine represents the primary configured resources of the virtual machine. For example, computationally intensive virtual machine instances typically allocate more CPU resources. When a user selects a virtual machine instance for a data-intensive computing task, it is usually preferred to select the computationally intensive virtual machine instance to run the task. The execution time of a task is determined by the length of the task and the computing power of the virtual machine it occupies. In the case of a certain task length, the stronger the computing power of the virtual machine,

the shorter the corresponding execution time. The computational performance of a virtual machine is usually based on its specific resource configuration and is determined by the performance of the physical machine it maps to. In general, the physical machines in cloud data centers have different computing capabilities.

4.3.2 The simulation configuration

In the simulation environment, 25 physical host of each specification were selected, a total number of Host is 100. According to the literature (Beloglazov & Buyya, 2012) (Baker, 2019), the energy consumption of various host is shown in the following table 4.2. Basic energy consumption accounts for 60% of peak energy consumption.

Table 4.2: Physical Host hardware configuration

Parameter	Type 1	Type 2	Type 3	Type 4
Host	HP ProLiant SL390S G7 Intel Xeon 5640	HP ProLiant BL460c G6 Intel Xeon 5630	HP ProLiant ML110 G5 Intel Xeon 3075	HP ProLiant ML110 G4 Intel Xeon 3040
CPU Frequency (MHz)	3060	2530	2600	1860
RAM(GB)	16	8	4	4
I/O	500 MB/s	400 MB/s	300 MB/s	300 MB/s
Basic Energy Consumption(W)	342	192	93.7	86
Peak Energy Consumption(W)	570	320	156.2	143.3
Virtual machine number	4	6	3	2
Frequency	1843	3067	2048	2500

4.3.3 The simulation results

To verify the effectiveness of the algorithm, the comparison algorithm will include simple assignment algorithm (SIMPLE), random assignment algorithm (RANDOM), resource balancing algorithm (BALANCE) and DQ-TSA algorithm.

SIMPLE algorithm is to assign tasks to the virtual machine in the order in which they come. After each virtual machine is assigned a task, it can be re-assigned (Calheiros, Ranjan, Beloglazov, De Rose & Buyya, 2011).

RANDOM algorithm assigns tasks to the virtual machine with equal probability in the order they arrive (Pars & Maleki, 2009).

The BALANCE algorithm places the demand of CPU resources under dynamic workload, and the CPU demand changes as the task execution progresses (a cloud task based on progress). The change model (strategy) is determined by the utilization data in the workload file (Lin, Xu, He & Li, 2017).

Time, energy consumption and host SLA violation rate were evaluated. The host SLA violation rate is a measure built into Cloudsim that indicates that the CPU resource requirements of the running task exceed the allocated CPU resources within a certain period of time. The SLA violation rate is calculated by dividing the host SLA violation time by the host execution time.

System energy consumption

As shown in Figure 4.2, the system energy consumption of the four scheduling methods will increase with the increase of the task volume. However, the DQ-TSA algorithm proposed in this paper has a low energy consumption.

Among them, RANDOM algorithm has the highest energy consumption, and sometimes the energy consumption generated by this algorithm is more than twice that of DQ-TSA algorithm.

SIMPLE algorithm and BALANCE algorithm have little difference in system energy consumption. Even so, the energy consumption of the DQ-TSA algorithm was only 55% of that of the SIMPLE algorithm and 53% of that of the BALANCE algorithm.

Simulation time

As shown in Figure 4.3, SIMPLE algorithm consumes the most time, while DQ-TSA algorithm consumes the least. The graphs of the RANDOM algorithm and

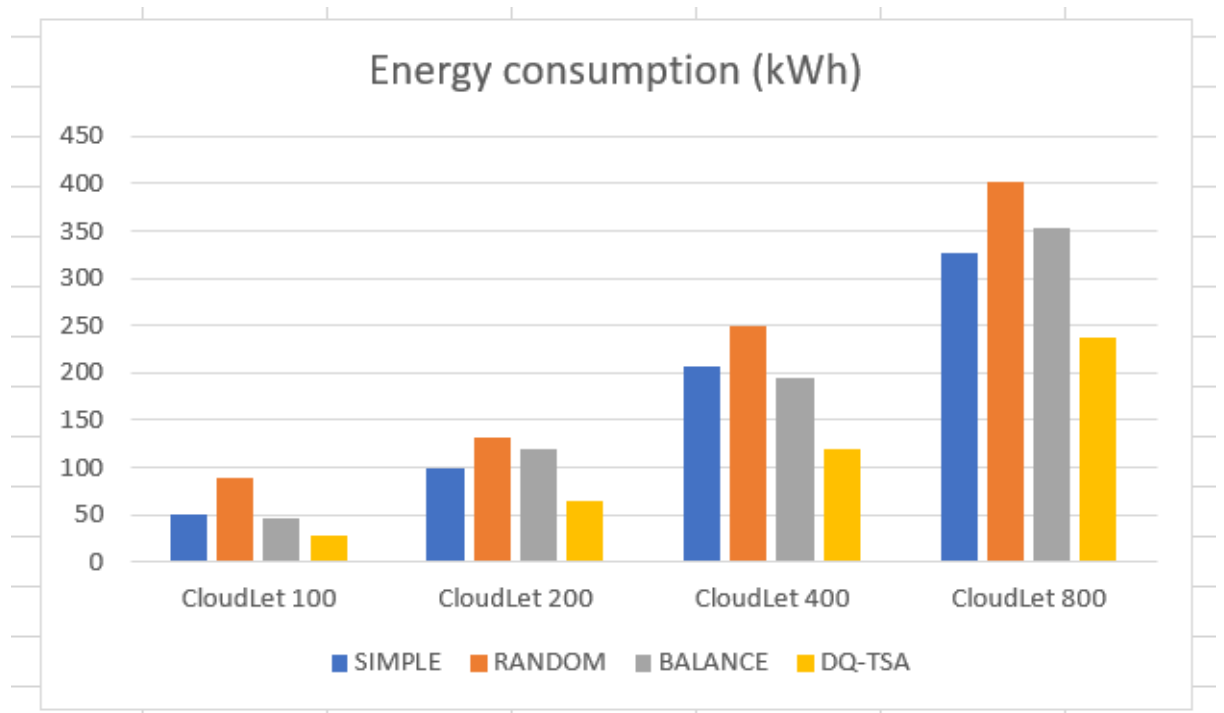


Figure 4.2: Compare the energy consumption of different algorithms

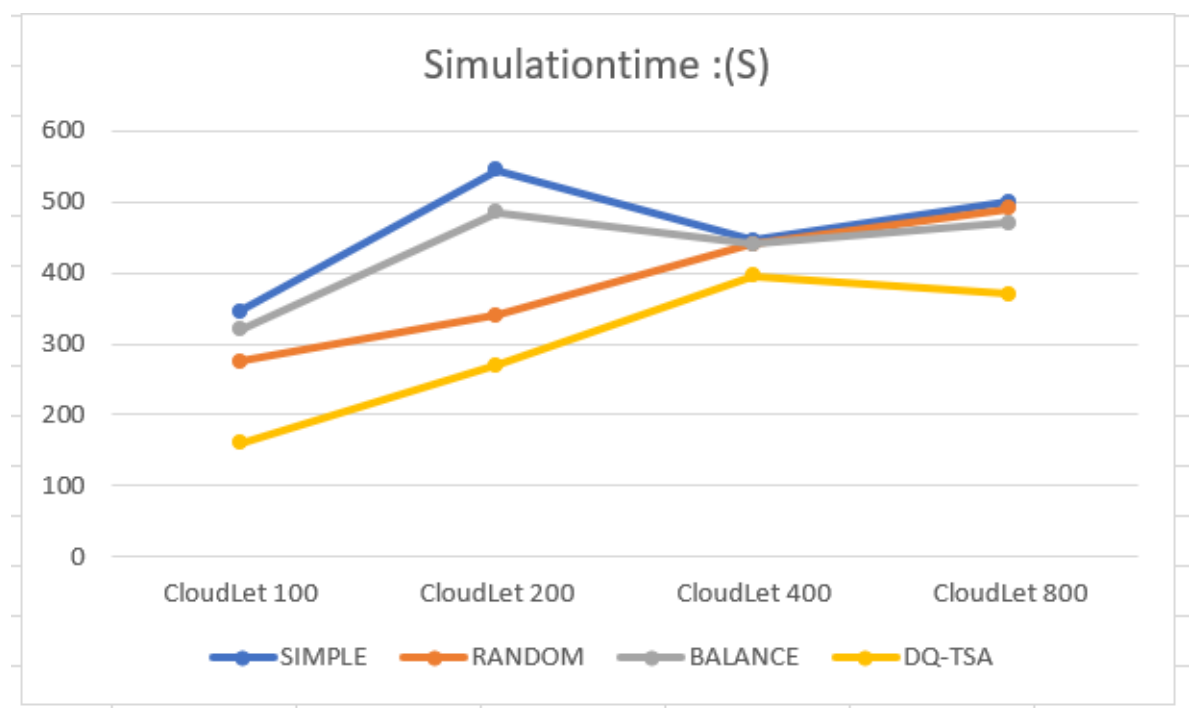


Figure 4.3: Compare the simulation time of different algorithms

BALANCE algorithm are basically familiar with each other. Both of them reach the peak value when the number of tasks is 200, and then drop to the lowest value when the number of tasks is 400, and then slowly rise when the number of tasks reaches 800. The time use curve of RANDOM algorithm is relatively smooth, with no big fluctuation. It all rises with the increase of the number of tasks. The increase of time is directly proportional to the increase of the assignment book. The time-consuming curve of DQ-TSA algorithm is different from the others. It goes up from cloudlet 100 to 400, and then peaks at 400, and then it starts to go down, and then it goes down from cloudlet 400 to 800. This indicates that the DQ-TSA algorithm will perform better with more tasks.

SLA violation

As can be seen from Chart 4.4 and 4.5 , both the DQ-TSA algorithm and BALANCE algorithm have a very good performance in teams of SLA violation. The BALANCE algorithm is superior to the DQ-TSA algorithm when the number of tasks is small(100-200). However, when the number of tasks increases to 800, the DQ-TSA algorithm is still more ideal.

Summary

DQ-TSA algorithm had the best performance in terms of energy consumption, time and SLA violation. It differs from other algorithms in that it already ranks the data that needs to be processed by the quality of the data before the cloud task is submitted, following the principle that high-level data is allocated to high-performance processing machines. Moreover, the energy consumption of DQ-TSA algorithm is about half that of the traditional SIMPLE algorithm when the number of tasks is larger.

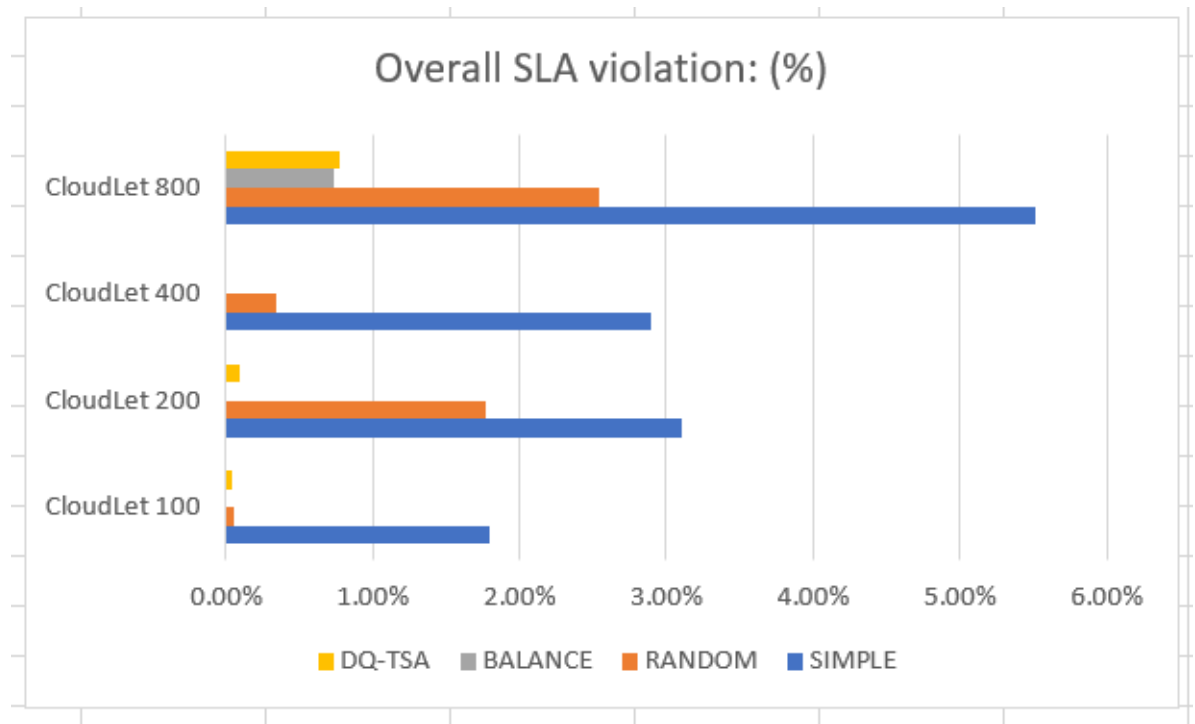


Figure 4.4: Compare the overall SLA violation of different algorithms

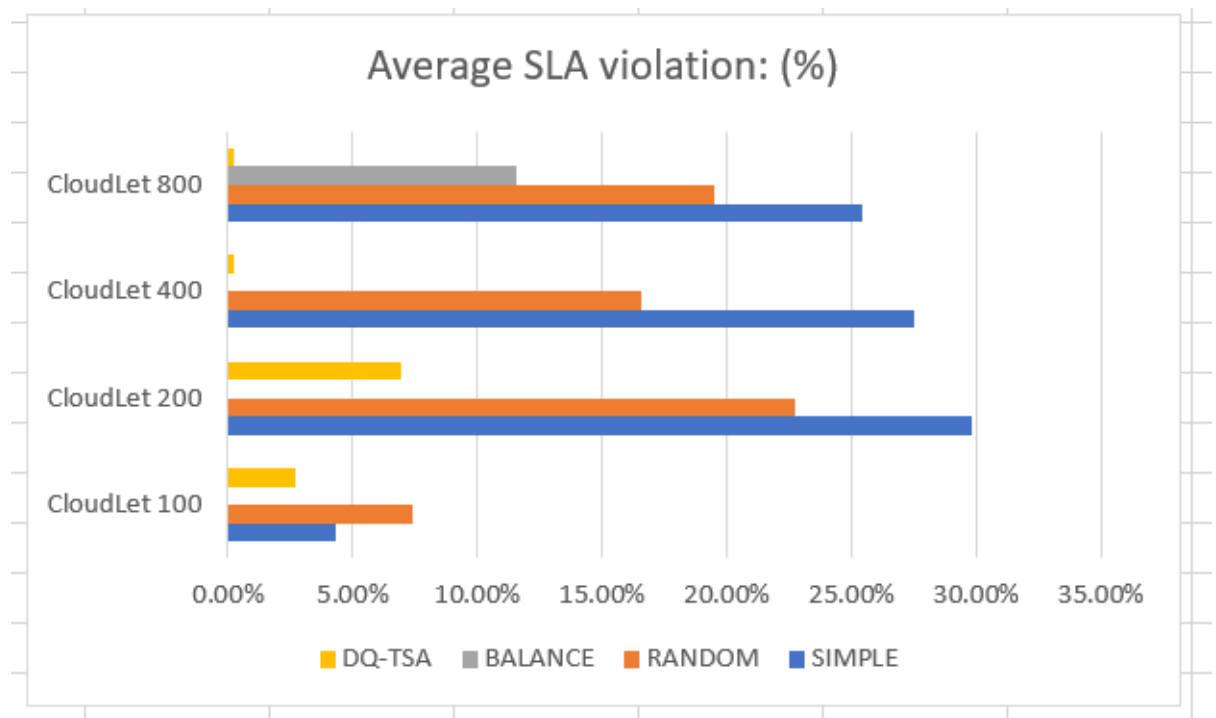


Figure 4.5: Compare the average SLA violation of different algorithms

4.4 Case study 2: DQ-HSA algorithm

This thesis used the cloud storage workload generator Mimesis (Abad, Roberts, Lu & Campbell, 2012), which can help generate more complex cloud workloads, especially file-granular cloud storage workload testing. Meanwhile, Cloud storage system (HDFS) without integrated data classification algorithm, hot-spot sensing storage placement algorithm (Ha Daap) (Xiong, Du, Jin & Luo, 2018), and DQ-HAS algorithm based on data quality proposed in this paper are implemented in Cloud Sim for final performance comparison.

4.4.1 The simulation configuration

Based on the extended Cloud Sim platform, this paper simulates the hierarchical Cloud storage model proposed above, and realizes the final data optimization storage placement process by improving the open MOEA/D toolkit. Storage device performance is an important factor affecting cloud storage system. Table 4.3 shows various performance parameters of storage device in the process of simulation test. In the experiment in this paper, a series of storage devices with different performance were simulated through CloudSim, including a small number of high-performance memory SSD for storing hot file data and a large number of low-performance memory HDD for storing hot file data. At the same time, all the devices have some differences in various storage performance indicators. The specific storage device configuration is shown in table 4.3.

4.4.2 Workload

Due to the need to test the hierarchical cloud storage scheme proposed in this article, this paper uses the workload in the cloud storage platform as the data set used for the test. Therefore, in addition to the storage device performance configuration

Table 4.3: Storage device configuration

Type	Capacity	IOPS	Latency	Transfer Rated	Power	Count
SSD	5	5000	15	300	12.6	1
	8	4000	18	290	11.2	2
	10	3000	20	280	9.6	3
HDD	15	250	30	200	4.3	4
	28	200	40	180	3.8	6
	20	150	50	160	3	8

described above, workload is another important factor affecting the performance of cloud storage systems. Xie and Sun (2009) found that in the actual cloud environment, the workload had complex data access characteristics, and the experimental scheme of existing methods could not reach the complexity of data access in the actual workload. To better describe the actual workload in a cloud storage cluster, the load tests in this article used the load simulation generator Mimesis (Abad, Roberts et al., 2012) to generate a more realistic workload trace.

By improving the load simulation generator Mimesis, the simulation test was completed in the Cloud sim-based simulation hierarchical Cloud storage system, using the file access load with diverse statistical characteristics. As a hierarchical storage solution associated with this article content, in the traditional hierarchical storage method, generic data storage Load test tool has the Load Runner, Post mark, Io zone etc, this part of the tool is limited to direct to Load test data storage, and file storage system including data write, random, speaking, reading and writing.

Beside that, both Ceph and Hadoop have a cloud workload trace collection mechanism, namely Bench Mark collection Device. Mimesis focuses on the workload generation implementation of the data storage part based on Bench Mark's implementation, which can generate load data sets with the same statistical characteristics as

Yahoo Home02 and EECS data sets. The advantages of applying Mimesis to test load generation over other workload data sets are:

1) Mimesis focuses on data storage load generation in the cloud environment, which has the same statistical characteristics as Yahoo trace. Mimesis is also capable of generating complex data metadata information structures and data file directory tree structures for the purposes of this article.

2) Mimesis is implemented in Java, which provides an open source and independent coding structure. All the load generation modules provide flexible statistical feature parameter setting, which is easy to simulate the cloud storage working environment under different conditions. The generated workload is suitable for simulation testing.

According to Abad, Roberts et al. (2012) and Xie and Sun (2009) and others cloud workload statistics characteristics of analysis, this paper improved the Mimesis of part of the code, the final test pass and parameter configuration, characteristics of different data access to reflect the complex cloud workload on the system, the influence of this scheme is presented in table 4.4 main test load characteristic parameters, the parameters of the concrete content as shown below.

1) Number of documents. The load test in this paper starts from 1000 files and has a high test load volume.

2) File size and quantity distribution. Used to describe how many files' data sizes fall within a specific interval in a workload. In particular, the definition range of small files and the skew of the number of small files are set based on the fact that most small files are in the actual file access load

3) The distribution of files and visits is the main load feature of file access. Based on the feature performance of the actual load (zipf-like feature), the experiment in this paper gives the skewness setting of multiple groups of file access.

4) File read-write ratio. The default read-write ratio of the file in the load test in this paper is set to 0.7:0.3. On this basis, through the random optimization of a certain

Table 4.4: Workload parameters

Parameter	Value
Distribution of the number of file	[1000, 3000, 5000, 8000, 10000, 15000] Default:8000
Distribution of number of file size	Size interval:[0K-40G] FSC:Most file size concentrated [0K-100M] Skewness of FSC: [0.1-0.3]
Distribution of number of file access(zipf like)	X:Y=0.2:0.8, 0.25:0.75, 0.3:0.7, 0.35:0.65 Default:0.3:0.7
Read and Write rated	R:W=0.7:0.3
Advance storage rate of file	≤ 0.1
Delete rate of file	≤ 0.05

ratio, the characteristics of the file being written once and read many times in the actual load process are simulated.

5) File pre-storage rate, and file deletion rate in the load process. By setting a certain amount of pre-stored files, it avoids the disadvantage of starting from 0 in the traditional hierarchical storage scheme in the test, and then by setting a low file deletion rate, it can describe the diversified storage life cycle of files in the actual load .

In addition, when testing, this paper simulates a single statistical feature and combines the correlation effects of multiple statistical features to cover the complexity of the actual workload as much as possible. The specific test application can be seen in the following section.

4.4.3 The simulation results

In this chapter, the file volume and file access distribution in the workload characteristic variables are used as independent variables during performance testing. During the testing process, the system energy consumption, average migration cost, and load changes are discussed as goals. HDFS default storage is analyzed. The storage performance of the process, Ha Daap algorithm, and the DQ-HST algorithm proposed in this thesis.

System energy consumption

As shown in Figure 4.6, in different file load tests, the system energy consumption of the three storage methods increases with the increase of file size. In general, the DQ-HST algorithm proposed in this paper has lower system energy consumption than HDFS and Ha Daap. Among them, when the amount of files is low, the energy consumption level of the three storage methods differs little. When the amount of files reaches a certain level, HDFS system has the highest power consumption. Compared with HDFS, Ha Daap and DQ-HST have obvious energy consumption reduction effect, and compared with Ha Daap, DQ-HST algorithm has lower system energy consumption.

In particular, as shown in Figure 4.6, along with the increase in the file, the Ha Daap and DQ-HST node in the fifth contrast difference of system energy consumption rate is reduced, this is because when a file is high (roughly around 10000), in order to close to the actual work load more, when generate simulated load, this article to a certain extent, to broaden the definition of small files interval, correspondingly reduced the number of small files centralized skewness. However, at the 6th comparison node, compared with HDFS and Ha Daap, the energy consumption of DQ-HST is the highest, among which, the energy consumption of DQ-HST is 32.84% lower than THAT of HDFS and 17.32% lower than that of Ha Daap. This shows that DQ-HST has a better

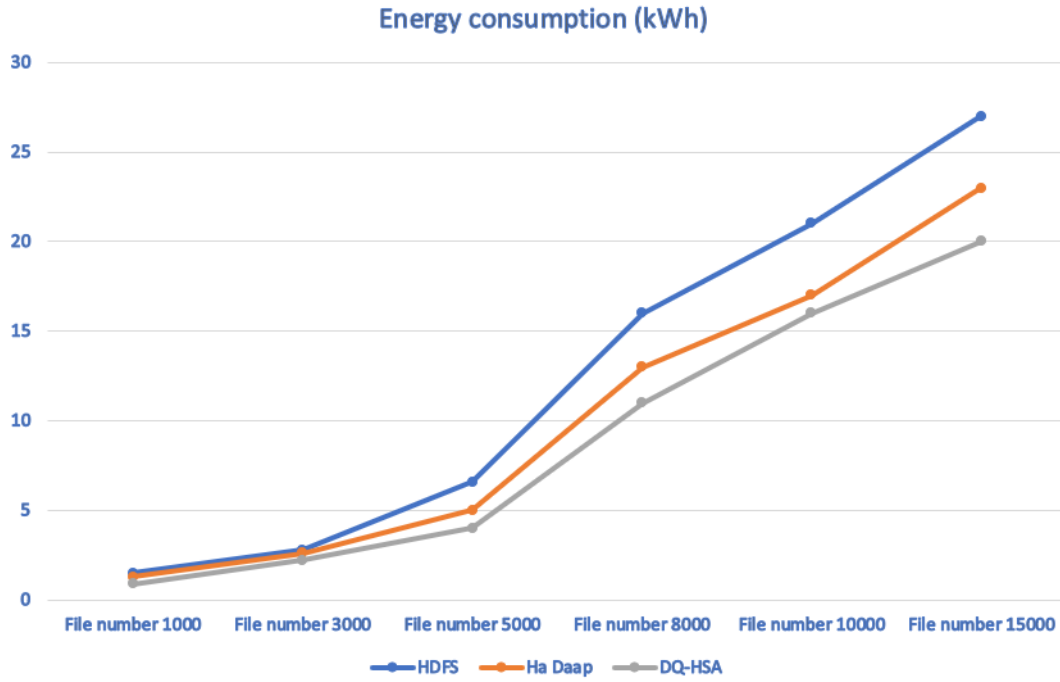


Figure 4.6: The system energy consumption of the algorithm under different file sizes

energy-saving effect in the case of large file size.

As shown in Figure 4.7, with the increase of frequency between files and visit, the system energy consumption of HDFS basically remains the same, while that of Ha Daap and DQ-HST obviously decreases. While both Ha Daap and DQ-HST regard reducing system energy consumption as the optimization goal, as DQ-HST realizes a more fine-grained data heat pre-classification process, DQ-HST has a better energy saving effect when the skew degree of files and visits increases. As can be seen from Figure 4.7, when the file access concentration is high, the system energy consumption of DQ-HST is 79.23% lower than that of HDFS and 28.36% lower than that of Ha Daap.

System time consumption

As shown in Figure 4.8, with the increase of file size, the system service time of the three storage schemes increases accordingly. Compared with HDFS, Ha Daap and

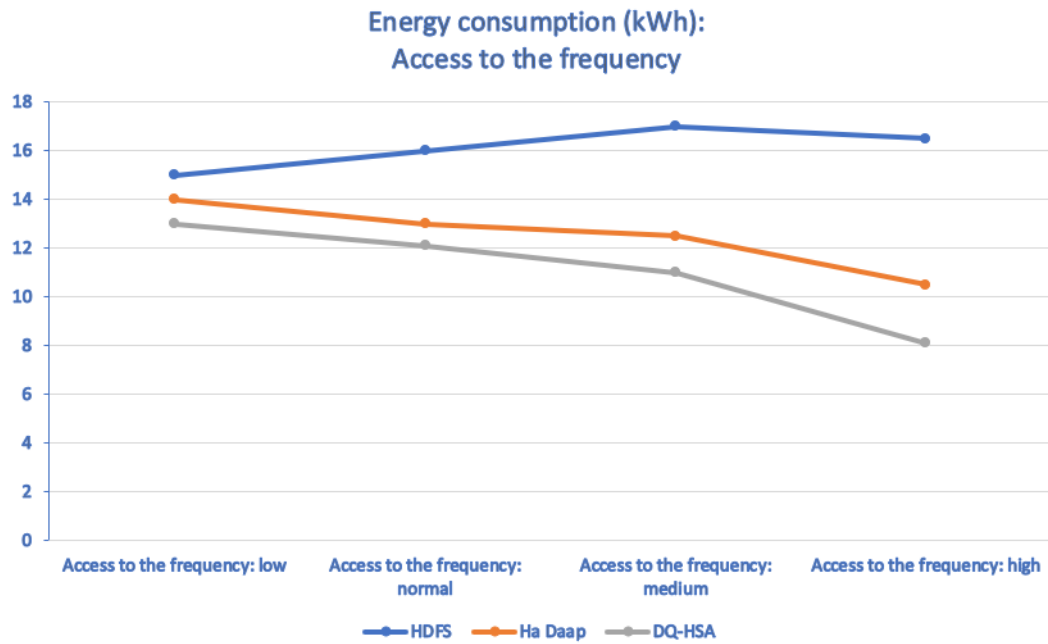


Figure 4.7: The system energy consumption of the algorithm under different file access frequencies

DQ-HST have lower system service time. Meanwhile, the system service time of Ha Daap and DQ-HST is not different. The Figure 4.8 shows that with the increase of file size, the average service time of DQ-HST decreases by 41.5% compared with HDFS and 7.1% compared with Ha Daap.

As shown in Figure4.9, the system service time of HDFS basically remains unchanged, and the system service time of Ha Daap and DQ-HST decreases accordingly. Meanwhile, the DQ-HST system has the lowest service time. This is because DQ-HST is based on the rule of file access and evaluates the heat through multiple file access attributes. The Figure 4.9 shows that DQ-HST is very sensitive to the system load in the access set. Compared with HDFS, system service time decreases by 34.4% on average, and system service time decreases by 19.63% on average compared with Ha Daap.

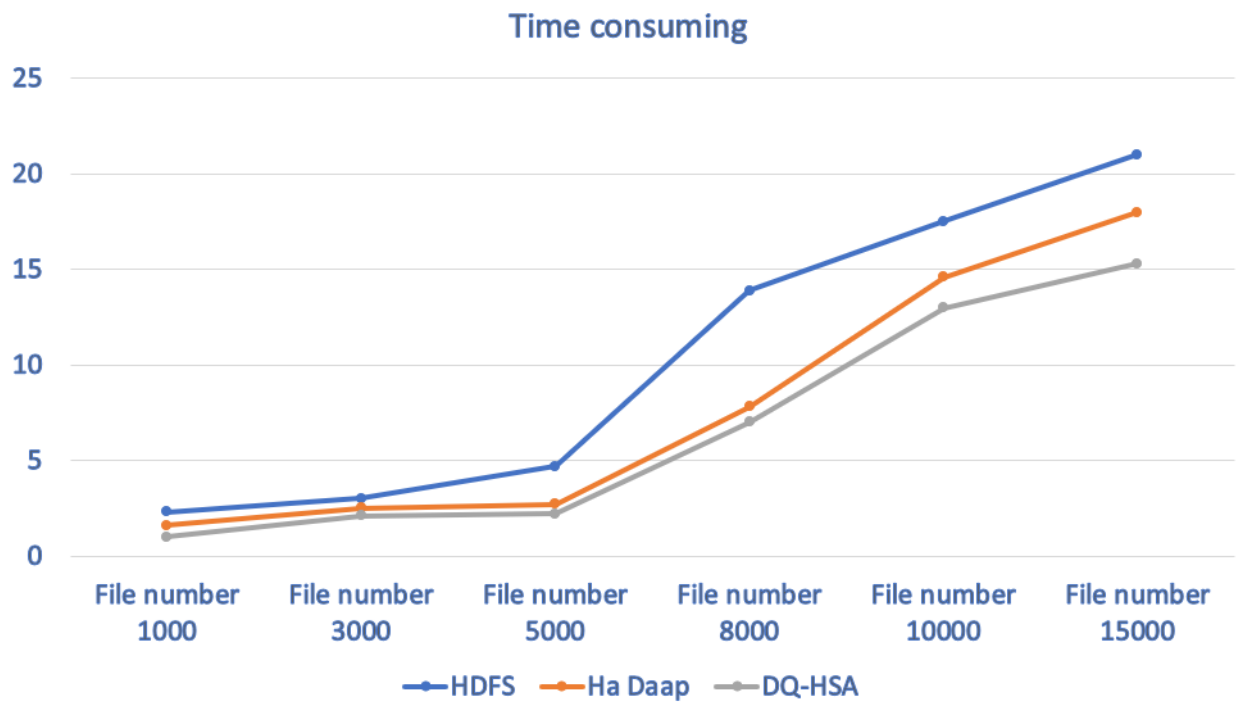


Figure 4.8: Time comparison between different file volumes

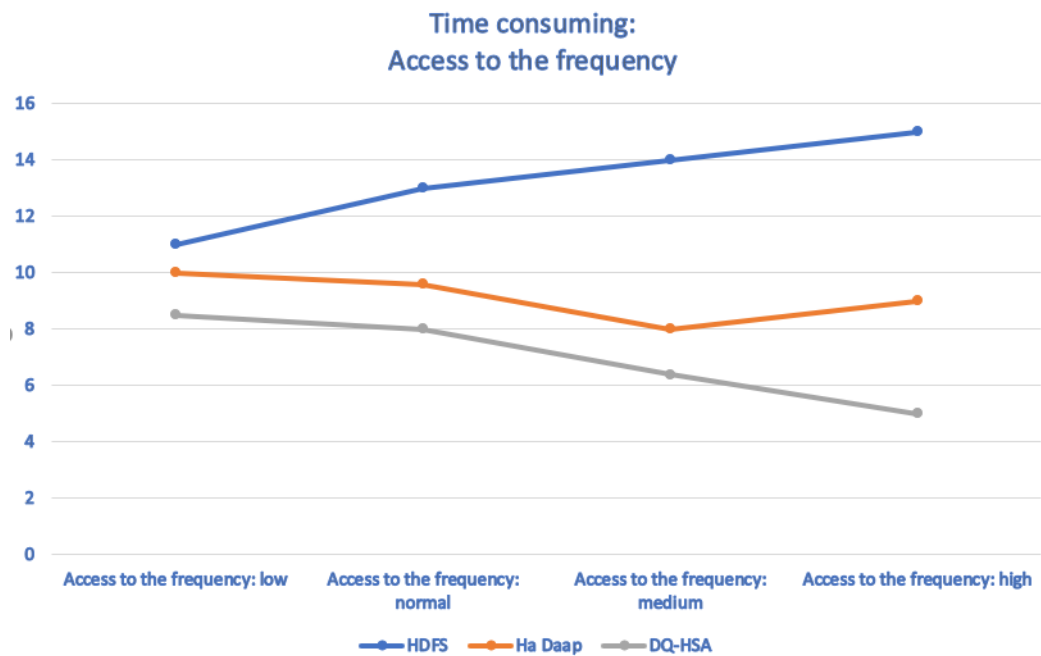


Figure 4.9: Time comparison between different file visits

Workload variation

As shown in Figure 4.10, the load changes of the three storage schemes are not significantly different in the early stage (compare node 1 to node 3). In the middle and late stage (comparison node 4 6), the load change rate of the three storage schemes decreased. The Figure 4.10 shows that when the system file volume is high, the load change rate of DQ-HST is the lowest, which is 45.57% lower than HDFS and 19.21% lower than Ha Daap.



Figure 4.10: Workload comparison between different file volumes

As shown in Figure 4.11, the load change rate of HDFS increases with the increase of skew between files and visits. The load change rate of Ha Daap increases once in the middle period, and then decreases gradually. This is because when the skew of files and visits increases, the hotspot awareness algorithm adopted by Ha Daap also increases its responsiveness to the centralized access load, data mobility and load change rate. Finally, as the skew of files and visits continues to increase, the load change rate decreases due to Ha Daap's dual placement scheduling of hot and cold

data. Compared to the HDFS and Ha Daap, due to the heat DQ - HST has realized the data evaluation classification process, the data storage device when the choice is not directly selecting a particular goal, but adopting the processing mode of regional selection through multi-objective optimization algorithm to choose the final goal of storage, therefore, DQ-HST load rate as the file with the traffic steadily decreased with the rise of skewness. As can be seen from Figure 4.11, when the skew between files and page views increases, the load change rate of DQ-HST is about 45% lower than HDFS and 21.29% lower than Ha Daap.

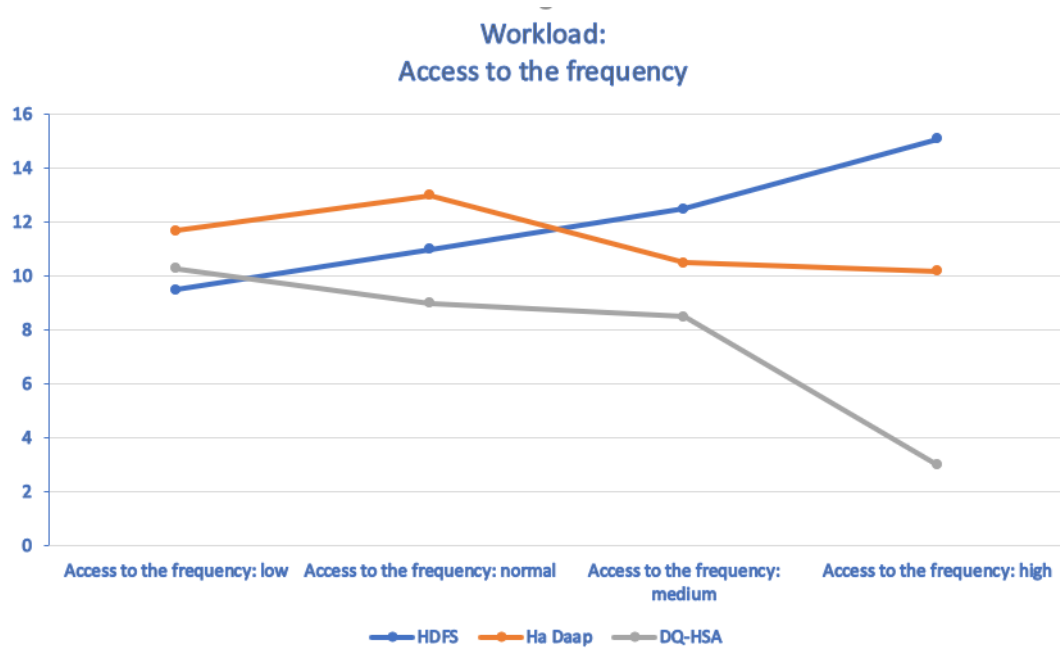


Figure 4.11: Workload between comparison between different file visits

Summary

This chapter introduces the test environment of the final experiment. Then the generating tool of the test workload is introduced, several important load statistical features are determined, and the parameter configuration of each load statistical feature is given. Finally, gives the HDFS, Ha Daap and this DQ-HST algorithm under different

experimental configuration of various performance test results, from the point of the final results, this paper puts forward the DQ-HST algorithm saves energy consumption to a certain extent, reduce the service time, improve the system load balance, and increase the load characteristics of complexity still has a stable performance improvements in performance.

Chapter 5

Conclusion and Future Work

This chapter mainly summarizes the work done in this paper and looks into the future. Summarizes the The deficiencies of the paper, and give the future solution.

5.1 Conclusion

With the advent of the era of big data, great changes have taken place in people's lives. More and more science and technology are developing around data science. Data not only brings convenience to human life, but also has an impact on environmental pollution. For example, the rise of more and more data centers. At present, many researches are improving various algorithms to improve the energy efficiency of data centers, and few literature have proposed the relationship between data classification and energy consumption.

This thesis puts forward the concept of data hierarchical measurement, analyzes the data characteristics under the big data environment through the sorting of big data technology. Puts forward the general index of data quality under the big data environment, defines the formularized measurement method of the index system, and provides effective guidance for data quality management and evaluation. In addition, in the

life cycle of big data, two parts of calculation and storage are selected to simulate the concept of data classification. Cloud task scheduling algorithm DQ-TSA and hierarchical storage algorithm DQ-HSA are respectively proposed based on data classification. The main idea of both algorithms is to level the data before it is processed (stored) so that the higher the level of data, the better the physical resources (host performance, storage performance, bandwidth, etc.). Simulation experiments in CloudSim show that both DQ-TSA and DQ-HST have excellent performance in energy saving.

5.2 Future Work

In this practice of our thesis, more factors should be taken into account, such as the energy consumption of equipment other than IT equipment in the data center. And the simulation experiments in this article is only selected the data life cycle of the computing and storage the feasibility of the two parts to verify the theory, in the future in the study of simulation experiment can be expanded range, starting from the data acquisition phase, for example, will collect data terminal equipment also use data classification concept to define, different terminal equipment acquisition data quality is also different.

References

- Abad, C. L., Luu, H., Roberts, N., Lee, K., Lu, Y. & Campbell, R. H. (2012). Metadata traces and workload models for evaluating big storage systems. In *2012 IEEE Fifth International Conference on Utility and Cloud Computing* (pp. 125–132).
- Abad, C. L., Roberts, N., Lu, Y. & Campbell, R. H. (2012). A storage-centric analysis of mapreduce workloads: File popularity, temporal locality and arrival patterns. In *2012 IEEE International Symposium on Workload Characterization (IISWC)* (pp. 100–109).
- Akula, G. S. & Potluri, A. (2014). Heuristics for migration with consolidation of ensembles of virtual machines. In *2014 Sixth International Conference on Communication Systems and Networks (ComSNETS)* (pp. 1–4).
- Alan, I., Arslan, E. & Kosar, T. (2014). Energy-aware data transfer tuning. In (pp. 626–634). IEEE. doi: 10.1109/CCGrid.2014.117
- AL-Hazemi, F., Mohammed, A. F. Y., Laku, L. I. Y. & Alanazi, R. (2019). Pue or gpue: A carbon-aware metric for data centers. In (pp. 38–41). IEEE. doi: 10.23919/ICACT.2019.8701895
- Atat, R., Liu, L., Wu, J., Li, G., Ye, C. & Yang, Y. (2018). Big data meet cyber-physical systems: A panoramic survey. *IEEE Access*, 6, 73603–73636.
- Azevedo, A. I. R. L. & Santos, M. F. (2008). Kdd, semma and crisp-dm: a parallel overview. *IADS-DM*.
- Bahwairath, K., Benkhelifa, E., Jararweh, Y., Tawalbeh, M. A. et al. (2016). Experimental comparison of simulation tools for efficient cloud and mobile cloud computing applications. *EURASIP Journal on Information Security*, 2016(1), 15.
- Baker, R. J. (2019). *Cmos: circuit design, layout, and simulation*. John Wiley & Sons.
- Barabási, A.-L. & Gelman, A. (2010). Bursts: The hidden pattern behind everything we do. *Physics Today*, 63(5), 46.
- Batini, C. & Scannapieco, M. (2016). Methodologies for information quality assessment and improvement. In *Data and information quality* (pp. 353–402). Springer.
- Beloglazov, A., Abawajy, J. & Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future generation computer systems*, 28(5), 755–768.
- Beloglazov, A. & Buyya, R. (2012). Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurrency and Computation: Practice*

- and Experience*, 24(13), 1397–1420.
- Beloglazov, A., Buyya, R., Lee, Y. C. & Zomaya, A. (2011). A taxonomy and survey of energy-efficient data centers and cloud computing systems. In *Advances in computers* (Vol. 82, pp. 47–111). Elsevier.
- Bihl, T. J., Young II, W. A. & Weckman, G. R. (2016). Defining, understanding, and addressing big data. *International Journal of Business Analytics (IJBAN)*, 3(2), 1–32.
- Bircher, W. L. & John, L. K. (2007). Complete system power estimation: A trickle-down approach based on performance events. In *2007 IEEE International Symposium on Performance Analysis of Systems & Software* (pp. 158–168).
- Bohra, A. E. H. & Chaudhary, V. (2010). Vmeter: Power modelling for virtualized clouds. In *2010 IEEE International Symposium on Parallel & Distributed Processing, Workshops and PhD Forum (IPDPSW)* (pp. 1–8).
- Bovee, M., Srivastava, R. P. & Mak, B. (2003). A conceptual framework and belief-function approach to assessing overall information quality. *International Journal of Intelligent Systems*, 18(1), 51–74.
- Buyya, R., Ranjan, R. & Calheiros, R. N. (2009). Modeling and simulation of scalable cloud computing environments and the cloudsims toolkit: Challenges and opportunities. In *2009 International Conference on High Performance Computing & Simulation* (pp. 1–11).
- Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A. & Buyya, R. (2011). Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience*, 41(1), 23–50.
- Carvalho, R. d., Saldanha, R. R., Gomes, B., Lisboa, A. C. & Martins, A. (2012). A multi-objective evolutionary algorithm based on decomposition for optimal design of yagi-uda antennas. *IEEE Transactions on Magnetics*, 48(2), 803–806.
- Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., ... Gruber, R. E. (2008). Bigtable: A distributed storage system for structured data. *ACM Transactions on Computer Systems (TOCS)*, 26(2), 1–26.
- Chang, V. & Gütl, C. (2010). Generation y learning in the 21st century: integration of virtual worlds and cloud computing services. In *Global learn* (pp. 1888–1897).
- Chao, H., Chen, Y. & Wu, J. (2011). Power saving for machine to machine communications in cellular networks. In *2011 IEEE Globecom Workshops (GC Wkshps)* (pp. 389–393).
- Chen, F., Grundy, J., Yang, Y., Schneider, J.-G. & He, Q. (2013). Experimental analysis of task-based energy consumption in cloud computing systems. In *Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering* (pp. 295–306).
- Chen, L., He, Y., Yang, Y., Niu, S. & Ren, H. (2017). The research status and development trend of additive manufacturing technology. *The International Journal of Advanced Manufacturing Technology*, 89(9-12), 3651–3660.
- Davenport, T. H. & Patil, D. (2012). Data scientist. *Harvard business review*, 90(5), 70–76.

- Dean, J. & Ghemawat, S. (2008). Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1), 107–113.
- Deelman, E., Vahi, K., Juve, G., Rynge, M., Callaghan, S., Maechling, P. J., ... others (2015). Pegasus, a workflow management system for science automation. *Future Generation Computer Systems*, 46, 17–35.
- Dhiman, G., Mihic, K. & Rosing, T. (2010). A system for online power prediction in virtualized environments using gaussian mixture models. In *Proceedings of the 47th design automation conference* (pp. 807–812).
- Economou, D., Rivoire, S., Kozyrakis, C. & Ranganathan, P. (2006). Full-system power analysis and modeling for server environments..
- Elnozahy, E. M., Kistler, M. & Rajamony, R. (2002). Energy-efficient server clusters. In *International workshop on power-aware computer systems* (pp. 179–197).
- Fan, X., Weber, W.-D. & Barroso, L. A. (2007). Power provisioning for a warehouse-sized computer. *ACM SIGARCH computer architecture news*, 35(2), 13–23.
- Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, 17(3), 37–37.
- Fox, C., Levitin, A. & Redman, T. (1994). The notion of data and its quality dimensions. *Information processing & management*, 30(1), 9–19.
- Frej, M. B. H., Dichter, J. & Gupta, N. (2018). Light-weight accountable privacy preserving (lapp) protocol to determine dishonest role of third party auditor in cloud auditing. In *2018 ieee international conference on consumer electronics (icce)* (pp. 1–6).
- Gantz, J. & Reinsel, D. (2012). The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east (2012). URL: <http://www.emc.com/collateral/analyst-reports/idc-the-digital-universein-2020.pdf>.
- Gao, Y., Guan, H., Qi, Z., Wang, B. & Liu, L. (2013). Quality of service aware power management for virtualized data centers. *Journal of Systems Architecture*, 59(4-5), 245–259.
- Gawali, M. (June 2014). Esdl – virtual machine administration.
doi: <http://mahenswap.blogspot.com/2014/06/virtual-machine-administration.html>
- Ge, R., Feng, X. & Cameron, K. W. (2009). Modeling and evaluating energy-performance efficiency of parallel processing on multicore based power aware systems. In *2009 ieee international symposium on parallel & distributed processing* (pp. 1–8).
- Geng, X., Mao, Y., Xiong, M. & Liu, Y. (2019). An improved task scheduling algorithm for scientific workflow in cloud computing environment. *Cluster Computing*, 22(3), 7539–7548.
- Ghemawat, S., Gobioff, H. & Leung, S.-T. (2003). The google file system. In *Proceedings of the nineteenth acm symposium on operating systems principles* (pp. 29–43).
- Graham-Rowe, D., Goldston, D., Doctorow, C., Waldrop, M., Lynch, C., Frankel, F., ... others (2008). Big data: science in the petabyte era. *Nature*, 455(7209), 8–9.
- Gupta, H., Vahid Dastjerdi, A., Ghosh, S. K. & Buyya, R. (2017). ifogsim: A toolkit for

- modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments. *Software: Practice and Experience*, 47(9), 1275–1296.
- Gupta, V., Nathuji, R. & Schwan, K. (2011). An analysis of power reduction in datacenters using heterogeneous chip multiprocessors. *ACM SIGMETRICS Performance Evaluation Review*, 39(3), 87–91.
- Hale, J. S. (2013). Amazon cloud drive forensic analysis. *Digital Investigation*, 10(3), 259–265.
- Hazas, M., Morley, J., Bates, O. & Friday, A. (2016). Are there limits to growth in data traffic?: On time use, data generation and speed. In *Proceedings of the second workshop on computing within limits* (p. 14).
- Herzog, C. (2013). Standardization bodies, initiatives and their relation to green it focused on the data centre side. In *European conference on energy efficiency in large scale distributed systems* (pp. 289–299).
- Hey, T., Tansley, S., Tolle, K. et al. (2009). *The fourth paradigm: data-intensive scientific discovery* (Vol. 1). Microsoft research Redmond, WA.
- Hidalgo-León, R., Siguenza, D., Sanchez, C., León, J., Jácome-Ruiz, P., Wu, J. & Ortiz, D. (2017). A survey of battery energy storage system (bess), applications and environmental impacts in power systems. In *2017 ieee second ecuador technical chapters meeting (etcm)* (pp. 1–6).
- Hidalgo-León, R., Urquizo, J., Macías, J., Siguenza, D., Singh, P., Wu, J. & Soriano, G. (2018). Energy harvesting technologies: Analysis of their potential for supplying power to sensors in buildings. In *2018 ieee third ecuador technical chapters meeting (etcm)* (pp. 1–6).
- Horvath, T. & Skadron, K. (2008). Multi-mode energy management for multi-tier server clusters. In *2008 international conference on parallel architectures and compilation techniques (pact)* (pp. 270–279).
- Jain, R., Molnar, D. & Ramzan, Z. (2005). Towards understanding algorithmic factors affecting energy consumption: switching complexity, randomness, and preliminary experiments. In *Proceedings of the 2005 joint workshop on foundations of mobile computing* (pp. 70–79).
- Jarke, M., Lenzerini, M., Vassiliou, Y. & Vassiliadis, P. (2013). *Fundamentals of data warehouses*. Springer Science & Business Media.
- Jaureguialzo, E. (2011). Pue: The green grid metric for evaluating the energy efficiency in dc (data center). measurement method using the power demand. In (pp. 1–8). IEEE. doi: 10.1109/INTLEC.2011.6099718
- Jiang, Z., Lu, C., Cai, Y., Jiang, Z. & Ma, C. (2013). Vpower: Metering power consumption of vm. In *2013 ieee 4th international conference on software engineering and service science* (pp. 483–486).
- Kakoulli, E. & Herodotou, H. (2017). Octopusfs: A distributed file system with tiered storage management. In *Proceedings of the 2017 acm international conference on management of data* (pp. 65–78).
- Kambatla, K., Kollias, G., Kumar, V. & Grama, A. (2014). Trends in big data analytics. *Journal of Parallel and Distributed Computing*, 74(7), 2561–2573.

- Kansal, A., Zhao, F., Liu, J., Kothari, N. & Bhattacharya, A. A. (2010). Virtual machine power metering and provisioning. In *Proceedings of the 1st acm symposium on cloud computing* (pp. 39–50).
- Kerr, K., Norris, T. & Stockdale, R. (2007). Data quality information and decision making: a healthcare case study. *ACIS 2007 Proceedings*, 98.
- Kliazovich, D., Bouvry, P. & Khan, S. U. (2012). Greencloud: a packet-level simulator of energy-aware cloud computing data centers. *The Journal of Supercomputing*, 62(3), 1263–1283.
- Lent, R. (2013). A model for network server performance and power consumption. *Sustainable Computing: Informatics and Systems*, 3(2), 80–93.
- Lewis, A. W., Ghosh, S. & Tzeng, N.-F. (2008). Run-time energy consumption estimation based on workload in server systems. *HotPower*, 8, 17–21.
- Li, H., Casale, G. & Ellahi, T. (2010). Sla-driven planning and optimization of enterprise applications. In *Proceedings of the first joint wosp/sipew international conference on performance engineering* (pp. 117–128).
- Li, Y., Wang, Y., Yin, B. & Guan, L. (2012). An online power metering model for cloud environment. In *2012 ieee 11th international symposium on network computing and applications* (pp. 175–180).
- Lim, S.-H., Sharma, B., Nam, G., Kim, E. K. & Das, C. R. (2009). Mdcsim: A multi-tier data center simulation, platform. In *2009 ieee international conference on cluster computing and workshops* (pp. 1–9).
- Lin, W., Xu, S., He, L. & Li, J. (2017). Multi-resource scheduling and power simulation for cloud computing. *Information Sciences*, 397, 168–186.
- Long, S.-Q., Zhao, Y.-L. & Chen, W. (2014). Morm: A multi-objective optimized replication management strategy for cloud storage cluster. *Journal of Systems Architecture*, 60(2), 234–244.
- Lyman, P. (2003). How much information 2003? <http://www.sims.berkeley.edu/research/projects/how-much-info-2003/>.
- Majidpour, J. & Hasanzadeh, H. (2020). Application of deep learning to enhance the accuracy of intrusion detection in modern computer networks. *Bulletin of Electrical Engineering and Informatics*, 9(3), 1137–1148.
- Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C. & Byers, A. H. (2011). *Big data: The next frontier for innovation, competition* (Tech. Rep.). and productivity. Technical report, McKinsey Global Institute.
- Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J. & Ghalsasi, A. (2011). Cloud computing—the business perspective. *Decision support systems*, 51(1), 176–189.
- Mayer-Schönberger, V. & Cukier, K. (2013). *Big data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt.
- McCullough, J. C., Agarwal, Y., Chandrashekar, J., Kuppuswamy, S., Snoeren, A. C. & Gupta, R. K. (2011). Evaluating the effectiveness of model-based power characterization. In *Usenix annual technical conf* (Vol. 20).
- McGilvray, D. (2008). *Executing data quality projects: Ten steps to quality data and trusted information (tm)*. Elsevier.

- Morley, J., Widdicks, K. & Hazas, M. (2018). Digitalisation, energy and data demand: The impact of internet traffic on overall and peak electricity consumption. *Energy Research & Social Science*, 38, 128–137.
- Naumann, F. (2003). *Quality-driven query answering for integrated information systems* (Vol. 2261). Springer.
- Navimipour, N. J. & Milani, F. S. (2015). Task scheduling in the cloud computing based on the cuckoo search algorithm. *International Journal of Modeling and Optimization*, 5(1), 44.
- Pars, S. & Maleki, R.-R. (2009). A new task scheduling algorithm in grid environment. *International Journal of Digital Content Technology and its Applications*, 3(4), 152–160.
- Patterson, M. G. (1996). What is energy efficiency?: Concepts, indicators and methodological issues. *Energy policy*, 24(5), 377–390.
- Ranky, P. (2010). *Introduction to sustainable green engineering system analysis design*. IEEE.
- Redman, T. C. (1995). Improve data quality for competitive advantage. *MIT Sloan Management Review*, 36(2), 99.
- Redman, T. C. & Blanton, A. (1997). *Data quality for the information age*. Artech House, Inc.
- Rivoire, S., Ranganathan, P. & Kozyrakis, C. (2008). A comparison of high-level full-system power models. *HotPower*, 8(2), 32–39.
- Roy, S., Rudra, A. & Verma, A. (2013). An energy complexity model for algorithms. In *Proceedings of the 4th conference on innovations in theoretical computer science* (pp. 283–304).
- Salzmann, V. S. (2000). Are public records really public: The collision between the right to privacy and the release of public court records over the internet. *Baylor L. Rev.*, 52, 355.
- Segaran, T. & Hammerbacher, J. (2009). *Beautiful data: the stories behind elegant data solutions*. " O'Reilly Media, Inc."
- Sego, L. H., Márquez, A., Rawson, A., Cader, T., Fox, K., Gustafson Jr, W. I. & Mundy, C. J. (2012). Implementing the data center energy productivity metric. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 8(4), 1–22.
- Song, S. L., Barker, K. & Kerbyson, D. (2013). Unified performance and power modeling of scientific workloads. In *Proceedings of the 1st international workshop on energy efficient supercomputing* (pp. 1–8).
- Sreenivas, V., Prathap, M. & Kemal, M. (2014). Load balancing techniques: Major challenge in cloud computing - a systematic review. In (pp. 1–6). IEEE. doi: 10.1109/ECS.2014.6892523
- Stephenson, D. (2018). *Big data demystified: How to use big data, data science and ai to make better business decisions and gain competitive advantage*. Pearson UK.
- Takaishi, D., Nishiyama, H., Kato, N. & Miura, R. (2014). Toward energy efficient big data gathering in densely distributed sensor networks. *IEEE Transactions on Emerging Topics in Computing*, 2(3), 388–397.

- Tang, C.-J. & Dai, M.-R. (2011). Dynamic computing resource adjustment for enhancing energy efficiency of cloud service data centers. In *2011 IEEE/SICE International Symposium on System Integration (SII)* (pp. 1159–1164).
- Tian, Y., Lin, C. & Li, K. (2014). Managing performance and power consumption tradeoff for multiple heterogeneous servers in cloud computing. *Cluster computing*, 17(3), 943–955.
- Toffler, A. & Alvin, T. (1980). *The third wave* (Vol. 484). Bantam books New York.
- Tudor, B. M. & Teo, Y. M. (2013). On understanding the energy consumption of arm-based multicore servers. In *Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems* (pp. 267–278).
- Vatsal, S. & Agarwal, S. (2019). Energy efficiency metrics for safeguarding the performance of data centre communication systems by green cloud solutions. In (pp. 136–140). IEEE. doi: 10.1109/ICoN-CuTE47290.2019.8991478
- Wand, Y. & Wang, R. Y. (1996). Anchoring data quality dimensions in ontological foundations. *Communications of the ACM*, 39(11), 86–95.
- Wang, R. Y., Storey, V. C. & Firth, C. P. (1995). A framework for analysis of data quality research. *IEEE transactions on knowledge and data engineering*, 7(4), 623–640.
- Wang, R. Y. & Strong, D. M. (1996). Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, 12(4), 5–33.
- Wangsom, P., Lavangnananda, K. & Bouvry, P. (2017). Measuring data locality ratio in virtual mapreduce cluster using workflowsim. In *2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE)* (pp. 1–6).
- Widdicks, K., Bates, O., Hazas, M., Friday, A. & Beresford, A. R. (2017). Demand around the clock: time use and data demand of mobile devices in everyday life. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (pp. 5361–5372).
- Wu, J., Guo, S., Huang, H., Liu, W. & Xiang, Y. (2018). Information and communications technologies for sustainable development goals: state-of-the-art, needs and perspectives. *IEEE Communications Surveys & Tutorials*, 20(3), 2389–2406.
- Wu, J., Guo, S., Li, J. & Zeng, D. (2016). Big data meet green challenges: Greening big data. *IEEE Systems Journal*, 10(3), 873–887.
- Wu, J., Rangan, S. & Zhang, H. (2016). *Green communications: theoretical fundamentals, algorithms, and applications*. CRC press.
- Xiao, P., Hu, Z., Liu, D., Yan, G. & Qu, X. (2013). Virtual machine power measuring technique with bounded error in cloud environments. *Journal of Network and Computer Applications*, 36(2), 818–828.
- Xie, T. & Sun, Y. (2009). A file assignment strategy independent of workload characteristic assumptions. *ACM Transactions on Storage (TOS)*, 5(3), 1–24.
- Xiong, R., Du, Y., Jin, J. & Luo, J. (2018). Hadaap: A hotness-aware data placement strategy for improving storage efficiency in heterogeneous hadoop clusters. *Concurrency and Computation: Practice and Experience*, 30(20), e4830.
- Yao, Y., Huang, L., Sharma, A., Golubchik, L. & Neely, M. (2012). Data centers power reduction: A two time scale approach for delay tolerant workloads. In

- 2012 proceedings ieee infocom* (pp. 1431–1439).
- You, X., Dong, C., Zhou, L., Huang, J. & Jiang, C. (2015). Anticipation-based green data classification strategy in cloud storage system. *Applied Mathematics & Information Sciences*, 9(4), 2151.
- Yu, C. & Lai, L. (2016). Study on metrics model for energy efficiency in data centers. In *2016 international conference on sensor network and computer engineering*.
- Zhang, P. & Zhou, M. (2017). Dynamic cloud task scheduling based on a two-stage strategy. *IEEE Transactions on Automation Science and Engineering*, 15(2), 772–783.
- Zhou, W., Feng, D., Tan, Z. & Zheng, Y. (2018). Improving big data storage performance in hybrid environment. *Journal of computational science*, 26, 409–418.