

Xi'an Technological University (XATU)

Xi'an Technological University was founded in 1955. After more than fifty years of development, the school has become a multidisciplinary university with distinctive features and great impact. The school covers an area of 1290 mu, with building area of nearly 475,000 square meters, 1773 full-time regular teachers, including 126 professors, 483 teachers with senior professional titles; 430 teachers with doctor's degree.

At present there are 14 secondary colleges, 4 direct departments, one industrial center, more than 1,700 graduate students and 19000 undergraduates of various disciplines. The university has three doctoral degree disciplines: "Optical Engineering", "Materials Processing Engineering" and "Mechanical Engineering", two doctoral degree support disciplines: "Computer Science and Technology", "Management Science and Engineering", 6 subjects and disciplines with characteristics at the provincial level, 15 master degree authorization disciplines at first level, 62 master degree authorization centers and 13 master degree authorization centers for Master of Engineering and Master of Business Administration (MBA), 46 undergraduate majors, covering Engineering, Science, Management Science, Economics, Science of Law, Literature, Education and other seven disciplines. In the process of discipline construction and development, the school focus on highlighting science industry characteristics, take the road of production, teaching and research integration, promote the full integration of school and the local economic development, meets development requirements of weapons industry. The school has already had two national key research base, an international cooperative research centre and 11 provincial and ministerial key laboratories, with the ability to undertake large-scale military and civilian research projects in Optics, Machinery, Materials and other disciplines.

The school strengthens international exchange and cooperation, has established intercollegiate exchange relationship of research collaboration, student exchange programs with more than 20 renowned universities in more than 20 countries, such as The U. S., Japan, Britain, France, Germany, Australia and so on. Each year there are a number of teachers, students or the senior managers go abroad for learning, short-term training and investigation.

Website: www.xatu.edu.cn

Publisher: State and Provincial Joint Engineering Lab. of Advanced Network Monitoring and Control (ANMC)

Cooperate: Xi'an Technological University (CHINA)
West Virginia University (USA)
Huddersfield University of UK (UK)
Missouri Western State University (USA)
James Cook University of Australia (Australia)
National University of Singapore (Singapore)

Approval: Library of Congress of the United States
Shaanxi provincial Bureau of press, Publication, Radio and Television

Address: 4525 Downs Drive, St. Joseph, MO64507 , USA
No. 2 Xuefu Road,weiyang District, Xi'an, 710021, China

Telephone: +1-816-2715618 (USA) +86-29-86173290 (China)

Website: www.ijanmc.org

E-mail: ijanmc@jianmc.org
xxwlcncn@163.com

ISSN: 2470-8038

Print No. (China): 61-94101

Publication Date: Jun. 26, 2017



Editorial board

Editor in Chief

Professor Yaping Lei
Vice President
Xi'an Technological University, Xi'an, China

Associate Editor-in-Chief

Professor Wei Xiang
Electronic Systems and Internet of Things Engineering
College of Science and Engineering
James Cook University, AUSTRALIA

Houbing Song Ph.D.
Golden Bear Scholar and Professor
Department of Electrical and Computer Engineering
Director of Security and Optimization for Networked Globe Laboratory (SONG Lab)
West Virginia University, WV 25136 USA

Dr. Chance M. Glenn, Sr.
Professor and Dean
College of Engineering, Technology, and Physical Sciences
Alabama A&M University, Alabama 35762, USA

Professor Zhijie Xu
University of Huddersfield, UK
Queensgate Huddersfield HD1 3DH, UK

Professor Jianguo Wang
Vice Director and Dean
State and Provincial Joint Engineering Lab. of Advanced Network and Monitoring Control, CHINA
School of Computer Science and Engineering, Xi'an Technological University, Xi'an, CHINA

Administrator

Dr. & Prof. George Yang
Department of Engineering Technology
Missouri Western State University, St. Joseph, MO 64507, USA

Professor Zhongsheng Wang
Xi'an Technological University, CHINA
Vice Director
State and Provincial Joint Engineering Lab. of Advanced Network and Monitoring Control, CHINA

Associate Editors

Dr. & Prof. Yu Changyuan
Dept. of Electrical and Computer Engineering

National Univ. of Singapore (NUS)

Dr. Omar Zia
Professor and Director of Graduate Program
Department of Electrical and Computer Engineering Technology
Southern Polytechnic State University
Marietta, Ga 30060, USA

Dr. Liu Baolong
School of Computer Science and Engineering
Xi'an Technological University, CHINA

Dr. Mei Li
China university of Geosciences (Beijing)
29 Xueyuan Road, Haidian, Beijing 100083, P. R. CHINA

Dr. Ahmed Nabih Zaki Rashed
Professor, Electronics and Electrical Engineering
Menoufia University, Egypt

Dr. Rungun R Nathan
Assistant Professor in the Division of Engineering, Business and Computing
Penn State University - Berks, Reading, PA 19610, USA

Dr. Taohong Zhang
School of Computer & Communication Engineering
University of Science and Technology Beijing, CHINA

Dr. Haifa El-Sadi.
Assistant professor
Mechanical Engineering and Technology
Wentworth Institute of Technology, Boston, MA, USA

Huaping Yu
College of Computer Science
Yangtze University, Jingzhou, Hubei, CHINA

Tian Qichuan
School of Electric & Information Engineering
Beijing University of Civil Engineering & Architecture, Beijing, CHINA

Language Editor

Professor Gailin Liu
Xi'an Technological University, China

Dr. H.Y. Huang
Assistant Professor
Department of Foreign Language
The United States Military Academy
West Point, NY 10996, USA

CONTENTS

Optimal Pricing for Service Provision in an IaaS Cloud Market with Delay Sensitive Cloud Users/ <i>Gang Fang , Xianwei Li</i>	1	Research on Optimization of memory pool management for high concurrent service requests / <i>LIU Pingping, LU Zhaopan</i>	68
Improved K-means Algorithm Based on optimizing Initial Cluster Centers and Its Application/ <i>Xue Limyao, Wang Jianguo</i>	9	Image Watermarking Encryption Scheme Based on Fractional Order Chaotic System / <i>Dawei Ding , Zongzhi Li and Shujia Li</i>	79
Research and Implementation of Load Balancing Technology for Cloud Computing/ <i>Sun Hong, Wang Weifeng, Chen Shiping and Xu Liping</i>	17	The Design of Two Phase Chopping Regulation Voltage Soft Starter/ <i>Jingwen Chen*</i> , <i>Hongshe Dang</i>	90
Research of Virtual Network Classroom Collaborative Mechanism Based on Petri Net / <i>Shengquan Yang, Shujuan Huang</i>	29	Development of Levenberg-Marquardt Method Based Iteration Square Root Cubature Kalman Filter ant its applications/ <i>Jing Mu , Changyuan Wang</i>	98
A DCT Domain Image Watermarking Method Based on Matlab/ <i>Wu He-Jing</i>	38	The Prediction of Haze Based on BP Neural Network and Matlab/ <i>Ma Limei , Wang Fangwei</i>	107
A Mobile Terminal Security Strategy Based On the Cloud Storage/ <i>Wang Hui, Tang Junyong</i>	46	Multi Objective Optimization of Virtual Machine Migration Placement Based on Cloud Computing/ <i>Sun Hong , Tang Qing, Xu Liping and Chen Shiping</i>	120
Optimal Pricing Strategies for Resource Allocation in IaaS Cloud / <i>Zhengce Cai , Xianwei Li</i>	60		

Optimal Pricing for Service Provision in an IaaS Cloud Market with Delay Sensitive Cloud Users

Gang Fang¹, Xianwei Li^{2,3}

¹Trade Circulation Institute Anhui Business College Hefei, China
Email:1511154153@qq.com

²School of Information Engineering Suzhou University Suzhou, China

³Global Information and Telecommunication Institute Waseda University Tokyo, Japan
Email:lixianwei163@163.com

Abstract. Cloud computing has received a significant amount of attentions from both engineering and academic fields. Designing optimal pricing schemes of cloud services plays an important role for the success of cloud computing. How to set optimal prices of cloud resources in order to maximize these CSPs' revenues in an Infrastructure as a Service (IaaS) cloud market while at the same time meeting the cloud users' demand satisfaction is a challenging problem that CSPs should consider. However, most of the current works on cloud market are performed under the assumption that cloud users are not sensitive to delay, which is not practical. Towards this end, in this paper we study price-based service provision in an IaaS cloud market. Our simulations verify our analysis

Keywords: Price Competition, IaaS, CSP

1. Introduction

In recent years, cloud computing has received a significant amount of attentions from both engineering and academic fields and the use of cloud service is proliferating. Cloud computing can be defined by several ways, one widely adopted is proposed by Buyya et al.^[1]:

“A cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and the consumers”

Cloud services are mainly classified into three types^[2]: Infrastructure as a Service (IaaS), Software as a Service (SaaS) and Platform as a Service (PaaS). A recent study show that the market size of cloud computing will reach \$112 billion in 2018, in a large part due to IaaS cloud services^[3]. We focus on IaaS clouds in this paper, where CSPs deliver Infrastructure as a Service (IaaS) to cloud users. In the cloud computing environment, IaaS CSPs bundle their physical resources, such as CPU, memory and disk, into distinct types of virtual machine (VM) instances, according to their sizes and features, and offer them as services to users. Amazon EC2 is a public CSP which has hosted several types of VM instances (e.g. small, medium, large and extra large) based on the capacities of CPU, memory and disk^[4], the configurations of some VM instances are shown in Table 1. Cloud users purchase units of computing

time on VM instances to run their jobs.

The rapidly increasing demand for cloud resources from business and individuals is making resource management become the heart of CSPs' decision-process^[5], and pricing provides an effective approach to addressing this issue. Since the amount of resources that users' request is much smaller than the capacity of CSPs^[6], a rational user will subscribe to choose services from the one that maximizes its net reward, i.e., the utility which measures its satisfaction from using cloud service. With more and more IaaS CSPs beginning to provide cloud services, they compete with each other for existing and attract future cloud users. On one hand, CSPs want to charge more from users to maximize their revenues. On the other hand, if they set the prices of cloud services too high, they may have the risk of losing cloud users in the long run. Therefore, how to set the optimal prices to make the revenue maximized while attracting cloud users is a challenging problem, especially when CSPs have different cloud capacities. Furthermore, computing resources, such as CPU cycles and disk, are inherently perishable, that is, they are of no value if they are not utilized in time^[7]. In addition, even for the similar type of VM, different CSPs have different prices. For example, although Amazon EC2 m1.medium and Google n1-standard1 have the similar configurations (one virtual CPU and 3.75 GB RAM), they have different prices for one-hour usage.

Recent studies report that different IaaS CSPs process tasks with different completion time^[8]. From the perspective of cloud users, besides price quality of service (QoS) is also an important factor that affects the choice of them. Although QoS can be measured by several parameters, such as response time, availability and throughput, all of which can be determined by making use of the tool of queueing theory^{[9][10]}. We mainly focus on response time as the measurement of QoS^{[8][11]}.

Table 1 Configurations of Some Amazon EC2 VM Instances

Instance Types	Compute Unit	Storage (GB)	Memory (GiB)
c3.large	2	32SSD	3.75
c3.xlarge	4	80SSD	7.5
c3.2xlarge	8	160SSD	15
c3.4xlarge	16	32SSD	30
c3.8xlarge	32	80SSD	60

A significant amount of works have been devoted to resource management in cloud computing, but only a small fraction of them involved performance issues. In^[11], the authors we presented an aggressive virtualized resource management system for IaaS clouds based on reinforcement learning approach. Hong et al.^[7] investigated optimal resource allocation for cloud users in an IaaS cloud by developing a dynamic programming algorithm to minimize CSPs' costs. The authors in^[3] studied optimal resource allocation in a federated cloud, and they proposed a cloud federation mechanism that enables IaaS CSPs to maximize their profits. Kantere et al. studied the correlation between user demand and the price, and proposed a novel price-demand model to maximize the CSPs' profits^[8]. However, these previous works only considered delay-tolerant jobs ignoring delay which is of great important for users who run delay-sensitive jobs. This is because the delayed response time may discourage cloud users to subscribe cloud service or make them switch to other CSPs, which will cause revenue loss. Recent study shows

that every 100ms of latency cost Amazon 1 percent in sales and traffic dropped 20 percent if an extra 0.5 seconds happened in search page generation time in Google^[11].

Queuing theory are widely adopted to model CSPs' data centres and computing platforms. Feng et al. studied price competition in an oligopoly cloud market with multiple IaaS CSPs, each of which is modelled as an M/M/1 queue^[8]. Atmaca et al. proposed a G/G/c-like queuing model to represent a cloud computing system and compute expected performance indices. Their model has the advantage in that it can represent general distributions of workloads on the arrival and service patterns in the cloud computing systems^[12]. Khazaei et al. present an approximate model by using an M/G/m/m+K queue with general service time and Poisson arrivals to evaluate the performance of active VM instances^[13]. Based on^[13], similar model is also adopted by Chang et al. for the study of an IaaS cloud data center^[14]. A hierarchical stochastic model is proposed by in^[13] to analyze several factors such as variation in job arrival rates and buffer size that affect the quality of cloud service. Most of the aforementioned works are carried out under the assumption that there is an IaaS CSP in the cloud market, which is not realistic as cloud market is becoming more and more fierce with an increasing number of CSPs begin to provision cloud services. Only few works take competition between CSPs into account (such as^[8]) restricted to homogeneous cloud markets, that is, these IaaS CSPs have homogeneous cloud capacities. Without considering users' utilities, the heterogeneous cloud market is originally explored in^[15], where the authors analyze the price competition between a public CSP and a cloud broker.

In this paper, we study price competition in a heterogeneous IaaS cloud market by taking CSPs' heterogeneous cloud capacities into consideration. We consider a monopoly cloud market where a resource-constrained CSP modelled as an M/M/1 queuing system offers services to a potential stream of cloud users. Given the price of cloud service, we analyze cloud users' joining policy and show that there exists a unique equilibrium arrival rate to CSP.

2. System Model

In this section, we introduce the models of cloud users' and CSPs. As illustrated in Fig.1, we consider an IaaS cloud computing market with two CSPs to compete for a potential stream of cloud users. CSP1 has constrained cloud resources while CSP2 has sufficient cloud resources, that is, the cloud market is heterogeneous. One example is that CSP1 is an entrant CSP and CSP2 is an incumbent one

2.1 Cloud Users' Model

We assume that the tasks of users arrive at the cloud market with rate Λ following Poisson and they are served according to first-come-first-served (FCFS) queueing. According to recent studies for the analysis of cloud data centers, it is generally accepted that users' service requests arrive at the cloud servers follow Poisson distribution^[16]. Similar to^[14], we also assume that each job consists of one task, which is single-task job. Each user is supposed to carry a different task, therefore, we use task and user interchangeably. Upon arrival, each cloud user will make a decision to choose from one of the two CSP based on prices and quality of service (QoS) to buy cloud services to execute its task. The jobs of users can be classified into two types^[17]: interactive (delay-sensitive) jobs, such as web service, and batch (delay-tolerant) jobs, such as scientific applications. We focus on the study of delay-sensitive jobs. Based on the above assumptions, the utility that a cloud user get from using cloud service of CSP is denoted as

$$U = R - cw(\lambda) - p \quad (1)$$

where R is the reward from using cloud service, $w(\lambda)$ is the delay time in the cloud system of CSP, c is the delay cost per unit time and p is the per unit time price of VM instance of CSP. Similar utilities functions are widely used in the cloud computing literature^{[8][15][18]}.

2.2 CSP model

We model the CSP as an M/M/1 queue whose resource capacity is characterized by service rate μ (in tasks/s) as illustrated in Fig.2.

3. Monopoly Cloud Market

We study a monopoly cloud market, where there is a CSP provisioning cloud services to a potential stream of cloud users. Cloud users arrive at the cloud market with rate Λ . We analyze the relationship between the CSP and users as a two stage Stackelberg game, as illustrated in Figure 3. In the first stage, CSP sets optimal prices to maximize its revenue given the arrival rates of users. In the second stage, cloud users make their arrival rates decision based on the prices of cloud services. The Stackelberg game is solved by using backward induction method^[19].

A cloud user's net utility from using the cloud service is modeled as

$$U = R - p - cw(\lambda) \quad (2)$$

where $w(\lambda) = \frac{1}{\mu - \lambda}$ is the response time includes waiting time and processing time. We assume $\mu > \lambda$ in order to stabilize the queue.

To maximize his utility, a cloud user will pay to use this CSP's service if

$$U = R - p - cw(\lambda) \geq 0 \quad (3)$$

and refuse to use it otherwise.

Similar to the existing works [8] [20] we consider the equilibrium case, which means

$$R - p - c \frac{1}{\mu - \lambda} = 0 \quad (4)$$

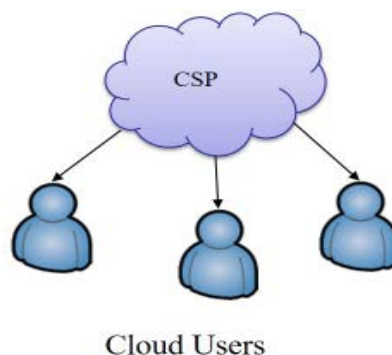


Figure.1 An IaaS cloud market

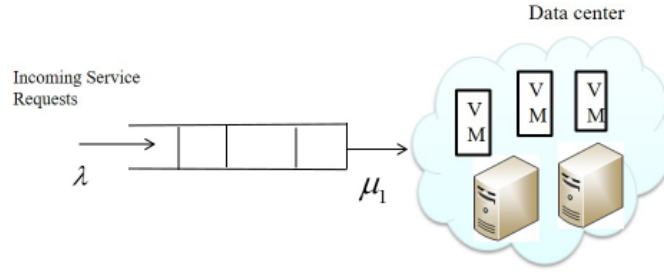


Figure.2 CSP1 is modeled as an M/M/1 queue

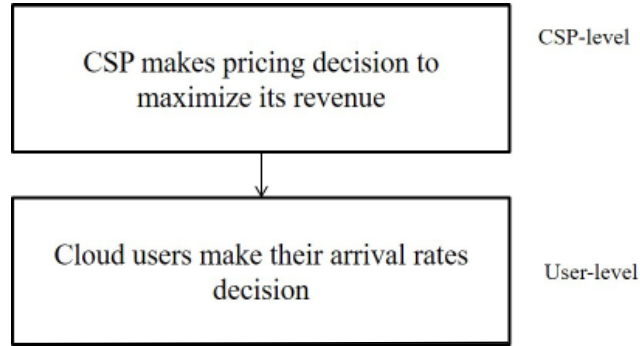


Figure.3 A Two-Stage Stackelberg Game

From the Eq. (4), we get

$$\lambda = \mu - \frac{c}{R-p} \quad (5)$$

If the CSP cannot take the whole cloud market in equilibrium, otherwise. So the actual market share of the CSP is

$$\lambda = \min\{\Lambda, \mu - \frac{c}{R-p}\} \quad (6)$$

The revenue of the CSP per unit time is

$$\max_{0 < p < R - \frac{c}{\mu - \lambda}} \pi = p\lambda \quad (7)$$

where λ is given by (6).

The equilibrium price p^* is equal to the first-order price, the form of which is

$$p_m^* = R - \sqrt{\frac{cR}{\mu}} \quad (8)$$

with the corresponding market share is

$$\lambda = \min\{\Lambda, \mu - \sqrt{\frac{\mu c}{R}}\} \quad (9)$$

If the CSP can take the entire cloud market, e.g., $\lambda = \Lambda$, then the market capture price is

$$p_{\Lambda} = R - c \frac{1}{\mu - \Lambda} \tag{10}$$

4. Performance Evaluation

In this section, we do simulations to verify our analysis in the previous sections. In particular, we analyze cloud users' equilibrium arrival rates and CSP's revenue to several parameters, such as reward values, delay cost and service rate.

4.1 Cloud Users' Equilibrium Arrival Rates versus Prices

We first analyze how users' equilibrium arrival rates versus prices of cloud services p . As shown in Figure 4, equilibrium arrival rates not only decrease with increasing values of prices, but also decrease with delay cost value c increasing.

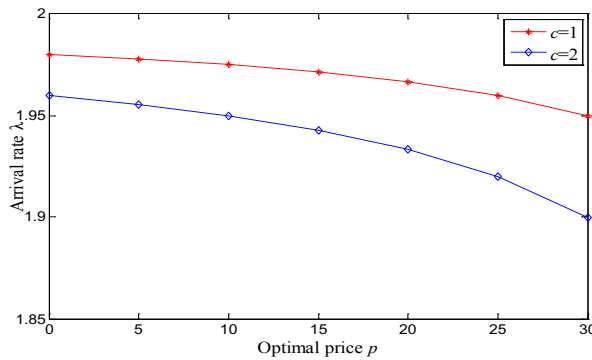


Figure.4 Cloud users' arrival rate vs optimal price p with $r=50$, $\mu =2$.

4.2 CSP's Revenues versus Service Rates

We next analyze how CSP's revenues vary with service rates.

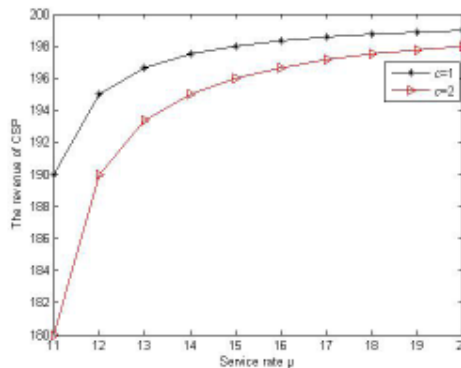


Figure.5 The revenue of CSP versus service rate with $R=20$, $\Lambda =10$.

5. Conclusions

We studied duopoly price competition in an IaaS cloud market in this paper. We model the interactions between CSPs and users a two-stage Stackelberg game, where CSPs set optimal prices to make revenues maximized in the first stage, then cloud users make their arrival rates decision in the second stage. We

consider two cloud market cases. The first case is the total arrival rates of the two CSPs is smaller than the market size, and the second case is the total arrival rates of the two CSPs is equal to the market size.

In future works, we will extend our study to duopoly and oligopoly cloud market and study other pricing schemes, such as reservation and spot pricing schemes. We will also study how to segment cloud resources with different pricing schemes.

Acknowledgment

This paper is supported by the following projects, Anhui Key research projects of Humanities and Social Sciences (SK2016A0207), and Suzhou Regional Collaborative Innovation Center (2016szxt05).

References

- [1] R. Buyya, C.S. Yeo, and S. Venugopal, "Market Oriented Cloud Computing: Vision, Hype, and Reality for Delivering it Services as Computing Utilities", Proc. 10th IEEE Conference on High Performance Computing and Communications (HPCC 2008), pp. 5-13, Sept. 2008.
- [2] D. Bruneo, "A stochastic model to investigate data center performance and QoS in IaaS cloud computing systems", IEEE Trans. Parallel Distrib. Syst., vol.25, no.3, pp.560 – 569, March 2014.
- [3] L. Zheng, Carlee Joe-Wong, and C. G. Brinton et al. "On the Viability of a Cloud Virtual Service Provider", Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science (SIGMETRICS 2016), Antibes Juan-les-Pins, France, pp. 235-248, June 2016.
- [4] Amazon EC2 Pricing. <http://aws.amazon.com/cn/ec2/pricing/>.
- [5] L. Mashayekhy, M. M. Nejad, and D. Grosu, "Physical machine resource management in clouds: A mechanism design approach", IEEE Trans. Cloud Comput., vol.3, no.3, pp.247 – 260, July-Sep. 2015.
- [6] T.T. Huu and C.K. Tham, "A novel model for competition and cooperation among cloud providers", IEEE Trans. Comput., vol.2, no.3, pp.251 – 265, July-Sep. 2014.
- [7] H.Xu, B.Li, "Maximizing revenue with dynamic pricing: the infinite horizon case", Proc. IEEE Conference on Communications (ICC 2012), Ottawa, Canada, pp. 2929-2933, June 2012.
- [8] Y. Feng, B. Li, and B. Li, "Price competition in an oligopoly market with multiple IaaS cloud providers", IEEE Trans. Comput., vol. 63, no. 1, pp. 59-73, Jan. 2014.
- [9] J. Liu, Y. Zhang, and Y. Zhou et al., "Aggressive resource provisioning for ensuring qos in virtualized environments", IEEE Trans.Cloud Comput., vol.3, no.2, pp.119 – 131, April-June 2015.
- [10] T. Atmaca, T. Begin, and A. Brandwajn et al., "Performance Evaluation of Cloud Computing Centers with General Arrivals and Service", IEEE Trans. Parallel Distrib. Syst., vol.27, no.8, pp.2341 – 2348, Aug. 2016.
- [11] J. Liu, Y. Zhang, and Y. Zhou et al., "Aggressive resource provisioning for ensuring qos in virtualized environments", IEEE Trans.Cloud Comput., vol.3, no.2, pp.119 – 131, April-June 2015.
- [12] T. Atmaca, T. Begin, and A. Brandwajn et al., "Performance Evaluation of Cloud Computing Centers with General Arrivals and Service", IEEE Trans. Parallel Distrib. Syst., vol.27, no.8, pp.2341 – 2348, Aug. 2016.
- [13] H. Khazaei, J. Mistic, and V. B. Mistic, "Performance analysis of cloud computing centers using M/G/m/m+r queuing systems", IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 5, pp. 936 – 943, May 2012.
- [14] X. Chang, B. Wang and J. K. Muppala et al., "Modeling Active Virtual Machines on IaaS Clouds Using an M/G/

m/m+k Queue” , IEEE T. Serv. Comput., vol. 9, no. 3, pp. 408 – 420, May 2016.

[15] C.T. Do, N.H. Tran, and E.N. Huh et al., “Dynamics of service selection and provider pricing game in heterogeneous cloud market” , Journal of Network and Computer Applications, vol.69, pp.152 – 165, July 2016.

[16] M. Liu, W. Dou, and S. Yu et al., “A decentralized cloud firewall framework with resources provisioning cost optimization” , IEEE Trans. Parallel Distrib. Syst., vol.26, no.3, pp.621 – 631, March 2015.

[17] Z. Liu, M. Lin, and A. Wierman et al., “Greening geographical load balancing” , IEEE/ACM Trans. Netw., vol.23, no.2, pp.657 – 671, April 2015.

[18] J. Chen, C. Wang, and B. Zhou et al., “Tradeoffs between profit and customer satisfaction for service provisioning in the cloud” , Proc. Of the 20th international symposium on High performance distributed computing (HPDC 2011), San Jose, California, USA, pp.229 – 238, June 2011.

[19] D. Fudenberg and J. Tirole, “Game theory” , MIT Press, Cambridge, MA, USA, 1991.

[20] C. Liu, K. Li, and C. Xu et al., “Strategy Configurations of Multiple Users competition for cloud service reservation” , IEEE Trans. Parallel Distrib. Syst., in press.

Improved K-means Algorithm Based on optimizing Initial Cluster Centers and Its Application

Xue Linyao, Wang Jianguo

School of Computer Science and Engineering,

Xi'an Technological University, Xi'an, 710032 China

Email: xuelinyaoyao@foxmail.com

Abstract. Data mining is a process of data grouping or partitioning from the large and complex data, and the clustering analysis is an important research field in data mining. The K-means algorithm is considered to be the most important unsupervised machine learning method in clustering, which can divide all the data into k subclasses that are very different from each other. By constantly iterating, the distance between each data object and the center of its subclass is minimized. Because K-means algorithm is simple and efficient, it is applied to data mining, knowledge discovery and other fields. However, the algorithm has its inherent shortcomings, such as the K value in the K-means algorithm needs to be given in advance; clustering results are highly dependent on the selection of initial clustering centers and so on. In order to adapt to the historical data clustering of the geological disaster monitoring system, this paper presents a method to optimize the initial clustering center and the method of isolating points. The experimental results show that the improved k-means algorithm is better than the traditional clustering in terms of accuracy and stability, and the experimental results are closer to the actual data distribution.

Keywords: Clustering analysis, improved K-means algorithm, geological disaster monitoring data

1. Introduction

The occurrence of geological disasters caused great casualties to humans, the main reasons include landslides and debris flow and rainfall and so on. And these geological disasters always cause many local public facilities to be damaged by large and small, and brought great damage to the people and their property. Also, there are still many such cases in China. Faced with such a severe threat of geological disasters, the state and the government on the prevention and control of geological disasters into a lot of human and material resources, and achieved remarkable results. With the progress of technology and high development of information technology, many new detection equipments have been put into the geological disaster real-time detection, such as GPS, secondary sound wave monitoring, radar and so on.

With the development of geological hazard detection technology, the amount of the monitoring data grew by leaps and bounds, data types are becoming more and more complex as well. K-means algorithm is a clustering algorithm based on the classification of the classic algorithm, the algorithm in the industrial and commercial applications more widely. As we all know, it both has many advantages and many disadvantages. The research on the deficiency of K-means algorithm is divided into two branches: 1) the

number of initial clustering centers K ; 2) the choice of initial clustering center. In this paper, we mainly study the latter, and propose a new initial clustering center algorithm.

The data source of the study is the historical data detected by the geological disaster monitoring system, and 2000 records are randomly selected from the rainfall data of different areas in Shaanxi Province as the research object, which are served as a representative sample of the improved K-means clustering algorithm. The experimental results show that the algorithm is better than the traditional clustering in terms of accuracy and stability, and the experimental results are closer to the actual data distribution.

2. Brief and Research Status of K-Means Algorithm

2.1 Overview of K-means algorithm

The K-means algorithm is a classical unsupervised clustering algorithm. The purpose is to divide a given data set containing N objects into K clusters so that the objects in the cluster are as similar as possible, and the objects between clusters are as similar as possible. Set the sample set $X = \{x_1, x_2, x_3, \dots, x_n\}$, n is the number of samples. The idea of the K-means algorithm is: Firstly, k data objects are randomly selected from the sample set X as the initial clustering center; Secondly, according to the degree of similarity between each data object and k clustering centers, it is allocated to the most similar clusters; Then recalculate the average of each new cluster and use it as the next iteration of the clustering center, and repeat the process until the updated cluster center is consistent with the update, that is, the criterion function E converges.

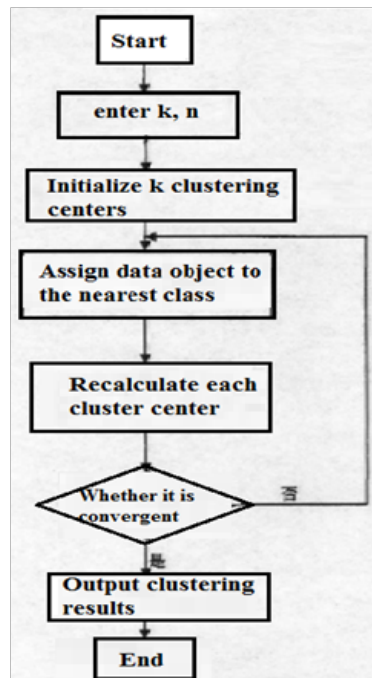


Figure.1 K-means flow

The goal is to make the object similarity in the cluster the largest, and the similarity between the objects is the smallest. The degree of similarity between the data can be determined by calculating the Euclidean distance between the data. For the n -dimensional real vector space, the Euclidean distance of two points is defined as:

$$d(x, y) = \sqrt{(x_i - y_i)^2} \quad (1)$$

Here, x_i and y_i are the attribute values of x and y respectively, and the criterion function is defined as:

$$E = \sum_{i=0}^n \sum_{x \in C_i} |x - \bar{x}_i|^2 \quad (2)$$

Here, k is the total number of clusters, and \bar{x}_i is the center of cluster c . The flow of K-means algorithm is shown in Figure 1.

2.2 Research status quo of K-means algorithm

For the advantages of K-means algorithm, it has been widely used in practice, but there are many shortcomings as well. In order to get better clustering effect, many researchers have explored the shortcomings of improving K-means. Aiming at the shortcomings of K-means algorithm in selecting the initial point, many scholars have proposed an improved method. Duan Guiqin^[1] uses the method of product based on mean and maximum distance to optimize the initial clustering center. The algorithm first selects the set of data objects which are the farthest from the sample set to join the clustering center, and then the set of mean and current poly The largest data object of the class center is added to the clustering center set, which improves the accuracy. Yi Baolin^[2] et al. proposed another improved K-means algorithm, which first calculates the density of the region to which the data object belongs, and then selects k points as the initial center in the high density region. The experimental results show that the algorithm reduces the initial center point Impact. Yiu-Ming Cheng^[3] and others proposed a new clustering technique called K^* -means algorithm. The algorithm consists of two separate steps. A center point is provided for each cluster in the first step; and then adjust the unit through adaptive learning rules in the second step. The algorithm overcomes the shortcomings of K-means algorithm initial center sensitivity and K value blindness, but the calculation is complicated. Xie and others^[4] proposed a k -means algorithm to optimize the initial clustering center by using the minimum variance based on the sample space distribution compactness information. The algorithm chooses the samples with the smallest variance and a distance away from each other as the initial clustering center. Liu Jiaying et al.^[5] proposed a radius-based k -means + λ algorithm. When selecting the initial center point of the cluster, the distance ratio between points is calculated from the λ parameter and rounded at a specific distance. In the circle, an initialized center point is selected according to the distance ratio, and the algorithm has higher performance in error rate and operation time. Ren Jiangtao^[6] proposed an improved K-means algorithm for text clustering, which is improved by using feature selection and dimension reduction, sparse vector selection, initial center point search based on density and spreading, Class accuracy, stability and other aspects have improved.

2.3 The analysis of shortcomings of K-means algorithm

- 1) The K value in the K-means algorithm needs to be given in advance. According to the K value determined in advance, the clustering samples are classified into K class, so that the sum of squares of all the samples in the clustering domain to the clustering center is minimized.
- 2) Clustering results are highly dependent on the selection of initial clustering centers. The K-means algorithm uses the stochastic method to select the initial clustering center. If the initial clustering center is chosen improperly, it is difficult to obtain the ideal clustering effect. This dependence on the initial value may lead to the instability of the clustering results, and it is easy to fall into the local optimal rather than the global optimal results.

3) Sensitive to noise and isolated points.

3. Improvement of K-Means Algorithm and Its Application

3.1 The selection of data object in Cluster analysis

The preliminary data are collected firstly when data selecting, then know about the characteristics of data to identify the quality of the data and to find a basic observation of the data or assume the implied information to monitor the subset of data of interest. The data object segmentation variable determines the formation of clustering, which in turn affects the correct interpretation of the clustering results, and ultimately affects the stability of the clustering clusters after the new data objects are added. Before the K-means clustering related data mining, the sample data set related to the data mining clustering analysis should be extracted from the original data object set, and it is not necessary to use all the historical data. In addition, we should pay attention to the quality of data, only high-quality data to the correct analysis of conclusions everywhere, to provide a scientific basis for clustering.

The source of this research object is the historical monitoring data of the geological disaster monitoring system. From the records of geological monitoring data from 2015 to 2016, a representative sample of K-means clustering algorithm for this improved algorithm is selected as the object of study in 2000, and the two samples of 3D rainfall are randomly selected in different regions.

The sample data attributes show as table1:

Table 1 The sample data attributes

Field number	Field name	Field code	Type of data
1	Id	xx	Number
2	Sno	yy	Varchar
3	Type	type	Varchar
4	Gettime	time	Datetime
5	Alarm Level	alarm	Integer
6	Value	value	Double
7	Day Value	d_value	Double

For the cluster analysis, there are obviously redundant ones in the data attributes of the above geological hazard monitoring system, and it does not have the objectivity of the cluster analysis data. Therefore, the redundant ones should be eliminated. Finally, only four data object attributes reflecting the characteristics of rainfall data are selected as the research object. The optimized data attributes show as table2:

Table 2 The optimized data attributes

Field number	Field name	Field code	Type of data
1	Id	xx	Number
2	Sno	yy	Varchar
3	Gettime	time	Datetime
4	Day Value	d_value	Double

3.2 Improvement of K-means algorithm

For the above geological disaster monitoring system rainfall data characteristics, the K-means algorithm is very sensitive to the initialization center, and the initial clustering center is very easy to make the clustering result into the local optimum and the influence of the isolated point is large. The algorithm is based on the small cluster with the largest variance and can be divided into two clusters with different variance. The algorithm of initializing center is proposed. In addition, a method of isolating points has been proposed. The idea of this algorithm is to first find out the two points furthest from the sample point as the initial center point, and then divide the other sample points into the cluster to which the nearest center point belongs, and determine the number of points within the cluster. And whether the corresponding initial clustering center is an isolated point, and finally select the next object to be split according to the variance within the cluster and update the initial cluster center according to certain rules. The above steps are repeated until the number of cluster centers is satisfied.

1) Initial clustering center selection algorithm

$X=\{x_1,x_2,x_3,\dots,x_n\}$, n is the number of samples. $d(x_i, x_j)$ ($i, j \in \{1,2,\dots,n\}$) is the Euclidean distance between the data points x_i and x_j , c_i ($i \in \{1,2,\dots,n\}$) is the clustering center, Q is the data object that will be spited, S is the number of clustering centers.

The initial clustering center selection algorithm is as follows:

Input: data set X , number of clusters k , threshold u

Output: cluster center set C and isolated point set D

(1) Let $Q=X=\{x_1,x_2,x_3,\dots,x_n\};S=0;$

(2) Calculate the Euclidean distance $d(x_i,x_j)$ between the two data points in W , and find the two points x_i,x_j , which are marked as c_i, c_j , and

let:

$S = S + 2;$

$$Q_i = \{x_p \mid d(x_p, x_i) < d(x_p, x_j), x_p \in Q\},$$

$$Q_j = \{x_p \mid d(x_p, x_j) < d(x_p, x_i), x_p \in Q\},$$

which means Q is divided by the cluster, and Q_i and Q_j become the split clusters.

(3) If the number of data objects in Q_i or Q_j is less than u , the selected initial center x_i or x_j is an isolated

point. Remove x_i or x_j from Q , remove c_i or c_j in set C , and add x_i or x_j to D , return to step 1;

(4) If the number of data objects in the set C is less than k , find the set Q_p with the largest variance in the splitting cluster and let $Q=Q_pQ_p$, $S=S-1$, then remove c_{pcp} the set C ;

(5) Calculate the mean of all the objects in the split cluster, and the resulting mean is k initial clustering centers.

2) Improved K-means algorithm

Data set $X=\{x_1,x_2,x_3,\dots,x_n\}$, there are n objects. $C_{o,i}$ represents the i -th cluster center of the previous round, $C_{n,i}$ represents the new cluster center calculated in current time, and the algorithm is described as follows:

Input: data set X , number of clusters k , threshold u

Output: k clusters and the number of bands

(1) Call the improved initialization center selection algorithm to get the initialization center, if there is an isolated point will be isolated points alone in a class, do not participate in the follow-up clustering algorithm;

(2) Calculate the distance between all data objects and k cluster centers, and assign the text to the nearest cluster;

(3) Calculate the mean of each cluster to obtain a new round of cluster center;

(4) If $E' = \sum_{i=1}^k \sum_{x \in c_i} |C_{o,i} - C_{n,i}|^2 < 10^{-10} E' = \sum_{i=1}^k \sum_{x \in c_i} |C_{o,i} - C_{n,i}|^2 < 10^{-10}$, then the iteration is terminated, otherwise it returns to 2). (note: E' is the measure function)

4. Experiment Analysis

4.1 Experimental description

The data set selected from the experiment comes from the rainfall data collected in the geological hazard detection system and the rainfall data set after the artificial noise is added. The experimental environment is: Inter(R)Core(TM)i3-2330M,4G RAM, 250G hard disk, Win7 operating system.

In order to verify the validity and stability of the improved algorithm, the original k -means algorithm, the algorithm in literature^[4] and the improved algorithm are analyzed and compared under the rainfall data set. In order to further verify the superiority of the algorithm in dealing with isolated points, the algorithm is compared with other algorithms on the rainfall data set after adding noise. The clustering results are clustered and criterion function changes and the clustering time are used to evaluate the clustering results.

4.2 Experimental results and analysis

The clustering criterion function of the two algorithms will decrease with the increase of the number of the adjustment of the cluster until the final convergence, and the more compact the two curves, the higher the accuracy of the corresponding clustering results The vice versa. Figure 2 is the comparison of the traditional k -means algorithm and the improved algorithm standard function values with the clustering canroids adjustment and constantly changing the comparison chart.; In order to test the

speed of the improved algorithm in this paper, three samples were randomly selected from the historical data of the geological hazard system, and the sample capacity was 5000, 10000 and 18000 respectively. The experimental results are shown in Figure 3.

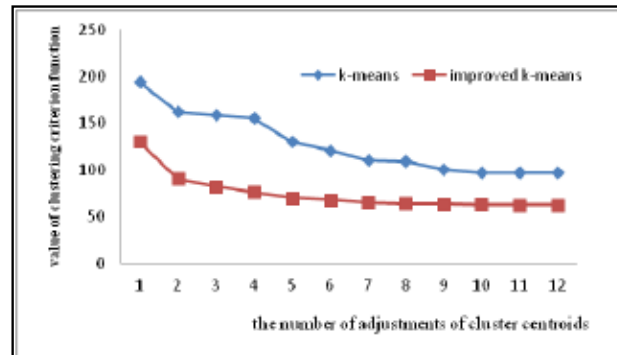


Figure.2 The comparison of criterion function changes trend graph

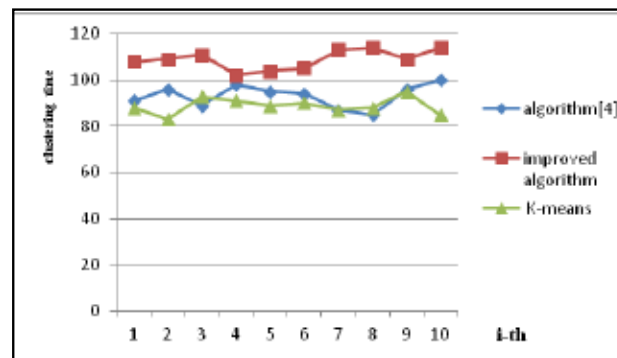


Figure.3 The comparison of clustering times in artificial data sets

According to the comparison of criterion function change trend graph in the rainfall data set in Figure 6, the clustering criterion function of the improved algorithm is superior to the clustering criterion function value of the traditional k-means algorithm because the data objects in the optimized cluster are more compact and independent in each iteration process, the criterion function value is significantly lower than the traditional k-means algorithm, which also further validates the superiority of this algorithm; Figure7 shows that the traditional k-means algorithm is running at the fastest speed, and the speed of this algorithm is slightly lower than that of algorithm^[4].

5. Conclusion

Aiming at the instability of the clustering results caused by the random clustering of the traditional k-means algorithm and the effect of the isolated points on the clustering results, the authors of this paper have the advantages of small distance from the large sample points to the same cluster. The clustering algorithm with the largest variance of variance can be split into two clusters with relatively small variance, a k-means clustering algorithm is proposed to optimize the initial clustering center. Simulation experiments in geological hazard systems and artificial data sets with the same proportion of noise show that the proposed algorithm improves the accuracy and clustering error compared with the traditional k-means algorithm and the other two optimization initial center algorithms. However, the initial algorithm of the algorithm is somewhat complicated, and it takes too much time in the selection of the central problem. In the future work, it will be further improved, and it will be tried in all respects.

References

- [1] Zhai D H, Yu J, Gao F, et al. k-means text clustering algorithm based on initial cluster centers selection according to maximum distance [J]. Application Research of Computers, 2014, 31(3):379 – 382.
- [2] Baolin Yi, Haiquan Qiao, Fan Yang, Chenwei Xu. An Improved Initialization Center Algorithm for K-Means Clustering[C]. Computational Intelligence and Software Engineering, 2010, pp:1-4.
- [3] Redmond S J, Heneghan C. A method for initializing the K-means clustering algorithm using kd-trees[J]. Pattern recognition letters, 2007, 28(8):965-973.
- [4] Liu J X, Zhu G H, Xi M. A k-means Algorithm based on the radius [J]. Journal of Guilin University of Electronic Technology, 2013, 33(2):134-138.
- [5] Habibpour R, Khalipour K. A new k-means and K-nearest-neighbor algorithms for text document clustering [J]. International Journal of Academic Research Part A, 2014, 6(3) : 79 – 84.
- [6] Data mining techniques and applications – A decade review from 2000 to 2011 [J]. Shu-Hsien Liao, Pei-Hui Chu, Pei-Yuan Hsiao. Expert Systems With Applications . 2012 (12)
- [7] Application of Improved K-Means Clustering Algorithm in Transit Data Collection. Ying Wu, Chun long Yao. 2010 3rd International Conference on Biomedical Engineering and Informatics (BMET). 2010.
- [8] Zhou A W, Yu Y F. The research about clustering algorithm of K-means [J]. Computer Technology and Development, 2011, 21(2):62-65.
- [9] Duan G Q. Auto generation cloud optimization based on genetic algorithm [J]. Computer and Digital Engineering, 2015, 43(3):379-382.
- [10] Wang C L, Zhang J X. Improved k-means algorithm based on latent Dirichlet allocation for text clustering [J]. Journal of Computer Applications, 2014, 34(1):249-254.
- [11] Deepa V K, Geetha J R R. Rapid development of applications in data mining [C]. Green High Performance Computing (ICGHPC), 2013, pp:1-4.
- [12] Sharma S, Agrawal J, Agarwal S, et al. Machine learning techniques for data mining: A survey [C]. Computational Intelligence and Computing Research (ICCIC), 2013, pp:1-6.
- [13] Efficient Data Clustering Algorithms: Improvements over Kmeans [J]. Mohamed Abubaker, Wesam Ashour. International Journal of Intelligent Systems and Applications (IJISA) . 2013 (3).
- [14] Fahad A, Alshatri N, Tari Z, Alamri A. A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis [C]. Emerging Topics in Computing. 2014:267-279.
- [15] Abubaker M, Ashour Wesam. Efficient data clustering algorithm algorithms: improvements over k-means [J]. International Journal of Intelligent Systems and Applications. 2013(3):37-49.
- [16] Tang Zhaoxia, Zhang Hui. Improved K-means Clustering Algorithm Based on Genetic Algorithm [C]. Telkomnika Indonesian Journal of Electrical Engineering. 2014, pp:1917-1923.
- [17] Optimal variable weighting for ultrametric and additive tree clustering [J]. Geert Soete. Quality and Quantity . 1986 (2).

Research and Implementation of Load Balancing Technology for Cloud Computing

Sun Hong¹, Wang Weifeng², Chen Shiping³ and Xu Liping⁴

**¹University of Shanghai For Science and Technology, China,
sunhong@usst.edu.cn**

**²University of Shanghai For Science and Technology, China,
wwfhuo@163.com**

**³University of Shanghai For Science and Technology, China,
chensp@usst.edu.cn**

**⁴University of Shanghai For Science and Technology, China,
Email:5850487@qq.com**

Abstract. This article selects load balancing system technology to analyze, combines the live migration technology of virtual machine, and studies the frame of virtual machine live migration as well as the mathematical model applied to concrete process of the migration. The article presents that the process of combining the specific strategies of load balance to the frame of live migration, that the simulation experiment and conclusion to this total process. The study takes Eucalyptus as experimental platform, and decides initial to take Xen as experimental virtual machine. The article's innovations are to optimize the design of virtual machine live migration in cloud environment, and to combine the specific strategies of load balance to the process of live migration. After designing and analyzing the specific strategies of load balance modularization, according to rationalizations that the technology of live migration can apply to the specific strategies of load balance, the study sets the measure index to evaluate load balance, and solve consequently the problem of load in cloud environment, which is realized in the process of live migration. It turned out that algorithm fusion raised has obvious performance advantages.

Keywords: Cloud Computing, Visualization, Modularization, Live Migration, Load Balance

1. Introduction

With the fast development of the Internet and network, people rely more and more increasingly on the access to the Internet for getting information. Amount of data transmission in network and user request appears exploded increase, which raises a higher claim to server processing ability, and makes server sent answer-response in the shortest possible time on the basis that servers accept reasonably client request to improve user experience optimization. A huge amount of data and access request does cry for updating servers, which can response fleetly, are easy-to-use, and have high expansibility to expand network bandwidth in response to user request timely. The appearance of virtual machine live migration is the very effective approach to settle load imbalance. By means that the total virtual machine running state can mutual transfer smoothly between two physical hosts of random clusters, of course, which is processing and does not have any stagnant feelings for the users when it is necessary to migrate, virtual machine live migration can help cloud environment maintainer make full use of node server in the

cluster and dynamically achieve load balancing of cloud resources.

The traditional load balance algorithm is based on task control allocation, when applied to the cloud environment, it has some disadvantages: small task control allocation particles, node load conditions vary greatly, load information cannot be updated in a timely manner, load balance algorithm^[1] will be misguided. Task control allocation needs a central dispatcher, which is in full charge of dispatch and migration, the huge amount of dispatch and migration will make central dispatcher busy and cause disorders easily, which is also a major bottleneck of system performance. There will be more computational tasks in cloud computing environment, the complexity of allocation algorithm will face greater challenges and the availability of algorithm needs more attention.

2. Framework Design of Dynamic of Virtual Machine in Cloud Environment

2.1 Basic framework for dynamic migration of virtual machine

The basic migration structure is implemented by four modules including monitor migration, operation migration, freezing and target domain arousal with the four functions of the corresponding modules to achieve^[2]. As shown in figure 1.

Monitor migration module: The primary function of the primary module is to determine the source machine of the migration, the start time of the migration, and the target machine of the migration. The working mode of monitor migration module is determined by the purpose of migration. In order to ensure that the load of each node is balanced, the monitor signal is set in the virtual machine management program, the monitoring load operation of various nodes determines whether the monitor signal needs to migrate.

Operation migration module: The module is the most important module, and undertakes the most migration of the virtual machine and most work of the migration process. After the start of the migration, the module collects running information of the source machine, and at the same time sends "frozen" signal to freezing module to make the source machine downtime. Then the module continues to copy the remaining pages, after the end of the copy, it sends a wake-up call to wake module of the destination machine. this process is the key part of the whole migration process, and directly influences the downtime in the migration process and migration time.

Freezing module: The module is mainly responsible for how to solve the system to provide users with uninterrupted service^[3] to make users feel the service without interruption.

Target domain arousal module: The function of this module is to determine the time to wake up the destination machine, to make sure that the arousal target machine is in line with the source machine on service, and how to maintain the consistency of the target domain and the original domain on service. After downtime, the module is operated to copy the remaining memory page and sends a wake-up call to wake module of the destination machine after the end of the copy.

The direct consequence of downtime is to make the connecting device interrupt, peripheral device cannot connect to the virtual machine, which will certainly cause peripheral service is not timely or presents a variety of transmission errors.

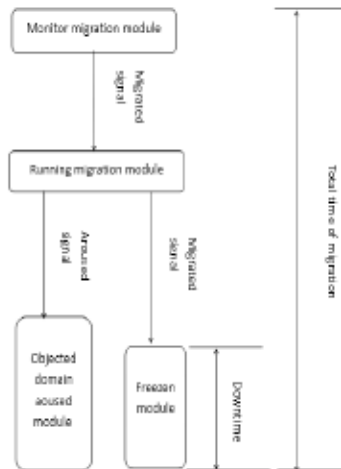


Figure.1 Basic dynamic migration module

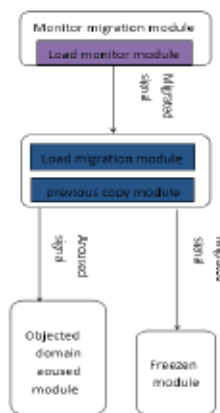


Figure.2 Dynamic migration framework optimization modules

2.2 Dynamic migration framework optimization of virtual machine

In order to increase the rate of load application and make migration process more smooth and effective, we make the frame design of the virtual machine dynamic migration optimized, and add 2 module about the implementation of the load balancing. One load monitor module adds to the original monitor migration module, which is devoted to marking the operation information of current virtual machine, setting the trigger condition for migration, and paving the way for a subsequent migration to select the appropriate load. The other is mainly in charge of location and selection strategy after the start of the virtual machine migration. As shown in figure 2, Grey patterns mark three modules.

3. Implementation of Resource Load Balancing Algorithm in Cloud Computing Environment

3.1 Overview of load balancing

Mutual use of servers is increasingly frequent, because there is a huge gap between the effective use of resources on the servers. Some servers are often in a competition for resources state, some for a long time are rarely even without the use of resources, which greatly reduces the resource utilization of corresponding range of system and leads to the severe decrease of performance of the overall cluster system. Load balance can be directly implemented by hardware level, for example increasing physical

device, and by software level, such as setting the relevant protocol and using specific software. Hardware implementation is to install a load balancer connected to the external network, through which the server for user access in the load application and user can access the resources. Hardware can make the processing capability of the cluster system stronger, at the same time, and promote the equalization performance, but it cannot effectively master real-time status of server, because it parses the data flow only from the network layer, which is not flexible enough.

An effective load balancing algorithm not only be able to assign the load evenly to every server, which can reduce the users' waiting time, but also need to migrate the load above the node over the load value to the node that does not cross the threshold or relatively easy to operate^[4].

Load balancing problem is a classic combinatorial optimization problem. That distribution and redistribution of tasks and resources on each node ultimately make each node load benefit roughly in balance and improve the overall system performance is load balancing. Load balancing algorithm, with respect to the load sharing algorithm, has a higher goal performing the more efficient allocation use of resources.

3.2 Classification of load balancing

Load balancing varies and should be classified on application scope. Classified on the task distribution and redistribution of the nodes to achieve the strategy of the algorithm, the load balancing technology can be classified as follows:

1) Implementation methods of hardware and software

Hardware load balancing primarily implements large system method in the high flow loss, and must increase the specific load balancer, of course, which will have better performance based on the increased cost. However software load balancing is applicable for some small and medium sized web sites or systems, software methods can be very convenient to be installed to the node server. The use of more commonly used URL redirection in computer networks or a technique based on the Internet such LVS to achieve a certain balanced load function can achieve the general equilibrium load demand.

2) Global and local methods

Classified according to the geographical distribution of the server, global load balancing technology is a certain degree of load balancing for multiple servers, which is distributed in the different regions. For access users, global load balancing technology automatically adjusts to the nearest point of the region by determining the location of the IP address. Local load balancing technology can control scheduling process some node of node server cluster in a regional scale at the time to a certain degree and make the node load relatively balanced. The technology can strengthen practical effect through node server designed and make network bandwidth be even distributed to every node server in server cluster.

3) Dynamic and static methods

The load balancing strategy in cloud computing is divided into two types, static state and dynamic state. The so-called dynamic decides from which overload node server chooses task and locates the target node to assign tasks according to the current state of the system. Once a node task of the system is overloaded, some tasks on this overloaded node will be migrated to other nodes to process, and to achieve the results of dynamic equilibrium. Certainly, task migration also brings additional consume

to the system. According to the simple system information, the mathematical function scheduling algorithm is used to select the source node and then locate the destination node and assign tasks and execute, which is the static performance. Static strategy implementation is relatively simple, but it is not fast enough and cannot dynamically adjust the information of each node as far as real-time response, and consequently some nodes utilization is very low. Most of the typical static load balancing strategies are based on the prediction model, for example algorithm based on inheritance, which can predict the trend of nodes according to the current information and historical information of nodes, and then give priority to high available future nodes to resource scheduling and task allocation.

3.3 Cloud resource load balancing strategy

There are a lot of physical servers in the cloud environment and these server specifications are not fully consistent. Through virtualization technology, a single physical node can be modeled as a number of computing machine entities, these virtual machines assign dynamically automatically to the user according to the user's requirements specification. But because of the user's requirement specification is not consistent, and the configuration of all the physical servers in the cluster is also not consistent. Traditional algorithm can balance load to a certain degree, but each algorithm itself have their obvious characteristics and deficiencies and exist such disadvantages, which makes the equalization effect disadvantaged and influences the service performance or causes other problems related ^[5]. At present, the main base of load balancing strategy has been divided into 4 types. NO.1 is ratio strategy, NO.2 is minimum number of connections strategy, NO.3 is round-robin strategy, NO.4 is fastest response time strategy.

Ratio strategy installs firstly the external request, and then pre-allocated to each load in a balanced state of the server. Cloud environment system has 4 servers, we can assume that the proportion of the probability of receiving a virtual machine migration request is 2:2:3:1, so each node server processing request is also different. The method is suitable to be responsible that the node server of load balance allocated according to the level of the hardware configuration, and set corresponding ratio depending on the corresponding processing efficiency, so that servers which have not the same properties can also be operated smoothly to prevent some servers overload and others in idle state.

Minimum number of connections strategy is that the hardware device responsible for the balanced effect monitors continuously and checks the number of connections on the relevant node server, selects a minimum number detected node as the purpose node of processing the request, which is suitable for application to long connections.

Round-robin strategy: The scheduler is applied to this strategy regardless of load status of destination node. As long as there is an external request, the request will be distributed to each node server, for example, there are 3 servers in the cluster system, and the request 3 servers will process are the same. Because of handling all transactions on average, this way is only suitable for the same hardware configuration nodes. This mode achieves relatively simple, the algorithm is also very simple to design, and the system overheads less, only to deal with the small business which have small differences and spend a short time.

Fastest response time strategy: The hardware devices send requests to be processed to each node continuously, which node is the fastest on response speed, the request is forwarded to the node server. The algorithm is suitable for stringent real-time response, but this method does not consider the load state of the destination node, which is easily prone to heavy load and causes stress to the

high configure servers.

3.4 Optimization algorithm of resource load balancing

Network nodes often contain memory resources, network bandwidth resources and other key resources. In order to describe accurately the use of various types of resources in the nodes, the load index vector is often used to describe. Each of these components is corresponding to a key resource in the node, which is used to represent the load usage. The algorithms mainly focus on three kinds of resources: CPU, memory and network bandwidth.

To ensure the efficient utilization of resources, we can comprehensively consider the cloud computing system, calculate the load of each server node, combine frame design with dynamic migration, make use of virtual machine migration technology, consider how to select the virtual machine from the overload node how to determine the migration destination and coordinate the load of different servers. This paper focuses on the load balancing part of the module diagram, figure 3 shown in figure.

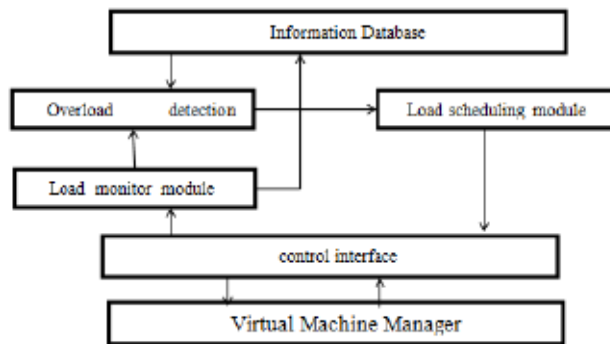


Figure.3 Load balancing algorithm module

Load monitor module: The resources between nodes are heterogeneous. In order to make the data value of the same kind of load can be compared, each component needs to be standardized and description undignified. The CPU load benefit is achieved by using the average utilization rate of all CPU on the nodes as well as the resource usage of the specification of the memory and network bandwidth. The most important of the three load benefits to do the following definition:

1) CPU Load Interest (CLI)

Calculate the average utilization rate of all CPU on the nodes, and reflect the CPU load of nodes by the average utilization rate. The physical CPU number of the node is m , the utilization rate of each CPU is c_i , the CPU load interest CLI of the node is expressed as:

$$CLI = \frac{\sum_{i=1}^m c_i}{m} \quad (1)$$

2) Memory Load Interest (MLI)

Load of a single virtual machine memory includes memory currently in use and memory for paging. The number of virtual machines on the node is m , $Vused_k$ represents the size of the virtual machine memory being used, $VChanged_k$ represents the size of memory of user page, $TotalV$ represents total memory of nodes and is also the sum of the former. Memory load of node:

$$MLI = \frac{\sum_{k=1}^m VUsed_k + VClanged_k}{TotalV} \quad (2)$$

3) Network bandwidth Load Interest (BLI)

The bandwidth load of a node is defined as the ratio of the sum of the bandwidth each virtual machine uses and the total bandwidth. The number of virtual machines on the node is m , $VnetBand_k$ represents bandwidth the virtual machine I is using, $TotalBand$ represents the maximum bandwidth of a node BLI:

$$BLI = \frac{\sum_{i=1}^m VnetBand_i}{TotalBand} \quad (3)$$

Load migration module: The migration process of virtual machine embraces the migration of the original host state and resources(internal memory, CPU and I/O device).Load migration module mainly adopts operation strategies to describe the current load conditions of each server node, and makes sure to record in real time and provides load index used by the server node resources. About the heterogeneous nodes that appear in the system, load operation strategy would also like to unify the description of its standardization and eliminate the heterogeneity, in order to facilitate the use of resources. Load operation strategies have selection rule, distribution rule and location rule^[6].

Usually, virtual machine nodes that need to be migrated often have more than one node in cloud computing systems. Equally, the destination machine of virtual machine migration has more than one server to meet the conditions of acceptance, If we take a physical node with the best performance in the current environment as the most suitable destination machine. All of the overload nodes choose the same destination node to migrate. This is bound to result in a sharp increase in the node load in a short period of time, severe cases will cause the destination node to collapse. The above phenomenon is called cluster effect^[7].

Firstly we identify a set of nodes that meet the low-load, select the destination node from the cluster when locating, and focus on computing node CPU computing and memory capacity of two performance indicators. Besides, if the node appears to have insufficient memory but the CPU computing capability remains or the two are in the opposite, the virtual machine on the physical node can't run properly, and also cause waste of resources. To avoid wasting as much as possible, balance the ratio of memory resources in physical node to CPU computing resources, and achieve optimal utilization, when selecting a destination node, we should mainly consider the matching degree between the virtual machine to be migrated and the destination node. That is the proportion of CPU consumption / memory consumption.

Table 1 Algorithm symbol and meaning

Symbol	Meaning
$(Cmigri) cost$	CPU usage rate of the node N_i in the virtual machine
$(Mmigri) cost$	The memory utilization of the node N_i in the virtual machine
$(C_i) cost$	CPU utilization ratio consumed by N_i
$(M_i) cost$	Memory utilization occupied by N_i
$(C_i) max$	CPU utilization rate migration trigger threshold of Node N_i
$(C_i) available$	CPU available rate of Node N_i
$(M_i) available$	Memory available of Node N_i
$(UCR_i)matched$	UCR matching threshold of node N_i
$N_{choosed}$	Destination node

$$(R_i)_{available} = Y C_i Y_{available} \times (M_i)_{available} \quad (4)$$

$$(UCR_i)_{available} = Y C_i Y_{available} \div (M_i)_{available} \quad (5)$$

Firstly, according to measurement index $UCR_{available}$ of the target node server, measurement index UCR_{cost} of the virtual machine and the performance of the target node, we can select k destination nodes meeting the requirements from the cluster center. Then the probability model of the $R_{available}$ value of the K labeled nodes is located. Suppose the current available resource capability for node I is $\&$, the probability of the node to accept the migrated virtual machine is P_i . Suppose the current available resource capability for node I is $(R_i)_{available}$. Then through the above description can be learned that the node allows to select the virtual machine to migrate to the probability of P_i as:

$$P_i = \frac{Y R_i Y_{available}}{\sum_{i=1}^k Y R_i Y_{available}} \quad (6)$$

Suppose the destination nodes' set is $\{N1, N2, N3, N4, N5\}$, its capable of utilizing resources $R_{available} = \{2.0, 2.0, 2.0, 3.0, 1.0\}$, and on the basis of the above formula to get the location probability $P = \{20\%, 20\%, 20\%, 30\%, 10\%\}$.

Lastly, when host select a destination node for virtual machine dynamic migration, using a RD random function to generate a arbitrary number in $[0, 1]$. Then according to the number of probability space is in which target node, and make sure the node that virtual machine migration finally choose. The probability that those physical hosts with a strong ability to use resources is being the target node of the virtual machine migrated is also large when there is a virtual machine triggered migration. The lower the utilization ratio of node resources, the greater the possibility of selection as a destination node and the smaller on the contrary. From these examples, we can see that the more idle state a node has and the greater the probability space occupying is, and the chance that generated random number is included in this probability range is larger. Thus the node is more able to be selected as a virtual machine migration destination node. In summary, to a certain extent, the probability mechanism prevents the occurrence of clustering effect, and the load balance of the server cluster in the cloud environment is better realized.

4. Experimental Results Analysis

4.1 Experimental environment and platform

We choose high modularized and rich-interface Eucalyptus suitable for C language as an experimental platform for the cloud environment. Eucalyptus platform is an open source project^[8]. It is a research result to study the global hot topic, cloud computing subject, and put into practice. It implements the IaaS service and enables users to allocate and manage physical resources through Xen or KVM virtualization technology. Eucalyptus interface can be connected to the SOAP and REST interface, if it is a cloud environment based on Eucalyptus platform, other visitors to the non cloud environment uses the SOAP interface or REST interface and can be connected to the common operation. Experiments using 4 PC machines to build a system that can be used as experiment, its topology structure as shown in Figure 4. Detailed configuration of each node as table 2 and table 3.

Table 2 Node hardware configuration

Type	Device	CPU	Memory	Hard disk
PC1	CC Cluster Controller	2.93GHz 32b dicaryon	1.85 G	320G
PC2	NC Node Controller	2.93GHz 32b dicaryon	1.85 G	320G
PC3	NC Node Controller	2.93GHz 32b dicaryon	1.85 G	320G
PC4	NC Node Controller	2.93GHz 32b dicaryon	1.85 G	320Ge

Table 3 Node software configuration

Type	OS	Platform and Module
PC1	Ubuntu10.10	MySQL 5.1+Eucalyptus 3.1+ Load balancing module
PC2	Ubuntu10.10	Eucalyptus 3.1+ Load monitoring module + Anomaly detection module +3th Xen3.4.2
PC3	Ubuntu10.10	Eucalyptus 3.1+ Load monitoring module + Anomaly detection module +4th Xen3.4.2
PC4	Ubuntu10.10	Eucalyptus 3.1+ Load monitoring module + Anomaly detection module +6th Xen3.4.2

4.2 Experimental results and analysis

Summarize the characteristics of the algorithm before the experiment. Firstly, the load balancing module is integrated into the dynamic migration framework, and the previous dynamic migration framework is only limited to the whole process of the migration, which is considered separately from resource scheduling. In this way, when dealing with the migration problem in the cloud environment it is easy to cause a phenomenon that after solving the problem of a unilateral and then producing a new problem; Secondly, new trigger rules for load prediction mechanism are used, which is different from the traditional trigger rules based on specific thresholds, and mainly solves that the transient load peak will trigger the frequent migration. Lastly, we should comprehensively consider the utilization of CPU and memory when selecting rules and locating rules. Especially when positioning rules, this algorithm chooses the probability mechanism used by migration destination nodes and prevents the occurrence of group effect. At the same time it realizes well the load balance of the server cluster in the cloud environment. In addition, the calculation of the location probability of each node does not impact relatively and mutually independent, Under these circumstances the balance of the cloud data center will be better.

The first set of simulation experiment designed carried out the test about migration trigger time. Select one of the nodes to monitor the CPU load utilization rate, and set the threshold of 0.7. Shown as figure 5, when t is equal to 8, once CPU load of nodes exceeds the threshold, the traditional trigger rules will immediately trigger a migration [9]. However, using predict trigger rules can partly predict that the node load is on the decline, which does not trigger the migration. In addition, when t is equal to 40, because the forecast data predicts the next trigger will have the trigger peak, so the first trigger did not immediately exercise. But when t is equal to 43, the trigger is migrated, and compared with the traditional algorithm based on threshold value, three migrations will be triggered during this period and cost vast virtual machine migration resources, which avoids the additional waste.

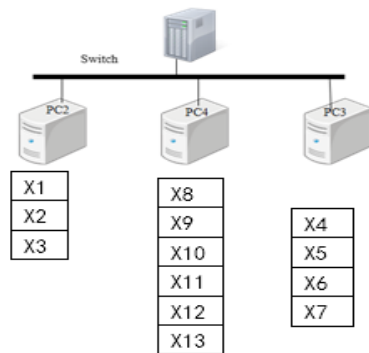


Figure.4 Experimental environment topology

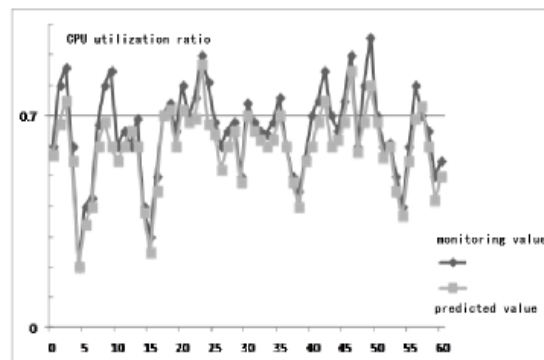


Figure.5 Compared with the traditional methods

In the emulation cloud environment of experiment 2, three machines have the same configuration on the nodes, and have initialized different timely load conditions. we set the use ratio of the system resource 70% and take it as overload limit of the load, at the same time, we stipulate that the request array have the same service type. Initialized load condition of each node need the artificial set, and are set the request array distributed by task scheduling. Each of the calculation of the system resource utilization and the rules of dynamic migration adopts the methods introduced by the above section. The system resource utilization shown as figure 6. We test on selecting a location algorithm of migration destination nodes and take no load balancing algorithm as the baseline, compared with the load balancing algorithm based on optimal adaptive rule. The load balancing of the system is evaluated by the standard deviation of the load benefit. Experimental results are shown in figure 7, figure 8 and figure 9.

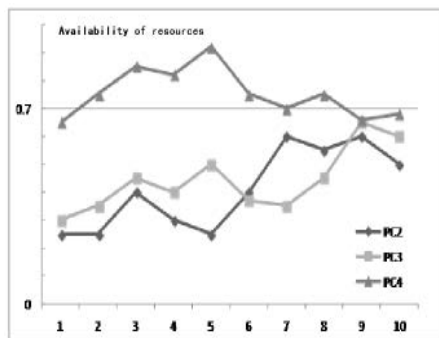


Figure.6 Node load variation

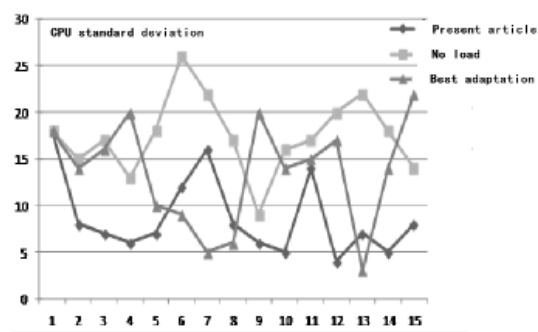


Figure.7 CPU load balancing

In the figure 6, from the initial PC4 start node has been in a super load state, start from the initial PC4 nodes has been in a over-load state, but there is no immediate migration equilibrium which is due to the role of the trigger rule. This experiment also verifies the effectiveness of the trigger rules based on the prediction from the side. When t is equal to 5, PC4 will trigger migration, and migrate the virtual machine to PC2, then the utilization rate of PC2 will obviously increase. Although PC4 has eased, due to the migration of virtual machines, and the utilization rate of PC2 and PC4 was higher than that of PC3. When t is equal to 8. The virtual machine on PC2 and PC4 migrates each load of them to PC3, At the end of this balanced period, the resource utilization rate of each node is similar and in the balanced state. Then approximate load balancing is achieved.

According to figure 7, we can see that CPU load standard deviation distribution is more balanced generally in this algorithm^[10]. Generally, the standard deviation is mainly distributed in the following 10. However the memory and network bandwidth have obvious advantages, the resource of each node in

the colony can be use reasonably. Shown as the follow figure.

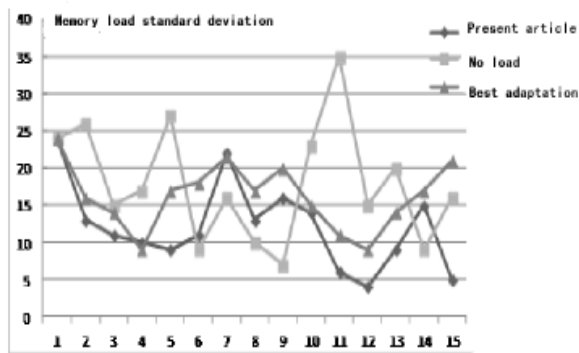


Figure.8 Memory load balancing

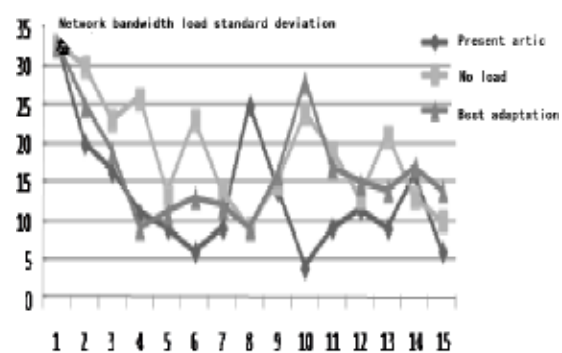


Figure.9 Network bandwidth load balance

We optimize and realize the design about load balance in cloud environment, and make relevant rules as far as platform completely modular, such as trigger rules, selection rules and location rules, and so on. The traditional methods based on the load balance system of cloud computing analyze anew the problem and improve it. The trigger rules is not simply based on threshold value, but on probability to predict. The selection and location rules both consider adequately CPU and memory usage, and analyze the best way of migration from overall consumption. The more available resource a node has, the easier the process that virtual machine is migrated to the most suitable destination node is. After the overall theoretical system is integrated, we conduct an experiment and analyze the result. Contrastive analyze CPU, memory, bandwidth, we can see the obvious performance advantages.

5. Summary and Outlook

The passage analyze concretely how load balance realize, design the concrete details of load balance module and has a further analysis that how to set up the frame of dynamic migration, and then set index of evaluating load balance. The characteristic of this aspect is the probability based, and solves the resource load in cloud environment, which is realized in the dynamic migration. The final result shows that algorithm fusion suggested has obvious performance advantages.

Now we just consider intensively the load balance of infrastructure later, and analyze concretely the location rules on load migration module. But about selection rules, we consider it from the ratio aspect, from consumption of overall migration, then from re-migrated aspect of available resource, and combine with the consumption about the increase of trigger rules to migration time. In addition, Information to initialize of algorithm is set by manual work. we collect load information with periodicity, and the periodicity is also set by manual work. We need to make sure the sampling period in the follow-up work.

References

- [1] Approach to cloud computing[M]. Beijing: Posts and telecom press, 2009:165.
- [2] Li yong: The study and analyze based on the dynamic migration technology of the virtual machine[Academic dissertation]. NUDT, 2007.
- [3] Li zhiwei, Wu qingbo, Tan yusong. The study of virtual machine dynamic migration based on the device agent machining. Application research of computers. The 26th volume, 2009.4.

- [4] Kaiqi Xiong, Harry Perros, Service Performance and Analysis in Cloud Computing, [C] In Proceeding of Congress on Services, July 2009: 693-700
- [5] Shi yangbin. A kind of load balancing algorithm based on the virtual machine live migration in cloud environment. [Master's thesis]. Shanghai: Fudan university, 2011.
- [6] Dina P, O Halloran D. The statistical properties of host load, In to appear in the Forth Workshop on Languages, Compilers, and Run-time Systems for Scalable Computers(LCR98) and CMU Tech[EB/OL].
- [7] Barnsley M. Fractals Everywhere[M]. NEW YORK: Academic Press, 1988:87.
- [8] Zhou wenyu, Chen huaping, Yang shoubao, Fangjun. The virtual machine cluster resource scheduling based on migration [J]. Journal of huazhong university of science and technology (JCR Science Edition), 2011, (39) SupplementI: 130-133.
- [9] Chang F, Dean J, Chomawath S. BigTable: A distributed storage system for structured data [J]. ACM Transactions on Computer Systems, 2008, 26(2): 1-26.
- [10] Chen guoliang, Sun guangzhong, Xu yun. Sabina chinensis integration research status and development trend of parallel computing [J]. Science, 2009, 54(8): 1043-1049.

Acknowledgements

This work was supported by the National natural Science Foundation of China (No.61472256, No. 61170277), Innovation Program of Shanghai Municipal Education Commission (No.12zz137), and the Huijiang Foundation (C14002).

Biographies

Sun Hong: female, Han, 1964-, from Beijing, China, Master, associate professor, School of Optical-Electrical and Computer Engineering University of Shanghai for Science and Technology, master tutor, associate professor direction of research; Business schools University of Shanghai for Science and Technology doctor graduate student; the main research direction: computer network communication and clouds computing, management science and engineering, Management Information and Decision Support System. Email: sunhong@usst.edu.cn, Telephone:13916902800

Wang Weifeng: male, Han, 1992-, master student, School of Optical-Electrical and Computer Engineering University of Shanghai for Science and Technology; the main research direction: cloud computing and management information system. Email: wwfhuo@163.com

Chen Shiping: male, Han, 1964-, from Zhejiang, China, professor, Ph.D. doctoral tutor Business schools University of Shanghai for Science and Technology, research direction: computer network communication and clouds computing, management science and engineering. Email: chensp@usst.edu.cn

Xu Liping: female, Han, 1986-, Master, associate professor, University of Shanghai for Science and Technology; the main research direction: cloud computing and management information system. Email: 5850487@qq.com

Research of Virtual Network Classroom Collaborative Mechanism Based on Petri Net

Shengquan Yang¹, Shujuan Huang²

School of Computer Science and Engineering

Xi'an Technological University, Xi'an, 710032 China

Email: ¹xaitysq@163.com; ²Shujuanhuang@163.com

Abstract. In order to keep multi-role communication action of virtual network classroom orderly and correctly at the same time, this paper proposes and studies its communication collaborative relationship based on Petri Net. Firstly, it introduces the basic theory and the system state change graph of the roles in virtual classroom, and discusses in detail the collaborative relationship between students and teacher in network environment. Especially by taking advantage of Petri Net tool, this paper in virtue of formalized method describes and analyzes collaborative relationship and collaborative mechanism between teacher and students, which exists in the virtual classroom. Finally, the collaborative mechanism has been realized successfully with the theory about process control concurrence view.

Keywords: Virtual Classroom, Collaborative Mechanism, Petri Net, Process Control

1. Introduction

The remote distance learning is a new generation educational pattern which is produced by the combination between computer network and the multimedia technologies today. It uses modern network and information technology to overcome geographic limitations of space, so that teachers, students can complete learning activities in different places. The modern distance learning is one kind of new education form which produces along with the present development of information technology, which is a principal means to construct people lifelong to study mode during the era of knowledge economy.

In distance learning interactive system, although teachers and students living in different places, but it feels like in a classroom, in which the teachers and students can see each other and be able to hear mutually. But because all activities are carry on under the network environment, the teacher is the teaching activity main body, he must have lots of qualifications such as that he can control the student to join, to make the student withdraw, to ask questions to students, and can cause the student to obtain the right to speak, and can cancel the student to speak jurisdictions and so on. The students may ask questions to the teacher at any time.

That is, in the entire teaching activities, each kind of activity which will occur will be concurrent, indefinite, and stochastic, therefore a collaborative mechanism must be studied successfully in order to suit the above characteristic, what's more to maintain the orderliness of the whole teaching and learning activities.

2. The Cooperative Relationship of Virtual Classroom

Virtual classroom is the local area network (LAN) or wide area network (WAN) space to create a virtual reality, interactive teaching and learning environment in order to achieve a variety of traditional classroom teaching function, which can provide a shared collaborative Classroom learning environment for a geographically dispersed network of online teachers and students to so that it can be a variety of real-time communication and collaboration ^[1].

In the virtual classroom, teacher and the students are acting with the traditional teaching in the same role, but in realizes specifically has the essential difference.

This kind of difference mainly displays in the virtual classroom teaching process, because in the long-distance teaching the teacher and the student usually are in the different place, the overwhelming majority students in the network region also possibly are scattered, which causes each kind of concurrent activity becomes very complex.

In order to accurately describe the synergy between teaching and learning activities, specific states which exist in teacher and students must be narrated clearly and concretely.

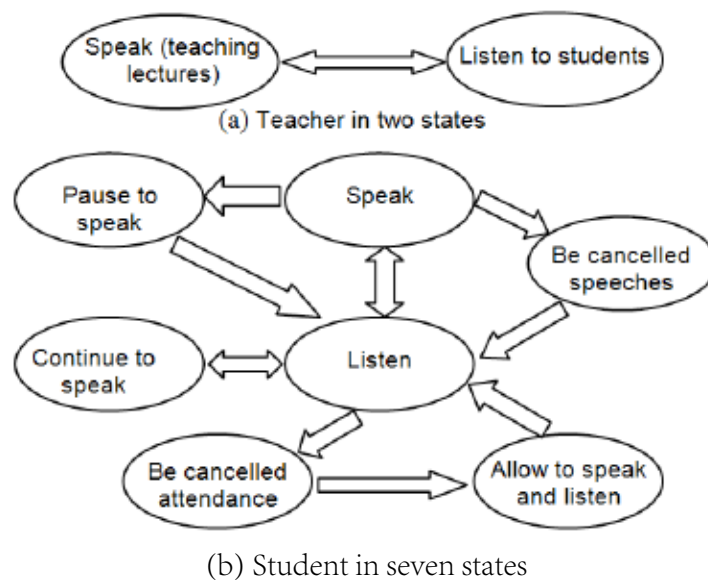


Figure.1 The states graph of teacher and student in virtual network classroom

In teaching activities, there are two kinds of the teacher's states (see Figure 1-a): speak (teaching lectures) and listen to students speak, however student's states (see Figure 1-b) are: listen, speak, pause to speak, continue to speak, be cancelled speeches, be cancelled attendance, allow to speak and listen, a total of seven states.

According to the changed states of teacher and student in figure 1, their collaborative relationships are described below :

In the entire teaching process, there is only a speaker allowed, which either is the teacher, either is the student; At any time each role is played in between speaking and listening state, and what's more their roles are transformed uncertainly in the two states. When the teacher is at the speech condition, the student listens, When the teacher asks questions or one student applies for speech and obtains the right

to speak, this student starts to speak, but the teacher listens at the time;

In order to control the entire ordering process of teaching students, the state change of students must be under the control of teachers, while teacher have rights to cancel the students to speak, to enable students to continue, abolish the “unpopular or unwelcome” students to speak, allowing students to listen and so on. In the network environment, all kinds of states in teaching will be a variety of unpredictable changes in concurrent operation, of which the appearance have many properties such as randomness, uncertainty and instability, etc. For example, students may ask questions at any time, a number of students to apply to speak, where there must be change between listen and speak, teachers and students how to coordinate and so on, it is necessary to control a variety of concurrent activities of a cooperative mechanism.

3. Petri Net Model of Cooperative Relations

Coordination mechanism among the virtual classroom is a typical computer supported collaborative work of the problem, which is abbreviated as CSCW. The so-called computer supported collaborative work that is more than one member of a group existed in some distributed network systems use multiple computers to work together to accomplish a task.

Because of this thinking is reflected in the information age groups, the way people work, interactive, distributed and collaborative nature of the objective requirements, it gives full play to the computer network as a potential communications media and superiority, which is being increasingly widely appreciated. That, computer supported cooperative work applied to the teaching field, is known as computer supported collaborative learning, abbreviated as CSCL.

Petri net is a useful tool of graphical representation, which has a combination of available models, and it has much unique strengths when needing to find on the description and analysis of the phenomenon. Petri net is also well and strict defined mathematical object; furthermore it may be appropriate not only to static structural analysis, but also to dynamic behavior analysis by way of the mathematical development of the Petri net analysis methods and techniques^{[3][4]}.

3.1 Establish the Petri net model

Definition 1: a triple-type $N = (S, T; F)$ can be called a net, if and only if

$$(1) S \cup T \neq \emptyset, S \cap T = \emptyset$$

$$(2) F \subseteq (S \times T) \cup (T \times S)$$

$$(3) \text{dom}(F) \cup \text{cod}(F) = S \cup T$$

$\forall x \in S \cup T$, Here: ${}^{\cdot}x = \{ y \mid (y \in S \cup T) \wedge ((y, x) \in F) \}$ and $x^{\cdot} = \{ y \mid (y \in S \cup T) \wedge ((x, y) \in F) \}$ are called the pre-set and rear-set.

Definition 2:

Quadruple $PN = (P, T, F, M)$ can be called Petri net, if and only if

1) $N = (P, T; F)$ is a net.

2) $M: S \rightarrow Z$ (set of non-negative integers) for the identity function, where M_0 is the initial marking

(that is initial state)

3) Firing rules: when transition (migration change) $t \in T$ can be called enabled under state M , if and only if $\forall s \in \cdot t : M(p) \geq 1$; From M the transition that t is enabled can lead to state changes, which is obtained subsequent identification M' after triggering.

Petri net is consists of four different elements, as follows: Place (P), with “O” expression, Transition (T), with “—” expression, the arc of direction that connect Place and Transition, and the token in the Place (Token, with a “•” expression).

Place is used to describe the logic state of the system, and transition is used for the action and production process of all events.

The input function (I) and the output function (O) expresses are used for contiguous function relations separately between the place and the transition,

If a Place is given a mark k (k is a non-negative integer), then the Place has k tokens, also known as the Place has been marked,

Thus a marked Petri net in the definition 2 can be decomposed into a quintuple $PN = (P, T, I, O, M)$, M is the Petri net state identification sets.

The collaborative mechanism among the virtual network classroom can be described into a Petri net, as shown in Figure 2:

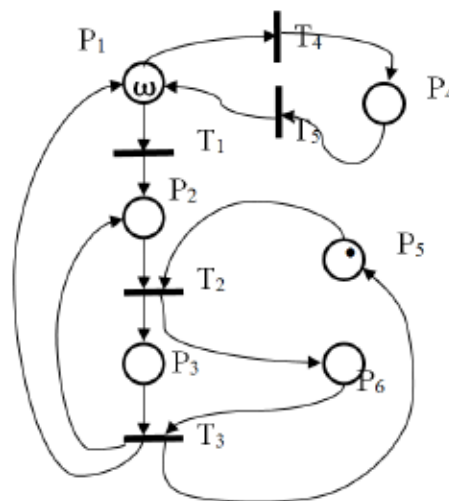


Figure.2 Petri net model of the collaborative mechanism among the virtual network classroom

The concrete description meaning see Table 1, including:

Place $P = \{ P_1, P_2, P_3, P_4, P_5, P_6 \}$,

Transition $T = \{ T_1, T_2, T_3, T_4, T_5 \}$,

Token $\omega = N+1$ in P_1 indicates N student and 1 teacher,

Token \bullet in P_5 indicates that one speak right resource is available.

Table 1 Petri net model concrete description meaning

Place	State or behavior	Transition	command or event
P1	the teacher or the student are at listen state	T1	Send command to apply for speak
P2	Execute the task of applying for speak right resources	T2	Send command to apply for speak
P3	With obtained the speak right, can execute speak task	T3	Send command to release the speak right resources
P4	Expressed that the student is at forbids to listen the teaching	T4	Send command to cancel the students listening
P5	Indicated that the speak right resources is at the idling state	T5	Send command to allow the students listening
P6	Indicated that the speak right resources is at the using state		

3.2 Analyze the Petri net model of the collaborative mechanism

The description of figure 2 is happen to the initial state of the system, which represents the teacher and student’ s token are in the P1 place.

Teacher work flow showing is as follows:

1) When teacher is teaching, the teacher needs to apply for the speak right to the system, then the T1 transition will work so that teacher token flows from the P1 place to the P2 place, which indicates that the teacher wants to get the speak right from the system, and by now if a student X is in the P3 place, then will enter (2), otherwise will change over to (3).

2) The teacher has the highest priority of the system, this now if a student X is in the P3 place, which namely indicates that the student X is speaking, then the system sends out an event of letting this student X release speak right unconditionally, so that the T3 transition occurs, the student X flows from the P3 place to the P1 place, simultaneously the right to speak resources flow from the P6 place to the P5 place, which namely indicates that the right to speak will change from busy state to idle condition.

3) Because the right to speak resources are at the idling condition, the system allows the teacher to obtain the speech firstly, then the T2 transition occurs, teacher token flows from the P2 place to P3 the place, which indicates that the teacher is at teaching or the commentary condition; Simultaneously the right to speak resources flow from the P5 place to the P6 place, which indicates that the right resources to speak is not available.

4) The teacher speaks a subject or a section of curricula, then puts forward a question or permits some student inquiry, or lets the student in waiting queue who was spoke ago, the teacher sends out the release speak right event, the T3 transition occurs, the teacher flows from the P3 place to the P1 place, simultaneously the speak right resources flows from the P6 place to the P5 place, which namely indicates that the right to speak will change from busy state to idle condition.

The system enters the next round of speak right resources competition recurrent state.

Student work flow showing is as follows:

1) When some student ask questions to teachers, the student must apply for the speak right to the system, T1 transition occurs, the student token X flow from P1 place to P2 place, which indicates that the student X is at applying for the right to speak condition.

2) At this time, if the teacher is lecturing or other student in his speech, this student X will be at the waiting status in the P2 place, until the right to speak is changed from the busy to idle (Idle) state, then transfers to (3) to execute.

3) Because the right to speak resources is at the release condition, according to the priority of students waiting to speak, the system adopts a first-input-first-out (FIFO) principle to serve for them, when student X 's priority is highest, T2 transition occurs, students X Token flows from the P2 place to the P3 place, indicating that X is at the speech or inquires some question to teacher; Simultaneously the right to speak resources flows from the P5 place to the P6 place, which indicates that the right to speak resources is occupied.

4) If this student X inquires to teacher, it needs teacher the gap-like to answer the question, then after the student X speaks, the system sends out the suspension speech event, the T3 transition occurs, the student X flows from the P3 place to the P2 place, and the system inserts the student to the first position of the waiting queue that applies for the speak right resources, which indicates that it has the highest priority in the queue of the waiting speech students, Simultaneously the right to speak resources flows from the P6 place to the P5 place, which namely indicates that the right to speak changes form by busy into the idle condition. By now it needs the teacher back and forth to answer questions for the topic, because teacher 's priority is highest, he does not use lining up, once the teacher requests to speak, the system enters teacher 's work flow immediately.

5) In the midway, if teacher wants to cancel the current student X speech, then the teacher sends out cancels the current student to speak the event, which forces this student X unconditional release right to speak, the T3 transition occurs, so that the student X must flow from the P3 place to the P1 place, and the student transforms to listening state, simultaneously the right to speak resources flow from the P6 place to the P5 place, which namely indicates that the right to speak changes from busy into idle condition.

The system enters the next round of speak right resources competition recurrent state.

6) During the students listening process, when there is "not welcome" student existence discovered by the teacher, the teacher will send out cancel listening event for the "undesirable" student, the T4 transition occurs, the student X flows from the P1 place to the P4 place, the student transforms into the condition of forbidding to listen.

7) When the student X sends out to the teacher an information that he is willing to observe the classroom discipline, and the teacher permits it 's again adding to listening, the teacher may sends out the event that joins the student X into listening, the T5 transition occurs, the student X from the P4 place flows to the P1 place, the student X transforms into the condition of permission listening.

3.3 model analysis of concurrent and conflict

According to the Petri net model knowledge which is described for collaborative mechanism of the virtual classroom, the student and the teacher are just like many advancements stochastically in the system interior, the concurrent movement ^[2].

But when there are a lot of students common in the application for speak right state, for example, to

identify the $M0 = (m, u, 0, 0, 1, 0)$, and T2 transition is enabled, the students, of which number is v , in the P2 place can flow into the P2 place, because:

$$M0 [T2] M1 = (m, u-v, v, 0, 1-v, 0)$$

And, $M_i[T]M_j$ expresses that the transition T stimulation (also calls ignition), causes the Petri net by to mark M_i to enter marks M_j .

But there is only one resources in the P5 place, it must guarantee that $1-v \geq 0$ is correct only, and then makes sense.

Therefore, $V \leq 1$, this means that any time a process can only be allowed to get the speech right. Therefore, we must resolve their conflicts arising from the common run-time, critical resources, and the conflict is the essence of competition^[5].

3.4 solution of resources competition conflict

The process of multiple concurrent accesses to critical resources must be controlled to make the system to normal operation.

With the aid of the process dispatching management game theory method in the operating system may solve this problem^[6].

The introduction of semaphore S (initial value is 1) plus Priority Power and events Event strategy. In S can be P operation and V operation.

Power and the Event is defined as: Power = 1 show that the teachers priority, Power = 0 shows that students priority, Event = 1 cancel the speech, Event = 2 suspend the speech, Event = 0 there is no immediate action;

P(S, Power, Event) definition is as follows:

1) When Power=1 and $S < 0$, if Event=1, system will eliminate the speech advancement resources of the current process, which causes it transform to listen state;

If Event=2, system will eliminate the speech advancement resources of the current process, which is inserted into the head of the blocking queue at once; If Power=0 or $S > 0$, system immediately enters (2);

2) $S = S - 1$;

3) If $S \geq 0$, then this process continues to carry on;

4) If $S < 0$, then the process is blocked, and it is inserted into the blocking queue of semaphore S, then the system re-schedule another process to put into operation.

V(S) definition is as follows:

1) $S = S + 1$;

2) If $S \geq 0$, then this process continues to carry on;

3) If $S \leq 0$, which shows that there are some process that are blocked, the system must wake up the first blocking process in queue of semaphore S, so that it can be entered the ready queue and continue the operation of the process.

4. Implementation of collaborative mechanism

IPremise: Makes the teacher process is ProcessT, and it has the highest priority; Makes the student i process is ProcessSi, and all students have the same level priority;

Strategy:

1) when process ProcessT is running and applying for the speak right resources, the system adopts “deprivation of way” , namely according to the event type which is sent out by ProcessT, if it cancels the current process ProcessSi to speak, the system will deprive ProcessSi of the right to speak resources, transforms ProcessSi to the listen condition; If suspends current ProcessSi speaking, the system will eliminate ProcessSi the right to speak resources, and insert process ProcessSi into the first place of the student waiting blocking speech queue.

2) When a ProcessSi is running and applying for the speak right resources, the system will take “non-deprivation mode” , which enables the release of the current process that is using speak right resources, the system according to the principle of first come first served (FIFO), will wake up the first place of the student waiting blocking speech queue, and make it obtain access to critical resources.

Finally an explanation point: When students are in the waiting blocking process speech queue, the state of the ProcessSi is listening, that indicates waiting speech student retains the right to listen, which is consistent with the actual classroom.

Implementation parts of the codes are as follows:

Teacher process control routine similar as follows:

ProcessT_Work

Begin

.....

P(S,1,1 or 2); //apply for speak right resources
// or cancel / suspend the current students to speak
T Process speak; //teaching

.....

T Process ask question; //for students
V(S); //release speak right resources

.....

End;

Every student process control routine similar as follows:

ProcessSi_Work(i)

Begin

.....

Si Listen to teacher’ s lecture; //in listening state

.....

P(S,0,0); // apply for speak right resources
Si Process speak; //for student
V(S); //release speak right resources

.....

End;

5. Conclusion

The Research of Virtual Classroom's Collaborative Mechanism discussed in this paper have decomposed the complex question into simple forms very much, which has highly versatile and scientific rigor, and it provides a better model and a good method for other similar research, so it has high practical and referenced value .

The author has utilized this model method successfully, which is designed for the actual development work of the Chinese some university distance learning system, and it has made the very good movement progress.

Acknowledgment

The Research is supported by the State and Provincial Joint Engineering Laboratory of Advanced Network, Monitoring and Control. (Financing projects No. GSYSJ2016014).

References

- [1] Yu Huang, Wenhui Hu, Xin Gao, Hart-pin Wang, "WSCI Formal Model Analysis Based on Petri Nets" , Computer Engineering & Science, vol. 31, pp. 60-63, October 2009
- [2] Yebai Li, Fuqi Mao, "Research of the Verification in Workflow Process Modeling on the Application of Petri Nets" , 2010. IC4E,10. International Conference on , Sanya, pp. 21 - 24, January 2010
- [3] Zouaghi, L.; Wagner, A.; Badreddin, E., "Hybrid, recursive, nested monitoring of control systems using Petri nets and particle filters" , Dependable Systems and Networks Workshops (DSN-W), 2010 International Conference on, Chicago, pp. 73 - 79, August 2010
- [4] Arpaia, P.; Fiscarelli, L.; La Commara, G.; Romano, F., "A Petri Net-Based Software Synchronizer for Automatic Measurement Systems" , Instrumentation and Measurement, IEEE Transactions on, vol. 60, pp. 319 - 328, January 2011
- [5] Mahgoub H Hammad, Alsadig Mohammed, Moawia E. Eldow . , "Design an electronic system use the audio fingerprint to access virtual classroom using Artificial Neural Networks" , Computer, Communications, and Control Technology (I4CT), 2015 International Conference on, Kuching, Malaysia, pp. 578-585, April 2015
- [6] Ziyue Ma , Yin Tong , Zhiwu Li , Alessandro Giua. "Basis Marking Representation of Petri Net Reachability Spaces and Its Application to the Reachability Problem" , IEEE Transactions on Automatic Control, vol. 3, pp. 1078 - 1093, May 2016

A DCT Domain Image Watermarking Method Based on Matlab

Wu He-Jing

Department of Computer Science & Electrical Engineering,

East University of Heilongjiang, Harbin, China

Email: 499917928@qq.com

Abstract. In the text, A method of image watermarking based on DCT (Discrete Cosine Transform) domain algorithm is proposed and verified in experiment by Matlab. The experimental result shows that the current method can achieve embedding with sound robustness and invisibility. From the experimental results, the quality of the watermarked image is almost no decline relative to the original.

Keywords: DCT, watermarking, matlab, embedding

1. Introduction

Along with the development of technologies about computer, network, and multimedia, the transmission of multimedia information such as music, image and video becomes more and more convenient, whereas the problem of copyright infringement has brought new opportunity for the effective protection of intellectual property. People invented a technique to hide the company logo, specific digital identifier and other information into the multimedia files for the sake of identification of ownership. Such a technique is called Digital Watermarking, which is a branch in the information hiding technology. The basic requirements are: 1) transparency, referring to that a certain amount of digital watermarking information is embedded in a digital media host, with the hidden data being imperceptible and without causing degradation to the original media; 2) robustness, referring to that the digital watermarking must be immune to the transformation applied on host media such as lossy compression, filtering and cropping, that is, the watermark information should not be lost due to some transformation applied to the host media; 3) safety, referring to that the digital watermark can resist all kinds of deliberate attack, and it is difficult to be copied or forged by others, as long as they do not know the secret key control algorithm.

This paper focuses on a theme on DCT-based image digital watermark design and implementation. Improve a digital image watermarking algorithm which is based on DCT transform and Arnold scrambling. It is ensure the security of the watermarking by Arnold scrambling the original watermark information. The experimental results show that it is hiding the information of gray image, and make an experimental simulation on some common image attacks.

2. Dct Transformation

Discrete Cosine Transform is based on orthogonal transform, which is one of the most commonly used linear transform in digital signal processing. It reflects the correlation properties of image signal. DCT algorithm is of moderate complexity, with medium energy consumption and has the good ability to energy compression. Thus, it is widely used in digital signal compression such as image

compression and other fields. JPEG compression is a standard established on the basis of the DCT transform. Watermarking algorithm of JPEG compression standard has greatly enhanced the ability to resist JPEG compression based on watermark. So the DCT transform in watermark technology is very important. DCT transform decompose the image into a spectrum of different spatial frequencies $F(u, v)$, with (u, v) known as the frequency domain coordinates. The inverse transform puts different spatial frequency components into the original image synthesis. In digital image processing often a two dimensional DCT transform is used. For a picture with the dimension $M \times N$, the DCT transform is:

$$F(u, v) = c(u)c(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cos \frac{\pi(2x+1)u}{2M} \cos \frac{\pi(2y+1)v}{2N} \quad (1)$$

$$c(u) = \begin{cases} \sqrt{1/M} & u = 0 \\ \sqrt{2/M} & u = 1, 2, \dots, M-1 \end{cases} \quad (2)$$

$$c(v) = \begin{cases} \sqrt{1/N} & v = 0 \\ \sqrt{2/N} & v = 1, 2, \dots, N-1 \end{cases} \quad (3)$$

The two dimensional inverse DCT transform formula is:

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} c(u)c(v) F(u, v) \cos \frac{\pi(2x+1)u}{2M} \cos \frac{\pi(2y+1)v}{2N} \quad (4)$$

Among them, x,y are spatial sampling values and u,v are the frequency domain sampling values. Because digital image is always measured by pixels, that is, $M=N$, in such cases, the two-dimensional DCT transform and inverse transform can be simplified as:

$$F(u, v) = c(u)c(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \frac{\pi(2x+1)u}{2N} \cos \frac{\pi(2y+1)v}{2N} \quad (5)$$

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} c(u)c(v) F(u, v) \cos \frac{\pi(2x+1)u}{2N} \cos \frac{\pi(2y+1)v}{2N}$$

3. Discrete Cosine Transform Watermarking Embedding Algorithm

Digital image watermarking algorithm that select the value of the two gray image as watermark information, choose the best one from different embedding coefficient. The vector image is 8*8 block, and the gray level digital watermark value directly embedded into the DCT transform domain vector in gray image, which realize the embedded watermarking. Specific methods are as follows: let I be the original image with the size of $M \times N$, W is the watermark image with the size of $P \times Q$, M and N are even times of P and Q, and the watermark W is loaded into the image I. The algorithm is divided into the following steps:

(1) I is divided into the block B with the size of $(M/8)*(N/8)$, and make a DCT transform;

(2) By Arnold transform, B is divided into the block V with the size of $(8P/M)*(8Q/N)$, and make a DCT transform;

(3) Watermark is embedded according to the multiplicative watermarking algorithm, which put the block into the carrier image in I, and the IDCT transform is performing and get a new image with watermark. The program code is as follows:

```

M=960;
N=120;
K=8;
I=zeros(M,M);
J=zeros(N,N);
BLOCK=zeros(K,K);
subplot(3,2,1);
I=imread( 'lena.jpg' );
I=rgb2gray(I);
I=imresize(I,[960,960], 'bicubic' );
imwrite(I, 'lena1.jpg', 'jpg' );
imshow(I);
title( 'original image' );
subplot(3,2,2);
J=imread( 'heida.jpg' );
J=im2bw(J,0.4);
J=imresize(J,[120,120], 'bicubic' );
imwrite(J, 'heida2.jpg', 'jpg' );
imshow(J);
title( 'the original image' );
for p=1:N
    for q=1:N
        x=(p-1)*K+1;
        y=(q-1)*K+1;

```

```

        BLOCK=I(x:x+K-1,y:y+K-1);    BLOCK=dct2(BLOCK);
    if J(p,q)==0
        a=-1;
    else
        a=1;
    end
    BLOCK=BLOCK*(1+a*0.03);
    BLOCK=idct2(BLOCK);
    I(x:x+K-1,y:y+K-1)=BLOCK;
end
end
subplot(2,2,1);
imshow(I);
title( ' watermark image' );
imwrite(I, ' watermarked.jpg' , ' jpg' );

```

4. Discrete Cosine Transform Watermark Extraction Algorithm

Set the image I as the carrier image with embedded watermark. Firstly, it need extract the watermark image from I, and the extraction process is the inverse of the embedded watermarking algorithm.

- (1) The image I is decomposed into the size of $(M/8)*(N/8)$;
- (2) Make a DCT transform for each block;
- (3) Make each block watermark extraction process according to the multiplicative inverse algorithm ;
- (4) Make the IDCT transform, and a watermark image is synthesized.

```

clear all
M=960;
N=120;
K=8;
A=imread( ' lena1.jpg' );
%I=imread( ' watermarked.jpg' );
%I=rgb2gray(I);

```

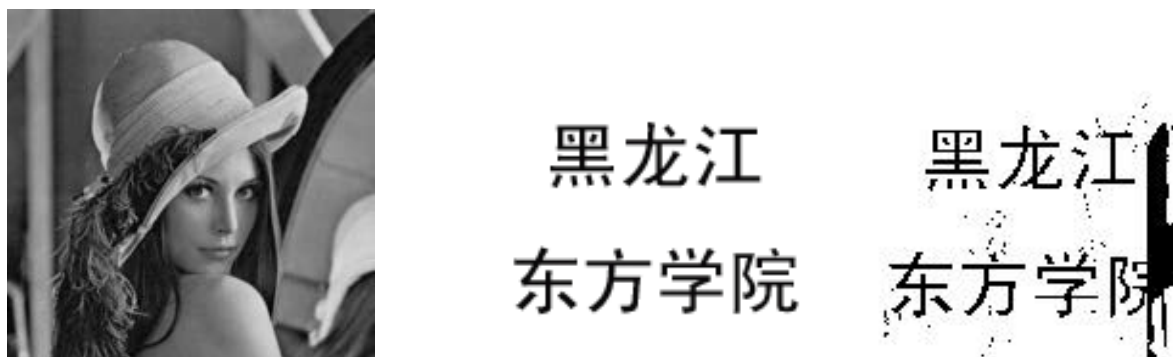
```

A=imresize(A,[960,960], 'bicubic' );
b=imread( 'watermarkedR.jpg' );
for p=1:N
    for q=1:N
        x=(p-1)*K+1;
        y=(q-1)*K+1;
        BLOCK1=A(x:x+K-1,y:y+K-1);
        BLOCK2=b(x:x+K-1,y:y+K-1);
        BLOCK1=idct2(BLOCK1);
        BLOCK2=idct2(BLOCK2);
        if BLOCK1(1,1)==0
            if BLOCK1(1,1)~=0
                a=(BLOCK2(1,1)/BLOCK1(1,1))-1;
                if a<0
                    W(p,q)=0;
                else
                    W(p,q)=1;
                end
            end
        end
    end
end
subplot(2,2,1);
imshow(W);
title( 'the extracted watermark image' );
imwrite(W, 'w1.jpg', 'jpg' );

```

5. Experimental Results And Analysis

In this paper, the experimental results are obtained based on MATLAB7.0 simulation. The original image uses the gray image of Lena, the watermark image is the two value image containing the words East University of Heilongjiang.



(a) The original carrier image (b) Two value watermark image (c) The watermarking image after

Figure.1 The watermark embedding test results

We can see from the experimental results, the quality of the watermarked image is almost no decline relative to the original, almost invisible.

Respectively, shear attacks、 noise attacks、 compression attacks are put on the watermarked image, and carries the watermark detection, get the following figure.



(a) gray image after shear 10% (b) 10% shear watermarked image

Figure.2 10% experimental results of shear



(a) Gray image after shear 15% (b) 15% shear watermarked image

Figure.3 15% experimental results of shear



(a) Gray image after shear 30%



(b) 30% shear watermarked image

Figure.4 30% experimental results of shear

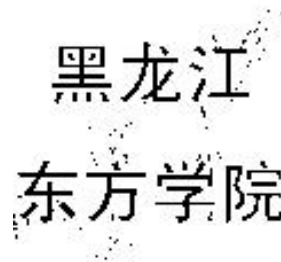
Table 1 normalized different shear ratio similarity coefficient

Shear ratio	5%	10%	15%	20%	30%	50%
NC	1	0.9980	0.9920	0.9880	0.9870	0.9827

From Table 1, we can see that according to the continuous change of different shear ratio, the extracted watermark image whose normalized similarity coefficient (NC) also showed different changes. According to the data in the above table and the extracted watermark image with different shear ratio, we can see that shearing off a portion of the image, but the relevant information from the original color image watermarking is still extracted.



(a)Gauss joined the variance of noise image 0.02

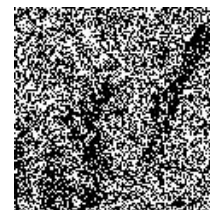


(b)From the variance of 0.02 images extracted watermark

Figure.5 Adding Gauss noise variance for experimental results of 0.02



(a)The compression factor of 70 gray image



(b)From a factor of 70 images extracted watermark

Figure.6 Experimental results for gray image compression factor of 70

Table 2 Different quality factor compression experimen

Quality factor(%)	90	80	70	60	50	40	30
NC	1	1	0.9930	0.9901	0.9850	0.9432	0.9120

We conducted various experiments to attack the algorithm, it is concluded that the algorithm can be well applied to copyright protection and authentication. The experiment results prove that the digital watermarking algorithm we used is a good robustness、transparency、low complexity algorithm.

In a word, research on the digital watermarking technique is developed in recent years and more active. The research in this area has achieved some results, but the workload is not enough. More research is needed to obtain greater progress in this field.

References

- [1] Hernandez J R, Amado M, Perez G F. DCT domain watermarking techniques for still images: Detector performance analysis and a new structure. *IEEE Trans on Image Processing*, 2000, 9(1):55-68.
- [2] A. Z. Tirkel, G. A. Rankin and R. van Schyndel. Electronic Watermark [C]. In: *Digital Image Computing, Technology and Applications-DICTA93*, Macquarie University, 1993:666-673.
- [3] R. G. Van Schyndel, A. Z. Tirkel and C. F. Osborne. A Digital Watermark [C]. *Proceedings of IEEE International Conference on Image Processing*, 1994, 2:86-90.
- [4] Sun X M, Luo G, Huang HJ. Component-based digital watermarking of Chinese texts [C]. *Proceedings of the third International Conference on Information Security*. Shanghai, China, 2004, 85:76-81.
- [5] P. S. Huang, C. S. Chiang, C. P. Chang. Robust Spatial Watermarking Technique for Color Images via Direct Saturation Adjustment [J]. *Proceedings of IEEE: Vision, Image and Signal Processing*, 2005, 152(5):561-571.
- [6] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn. Information Hiding - A Survey [J]. *Proceedings of IEEE*, 1999, 87(7):1062-1078.
- [7] G. Voyatzis and I. Pitas. The Use of Watermarks in the Protection of Digital Multimedia Products [J]. *Proceedings of the IEEE*, 1999, 87(7):1197-1207.
- [8] Nikolaidis N, Pitas I. Copyright Protection of Images Using Robust Digital Signatures [C]. *IEEE International Conference on Acoustics, Speech, and Signal Processing*. USA, 1996:2168-2171.
- [9] J. M. Acken. How Watermarking Adds Value to Digital Content [J]. *Communications of the ACM*, 1998, 41(7):74-77.

A Mobile Terminal Security Strategy Based On the Cloud Storage

Wang Hui, Tang Junyong

School of Computer Science and Engineering

Xi'an Technological University, Xi'an 710032, China

Email: 277019826@qq.com

Abstract. With the emergence of mass storage systems and the development of the Distributed File System, Cloud storage system has become the focus of the industry. The cloud storage services on mobile terminal have been putted on the agenda based on the rapid development of intelligent mobile terminal. Based on the analysis of the architecture of HDFS and Dynamo, a mobile Terminal Security strategy is presented in this paper. The database technology and the dynamic consistent hashing algorithm are adopted to deal with different target groups. According to the storage costs of nodes, data would be integrated scheduling by the storage system. Make full use of the advantages of AES(Advanced Encryption Standard) and RSA. A solution that combines AES and RSA encryption algorithm is proposed to implement the mobile terminal cloud storage security. Through the theoretical analysis and the simulation results, the cloud storage strategy proposed in this paper can make the cloud system achieve load balance. Moreover, multi-copy mechanism can improve the overall efficiency of the system.

Keywords: Cloud Storage, HDFS, Dynamo, Dynamic Consistent Hashing Algorithm, AES, RSA, Multi - Copy Mechanism

1. Introduction

With the rapid development of the Internet of things, more and more people are used to using mobile devices such as smart phones to surf the Internet, chat, browse news, shopping entertainment, and view all kinds of information. Traditional mobile cloud storage systems have lower storage density, the overall storage efficiency is low too. Traditional cloud storage systems do not adapt well to different application environment sand do not guarantee the integrity and confidentiality of cloud data. The cloud storage service does not guarantee that the data and operation of mobile users will not be lost, damaged, leaked, or illegally exploited by malicious or non-malicious. So it's very dangerous for sensitive data to be stored directly in the cloud. Simple encryption techniques have key management issues and can't support complex requirements such as query, parallel modification, and fine-grained authorization. As a result, a mobile cloud storage security technology solution is proposed in this paper, which enables reliable and secure cloud storage.

First, the distributed file system (the hadoop distributed file system, HDFS) and Dynamo would be compared in this paper, and then the dynamic consistency hash algorithm is introduced to realize the processing of data in different size. According to the storage cost of each storage node, select the optimal storage node to implement the access of mobile cloud storage. The relational database is used for storing indexes in small object files, and the class Dynamo system model is used to handle large object files.

The cloud storage system will choose the closest copy when the mobile terminal makes a request. This method can effectively improve the storage efficiency of the cloud system and ensure the load balance of the system. On the basis of implementing the mobile cloud storage, we make full use of advantages of AES and RSA algorithms, a cloud storage security scheme for mobile is proposed in this paper. The solution combines AES and RSA encryption algorithms to improve the shortcomings of the cloud storage system. The reliability model of the cloud storage system is also proposed in the paper. Finally, a series of simulation experiments show that the proposed cloud storage security technology scheme is a reliable scheme with higher security.

2. System Architecture of the Mobile Cloud Storage

Cloud Storage is developed on the basis of clustering techniques and embedded virtualized technologies, which is an extension of cloud computing. Grid technology, cluster technology and distributed system are used in cloud storage, which coordinated all different types of storage devices in the network. All these technologies and devices can be cooperated with cloud storage to provide the required data storage capabilities and related business visit. Cloud storage is not a single storage device. The nature of cloud storage is not storage. The essence of cloud storage is providing services. Different ways are taken to deal with different sizes of objects. In this way, system architecture of the mobile cloud storage is designed . System architecture of the mobile cloud storage is shown in figure 1.

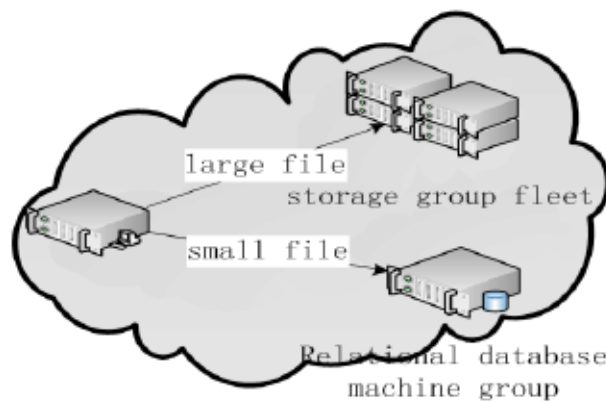


Figure.1 System Architecture of the Mobile Cloud Storage

Users access the network through a mobile terminal. The lowest load in the dispatcher group is selected by the mobile terminal and then communicates with it. Depending on the size of the file to determine whether control is handed over to the relational database machine group or to the storage group. Small files are handed over to the relational database machine group, and large files are processed by the storage group machine group. The copy mechanism was introduced to improve the reliability of the cloud storage system, multiple-copy mechanism can effectively improve system efficiency. When a mobile terminal makes a request, the closest copy can be chosen.

The primary copy would be selected first in traditional cloud storage system. The request is made to the standby copy only when the master copy is wrong. This process affects the speed of the traditional cloud storage system without considering the location of the copy.

Different storage policies and backup solutions are described in this article, which are used in the relational database machine group and the storage group machine group.

3. Storage Policies and Backup Plans

HDFS and Dynamo are reliable solutions that are commonly used in the cloud storage system. HDFS is a distributed file system that is suitable for running on common hardware. HDFS has good fault tolerance and can be used for inexpensive hardware. HDFS provides data access mechanisms with high throughput that can be widely applied to large data sets. Distributed file systems are developed on the infrastructure of the Apache Nutch search engine and apply to batch processing for data storage. HDFS emphasizes data throughput rather than response time for accessing data. The program in HDFS has a lot of data sets. File size of the HDFS is typically gigabyte to terabyte. As a result, terabytes of large files can be supported in HDFS through higher aggregated data bandwidth. And hundreds of nodal devices can be contained in a cluster, which allowing the terabytes of large files to be supported in it.

Dynamo is storage platform of amazon, and the key-value pair is used to store data in the key-value database schema. Dynamo has better availability and the higher extensibility. In Dynamo, the data are segmented according to the hash algorithm used in distributed file systems. And then all these segmented data are stored in separate nodes. The corresponding node is searched according to the hash value of the key, so that the read operation is realized in Dynamo. The consistency hash algorithm is used by Dynamo. At that time, it's not the exact hash value, but a range of hash values. When the hash value of the key is in this range, it will be searched clockwise along the loop, and the first node encountered is what we need. The consistency hash algorithm is improved by Dynamo, and in the ring, a set of devices are acted as a node rather than only one device is acted as a node. The synchronization mechanism is used to achieve the consistency of the data.

In HDFS the numbers of the copies are set to be 3. Whether the data would be stored in the node or not depends on the capacity of the node. The greater the capacity of the node, the greater the probability that the data will be stored in this node. So, when the capacity of the node is quite different, the nodes with large storage capacity in the system would be overloaded. The copy mechanism proposed in this paper can achieve load balancing, and the reliability and availability of cloud storage are also effectively improved. The system replication policy includes dynamic replica policies and static replica policies. The static copy strategy refers to the numbers of copies. The placement is fixed from the start to the data failure. The dynamic copy strategy is a strategy that system can adjust the numbers of copies in real time and their location, depending on performance requirements, load, and so on. The copy strategies for small files and large files are described as follows.

3.1 The copy strategy for a small file

Files that do not exceed 10MB are defined as small files. The SQL Server relational database is used as the copy strategy for small files. After receiving the file from the mobile terminal, the dispatcher will judge its size first, and then, once the small file is identified, it will be handed over to the relational database machine group. The correlation properties of the file are stored in the database table. The optimal node with the lightest load is dynamically selected by the database machine group. The lightest load database server in the machine group is selected to store the file, and keep a copy to ensure the reliability of the data. The IP address of the database server will be stored in the primary server. The IP address of the database server is retrieved and got from the primary server and then interacts with the database server. The storage processing pattern for small files is shown in figure 2.

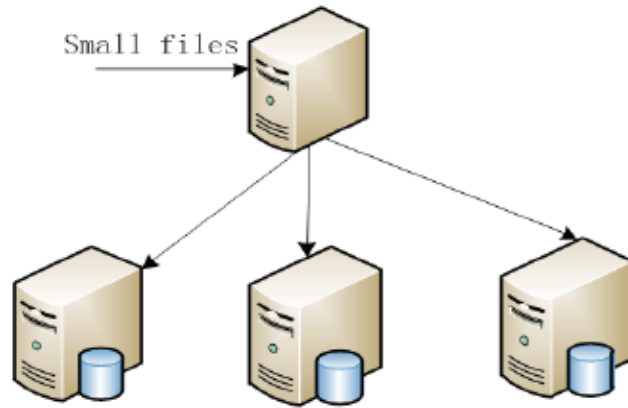


Figure.2 The Storage Processing Pattern for Small Files

The corresponding relationship is stored in the data table in the primary server, and the corresponding relationship is the file name and the server IP address. The data table is shown in table I.

Table 1 Data Tables in the Primary Server

Field	Typ	Length	Note
ID	int		Serial number
fname	varchar	255	Filename
IP	varchar	15	IP address

File names, file sizes, and content are stored in the database server. The file information table is shown in table 2.

Table 2 The File Information table

Field	Typ	Length	Note
ID	int		Serial number
fname	varchar	255	Filename
fsize	int		File size
creat	datetime		creation time
context	mediumtext		Content

3.2 The Copy strategy for large files

Files larger than 10MB are called large files. The storage group is used as a copy strategy for large files. The system architecture of the storage group is fully connected. The system architecture is shown in figure 3. The PC is used as a storage medium in the storage group. However, the reliability of the PC is not high, and it will even fail when the data is stored. Therefore, a copy is required to ensure that the data is reliable.

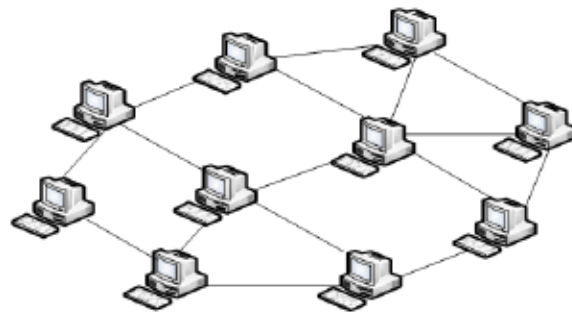


Figure.3 The System Architecture Diagram of the Storage Group

In the storage group system, all information about adjacent nodes are stored in every PC. The needed storage nodes can be found quickly through querying the information stored in the nodes.

The structure of storage space in the storage group is ring, and at the same time, the method of the unified addressing is adopted. In the storage group, the difference in performance of the PC can be offset by the virtual contiguous storage space. First, the hash Algorithm message-digest Algorithm 5 is used to implement system address conversion. The actual physical address is processed and converted to 32-bit information string through the MD5 algorithm, And then these information string are stored in the virtual continuous address. Thus, the differences in performance between devices will be offset. The loop storage structure of the storage group is shown in figure 4.

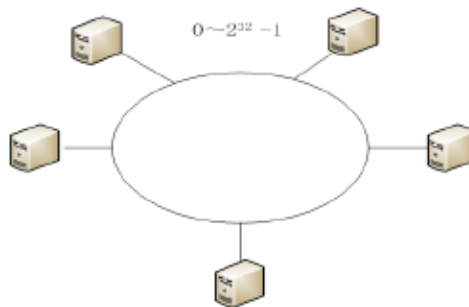


Figure.4 The Loop Storage Structure of the Storage Group

The converted address is mapped to the virtual storage space loop of the storage group through the MD5 algorithm. The device is found in the clockwise direction, and then the data is stored in the first PC mapped. The data is backed up to two adjacent PC. The larger the amount of data in the system, the more uniform the spatial distribution will be. The data are stored when the routing of the corresponding PC and adjacent PC are updated. The routing information table is shown in table 3.

Table 3 Routing Table

Field	Typ	Length	Note
ID	int		Serial number
fname	varchar	255	Filename
fsize	int		File size
IP	varchar	15	IP address

The IP address of the PC device where the file replica is located is stored in the IP field In the routing information table. The IP field is the routing information for the adjacent PC. Once a node fails, all the information stored in the node are backed up and the routing information of the adjacent node is modified in time. According to the principle of the consistency hash algorithm, the storage space of the new PC device will be mapped to the new virtual address space when a new device needs to be added to the storage group. The existing space on the ring will not be changed, and this method can be very effective in avoiding the vibration of the address space. Meanwhile, the routing information on the adjacent PC are updated. The process of adding a PC is as shown in figure 5.

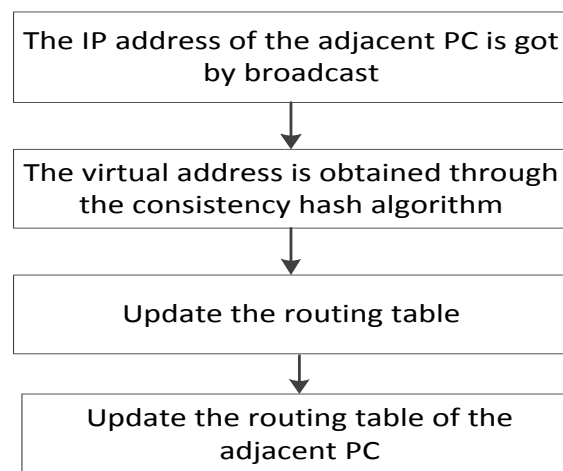


Figure.5 The Process of Adding A PC

The process of exiting the storage group is essentially the same as the process of joining. In the clockwise direction, there are three PC in the storage group, X, Y, and Z. If Y applied to get out of the loop, the information that was stored in the Y would be copied from X to Z, and the information that was stored in the Y would be copied from Z to X, and then, the data in Y-master backup is copied to the first device from z of the loop.

4. Security Design of the Cloud Storage

Cloud storage is a hot topic in industry and academia in recent years, and the security problems of the cloud storage would have been under scrutiny. The AES(Advanced Encryption Standard) algorithm is simple and the encryption of AES is fast. However, AES has problems with key allocation and confidentiality management. There is no need for secret allocation of keys in asymmetric encryption algorithms, and at the same time the security of the keys is easier. In addition, user authentication and digital signatures can be achieved through the RSA algorithm. To make full use of the advantages of the AES and RSA algorithms, a solution that combines AES and RSA encryption algorithms for mobile terminal cloud storage security design is proposed in the paper.

4.1 Encryption and decryption design for Mobile terminal

After the data is encrypted through AES, then the encryption key is encrypted by RSA. The encrypted key message are binded to the encrypted data. The message will then be stored in each node of the HDFS. This approach can improve the storage efficiency of the mobile cloud storage system and also solve the key distribution of single key cryptography. When the data on HDFS is read and downloaded,

the AES key is extracted from the cryptograph, then the decryption is obtained through the user’s private key, finally, the document is declassified and plain text is obtained. The process is as follows.

- 1) During data encryption uploads, users log in to the cloud storage system, Sending data requests to HDFS and encrypting the transfer. At the same time, a 128-bit AES encryption key is generated by the client random key generator.
- 2) On the mobile side, the data that needs to be transmitted is encrypted with the AES key, and the cipher text would be got.
- 3) The encryption KEY of the file is encrypted through the 128-bit RSA public KEY, and then the key cipher is obtained.
- 4) The key cipher are bound to the file cipher, In accordance with the file cipher, the file is stored in the HDFS file system with the corresponding tag bit and data length identification.
- 5) When the data is downloaded from an HDFS system in the cloud, the data are decrypted and downloaded. After the data are obtained, which are transferred from the HDFS system to the mobile end. The first bit of data is judged first by the system, and if it is zero means that the data is in plain text, the data are restored after removing the tag bits. On the contrary, if it is 1 means that the data is a cipher, it should be decrypted.
- 6) First, extract a 128-byte AES key cipher from the data, AES plaintext key are got by decrypted the user’s personal RSA private key.
- 7) The Cipher part of the stored file cipher is decrypted by AES through the AES key, and then the stored file plaintext is got

4.2 The data storage format of the cloud storage system

The data for the cloud storage system includes two storage formats, which are plaintext storage and crypto text storage. The storage format is shown in figure 6

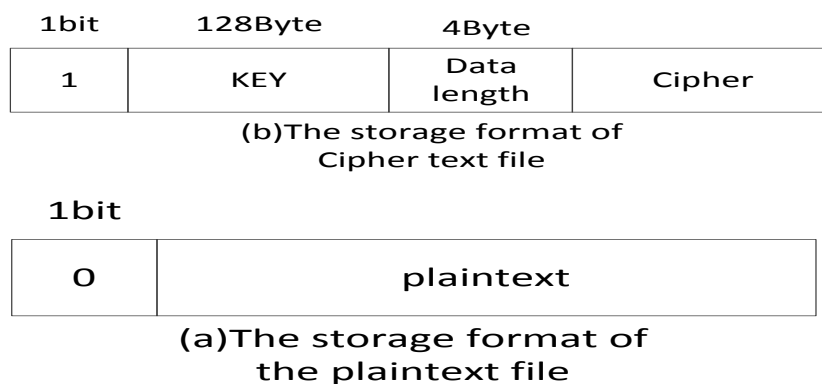


Figure.6 The Storage Format of the File

If the first digit in the storage format is 0, which means that the data is stored in plain text, on the contrary, if it is 1 means that the data is a cipher. The 128 bytes need to be added before the cipher text to store the AES cipher key encoded by RSA when the data is stored in cipher text. The field for the valid

length of the data is 4 bytes, and the field of Cipher represents the encrypted text by AES.

Each 1024-byte byte stream is encrypted with AES and is converted into a 1040-byte cipher stream. So the length of the Cipher section grows relative to the original plaintext data.

4.3 Security analysis

The security design of the mobile cloud storage is implemented in a combination of the AES algorithm and the RSA algorithm. The security is analyzed separately for the AES and RSA algorithms. The security of AES is analyzed through exhaustive attack, the differential attack, and interpolation attack when the AES key don't be known.

1) Exhaustive attack: The average complexity of the exhaustive key is 2^k-1 AES encryption, in which K is the length of the key. For the 128-bit key in this scheme, 2127 times of AES encryption are required and the calculation is very large, and obviously this method of attack is invalid.

2) Differential attack: The wide trajectory strategy adopted by the AES algorithm can effectively resist differential attacks. The prediction probability of the difference trajectory is between 2 and 150 after four rounds of transformation, it's between 2 and 300 after the eight transformations. So, enough times can be identified to make all the differential trajectory less than $1/2^n-1$, n is the number of blocks. This makes the difference attack fail.

3) Interpolation attack: F256 domain in AES algorithm, expansion is shown as blow:

$63+8FX127+B5X191+01X223+F4X239+25X247+F9X251+09X253+05X254$

Because the expansion is complex, the attack is also invalid.

Through this analysis, The AES algorithm is better immune to known attacks the unknown AES key, From the analysis above, we can learn that the AES algorithm is better immune to known attacks in case of not knowing the AES key. Also, the user's files in HDFS are stored in a certain size, and the security of the system can be further enhanced. Therefore, the main issue of security is the security of the AES file encryption key. How to manage and store file encryption keys is the key to determining the security of the solution.

In the design scenario presented in this article, Technology of one-timepad is used for file transfer storage. Each data stored has a different AES key, and the AES key is transparent to the user. In addition, the AES key for each file is encrypted by using the RSA algorithm. The encrypted AES key is bound to the file cipher and then stored in HDFS. The user must take care of his RSA private key throughout the process. The above encryption is done on the mobile side, which implements the file's cryptographic transfer and cryptographic storage.

And then the security of RSA is analyzed in detail. The security of RSA depends on the large integer factorization. The difficulty of attacking the RSA system is the difficulty of the large integer factorization. The Schroepel algorithm is a better factorization algorithm, and which is often used to analyze the problems of the large integer factorization. The number of operations required in decomposing the factor of decimal number n with different length by using Schroepel algorithm. The number of decomposing operations is shown in table IV, in which the factor of decimal number n with different length is decomposed by using Schroepel algorithm.

Table 4 The Number of Operations of Decomposition Factor By Using the Schroepfel

digits of Decimal number n	50	100	200	300	400
the number of operation	1.3×10^{10}	2.4×10^{15}	1.1×10^{23}	1.4×10^{29}	2.6×10^{34}

The longer the length of n is, the more difficult the factorization is in the RSA algorithm. For every ten bits of binary that are added, the time of decomposition is going to be doubled. And then the harder it is to decode the password, the more the strength of the encryption will be. A key length of 512, 1024, 2048 bit are often selected in RSA.

In the cloud storage security technology solution designed in this paper, the 16-byte AES key is the object of RSA encryption, the system will be highly secure once the key of 2048 bit is selected.

Assuming that the reliability of the cloud storage system is A. The time of encryption through different encryption algorithms is A_t , the encryption time A_t is reversed first, and after the normalization processing, A_j is got from A_t . The transfer rate of a file with the same size after the normalization processing is A_k , n is the copy number. The reliability model of the system is:

$$A = [1 - (1 - A_j)^n] [1 - (1 - A_k)^n] \quad (1)$$

It can be concluded through the analysis of the reliability model, when the value of A_j and A_k are more closer to 1, and the number of n is more larger, the cloud storage system will be more reliable and with higher security.

5. Experiment and Result Analysis

The Hadoop cluster built in this article consists of one namenode server and three datanode servers. The client submits the data through the namenode server. The configuration of the four datanode servers is: Intel dual core CPU G630@2700MHz; network environment for NetLink BCM5784 Gigabit Ethernet; The version of Hadoop is 1.0.3; The version of Linux is ubuntu 11.10; And JDK 1.6.0 _17 is used. The configuration of Client is Pentium/(R) Dual-core CPU E5200@2.5GHz. The mobile terminal is Huawei y635-cl00, Qualcomm Snapdragon CPU and 1 GB memory are used. A certain number of storage nodes are simulated by using CloudSim simulation. The data response tests are performed on file upload, file copy and file movement, for large files and small files respectively.

System is tested by using a smart mobile terminal. Experimental results demonstrate that the page is properly displayed, and the response time of the login page is basically completed within two seconds. The percentage of the response time for the transaction is shown in figure 7.

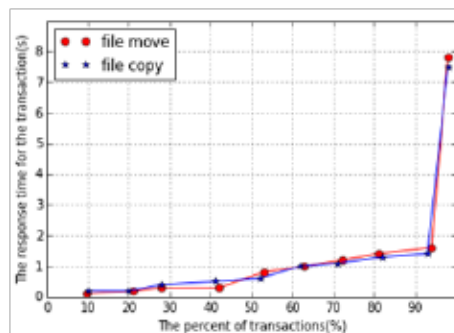


Figure.7 The Percentage of the Response Time for the Transaction

Through the analysis of figure 7, it can be known that 94 percent of transactions in a mobile cloud storage system can be implemented quickly within two seconds. The experimental results show that the system responds fast. The average response time of the transaction is obtained from the diagram.

Although one of the transaction response time is longer, but the response time for most other transactions is acceptable. When this happens, it is thought that the performance of the mobile cloud storage system is better.

After the encryption and decryption mechanism is introduced in the security of the mobile cloud storage, the security of the cloud storage is improved effectively.

There are two questions to be considered: The impact of encryption and decryption on file speed; The impact of encryption and decryption on the performance of the client host.

In the mobile cloud storage security technology proposed in this article, the method of encrypting and decrypting the file on the mobile end is used. The length of the file will change after the file is encrypted into a cipher file. According to the analysis of the file storage format in 4.2. The header of each cipher file needs to be added 128 bytes to store the AES secret key. In addition, when the AES file is encrypted, each 1024 bytes text will be encrypted and then changed to 1040 bytes cipher. In conclusion, after encrypting, the length of the cipher file is about 1.56 percent more than the file. The namenode and datanode in HDFS may be caused an additional cost of about 1.56 percent after encrypting.

For clientnode in HDFS, the time spent on file encryption and decryption is increased, and the performance is reduced eventually.

The effect of file encryption and decryption on the whole file transfer rate is mainly in two aspects: The time required to encrypt and decrypt the transmission file by using AES; The time spent on encrypting and decrypting the AES key by using RSA.

The experimental data are listed in table 5, which includes the time spent on encrypting the different sizes or different types files by using AES and the time spent on transmitting the file in HDFS.

Table 5 Time Comparison on AES Encryption and Decryption

File size (M)	File type	AES encryption (ms)	HDS upload (ms)	AES decryption (ms)	HDS download (ms)
3.07	pdf	1050	2685	370	2800
3.22	MP3	1178	2600	478	2830
23.8	mkv	3238	5930	2648	6290
25.8	doc	3140	5260	2163	6400
166.518	rmvb	23830	46400	16500	42460

The time that AES KEY is encrypted by using RSA are also tested. By using RSA the 128bit AES key was encrypted for an average time of 499ms and decrypted for an average time of 32ms. It can be concluded from the above test data, the time spent on encryption or decryption by using AES is regardless of the file type. The time comparison on AES encryption and decryption is shown in figure 8.

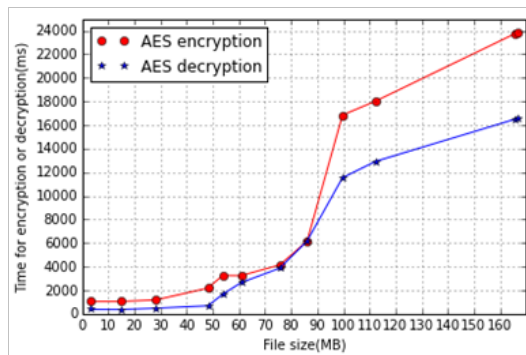


Figure.8 Time Comparison on AES Encryption and Decryption

The same size files are encrypted by different AES, RSA, or RSA + AES algorithms and then the encryption time is different, as shown in table X.

Table 6 Time Comparison for Different Algorithm Encryption

File size (M)	AES encryption (ms)	RSA encryption (ms)	AES+RSA encryption (ms)
3.04	1048	753	770
23.15	3230	2889	2901
167.58	23820	23598	23612

In the solution proposed in this paper , the time of AES key encrypted or decrypted by using RSA is relatively short. File transferring have little impact on total time loss and user experience. It may takes a relatively long time to encrypt files by using the AES, which cause a significant additional time overhead for HDFS. However, the encryption time that AES combined with RSA for encrypting the file was not significantly increased compared to RSA. Besides the impact on overall transmission rates, the impact of encryption and decryption on mobile performance is also important. The 167.58 MB file in table VI is the test case. Table 7 and table 8 are the test result.

Table 7 The Performance of the Mobile End Upload the Data

type of test	utilization rate of Mobile CPU (%)	transmission rate (Mbps)	utilization rate of Mobile /transmission rate
Raw data	16.436	3.57020	4.603663
After the encryption	38.432	2.36015	16.283711

Table 8 The Performance of the Mobile End Download the Data

type of test	utilization rate of Mobile CPU (%)	transmission rate (Mbps)	utilization rate of Mobile /transmission rate
Raw data	14.681	3.90135	3.763056
After the encryption	35.221	2.76147	12.754439

The ratio of CPU occupancy to upload speed is shown in table 7, which the data on the mobile side are tested before the encryption and after the encryption respectively. The ratio of CPU occupancy to download speed is shown in table 8, which the data on the mobile side are tested before the decryption and after the decryption respectively. It can be known from the table 7 and the table 8, if the encryption and decryption mechanism are used for HDFS transmission, then the CPU utilization will be increased by an average of 22% ~ 25% and the overall file transfer rate will be reduced by 30% ~ 35%. As we can see, when the encryption and the decryption mechanism are used, more than three times the performance loss can be caused on the mobile end side.

Although the encryption mechanism and decryption mechanism will cause some performance loss to the mobile end, the confidentiality of the data can be guaranteed. So it is acceptable from the perspective of user data security. Lots of time are spent during encryption or decryption, which can cause a drop in transmission rates. Two points of improvement are proposed for this situation:

- 1) The user can choose whether or not to encrypt the file. Important files are usually in the form of text or images, which are generally small and can choose to be encrypted. However, some larger files, such as video, audio, etc., users can choose whether to encrypt or not. The less important files are stored in plain text, which can improve the access efficiency of the files.
- 2) For cloud storage users, the file transfer and stored procedures are transparent. Therefore, the transport encryption buffer can be set up on the client for large file transferring. After the transfer request is submitted by the user, the file's decryption and transfer operations are implemented in the background. After the transfer is completed, only the prompt message can be given on the mobile end, which can improve the user experience.

Using the reliability model formula of the cloud storage system proposed in 4.3, combine the time required for processing the same size of file in table 6, when a different algorithm AES, RSA, or RSA + AES is used, the encryption time required for encrypting file, after that the encryption time is reversed, A_j then be got after the encryption time is normalized. In the same way, after normalizing, transfer rate A_k is got. If both A_j and A_k are closer to 1, and the number of copies in the cloud storage system is larger, then the reliability of the system is higher. According to the above analysis, the data in one hour is sampled continuously, combined with the data in table 6 and table 7, the reliability contrast diagram for the cloud storage system is shown in figure 9.

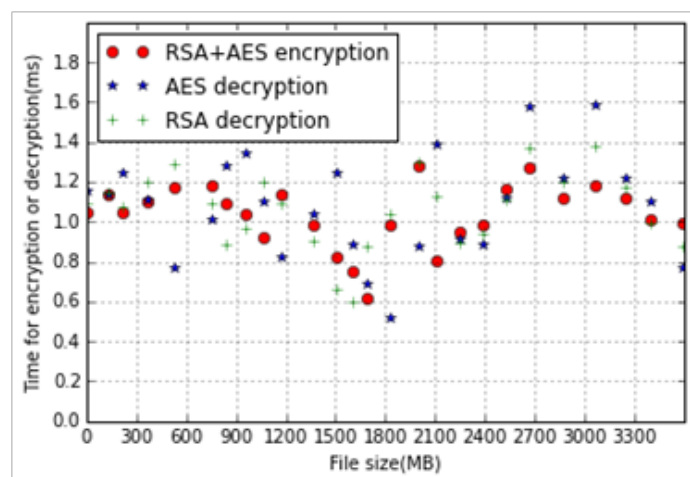


Figure.9 The Reliability Contrast Diagram

It can be known the reliability of the system through different algorithms by comparing the data in figure 9. By using the RSA encryption, the system's reliability values are almost maintained at 1. The reliability of the system is relatively high. That is, it has little impact on file transfer and user experience by using RSA encryption. It may take a relatively long time to encrypt files by using the AES. And the reliability of the system is very jitter. It is shown that the system reliability is low with AES encryption. However, the encryption time that AES combined with RSA for encrypting the file was not significantly increased compared to RSA. The value of the system reliability is consistent with the use of RSA, which can be maintained around 1. It is concluded that the system is relatively reliable by using AES and RSA encryption.

Through the simulation experiment, it is verified that the mobile cloud storage system has a good user experience. It is also verified that the multi-copy mechanism of the cloud storage system can effectively improve the efficiency of the cloud storage system. When a mobile terminal makes a request, the closest copy is selected and then the time can be saved effectively.

In the solution presented in this paper, the encryption and decryption performed by the mobile side has the following characteristics: transport security and storage security of the user data are guaranteed. The mobile side finishes the encryption before calculating the checksum, so the encryption will not break the HDFS data integrity check mechanism. In the entire distributed file storage system, the encryption and decryption are scattered to the various mobile devices. While this will cause some performance damage to the mobile, there is no additional performance penalty for namenode and datanode.

The solution enables the entire distributed file system to be protected by data privacy, and there is no significant performance penalty for multi-user, large access, and file access. In the current version of HDFS, the mobile user identity is given by the host operating system. The user authentication mechanism for the HDFS mobile is also very flawed. In the scheme proposed in this paper, the RSA algorithm and its public key library are introduced, which can create the prerequisite for solving the kind of problem.

6. Conclusion

Cloud storage security technologies for mobile terminals proposed in this paper, different storage policies are used for files in different sizes. Considering the storage efficiency of mobile, the load balancing effect of cloud storage system is improved, and the stability and extensibility of cloud storage system is improved.

In addition, in order to make the cloud storage system has higher reliability. According to the characteristics of HDFS data input output and integrity checking, the AES algorithm is used to encrypt the files uploaded by the user on the client side of HDFS. This ensures that the confidentiality of mobile user data in the cloud storage system. By using the RSA algorithm, the security of the AES key is guaranteed, and the issue of distribution and management of the AES single key password can be resolved. The two storage formats of the cloud file are designed to implement the user's own choice, reduce the number of copies, and ultimately improve the storage efficiency of the mobile cloud storage system. Finally, the reliability of the mobile cloud storage security technology scheme is verified through a series of simulation experiments.

The mobile cloud storage security technology scheme proposed in this paper has better security and reliability. But there are still many problems that have not been solved, and further research is needed.

In order to improve the user experience by setting up the encryption buffer. At the same time, PKI technology can be used to implement CA authentication and digital signatures for HDFS users to further enhance HDFS security.

Foundation Item

The Industrial research project of Science and Technology Department of Shaanxi Province(Grant No. 2016KTZDGY4-09)

References

- [1] IDZIOREK J, TANNIAN M, JACOBSON D. Attribution of Fraudulent Resource Consumption in the Cloud [J]. 2012 IEEE Fifth International Conference on Cloud Computing, 2012: 99-106.
- [2] TSAI T, THEERA -AMPORNPUNT N, BAGCHI S. A study of soft error consequences in hard disk drives [J].IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012), 2012:1-8.
- [3] Schmuck F B, Haskin R L.GPFS: A shared-disk file system for large computing clusters[C]//Proceedings of the Conference on File and Storage Technologies, January 28-30, 2002:231-244.
- [4] Namjoshi J, Gupte A. Service oriented architecture for cloud based travel reservation software as a service[C]// Proceedings of the 2009 IEEE International Conference on Cloud Computing(CLOUD' 09), Bangalore, India, Sep 21-25, 2009.Los Alamitos, CA, USA: IEEE Computer Society, 2009:147-150.
- [5] Goth G. Virtualization: Old technology offers huge new potential [J].IEEE Distributed Systems Online, 2007,8(2).
- [6] BOWERS K D, JUELS A, OPREA A. Proofs of retrievability : theory and implementation [J]. Proceedings of the 2009 ACM workshop on Cloud computing security (CCSW' 09), 2009:43.
- [7] ARMBRUST M, STOICA I, ZAHARIA M, et al. A view of cloud computing [C]. Communications of the ACM , 2010, 53(4):50.
- [8] MELL P, GRANCE T. NIST Special Publication 800 -145:The NIST Definition of Cloud Computing [J]. National Institute of Standards and Technology, 2011.
- [9] KARAME G O, CAPKUN S, MAURER U. Privacy -preserving outsourcing of brute-force key searches[J]. Proceedings of the 3rd ACM workshop on Cloud computing security workshop (CCSW' 11): 2011:101.
- [10] SCHIFFMAN J, MOYER T, VIJAYAKUMAR H, et al. Seeding clouds with trust anchors [J]. Proceedings of the 2010 ACM workshop on Cloud computing security workshop (CCSW' 10), 2010: 43

Optimal Pricing Strategies for Resource Allocation in IaaS Cloud

Zhengce Cai ^a, Xianwei Li ^{*b,c}

^a Department of Information Service, Anhui Business College, Hefei, China, 230000;

^b School of Information Engineering, Suzhou University, Suzhou, China, 234000;

^c Global Information and Telecommunication Institute, Waseda University, Tokyo, Japan

Email: *lixianwei163@163.com

Abstract. In cloud computing environment, pricing is an effective method for resource allocation and it provides efficient incentive for cloud providers to provide cloud services. In this paper, we investigate two pricing schemes for the allocation of cloud resources in a monopoly cloud market subject to constrained capacity of the cloud provider. There are two main pricing schemes, on-demand and reserved pricing mechanisms adopted by leading cloud providers, i.e., Amazon and Gogrid. We analyze how cloud users make their choice decisions to subscribe to cloud resources under different pricing schemes.

Keywords: Cloud computing, pricing schemes, resource allocation, revenue, utility

1. Introduction

Cloud computing has received a great deal of attention in both research and engineering fields, and the use of cloud services has become more and more wide. Fig .1 illustrates the application of cloud services ^[1]. The definition of cloud computing is an open topic and it can be defined by several ways, one popular recognized definition proposed by Buyya et al. ^[2] is:

Computers that are dynamically provisioned and presented as one or more unified computing resources based on “a cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized service-level agreements established through negotiation between the service provider and the consumers.

Cloud services can be categorized into three main types ^{[3][4]}, Infrastructure as a Service (IaaS), Software as a Service (SaaS) and Platform as a Service (PaaS), among which IaaS and SaaS are more commonly used, therefore, most of the current works focus on study resource allocation in IaaS and SaaS clouds. In IaaS cloud context, such as Amazon EC2, physical resources (memory, disk, and CPU et al) are virtualized into different types of virtual machines (VMs), and the computational resources are leased to cloud users in the form of VM instances, as shown in Fig .2. Table I shows some configurations of VM instances in Amazon EC2. In PaaS cloud context, such as Google App Engine, cloud users can develop and run their applications on a computing platform. In SaaS cloud context, cloud users can access the applications which are delivered over the Internet.

The rest of the paper is organized as follows. In the second section, we study optimal pricing for revenue maximization by making use of game theory to investigate the relationship between cloud provider and users. We do simulations to verify our analysis in the third section. Conclusions and future works are given in the last section.

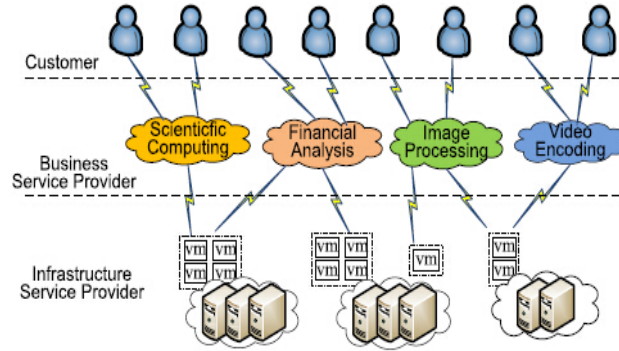


Figure.1 The uses application of cloud services ^[1].

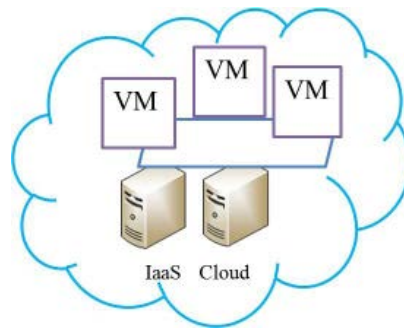


Figure.2 An illustration of IaaS cloud

2. Pricing for Revenue Maximization

We study optimal pricing for revenue maximization in a monopoly cloud market in this section. First, we analyze how to make choice decision given different prices and pricing schemes of VM instances. Then, we will study how to set optimal prices in order to maximize the revenue of cloud providers while meeting cloud users' satisfactions.

2.1 Cloud Users' Choice with Different Pricing Schemes

As illustrated in Table II and Table III, even for the same type of instance, cloud users have to make a decision about how to choose the suitable pricing schemes. In order to help cloud users make right decisions, we first analyze which pricing scheme cloud users should choose. Suppose that a cloud user wants to buy t hours to process his/her service requests. Take m1.small VM instance of Amazon EC2 as an example, the price that he/she has to pay is $\$0.048t$, and the reservation price per month is $\$21.9$. By setting $0.048t=21.9$, we can get $t \approx 456$ hours. This means that if total time of the usage of the m1.small VM instance are less than 456hours in one month, it is better for this cloud user to adopt on-demand pricing scheme, otherwise, it is better for this cloud user to adopt reservation pricing scheme. If this cloud user choose to buy the similar type of VM instance in Gogrid, for example, X-Small, the

total money that he/she has to pay is $0.03t$ under on-demand pricing scheme, and the reservation price is \$16.43 per month, by setting $0.03t=16.43$, we get $t \approx 548$ hours. This means that if total time of the usage of the X-Small VM instance are less than 548hours in one month, it is better for this cloud user to adopt on-demand pricing scheme, otherwise, it is better for this cloud user to adopt reservation pricing scheme. We summarize the above analysis results in Table IV. Fig .3 further illustrates the adoption option under on-demand and reservation pricing schemes.

2.2 Cloud Users' Decision Choices

We next study how to set optimal prices in order to maximize the revenue of cloud providers while meeting cloud users' satisfactions. Assume that there is a cloud provider with total capacity C selling cloud resources in the form of VM instances to a potential number of cloud users, and the price of per VM instance is charged with p \$/h under on-demand pricing scheme. Each cloud user is denoted by θ , which is uniformly distributed in $[0, 1]$.

Table 1 Configurations of Amazon EC2 VM Instances

Instance Types	CPU	Storage (GB)	Memory (GiB)
m1.small	1	160	1.7
m1.large	2	2*420	7.5
m2.xlarge	2	420	17.1
m3.xlarge	1	4	3.75
c3.2xlarge	8	160	15

Table 2 Prices of some Amazon EC2 VM instances

Instance Types	On-demand pricing(Hourly)	Reservation pricing(Monthly)
m1.small	\$0.048/h	\$21.9/m
m1.large	\$0.385/h	\$89.79/m
m2.xlarge	\$0.27/h	\$89.06/m
m3.xlarge	\$0.293/h	\$152.57/m
c3.2xlarge	\$0.462/h	\$234.33/m

Table 3 Choice decision for cloud users

Instance Types	Hourly	Monthly
m1.small	Usage <456hours	Usage \geq 456hours
X-Small	Usage <548 hours	Usage \geq 548 hours
Medium	Usage <456hours	Usage \geq 456hours
Large	Usage <581hours	Usage \geq 581hours
X-Large	Usage <546hours	Usage \geq 546hours

We model the interactions between cloud providers and users as a stakelberg game ^[10], which is illustrated in Fig 4. In the first stage, cloud provider makes their price decisions, and in the second stage, cloud users make their selection decision choices. We make use of the backward method and first analyze cloud users’ decision choices. Given the price of that VM instance p, a cloud user that requires xi VM instances will pay pxi, and his/her net benefit can be expressed as

$$\theta u(xi) - pxi \tag{1}$$

where u(xi) is the utility that this cloud user gets. We model u(xi) by the following utility function which is widely used in the literature ^[11],

$$u(xi) = xi^\alpha \tag{2}$$

where α is an elasticity parameter, which lies in (0,1). Fig. 5 illustrates how users’ utilities vary with different values of α . From this figure we can observe that α reflects the elastic demand of this cloud user, that is, the percentage change of demand to the percent change of price, and cloud user will get more utility if the value of the elastic parameter is higher. It is known that the elasticity of the above utility function is $1/(1 - \alpha)$.

For the cloud users, they will choose to subscribe cloud resources if and only if their net benefits are nonnegative, which implies that,

$$\theta x^\alpha - px \geq 0, \tag{3}$$

from which we can obtain a critical value θ_0 , and the cloud users whose values are distributed in $[0, \theta_0]$ will not to subscribe to the cloud resources, and the users whose values are distributed in $[\theta_0, 1]$ will choose to use the cloud services. The cloud users’ problem can be expressed by

$$\begin{aligned} &\max[\theta u(x) - px] \\ &\text{s.t. } x \geq 0 \end{aligned} \tag{4}$$

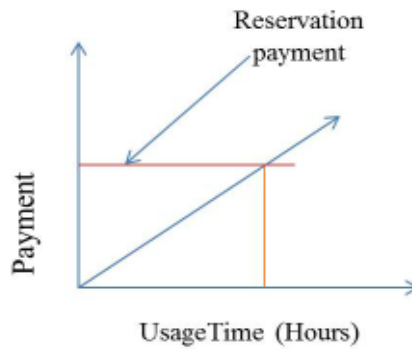


Figure.3 Illustration the adoption option.

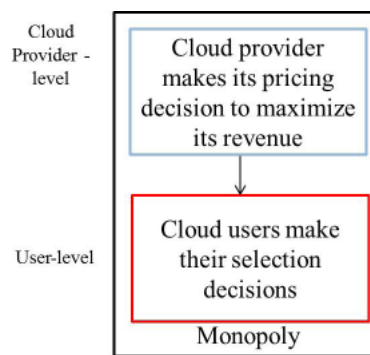


Figure.4 Stakelberg game in the monopoly cloud market.

2.3 Cloud Provider’s Pricing Decision

We will study how to set optimal prices in order to maximize the revenue of cloud providers while meeting cloud users’ satisfactions in this subsection. Pricing provides an economic incentive for cloud providers and it ensures the success of cloud computing.

For the cloud provider, its objective is to maximize its revenues, which is the total pay from cloud users. The cloud provider’s problem can be formulated as follows,

$$\max \pi = p \sum_{i=1}^N x_i \tag{5}$$

$$\text{s.t. } \sum_{i=1}^N x_i \leq C$$

$$\theta u(x_i) - px_i \geq 0$$

$$x_i \geq 0$$

The first constraint of problem (5) ensures that the total number of VM instances should not be over the capacity of this cloud provider, and the second constraint is to make sure that cloud users’ net benefit is nonnegative. From the observation of the second constraint of problem (5), we can find that $px_i = \theta u(x_i)$ should be satisfied to be optimal. Otherwise, the cloud provider can increase the price of this type of VM instance. Therefore, we can transform the problem (5) into an equivalent problem,

$$\begin{aligned} \max \pi &= \sum_{i=1}^N \theta u(x_i) \\ \text{s.t. } \sum_{i=1}^N x_i &\leq C \\ x_i &\geq 0 \end{aligned} \quad (6)$$

Now we get the cloud provider's revenue maximization problem, and we will do simulations to verify our analysis in the next section.

3. Simulation Results

In this section, we will do simulations to verify our analysis in the previous section. We first analyze users' decision choices, and then analyze cloud provider's revenue problem.

3.1 Cloud Users' Utilities

We first analyze how cloud users are sensitive to the prices of cloud resources. Fig.6 illustrates how cloud users' net benefits vary with different values of θ with $x=2$, $p=0.2$ and $\alpha =1/2$. We can observe from Fig.6 that not all the cloud users can get net benefit, therefore, some cloud users may not choose to subscribe to the cloud resources. The parameter θ reflects cloud users' willing to pay, which means that users with higher value of θ will be more willing to use cloud services. Fig.7 illustrates how cloud users' net benefits vary with the price charged by cloud provider with θ and other parameters are set the same as in Fig.6. We can observe that with price increasing, the net benefit of this cloud user will get less net benefit, and this implies that in order to encourage cloud users to use cloud services, cloud providers should set prices of cloud resources properly, otherwise, higher prices will discourage cloud users to pay to use cloud services.

4. Conclusions and Future Works

We studied resource allocation in a monopoly cloud market. We analyzed the choice decisions of cloud users in the Amazon EC2 and Gogrid clouds, and pointed out the right choice for cloud users when face on-demand and reservation pricing schemes. We not only analyzed how cloud users' utilities and net benefits vary with different types of cloud users and different number of VM instances, but also analyzed how the revenue of cloud provider varies with different prices and different number of VM instances. Future works will include resource allocation in a duopoly or oligopoly cloud market where there are more than one cloud providers.

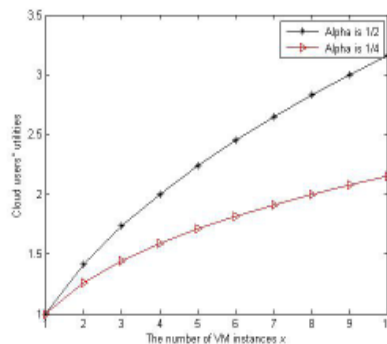


Figure.5 How cloud users' utilities vary with different values of α .

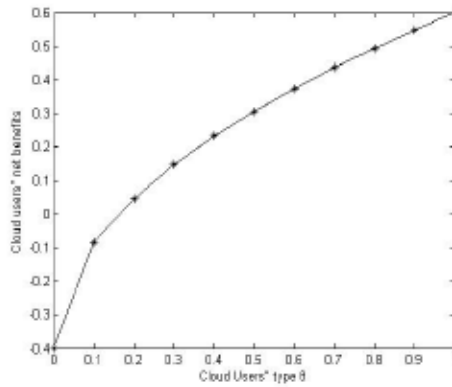


Figure.6 How cloud users’ net benefits vary with different values of θ .

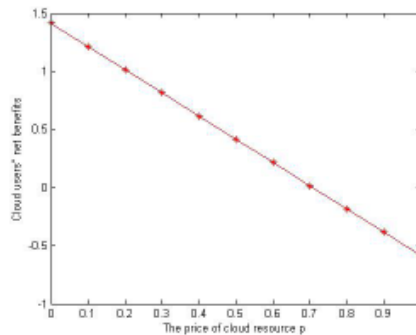


Figure.7 How cloud users’ net benefits vary with price of cloud resources p .

4.1 Cloud Provider’s Revenue Problem

We next analyze how the cloud provider sets its prices in order to maximize its revenue. From Eqs. (5) and (6) we know that the revenue of cloud provider is affected by cloud users’ choices, and we transform the revenue function into a function associated with cloud users’ utilities. Fig.8 shows that the revenue of the cloud provider varies with different number of VM instances, and the revenue will be higher if the price is set higher with the same number of VM instances. Fig.9 shows that cloud users are more willing to subscribe cloud services if they have higher values for using cloud services, and if cloud users have more elastic demand for cloud services, cloud provider will get more revenue. The above analysis imply that cloud provider can set higher prices for these cloud users who are more willing to pay and who have higher elastic demands.

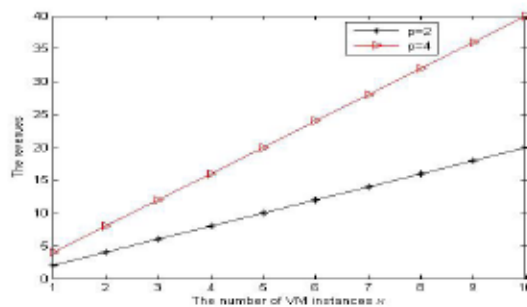


Figure.8 The revenues of cloud provider vary with different number of VM instances.

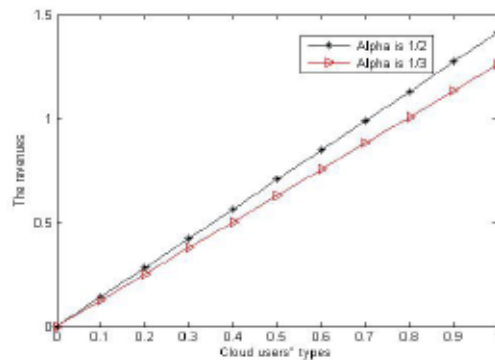


Figure.9 The revenues of cloud provider vary with different cloud users' types.

Acknowledgment

This paper is supported by the following projects, Anhui Key research projects of Humanities and Social Sciences (SK2016A0207), and Suzhou Regional Collaborative Innovation Center (2016szxt05).

References

- [1] J. Mei, K. Li, A. Ouyang et al. "A Profit Maximization Scheme with Guaranteed Quality of Service in Cloud Computing," in press.
- [2] R. Buyya, C.S. Yeo, and S. Venugopal, "Market Oriented Cloud Computing: Vision, Hype, and Reality for Delivering it Services as Computing Utilities," Proc. 10th IEEE Conference on High Performance Computing and Communications (HPCC 2008), Dalian, China, pp. 5-13, Sep. 2008.
- [3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. A view of cloud computing. *Commun. ACM*, 53(4):50 – 58, 2010.
- [4] S. K. Garg, S. Versteeg, and R. Buyya. A framework for ranking of cloud computing services. *Future Generation Computer Systems*, 29:1012 – 1023, 2013.
- [5] Amazon EC2. <http://aws.amazon.com/ec2/instance-types/>. GoGrid Pricing. <http://www.gogrid.com/pricing>.
- [6] Q. Wang, K. Ren, and X. Meng. When cloud meets ebay: Towards effective pricing for cloud computing. In *IEEE INFOCOM*, 2012.
- [7] H. Xu and B. Li. Dynamic cloud pricing for revenue maximization. *IEEE Transactions on Cloud Computing*, 1(2):158 – 171, July 2013.
- [8] Y. Feng, B. Li, and B. Li. Price competition in an oligopoly market with multiple iaaS cloud providers. *IEEE Transactions on Computers*, 63(1):59 – 73, Jan 2014.
- [9] D. Fudenberg and J. Tirole, *Game Theory*, MIT Press, Cambridge, USA, 1991.
- [10] S.Y. Yun, Y. Yi, D. H. Cho, et al. The economic effects of sharing femtocells. *IEEE Transactions on Selected Areas in Communications*, 30(3): 595-606, April. 2012.

Research on Optimization of memory pool management for high concurrent service requests

LIU Pingping, LU Zhaopan

School of Computer Science and Engineering

Xi'an Technological University, Xi'an 710021, China

Email: 1341369601@qq.com

Abstract. In order to quickly and accurately return the information to the user after the keyword are entered, and to effectively reduce the effect on the performance of the program when the search system allocates and deal locates memory frequently under the high concurrency, the Recoverable Fixed Length Memory Pool, Recoverable Variable Length Memory Pool and Allocate Not Free Memory Pool were designed. According to the different scenes features of the search engine. The result shows that, compared with the default system memory allocator, the efficiency of the Recoverable Fixed Length Memory Pool is increased by 70.20% ,the efficiency of the Recoverable Variable Length Memory Pool is increased by 13.84% and the efficiency of the Allocate Not Free Memory Pool is increased by 90.80%.

Keywords: High Concurrency, search engine, memory pool, distributor

1. Introductiong

The search engine is one of the most important applications of the Internet, which involved in information retrieval, distributed processing, semantic web, data mining etc. The reasonable data structure design, the index and the high concurrent system structure are all the factors that influence the query speed. The basic principle of the search engine has been very stable, but in terms of service, quality and performance needs to be optimized.

Most of traditional search engines use keyword matching mode, the system manages memory when the application is not released too frequently, but in the face of massive data processing and storage, search engines seem powerless. There are some drawbacks that directly using the system call Malloc/Free and New/Delete^[1] to distribute and release the memory. For example, calling the Malloc/New system in accordance with the "first match" and "best match" or other algorithms in free memory block table to find a free memory, the memory usage is not high; The system may need to merge free memory blocks when Free / Delete is called, which will result in extra time and space overhead; It is easy to produce a large number of memory fragments when used frequently, which reduces the efficiency and stability of the program; Memory more prone to leaks^[2] that caused by memory size continues to increase and memory exhausted. For memory allocation problem, Wang Xiaoyin, a professor of Xi' an University of Posts and telecommunications, analysis and research about the method and principle of the establishment of the memory pool in the article of Implementation and Application of the Memory Pool in Linux Kernel^[3]. Memory allocated in memory pool does not need to release, it will be released

when the memory pool is destroyed. Advantages: It speeds up memory allocation, when block of memory is enough, only conduct simple operation such as size judgment and pointer offset; Small memory payloads are high, require less additional information; The memory pool allocated memory usually do not need a separate release, but a unified recovery; In addition to using memory allocation functions instead of malloc, no other special conventions are used.

Therefore, to compare of the traditional search engine memory allocation and the memory pool allocation. In this paper, different memory pools are designed for different application scenarios, it manage memory allocation to get the fastest allocate and release speed. For the user's query, the system's memory management is completely taken over by the programmer, which is more conducive to investigate problem and optimize system, and quickly return a satisfied result for customer.

2. Principles and Key Technologies of Search Engine

Search engine is based on the information extracted from the web site to establish the database, search the relevant records of user query condition matching, and then return the results to the user according to a certain order. The working principle^[4] of search engine is divided into four steps: First, using web crawler technology^[5] to automatically grab the web page from the Internet, then analysis the original web page, and set up an index database, and finally searching and sorting in the index database. When there are multiple threads operating, if the system only has one CPU, it can not be carried out more than one thread at the same time, it can only divided the running time of CPU into several periods, then allocate the period of time to each thread, a thread code is run in a time, other threads are hanged up, this way we call concurrency. In the condition of high concurrency, the search system frequently allocate and recover memory will degrade the performance of program and the memory is used in a particular way, and pay the cost of performance on the function that is not required.

For long-running background service system, the performance decrease mainly due to the default memory management is a universal, and general memory management usually consider many factors, including the thread, size, recovery time, distribution, frequency and so on. For this reason, it is common to consider use of the memory pool to manage memory allocation, rather than simply using New/Delete, Malloc/Free for dynamic memory allocation. By designing a dedicated memory pool to allocate specific memory and optimize performance in different search application scenarios of search system, and to enhance the mass data storage and search speed, in order to solve the problem of universal memory.

2.1 Principle of memory pool

Memory pool^[6] is a way of memory allocation, is a device that can dynamically allocate memory. It can only be used by a specific kernel component (that is, the owner of the pool). Owners usually do not directly use the memory pool, when the common memory allocation fails, the kernel call a particular memory pool function to extract the memory pool, in order to get the extra memory. So the memory pool is only a memory of the kernel memory, used at a specific time.

As shown in figure 1, the memory pool contains a total of 4 memory blocks. When the memory pool is initially generated, only one block of memory is applied to the system, and the returned pointer acts as the head of the entire memory pool. After the application of the continuous demand for memory, memory pool judgment need to dynamically expand, then once again to apply for a new memory block of the system, and all of these memory blocks linked by pointers.

For the operating system, it has been allocated four equal-sized memory blocks for the application program. For example, on the fourth block of memory to enlarge, which contains a part of the memory pool information and three equal size memory pool units. The unit 1 and unit 3 are free, unit 2 has been allocated. When application program need to allocate a unit size of memory through the memory pool, only need a simple traversal of all pool size information, then locate quickly the free memory pool block unit. Then according to the size of the block position information directly locate the first free unit address, return the address and mark the next free unit; Marking directly the corresponding memory unit of the memory pool size information is free when the application program release a memory pool unit.

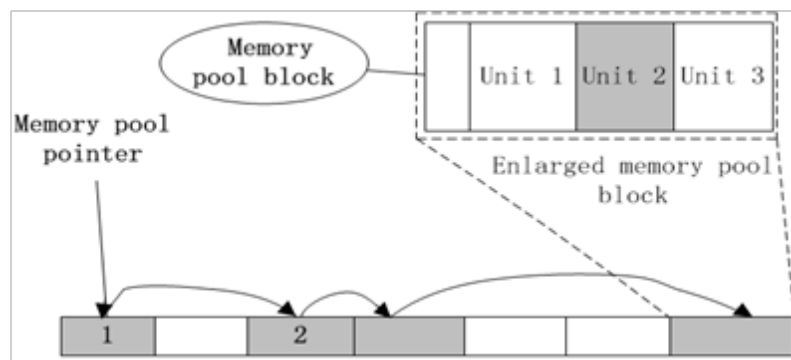


Figure.1 The Working Principle of Memory Pool

2.2 Small object allocation technology

Due to the application of memory block size of memory pool is uncertain, usually directly use the API of New and Malloc to apply for allocating memory. It is not effective for the small object allocation, when frequently used will cause a large amount of memory fragmentation and then reduce the performance, so the small object memory allocation technology^[7] suitable for the small object memory allocation is used here.

The size and number of blocks can be set in the construction period of the small object distributor. The Chunk layer contains logical information, it can configured and returned the block from memory. Once there is no free block in the Chunk layer, the function returns zero. Small Object Pool layer contains a vector, Chunk objects stored inside, the Chunk layer has been extended. There is a chunk queue, which stores all the information, there are two Chunk pointers, one pointing to the currently available Chunk, one pointing to the current with the release of a pointer.

2.3 Scene analysis of search engine system

In this paper, analyzing the characteristics of the three scenes, Fixed length scene, Size is not fixed scene and Multiple allocation scene, designing the corresponding memory pool.

1) Fixed length scene

In the existing search engine system, cache design takes advantage of the hash tables, Original system use the New and Delete functions for the allocation and release of each node of the hash table, and the size of the node is fixed, according to the allocation and release of the fixed size nodes, a memory pool is designed to improve the speed of cache allocation and release. A lot of places use the Map of STL^[9] in

the present search engine system, and the allocation and release of memory of each node in the Map is managed by the distributor in the STL, take over the fixed node memory allocation and release by itself, enhance efficiency, easy to debug.

Based on the above two scenarios, the common is that how to deal with node fixed size, design a small object dispenser to distribute and release the fixed size memory node.

2) Size is not fixed scene

In the cache management of the currents earch system, the search results are put into the cache, which is helpful for the next search, the size of each node in the cache is uncertain, and the time to enter the cache and propose cache can not be estimated. In the update module of the current search system, which manages the update and delete of the document, but the size of the document and the time of the update is unknown.

For this scene, it can design a recoverable variable length memory pool, the lock can be added to deal with base on the characteristics of cache multi thread^[10].

3) Multiple allocation scene

The current search engine will return a result within 10M size after input a keyword, and a lot of information that comes with results will be allocated and released by using New and Delete function, it cause that the New function used frequently, and affect efficiency and bring memory fragments^[11].

After analysis, the search engine return the result sat the same time, memory is frequently allocated, the number of release carried out only when the results of the query are returned, so the factor of frequent distribution should be considered, and the total capacity is not more than 10M, therefore, it is consider to allocate a large chunk of memory, after which all of the small memory is allocated, and finally released through the interface. Based on this scene design allocate not free memory pool.

3. Memory pool design and realize based on the high concurrency

Three scenarios are obtained by analyzing the current search engines: Hash table insert delete, Cache update and document update module, Query result return. Three memory pools are designed for the three scenarios: Recoverable Fixed Length Memory Pool and Recoverable Variable Length Memory Pool and Allocate Not Free Memory Pool were designed. The design structure of the search system memory pool is shown in Figure 2.

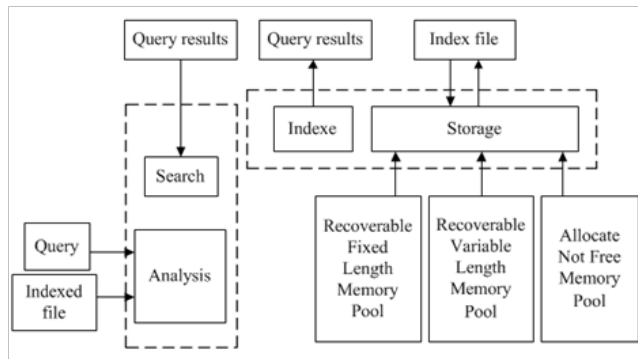


Figure.2 The Design Framework of Memory Pool of Search System

3.1 Recoverable Fixed Length Memory Pool

Recoverable Fixed Length Memory Pool is Small object distributor, it divided into 4 layers structure. As shown in Figure 3, the bottom layer is the Chunk object, each Chunk manages a large chunk of memory, which contains an integer number of fixed size block. Chunk contains logical information, the user can configure and return the block according to it. When the Chunk is no longer remaining blocks, the configuration fails and returns to zero. The second layer is Fix Allocate, which base on the first layer, using the known vector to expand the first layer and ensure that the size of the distribution can be extended. The third layer is Small Object Allocator, which provide universal distribution and return function. The third layer expand base on the second layer, it provide multiple second layer objects, it make the fixed length of the distribution technology turn into a variable length distribution technology. The fourth layer is Small Object, It made a package for the third layer, which provides a number of generic interfaces for the third layer and some common interface, extend it into a multi thread available distributor. Through layer by layer expansion, not only to ensure the release efficiency of the distribution, but also to better package the internal structure together, it not visible to the outside. By providing a common interface, to make it used like the operating system comes with the default memory.



Figure.3 The Structure of Small Object Distributor

3.2 Recoverable Variable Length Memory Pool

Recovery variable length memory pool is a multi-threaded, variable length, recyclable memory pool, similar to hash table. A linked list indicates an assignable size range, each element in the list is a specific size of memory block pointer, which point to a list of memory blocks, to find specific head pointer by aligning, and then assign a node outside in the list. The elements in the range will be allocated through the New, when released, it will be returned to the pool for the next allocation, and beyond the range of elements also be allocated through New, but when released, it directly call Delete, and return to the operating system. About the factors of thread, add lock to ensure the thread safety after the specified by the constructor. Mainly includes Block Header layer, tragCtrlUnit layer and RecycleLitePool layer. The structure of the graph is shown in figure 4.

Block Header layer is the bottom of the distribution structure, nCtrlIndex indicates the size of block

distribution, pNextBlock indicates the next block, the structure is linked list structure, the whole structure is Union type, which save space and improve efficiency. The tagCtrlUnit layer is a headpointer of each BlockHeader layer, and also contains a member that indicates the number of BlockHeader objects. RecycleLitePool layer contains the thread element, the lock element, thetagCtrlUnit layer pointer, some count elements and a memory distributor, default for the New distribution and delete function to delete.

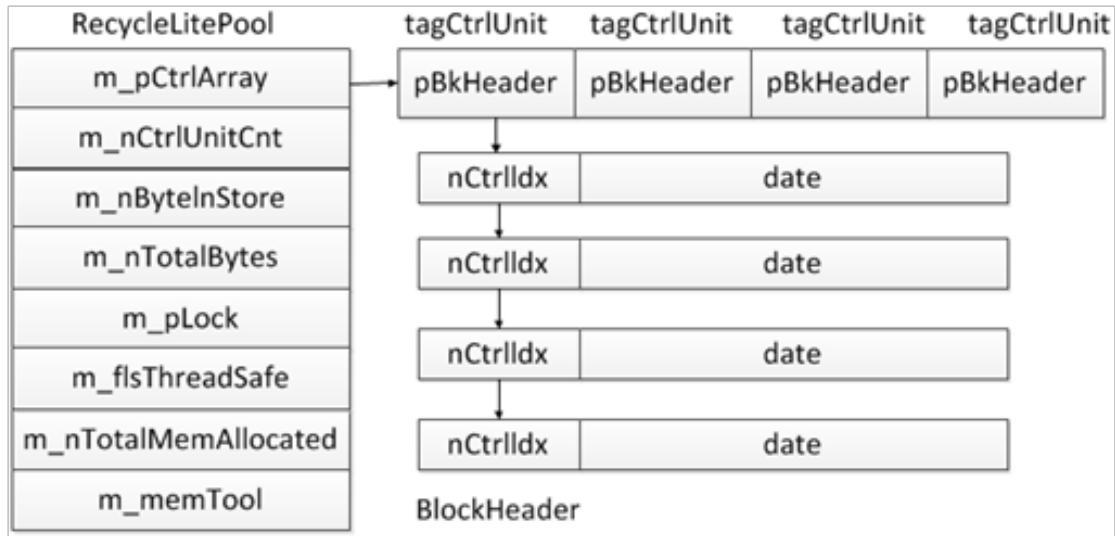


Figure.4 The Structure of Recyclelitepool

3.3 Allocate Not Free Memory Pool

Allocate Not Free Memory Pool is divided into four layers:

Memory Chunk layer: The bottom of the allocation block, there are three members inside, one indicates block size, one indicates location of initial address, one indicates currently available location.

Chain of Memory Chunk layer: Memory Chunk object is organized into a two-way linked list.

Simple Allocate Policy layer: This layer accept the request of distribution, change the size to be allocated not less than the size of Memory Chunk, and then added to the two-way linked list, the pointers of current distribution block point to the new block.

StagePool layer: This layer is the outermost layer, The default template parameter is Simple AllocatePolicy type, which provides external interface for distribution and release.

The overall structure of the StagePool contains a Chunk type pointer, which point to the currently allocated block, the allocation request are looked for from the current block every time, when the margin is not enough, it create a new block inserted into the list, select allocation strategy through the template parameter of Allocate Policy. The structure of the graph is shown in figure 5.

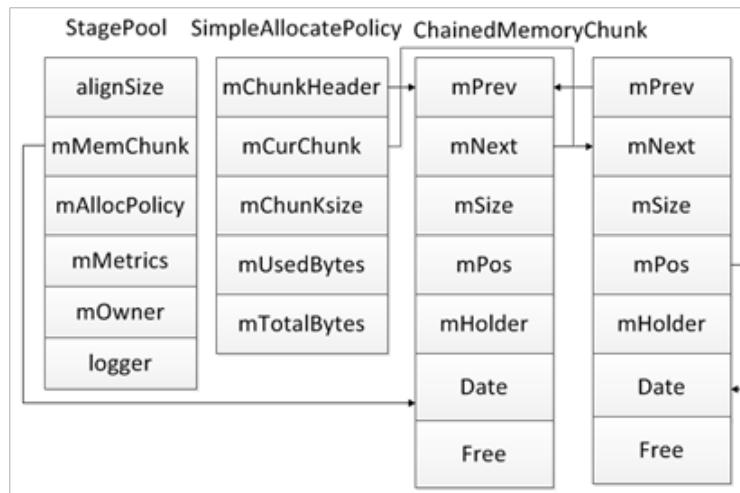


Figure.5 The Structure of Stagepool

4. Performance test and analysis

Through the Centos operating system, the compiler and debugging tools of vim, g++ and scons, Some scenarios are designed to simulate the actual scene of the search engine to test the performance differences between the default memory distributor and the designed distributor.

4.1 Performance test and analysis of the small objects distributor

1) For the small objects allocator test, when the amount of data is 100000, by testing the system function of New/Delete and the memory pool interface function of Allocate/Deallocate. Record 5 groups of data, as shown in Table 1, by analyzing and calculating the time difference, the small objects allocator is increased by 70.20% relative to New of system, and compared with the Delete, it is increased by 2.29%.

2) For the thread adapter hash table test, using the Node Allocator class to match the hash table, construct an identical class of Default Allocator, the internal partition function use New/Delete to achieve, using template parameters to match a hash table. For the single threaded test: For Node Allocator and Default Allocator, to applicate and release 50000 block, assuming that the program is running a fixed time of 10s, during this time repeated inserted and delete operation; Multi-threaded test: For Node Allocator and Default Allocator, to open the same number of threads, and execute thread function, to insert and delete data for corresponding hash tables. Table 2 indicates the test data of hash table. By analyzing and calculating: In the case of Single threaded, in terms of distribution, the efficiency of Node is increased by 22.61%, for the release, the efficiency of Node is reduced by 18.94%; In the case of multiple threads, in terms of distribution, the efficiency of the Node is increased by 13.80%, for the release, the efficiency of Node is reduced by 1.20%.

3) For the test of the small objects allocator adapter map container, to achieve a Small Object Alloator type, internal distribution released is achieved by Small ObjectPool, through the Map function, fit Small ObjectAllocator to Map, Map node distribution and release call interface of Allocate/Deallocate of Small ObjectPool; Similarly to achieve a NewAllocator type, internal distribution release is achieved by

New/Delete interface, also mapping to the Map through the constructed function; To compare with distributor type provided by the system. Single threaded test: Three map were inserted into 100000 data; Multi threaded test: Map data is inserted and emptied by three Map circulation within a certain time. Table 3 indicates the test data of adapter Map. By analyzing and calculating: In the case of Single threaded, in terms of Default, the efficiency is increased by 48.30%, for the New, the efficiency is increased by 54.50%; In the case of multiple threads, in terms of Default, the efficiency is increased by 35.90%, for the New, the efficiency is increased by 33.10%.

Table 1 Testing Data of the Small Objects Distributor(μ s)

Number of times	New allocation time	New release time	Small allocation time	Small release time
1	5318	2739	3174	2157
2	4662	1994	1271	1792
3	4856	2317	1350	1919
4	5093	3044	1381	1988
5	4899	1953	1253	1916
6	4768	2344	1329	3160
7	4835	2051	1743	1858
8	5852	2070	1388	1819
9	6412	2783	1328	1892
10	5119	2310	1271	1859
Average time	5181.4	2360.5	1548.8	2036

Table 2 Testing Data of Thread Adapter Hash Table(μ s)

Number of times	Single thread	Single thread	Single thread	Single thread	Multi-threaded	Multi-threaded	Multi-threaded	Multi-threaded
	default allocation	default release	node allocation	node release	ded default allocation	ded default release	ded node allocation	ded node release
1	42	39	33	48	141	130	121	131
2	43	39	33	48	143	130	123	131
3	43	40	33	47	147	127	119	125
4	43	40	33	48	147	133	131	134
5	43	39	33	48	141	127	123	131
6	43	40	33	48	140	125	121	128
7	43	40	33	48	143	129	126	131
8	43	39	33	48	140	125	119	125
9	43	40	34	39	141	125	123	129
10	43	40	34	49	141	127	122	129
Average time	42.9	39.6	33.2	47.1	142.4	127.8	122.8	129.4

Table 3 Testing data of Adapter Map Multi Thread(μ s)

Number of times	Single thread Default	Single thread Small	Single thread New	Multi-threaded Default	Multi-threaded Small	Multi-threaded New
1	18	10	23	340	229	378
2	19	10	21	381	247	375
3	18	10	19	391	261	361
4	17	9	20	404	241	356
5	19	9	19	419	252	383
6	18	9	19	388	251	377
7	16	9	19	367	260	374
8	18	8	19	373	249	401
9	17	8	22	435	257	366
10	16	9	19	396	248	360
Average time	17.6	9.1	20	389.4	249.5	373.1

4.2 Performance test and analysis of recoverable variable length memory pool

Given a set of arrays with assigned size from 1-10000, 4 threads, 5000 insert delete action, then use a variable length memory pool to assign and storage an array of pointers, to release and reallocate, cycle 20 times to get the test data results; Using Malloc to open the corresponding bytes of memory and assigning to another pointer array to storage. Under the same conditions, compare the time of distribution and release of the system function. Table 4 is the test data, by analyzing and calculating: In terms of New, the efficiency of RecycleLitePool is increased by 13.84%.

Table 4 Testing Data of Recoverable Variable Length Distributor(ms)

Number of times	New	RecycleLitePool
1	183	177
2	185	157
3	181	154
4	197	164
5	177	153
6	183	154
7	183	156
8	182	154
9	183	159
10	181	153
Average time	183.5	158.1

4.3 Performance test and analysis of allocate not free memory pool

Building a new distributor structure, alloc is interface of the distributor, using the space of distributor to allocate 4 bytes every time, allocated 10000 times; As a contrast, the system call the New function to allocate 4 bytes each time, to record the time of 50000 application action. Table 5 is the test data, by analyzing and calculating: In terms of New, the efficiency of StagePool is increased by 90.80%.

Table 5 Testing Data of Allocate Not Free Distributor(MS)

Number of times	StagePool	Newl
1	4	51
2	5	47
3	4	48
4	4	48
5	5	48
6	5	48
7	4	48
8	4	47
9	5	47
10	4	47
Average time	4.4	47.9

5. Conclusion

1)Three scenarios are obtained by analyzing the current search engines: Hash table insert delete, Cache update and document update module, Query result return. Three memory pools are designed for the three scenarios: Recoverable Fixed Length Memory Pool and Recoverable Variable Length Memory Pool and Allocate Not Free Memory Pool were designed.

2)Using the system default memory management function, malloc/free and new/delete. By analyzing of the various factors of the function. Allocating and freeing memory on the heap increases overhead. The design of the memory pool is applied to the search engine system. It optimize the internal memory management and improve the search speed. For the test of the three memory pool, Compared with the system's default memory, its efficiency are increased by 70.20%, 13.84%, 90.80%.

Sponsors or Supporters

This paper is partially supported by Special research project of Shaanxi Provincial Department of Education "16JK1376" .

Reference

[1] DAI Chunyan, XU Zhiwen. Discussion About Malloc/Free and New/Delete in C++[J].Science&Technology

of Baotou Steel (Group)Corporation, 2009(35):59

[2] LI Qian, PAN Minxue, LI Xuandong. Benchmark of Tools for Memory Leak [J]. Journal of Frontiers of Computer Science and Technology.2010(01)29.

[3] WANG Xiaoyin, CHEN Lijun. Implementation and Application of the Memory Pool in Linux Kernel [J]. Journal of Xi' an University of Posts and Telecommunications. 2011(04):40.

[4] QU Weihua, WANG Qun. Introduce and Analyzing of Search Engine Principle [J]. Computer Knowledgeand Technology. 2006(06):113.

[5] DUAN Bingying. Study and Design of Web Crawler in Search Engine [D]. Xidian University. 2014.

[6] GUO Bingxuan, ZHANG Jingli, ZHANG Zhichao. Algorithm of Spatial Data Scheduling Based on Memory Pool [J]. Computer Engineering. 2008,34(06):63.

[7] LIU Tao, NIE Xiaofeng, JING Jiwu, WANG Yuewu. Memory Management in Worm Simulation based on Small Object Memory Allocation Technique on The GTNetS [J]. Journal of Graduate University of Chinese Academy of Sciences. 2012,29(01):131.

[8] GUO Xufeng, YU Fang,LIU Zhongli. An Efficient Memory Built-in Self-Repair Method Based on Hash Table [J]. Acta Electronica Sinica. 2013(07):1371.

[9] LAI Xiangfang. Select The Appropriate STL Containers [J]. Digital Technology and Application. 2015(09):177.

[10] Alexandrescu A.Modern C ++ design: Generic Programming and Design Patterns Applied [M]. Boston: Addison-Wesley Professional. 2001.

[11] Robert W.P.Luka, Wai Lamb. Efficient In-Memory Extensible Inverted File [J]. Information Systems, 2007(32):733.

AuthorBrief

Liu Ping-Ping(1971-), female, Associate Professor, Xi' an Technological University, Research area: Artificial intelligence

Image Watermarking Encryption Scheme Based on Fractional Order Chaotic System

Dawei Ding¹, Zongzhi Li² and Shujia Li³

School of Electronic Information Engineering,

Anhui University, Hefei 230601, China.

Email: ¹ dwding@ahu.edu.cn, ² 493756119@qq.com,

³ 1352288370@qq.com

Abstract. Now the chaotic system and wavelet transform are more and more widely used in the watermarking technology. At the same time, the fractional order chaotic system has more complex dynamic characteristics than the integer order system. So a new image watermarking scheme based on the fractional order Chen chaotic system and discrete wavelet transform is proposed. Chaotic sequences generated by chaotic system are used to encrypt the watermark image, and the processed watermark information is embedded into the original image by the discrete wavelet transform. Finally, the security analysis of the proposed watermarking algorithm is presented. The experimental results show that the proposed watermarking scheme has high security, and it has stronger robustness and invisibility compared with the previous work.

Keywords: fractional order chaotic system, discrete wavelet transform(DWT), image watermarking, security, robustness, invisibility

1. Introduction

With the rapid development of network technology, all kinds of digital media are more convenient to spread through the network, the security of digital media becomes more and more important in the network. Copyright protection is one of the important aspects. Digital watermarking technique is a kind of information hiding technique, it can be used as a kind of more effective copyright protection of digital works and anti fake technology.

At present, the chaotic system is more and more widely used in the watermarking technology and has achieved good results, a considerable part of chaos-based image watermarking schemes are proposed^[1-6]. Poonkuntran and Rajesh proposed a new imperceptible image watermarking scheme for active authentication for images^[1]. The scheme used chaotic system to process the watermark, and used the integer transform to embed the watermark information. Tong Xiaojun et al. proposed an image watermarking technique based on scrambling and self restoration^[2]. A coupled chaotic map was used to scramble the original image block by block. Behnia et al. proposed an image watermarking scheme based on double chaotic map^[3]. One map was used to encrypt the embedded position, and another one was used to determine the pixels of the host image. Gao Tiegang et al. proposed an image watermark authentication method based on neural network with hyper chaotic characteristics^[4]. The method used the authentication password as the key, and the pixel value was used as the input of the neural network. Mooney et al. used the combination of white noise and chaotic sequence to encrypt the watermark^[5].

Gao Guangyong et al. used composite chaos to encrypt the watermarking image, and resisted the geometric attack based on a composite-chaos optimized support vector regression(SVR) model^[6].

Watermarking methods can mainly be divided into three types according to the restore information: non-blind, semi-blind and blind watermarking methods^[7]. Non-blind methods need all the information of the original image and the key. Semi-blind methods need watermark sequence and the key, and blind methods need key only.

Watermarking methods can be divided into two other types according to the embedding strategy:

1) Spatial domain watermarking: the value of the image element is changed directly, and the hidden content is added in the brightness of the image element, however, this method is easy to be obtained, and the robustness of the image processing is poor.

2) Transform domain watermarking: use a mathematical transformation to transform the image into the transform domain, and add the information by changing some transform coefficients of image, and then use the inverse transform to recover the hidden watermark information and image.

The advantages of using transform technique include the ability to ensure that the watermark is not visible and resistance to the corresponding lossy compression.

Keyvanpour et al. proposed a watermarking method based on chaotic map and operation of transform domain^[8]. The coding process was special and the key was generated by chaotic map, the wavelet quantization process was used to transfer the sequence. Zhang Dengyin et al. proposed a watermarking algorithm based on one-dimensional (1-D) chaotic map in wavelet transform (WT) domain^[9]. The watermark was encoded by a chaotic sequence and embedded into the low-and intermediate-frequency bands of three-layer WT domain. Barni et al. proposed a watermarking method based on discrete wavelet transform, the embedded operation was done in the high frequency part^[10]. In addition, there are many examples of the combination of wavelet transform and other operations^[11-13]. Therefore, it is feasible to use discrete wavelet transform(DWT) and chaotic system to encrypt the watermark.

At the same time, the research shows that the low dimensional chaotic system has the defects of the limited key space and the worrying security, but the high dimensional chaotic systems have higher complexity, randomness and unpredictability, and it can better resist the attack of phase space reconstruction and other methods^[14]. The Chen's system is a three-dimensional chaotic system with complex topology than Lorenz attractor. The fractional order chaotic dynamics system has more complex and richer dynamic characteristics than the integer order system, and it has the advantage of increasing the randomness and unpredictability, Moreover, the fractional order system can also provide more key parameters and increase the key space for the encryption system, so it will improve the encryption effect of the system.

Inspired by above analysis, a new image watermarking scheme based on the fractional order Chen chaotic system and discrete wavelet transform is proposed. Firstly chaotic sequences generated by chaotic system are used to encrypt the watermark image. Then the processed watermark information is embedded into the original image by the discrete wavelet transform.

The main content of the study is as follows. In Section 2, the related theoretical works are introduced in detail. In Section 3, the process of the proposed watermarking algorithm is described in detail. Experimental results and security analysis are given in Section 4. The final conclusion is shown in Section 5.

2. Related Works

2.1 The fractional-order Chen's chaotic system

Consider the fractional-order Chen's chaotic system^[15] described by

$$\begin{cases} \frac{d^\alpha x}{dt^\alpha} = a(y-x) \\ \frac{d^\beta y}{dt^\beta} = (c-a)x - xz + cy \\ \frac{d^\gamma z}{dt^\gamma} = xy - bz \end{cases} \quad (1)$$

where α, β, γ are fractional derivative orders, $(x, y, z) \in \mathbb{R}^3$ are state variables, $a > 0, b > 0, c > 0$ are parameters of the system.

The Grunwald-Letnikov of fractional calculus[16] is defined as:

$${}_a D_t^\nu f(x) = \lim_{h \rightarrow 0} \frac{1}{h^\nu} \sum_{j=0}^{[(t-a)/h]} (-1)^j \frac{\Gamma(\nu+1)}{j! \Gamma(\nu-j+1)} f(x-jh); \nu > 0 \quad (2)$$

where a and t are lower bound and upper limit of integral, ν is fractional derivative order, h is integration time step, $[x]$ represents integer part of variable x . Its mathematical expression is shown in the Eq.3:

$${}_a D_t^\alpha f(t) = \lim_{h \rightarrow 0} h^{-\alpha} \sum_{j=0}^{[(t-a)/h]} (-1)^j \binom{\alpha}{j} f(t-jh) \quad (3)$$

where $\binom{\alpha}{j} = \frac{\alpha(\alpha-1)\dots(\alpha-j+1)}{j!}$

Simplified Eq. 3 get Eq. 4:

$${}_0 D_t^\alpha y(t_m) = h^{-\alpha} \sum_{j=0}^m \omega_j^{(\alpha)} y_{m-j} \quad (4)$$

where $\omega_j^{(\alpha)} = (-1)^j \binom{\alpha}{j}; j = 0, 1, 2, \dots$.

According to Eq. 4, change Eq. 1 :

$$\begin{cases} h^{-\alpha} \sum_{j=0}^m \omega_j^{(\alpha)} x_{m-j} = a(y_m - x_m) \\ h^{-\beta} \sum_{j=0}^m \omega_j^{(\beta)} y_{m-j} = (c-a)x_m - x_m z_m + cy_m \\ h^{-\gamma} \sum_{j=0}^m \omega_j^{(\gamma)} z_{m-j} = x_m y_m - bz_m \end{cases} \quad (5)$$

Simplified Eq. 5:

$$\begin{cases} x_m = (ah^\alpha y_m - \sum_{j=1}^m \omega_j^{(\alpha)} x_{m-j}) / (1 + ah^\alpha) \\ y_m = (h^\beta (c - a - z_m)x_m - \sum_{j=1}^m \omega_j^{(\beta)} y_{m-j}) / (1 - ch^\beta) \\ z_m = (h^\gamma x_m y_m - \sum_{j=1}^m \omega_j^{(\gamma)} z_{m-j}) / (1 + bh^\gamma) \end{cases} \quad (6)$$

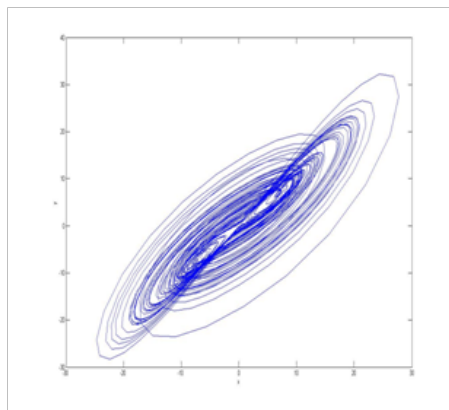
where x_m, y_m, z_m are implied. The iterative algorithm is used to express them:

$$\begin{cases} x_m^{(l)} = (ah^\alpha y_m^{(l-1)} - \sum_{j=1}^m \omega_j^{(\alpha)} x_{m-j}^{(l-1)}) / (1 + ah^\alpha) \\ y_m^{(l)} = (h^\beta (c - a - z_m^{(l-1)})x_m^{(l-1)} - \sum_{j=1}^m \omega_j^{(\beta)} y_{m-j}^{(l-1)}) / (1 - ch^\beta) \\ z_m^{(l)} = (h^\gamma x_m^{(l-1)} y_m^{(l-1)} - \sum_{j=1}^m \omega_j^{(\gamma)} z_{m-j}^{(l-1)}) / (1 + bh^\gamma) \end{cases} \quad (7)$$

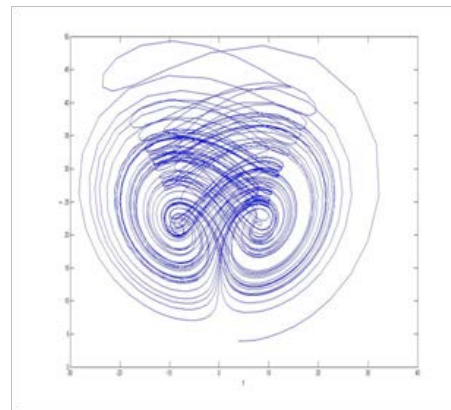
where l is iteration number .

When $|x_m^{(l)} - x_m^{(l-1)}| < \delta, |y_m^{(l)} - y_m^{(l-1)}| < \delta, |z_m^{(l)} - z_m^{(l-1)}| < \delta$ (δ is very small, such as $\delta = 10^{-5}$), $x_m^{(l)} = x_m^{(l-1)}, y_m^{(l)} = y_m^{(l-1)}, z_m^{(l)} = z_m^{(l-1)}$. System will exhibit chaotic behavior when initial conditions are set as: $h = 0.01$, $(\alpha, \beta, \gamma) = (0.97, 0.98, 0.99)$, $a = 35, b = 3, c = 28$, $(x_0, y_0, z_0) = (1, 3, 4)$. The projections of the attractor are shown in Fig.1.

Chaotic system will produce chaotic sequence, and these sequences are used to encrypt watermark image. The result will produce a chaotic encrypted image, which will be then used for embedding the wavelet coefficients^[17].



(a) x-y plane



(b) y-z plane

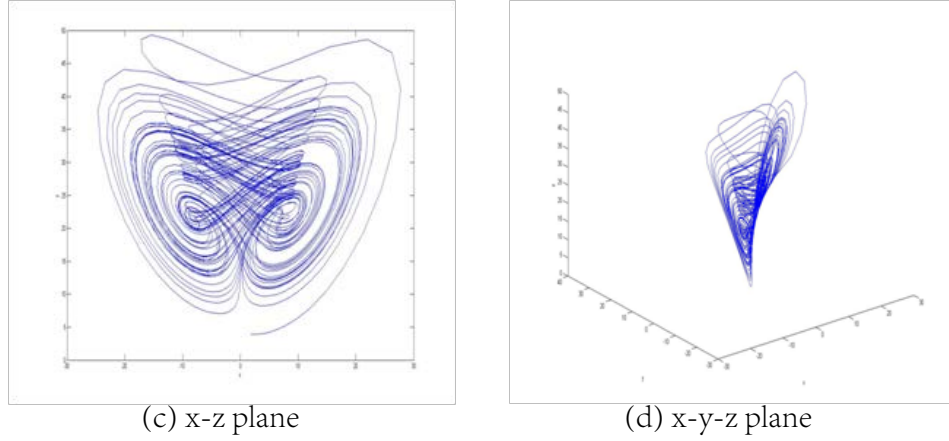


Figure.1 The attractors of system described in Eq. (1)

2.2 Discrete wavelet transform

In the process of image processing, the lossy compression can cause damage to the digital watermark. So the characteristics of lossy compression must be used to find the maximum robustness in the process of embedding and extracting digital watermark. The DWT is a local transformation and has the ability of multi scale analysis. By using wavelet transform, the original image sequence can be decomposed into multi frequency sub images which can adapt to the visual characteristics of the human eye, and the watermark embedding and detection can be carried out in a plurality of levels. Wavelet transform domain digital watermarking method has the advantages of both temporal and spatial domain method and DCT transform domain method.

A two-dimensional image can be decomposed into different frequency components, and the image can be decomposed into 4 parts at each level of the transformation. For example, the first level of decomposition, i.e. LL_1, LH_1, HL_1, HH_1 ^[18]. A wide range of information is contained in the low frequency component LL_1 part. LH_1, HL_1, HH_1 are high frequency components which contain the specific details. Wavelet decomposition can continue be used to decompose LL_1 to get LL_2, LH_2, HL_2, HH_2 . Repeat this process until the required decomposition level is obtained, i.e. LL_n , where n represents the decomposition level. These wavelet coefficients can be used in the future to restore the original image, this inverse process of DWT is known as IDWT.

3. Proposed Watermarking Algorithm

This part gives a detailed introduction of the watermark embedding algorithm and extraction algorithm. The size of the original image I is $M \times N$ and the size of binary watermark image W is $m \times n$.

3.1 Embedding watermarking

The different fractional derivative orders and initial conditions for Eq.(1) are given as:

$$(x_0, y_0, z_0), (x_1, y_1, z_1), (\alpha_0, \beta_0, \gamma_0), (\alpha_1, \beta_1, \gamma_1)$$

The specific steps of the embedding watermarking algorithm are as follows:

Step1. Perform operations on the original image according to a two level DWT, and four parts are obtained, i.e. LL_2, LH_2, HL_2, HH_2 . Embedding operation is performed on this four parts.

Step2. The chaotic system can produce two chaotic sequences by input the key, and the watermarking

will be scrambled and encrypted. It is defined as U .

Step3. The encrypted binary watermarking is embedded into the original image according to the formula below:

$$I_2'(i, j) = I_2(i, j) + \alpha U(i, j) \quad (8)$$

where α represents visibility factor, its value is 0.05 for proposed scheme, $I_2(i, j)$ represents the second level wavelet coefficient. Embedding computing in four parts is all like this, i.e. LL_2, LH_2, HL_2, HH_2 .

Step4. Perform operations on each part according to a two level IDWT of $I_2'(i, j)$, the watermarked image for each part $I_2''(i, j)$ is obtained.

Step5. Combine four parts to get the watermarked image.

The flow chart of embedding watermarking is shown in Fig. 2

3.2 Extracting watermarking

The process of extracting watermarking is the reversed order of the embedding procedure. It can be briefly introduced as follows:

Step1. Perform operations on the watermarked image according to a two level DWT and extract all the parts.

Step2. Perform operations on the original image according to a two level DWT.

Step3. With the help of the chaotic system, chaotic sequences will be generated.

Step4. Extract wavelet coefficients of the embedded watermarking, All four parts are calculated according to the formula below:

$$U'(i, j) = (I_2''(i, j) - I_2(i, j)) / \alpha \quad (9)$$

Step5. Use chaotic sequence to decrypt the encrypted watermarking

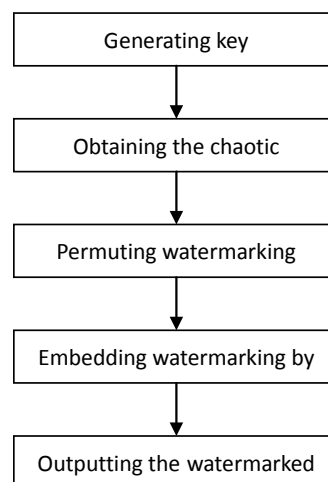


Figure.2 The flowchart of embedding watermarking

4. Experimental Results and Security Analysis

This section presents the experimental results and security analysis of the proposed algorithm. Firstly, the experimental results are given and the embedding efficiency is calculated, and the results of the proposed scheme under various different attacks are also given. Then the test results of encryption security for proposed scheme are given, such as the grey histogram, the space of key, key sensitivity.

4.1 Experimental results

Watermarking scheme usually need to satisfy some properties, such as “embedding efficiency” and “attacks” . The experimental results of these properties are as follows.

1) Experimental results

In this section, the standard Lena image with 256×256 is used as host image and binary logo with 64×64 is used as watermark image. Initial conditions are set as:

$$(x_0, y_0, z_0) = (1, 3, 4) , (x_1, y_1, z_1) = (2, 7, 5) , (\alpha_0, \beta_0, \gamma_0) = (0.97, 0.98, 0.99) ,$$

$$(\alpha_1, \beta_1, \gamma_1) = (0.97, 0.98, 0.99) .$$

The results of watermarking embedding and extraction are obtained as shown in Fig 3.

The embedding of watermarking can be seen as effective if raw data and processed data cannot be distinguished. In order to show the effect of the proposed scheme more directly, the peak signal-to-noise ratio (PSNR) was used to evaluate the image quality, the calculation formula is as follows:

$$PSNR = 10 \times \log_{10} \frac{255^2}{MSE} (dB) \quad (10)$$

The mean squared error (MSE) between the original image and watermarked image can be defined as:

$$MSE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I(i, j) - I_2(i, j)]^2 \quad (11)$$

where $I(i, j)$ and $I_2(i, j)$ represent the pixel values on the location (i, j) , while the image size is $M \times N$.

In this study, the bit error rate (BER) of extracted watermarking is used to test reliability, the calculation formula is as follows:

$$BER = \frac{B}{M \times N} \times 100 \quad (12)$$

where B represents the number of erroneously detected bits, and the size of extracted watermark image is $M \times N$.

The PSNR value of the watermarked image is 41.33 dB, and the BER value of the extracted watermarking is zero. Therefore, there is almost no obvious perceptual distortion between original image and watermarked image; the process of embedding watermarking does not affect the quality of image.

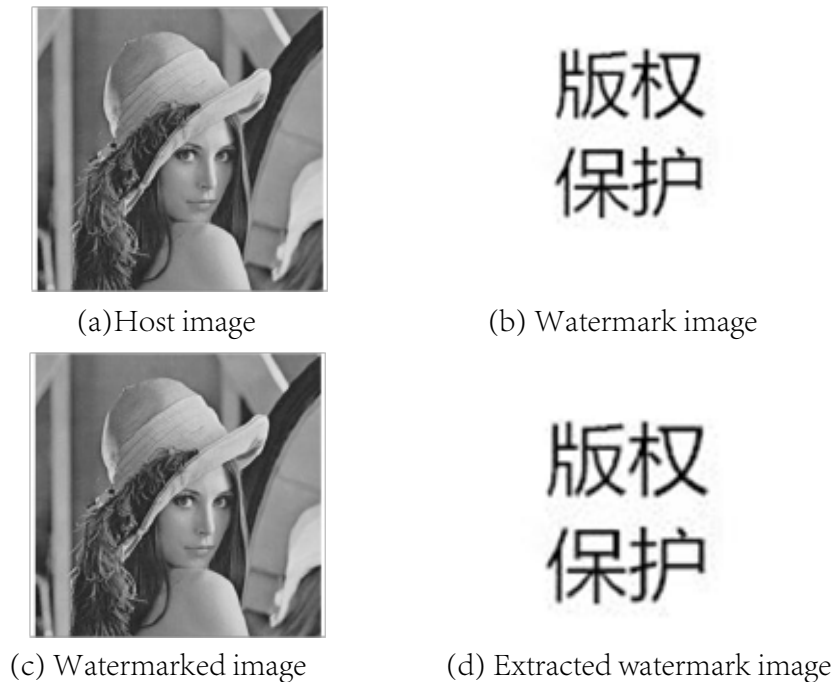


Figure.3 Experimental results

2) Attacks

In order to test the robustness, several different attacks are given. Common signal processing attacks include:

(1) **JPEG compression:** JPEG is the abbreviation of Joint Photographic Experts Group, JPEG image compression algorithm can provide good compression performance, and has a good reconstruction quality, it is widely used in the field of image and video processing [19, 20]. The compression ratio of the proposed scheme is 50:1.

(2) **Filtering:** Filtering can filter out the specific band frequency of the signal, it is an important measure to restrain and prevent interference.

(3) **Noise addition:** The probability density function of Gaussian noise obeys Gauss distribution (i.e. normal distribution). If the amplitude distribution of a noise obeys Gauss distribution, and its power spectrum density is uniformly distributed, it is called Gaussian white noise^[21, 22, 23]. The proposed scheme adds the Gaussian noise, whose mean value is 0 and the variance is 0.01.

(4) **Histogram equalization:** Histogram equalization is a method to adjust the contrast in the field of image processing using image histogram^[21, 22, 23].

(5) **Contrast adjustment:** Contrast of the watermarked image is improved by 50%.

(6) **Gamma correction:** Gamma correction can edit the gamma curve of image, recognize the dark part and light part of the image signal, and increase the proportion of the image. The gamma value of the proposed theme is reduced to 0.6.

The test results for watermarked image are given in Table 1. It can be clearly displayed from the Table 1 that the proposed scheme performs better.

Table 1 Comparison result of PSNR values between proposed scheme and previous work

Attacks	PSNR[dB]	
	Proposed scheme	Rawat ea al[7]
Gaussian Noise	19.69	13.37
Contrast Enhancement	21.51	19.008
Average Filtering	37.24	29.26
Median Filtering	32.05	31.03
Gamma Correction	17.25	15.43
Histogram Equalization	25.71	19.4
JPEG(Q=50)	39.26	34.96

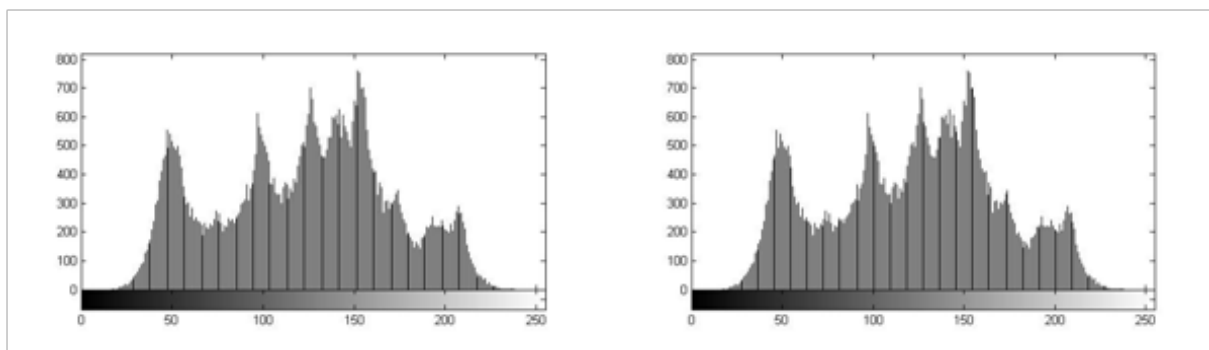
4.2 Security analysis and simulation for proposed scheme

1) The grey histogram analysis

The histogram reflects the basic statistical characteristics of the image. Compare the grey histograms of the original image and the watermarked image, the statistical performance is analyzed. Fig.4 shows the histograms of the original image and the watermarked image. From the figure, it is clear that two histograms are almost the same. The real information of the watermark image has been well hidden, it is not easy to use statistical characteristics to attack the watermarked image. So, the proposed algorithm can well resist statistical attack.

2) The space of key

For an encryption scheme, key space should be enough large to resist brute-force attack. In proposed scheme, the initial key consists of many elements. So the key of the system can be set $(\alpha_0, \gamma_0, a, b)$, each key parameter is independent of each other. In practical design, a and b are impossible to be infinitely large, their range can be set $0 < a, b < 100$. According to the precision of the double precision floating point of the computer, the scheme takes 8 bytes and 15 effective numbers to analyze the data. So the key space is equivalent to $10^{14} \times 10^{14} \times 10^{15} \times 10^{15} = 10^{58}$, which is able to resist the brute-force attack.



(a) The histogram of the original image

(b) The histogram of the watermarked image

Figure.4 The grey histograms

3) Key sensitivity

A good encryption scheme needs not only a large key space, but also it must be sensitive to the key parameters. Only in this way can it be able to resist the differential attack. In the test, the key used in the scheme is (0.97, 0.99, 35, 3). In order to test the sensitivity of the algorithm to the key, some error keys are used to extract the watermark image. As can be clearly seen from Fig.5, the proposed scheme is very sensitive to the key parameters, even if a single key parameter is only 0.001 of the deviation, it will lead to a completely different extraction result.



(a) Extracted watermark by the correct key: (0.97,0.99,35,3) (b) Extracted watermark by the wrong key: (0.971,0.99,35,3) (c) Extracted watermark by the wrong key: (0.97,0.99,35,3.001)

Figure.5 Extraction result

5. Conclusion

Through research, a new image watermarking scheme based on the fractional order Chen chaotic system and discrete wavelet transform is proposed. The fractional order Chen chaotic system is used to increase the overall complexity of the algorithm. Chaotic system is used to deal with the digital watermarking, and the watermarking information is embedded into the original image which is processed by discrete wavelet transform. By analyzing and comparing the experimental results show that the proposed watermarking scheme has high security and stronger robustness and invisibility. All these characteristics demonstrate that the proposed scheme is in favor of image watermarking encryption.

Acknowledgment

This work was supported by National Nature Science Foundation of China (No: 61201227).

References

- [1] S. Poonkuntran, R.S. Rajesh. "Chaotic model based semi fragile watermarking using integer transforms for digital fundus image authentication", *Multimedia Tools & Applications*, vol. 68, no.1, pp. 79-93, 2014.
- [2] X.J.Tong, Y.Liu, M.Zhang, et al. "A novel chaos-based fragile watermarking for image tampering detection and self-recovery", *Signal Process Image Commun*, vol. 28, no.3, pp. 301-308, 2013.
- [3] S.Behnia, M.Teshnehlal, P.Ayubi. "Multiple-watermarking scheme based on improved chaotic maps", *Communications in Nonlinear Science & Numerical Simulation*, vol. 15, no.9, pp. 2469-2478, 2010.
- [4] T.G.Gao, Q.L.Gu, S.Emmanuel. "A novel image authentication scheme based on hyper-chaotic cell neural network", *Chaos Solitons&Fractals*, vol. 42, no.1, pp. 548-553, 2009.
- [5] A.Mooney, J.G.Keating, I.Pitas. "A comparative study of chaotic and white noise signals in digital watermarking", *Chaos Solitons&Fractals*, vol. 35, no.5, pp. 913-921, 2008.
- [6] G.Y.Gao, G.P.Jiang. "Zero-bit watermarking resisting geometric attacks based on composite-chaos optimized SVR model", *The Journal of China Universities of Posts and Telecommunications*, vol. 18, no.2, pp. 94-101, 2011.

- [7] S.Rawat, B.Raman. “A publicly verifiable lossless watermarking scheme for copyright protection and ownership assertion”, *AEU-International Journal of Electronics and Communications*, vol. 66 ,no.11,pp. 955-962,2012.
- [8] M.R.Keyvanpour, F.M.Bayat. “Blind image watermarking method based on chaotic key and dynamic coefficient quantization in the DWT domain”, *Mathematical&Computer Modelling*, vol. 58,pp. 56-67,2013.
- [9] D.Y.Zhao, J.P.Chen, J.C.Sun. “Design and implementation of improved watermarking system in WT domain”, *The Journal of China Universities of Posts and Telecommunications*, vol. 14,no.2,pp. 58-63,2007.
- [10] M.Barni, F.Bartolini, A.Piva. “Improved wavelet-based watermarking through pixel-wise masking”, *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, vol. 10,no.5,pp. 783-791,2001.
- [11] B.Y.Lei, D.Ni, S.P.Chen, et al. “Optimal image watermarking scheme based on chaotic map and quaternion wavelet transform”, *Nonlinear Dynamics*, vol. 78,no.4,pp. 2897-2907,2014.
- [12] O.Benrhouma, H.Hermassi, S.Belghith. “Tamper detection and self-recovery scheme by DWT watermarking”, *Nonlinear Dynamics*, vol. 79,no.3,pp. 1-17,2014.
- [13] J.Song, Z.Zhang. “A Digital Watermark Method Based on SVD in Wavelet Domain”, *International Journal of Advancements in Computing Technology*, vol. 3,no.8,pp. 205-214,2011.
- [14] G.P.Tang, X.F.Liao, Y.Chen. “A novel method for designing S-boxes based on chaotic maps”, *Chaos Solitons&Fractals*, vol. 23,no.2,pp. 413-419,2005.
- [15] C.P.Li, G.J.Peng. “Chaos in Chen’s system with a fractional order”, *Chaos Solitons&Fractals*, vol.22,no.2,pp. 443-450,2004.
- [16] S.M.Kenneth, R.Bertram. “An introduction to the fractional calculus and fractional differential equations”, *Wiley-Interscience*, vol. 65,no.9,pp. 1000-1003,1993.
- [17] J.H.Song, J.W.Song, Y.H.Bao. “A Blind Digital Watermark Method Based on SVD and Chaos”, *Procedia Engineering*, vol. 29,no.29,pp. 285-289,2012.
- [18] T.H.Chen, G.B.Horng, W.B.Lee. “A publicly verifiable copyright-proving scheme resistant to malicious attacks”, *IEEE Transactions on Industrial Electronics*, vol. 52,no.1,pp. 327-334,2005.
- [19] W.B.Pennebaker, J.L.Mitchell. *JPEG Still Image Data Compression Standard*, New York: Van Nostrand Reinhold, 1993.
- [20] T.Acharya, P.S.Tsai. *JPEG2000 Standard for Image Compression: Concepts, Algorithms and VLSI Architectures*,New York: John Wiley & Sons, 2004.
- [21] R.C.Gonzalez, R.E.Woods. *Digital Image Processing*, New York: Addison-Wesley Longman Publishing Co., Inc., 2001.
- [22] W.K.Pratt. *Digital Image Processing*,New York: Wiley & Sons, 1991.
- [23] A.Rosenfeld A, A.C.Kak. *Digital Picture Processing*,Cambridge, Massachusetts: Academic Press,1982.

Author Brief and Sponsors:

Dawei Ding, he is an associate professor with School of Electronics and Information Engineering at Anhui University, Hefei, China. His research area include communications networks, the nonlinear circuit network, the network congestion control, non- linear dynamics and chaos, bifurcation, etc..

The Design of Two Phase Chopping Regulation Voltage Soft Starter

Jingwen Chen*, Hongshe Dang

School of Electrical and Information Engineering, Shaanxi University of Science & Technology, Xi'an, 710021, China

***Address correspondence to this author at xuefu Road, Xi'an, China. postcode:710021
E-mail:15991663575@163.com**

Abstract. As for the complexity of the trigger pulse control method for the generally-used thyristor soft starter, The poverty of the continuity of the stator current, the shortage of the waveform distortion and the higher harmonic content. A type with IGBT to realize AC chopping regulation voltage soft starter has been designed, using the two-phase AC power control to achieve the aim of three-phase AC power control, ensuring the effect and saving the costs. The paper makes use of the MATLAB to analyze of the two phase chopping regulating all control type soft starter start performance with simulation, and designs the soft starter of the hardware circuit and the software programs.

Keywords: IGBT, Two phase chopping, Asynchronous motor, Soft starter

1. Introductiong

In the application of the motor, the starting of the motor problem is particularly important. When the asynchronous motor starts, transient current shock is large, generally up to 4~8 times than the motor rated current, and even larger. Too large starting impact current will cause adverse effect on the motor itself, the normal operation of the other electrical equipment and power grid.

Because of the half control of thyristor, the discontinuous current, currently used thyristor voltage regulation of the soft starter will produce a lot of low harmonic, causing harmonic pollution, making the output voltage waveform distort serious and influencing the dynamic characteristics of the motor^[1-3]. Because of using all control devices, the two phase chopping regulating all control type soft starter can be shut off by self and can achieve the stator voltage stepless regulation by changing duty ratios of the trigger pulse. Its advantage is that IGBT trigger pulse does not need to link the phase of the three-phase BUS voltage, and has no need to test the zero crossing point. It regulates voltage by adjusting the duty ratios of trigger pulse without calculating the firing angle; the control algorithm and the implementation method are relatively simple. In the use of the freewheeling circuit, The motor current and voltage waveform become much closer to the sine wave, and the waveform distortion rate becomes low^[4-5].

2. The main circuit of two phase chopping regulating soft starter

The two phase chopping regulating all control type soft starter is by controlling the duty ratio of all control devices IGBT trigger pulse to adjust the input voltage of stator from power grid, so as to adjust the size of the starting electromagnetic torque. Specifically it's adjusting the three-phase output voltage size by controlling the output voltage of the two phase. The main circuit topology structure is shown in

Fig.1 The specific voltage regulation methods are as follows:

The main circuit structure adopts four power diodes, an insulated gate bipolar transistor and its corresponding protection circuit, to form two groups of basic single-phase ac voltage regulation circuit (1), (2), respectively concatenated in the power supply phase A and phase B of the three-phase ac asynchronous motor. In addition, it adopts six power diodes, an insulated gate bipolar transistor and its corresponding protection circuit, to form a three-phase freewheel bridge (3). The three-phase freewheel bridge connects ac voltage regulation circuit to three-phase ac asynchronous motor stator winding. In order to detect and control the starting current, set up current transformer (12), (13), (14) to A, B, C phase power line of the three-phase ac asynchronous motor respectively. Then the current signal tested by the current transformer will be send to the microprocessor control system (5).

On the main circuit control, the control pulse generated by the microprocessor control system has two ways, one triggers the insulated gate bipolar transistor (8), (7) of two sets of ac voltage regulation (1), (2) shown in diagram, the other triggers the insulated gate bipolar transistor (6) of the three-phase freewheel bridge dc side. The positive and negative of these two pulses are exactly opposite, and they are complementary pulse. Therefore, when (8), (7) trigger on, (6) shuts off, then three-phase ac power supplies to three-phase ac asynchronous motor; when the (8), (7) shut off, (6) triggers on, then three-phase ac asynchronous motor stator current follow current by three-phase freewheel bridge.

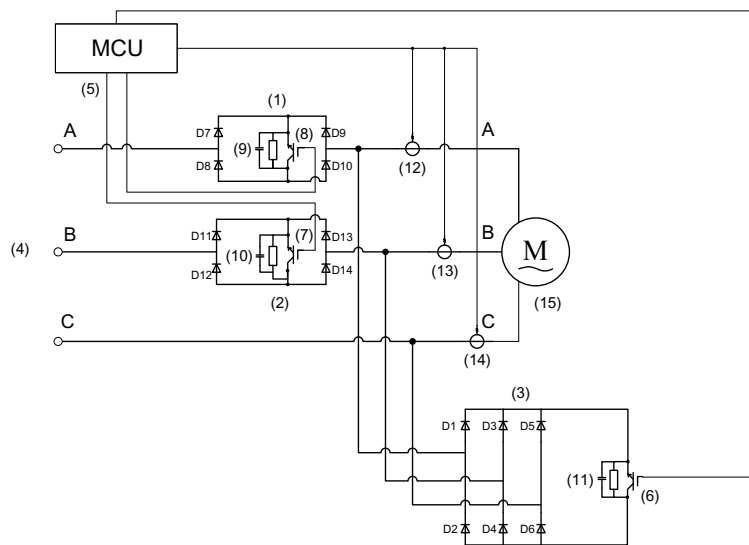


Figure.1 The main circuit structure of two phase chopping regulating soft starter

3. Start modes

The soft starter has a variety of start modes: voltage ramp start, voltage ramp start with pulse kick, current-limiting start, dual ramp start, etc. The maximum allowable time during start process can be set as starting time parameter, and starting time limit range is: 10-120 s, the default value is 60s. The start-up way is as follows:

For voltage ramp start, the user can set up an initial starting torque, and this initial starting torque corresponds to an initial starting voltage. The voltage of the motor increases at a fixed growth rate from the initial starting voltage by a constant speed, until up to the rated voltage. And the growth rate of voltage can be adjusted to adapt to different starting times.

Voltage ramp start with pulse kick is the mode that gives the motor with a higher starting voltage for a short period of time in the motor starting moment, making the motor produce a large instantaneous torque to overcome the static friction torque under the static state, and making the motor turn up. Its main application is on overloading starting motor.

Current-limiting start provides the motor decompression start with a fixed current, used to limit the maximum starting current, which is mainly used in the occasion of the need to limit the impact current when starting^[6-8]. Current-limiting start needs to cooperate with the effective value of current detection circuit. The program contains the PID current limit control algorithm, fuzzy control algorithm or combining fuzzy and PID control algorithm, etc.^[4], which makes the motor current won't exceed the maximum value during the process of start, as the voltage ramp starts without a maximum value range, providing a kick torque start. The current-limiting start current limit value can be set in 2 ~ 4 times of the rated current.

Dual ramp start, similar to voltage ramp start, corresponds to two different rates of voltage rise. These two start solutions are set in advance. According to their practical application, users could store two commonest start modes in the soft starter, in order to set the starting parameters quickly to start motor. Slope 1 starts from 50% of the full voltage, lasts 100s, and linear increases to full voltage. Slope 2 starts from 70% of the full voltage, lasts 120s, and linear increases to full voltage. The motor can also firstly start by the slope 2. The initial torque is higher, but the voltage rise slowly, which makes the current of the motor change not severe. When the motor speeds up, the current will be reduced, making the motor rise to full voltage rapidly, and completing the start.

4. The Simulation Analysis of Two Phase Chopping Regulating All Control Type Soft Start

Based on the main circuit of two phase chopping regulating all control type soft starter shown in the figure.1, the simulation model shown in Fig.2 is built. Simulation parameter settings are as follows: IGBT switching frequency is 1KHz, trigger pulse duty ratio 0.5, three-phase ac power source 380V, 50 Hz, asynchronous motor rated power 15kw, rated voltage 400V, rated frequency 50Hz, and the load torque 85N · m.

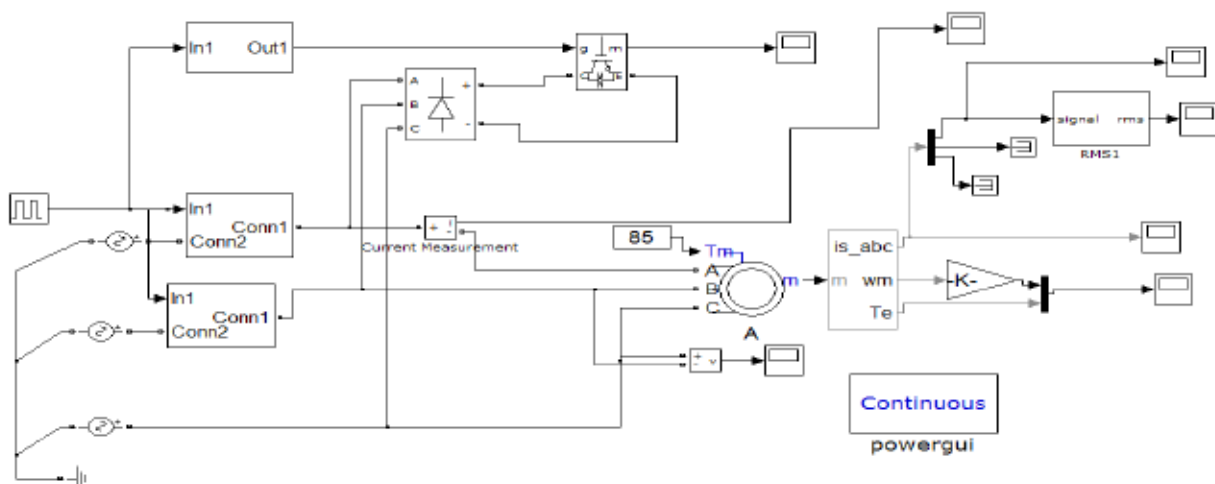


Figure.2 The main circuit model of two phase chopping regulating soft starter

The stator current simulation result of two phase chopping regulating all control type soft starter is shown in Fig.3, it can be shown from the diagram that the stator current fluctuation is significantly decreased than that of thyristor regulating soft start, and the biggest stator current is about 2 times than that of stable operation when starts^[9-10].

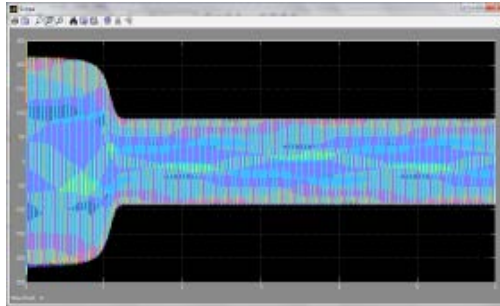


Figure.3 The stator current simulation result of two phase chopping regulating soft starter

Fig.4 is the speed and torque simulation results of two phase chopping regulating all control type soft starter. It can be reflected: the revolving speed of two phase chopping regulating all control type soft starter rises faster, and the rising process is smoother. The time to stabilization is also ahead of thyristor regulating soft starter. The starting torque of two phase chopping regulating all control type soft starter is smoother, there's no peak impact torque.

Fig.5 is the stator current local amplification figure of two phase chopping regulating all control type soft starter. It can indicated from the picture that as IGBT works under the high switch frequency, the stator current waveform is very close to the sine wave, and the stator current is continuous. Discontinuous phenomenon hardly appears for a while near zero.

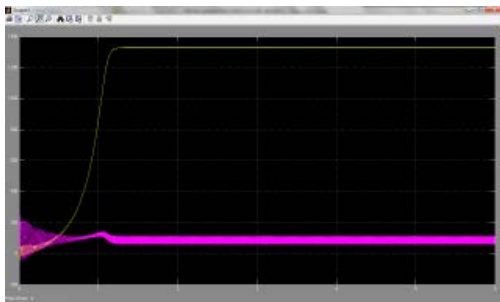


Figure.4 The speed and torque simulation results of two phase chopping regulating soft starter

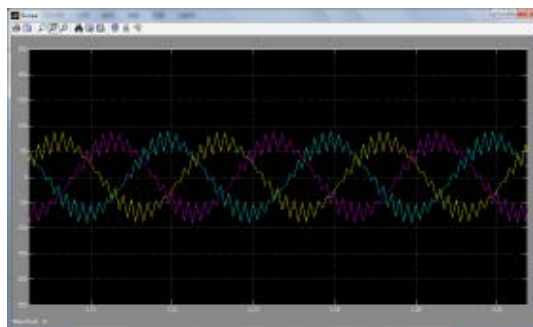


Figure.5 The stator current local amplification figure of two phase chopping regulating soft starter

Fig.6 is the FFT analysis of two phase chopping regulating soft starter A phase stator current simulation results. It can be presented that the main harmonic frequency is 19, 21, 59, 61, and the total harmonic factor is only 6.15%, compared with thyristor regulating soft start. The total harmonic factor has fallen by half. The harmonic content of 19 and 21 is higher, and they belong to higher harmonic, easy to be filtered through the filter circuit.

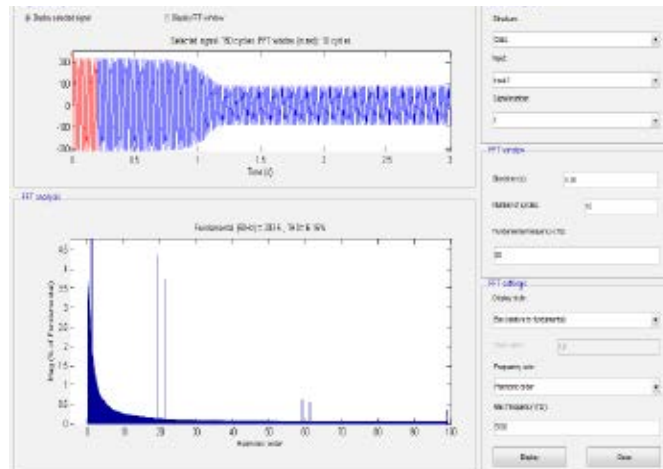


Figure.6 the FFT analysis of two phase chopping regulating soft starter A phase stator current

Through the analysis of simulation results, it confirms that the two phase chopping regulating all control type soft starter has the advantages: small starting current, smooth starting torque, low total harmonic content, and low harmonic content, compared with thyristor regulating soft starter.

5. The Hardware Circuit Design of Two Phase Chopping Regulating All Control Type Soft Starter

The hardware schematic diagram of two phase chopping regulating soft starter is shown in Fig.7. It mainly includes several main parts: the main control chip STM32^[5], communication circuit, current detection circuit, USB to serial communication circuit, the CPLD circuit, voltage detection circuit, drive circuit, power circuit. Every subcircuit needs to be connected with the corresponding power supply circuit, without showing the specific connection in the picture.

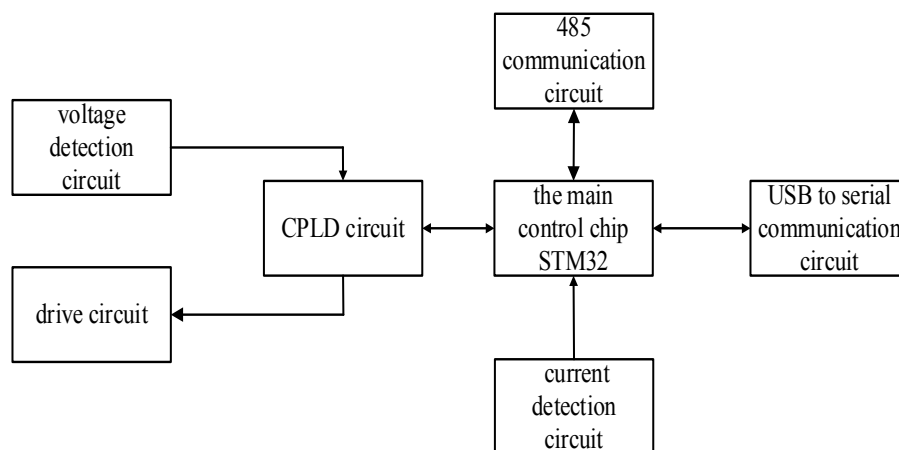


Figure.7 The schematic diagram of two phase chopping regulating soft starter

6. The Software Design of Two Phase Chopping Regulating All Control Type Soft Start System

6.1 The design of main program

As the central control system, two phase chopping regulating soft start main program, has command and deployment effects to the behavior of each program of the control system. The main program flow chart of control system is shown in Fig.8. Simple its structure is, it is the soul of the whole control system, and has irreplaceable importance. The detailed working process is displayed below.

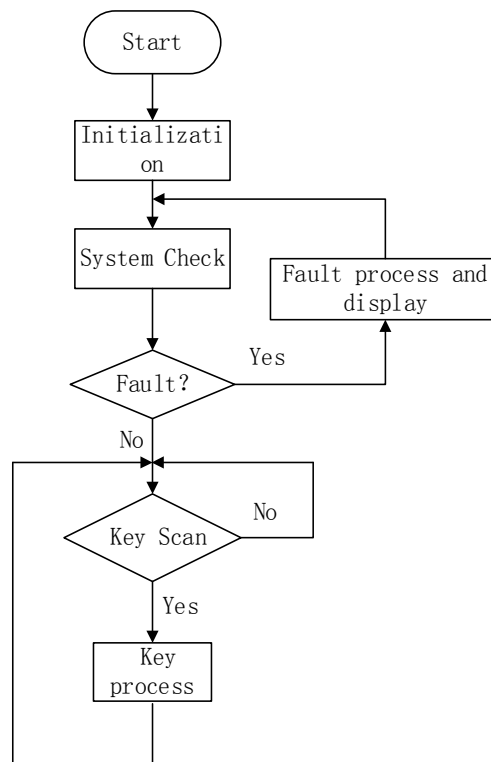


Figure.8 The main program flow chart of control system

First of all, the control system must initialize the main control chip internal various registers and each external pin operation after power-on. Then run a series of system detection subroutines by reading the corresponding parameters of pin, to judge whether the current environment of the motor could start. The operation of the main subroutines on the stage includes phase sequence detection, lack of phase detection, temperature detection. If the detect results conform to the motor starting, it will enter to scan key link. This link uses to monitor whether the control signals are input by the human through the keyboard. If so, the process enters the key processing steps, then the relevant subroutines process the control of the input signal, and wait for starting. If the output of detection subroutines does not conform to the motor starting, the system will enter the fault process which will be displayed. After the fault processed, test whether it conforms to the motor starting conditions.

6.2 The design of main program

Current-limiting starting mode is a more detailed starting mode, which highlights the current size

control in the process of motor starting. It needs to feedback the real-time current to the main control system in the process of motor starting, then by the comparing the value of the real-time current with the set current, the main control system generates control signal, adjusts the IGBT trigger pulse duty ratio, and then makes the motor starting current near the set value within a range. This way of starting is relatively complex.

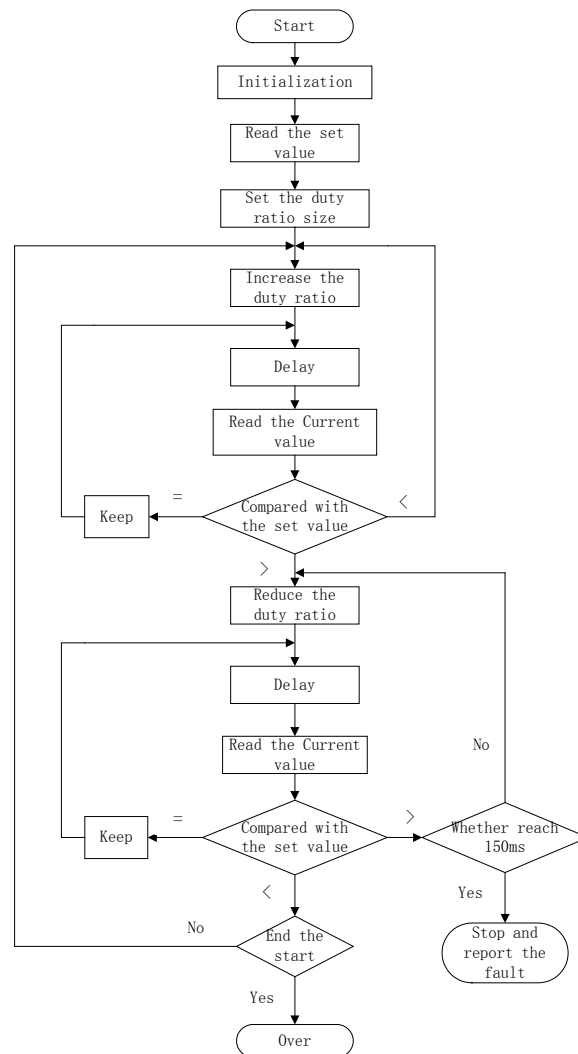


Figure.9 Flow chart of current limiting soft start

The biggest starting current of the current-limiting starting is generally no more than four times of the electric motor rated current. Usually the current limit control algorithm has PID method, fuzzy control method, the combination of PID and fuzzy control method, and so on. This article uses the fuzzy control method. Flow chart of current limiting soft start is shown in Fig.9.

7. Conclusion

This paper proposes the two phase chopping regulating all control type soft starter. Its starting current, starting torque, total harmonic content and the stator current waveform, are all superior to the thyristor voltage regulating soft starter. The paper proposed the hardware circuit and the software program of the two phase chopping regulating all control type soft starter. The harmonic content of the two phase chopping regulating all control type soft starter is lower. Compared with thyristor regulating soft starter, the starting current is closer to the sine wave, and with good continuity, Control algorithm is simple, the protection circuit is complete, the system has high stability and easy maintenance, and Large starting

torque is continuous adjustable. With use less of power electronic devices, the equipment controls the two phase to achieve the purpose of control three phase, saving costs. It's a method of ac asynchronous motor soft start worth promoting.

Acknowledgment

The authors wish to thank Shaanxi provincial government. This work was supported in part by Industrial research project of Science and Technology Department of Shaanxi Province. Fund number:2015GY074.

Reference

- [1] Hu Hongming, "The study of asynchronous motor soft start", master thesis, Huazhong university of science and technology, wuhan, china. June 1, 2010.
- [2] Sun Jinji, Fang Jiancheng, Wang Jianmin, "The shock in the process of asynchronous motor soft start", Transactions of China Electrotechnical Society, vol. 22, pp. 15-21, February 2007.
- [3] Fan Liping, Zhang Liang, "Fuzzy simulation of asynchronous motor soft start", Transactions of Power System and Automation, vol. 23, pp. 123-126, March 2011.
- [4] Fan Liping, Hu Wenhao, "The study of combination of filtering function asynchronous motor soft start", Electric Drive, vol. 41, pp. 61-64, September 2011.
- [5] Meng Yanjing, Gong Wenzhan, "The hardware design of the intelligent soft starter control system", Small & Special Electrical Machines, vol. 40, pp. 64-72, January 2012.
- [6] Shicheng Zheng, Hong Zhu, Xiaohu Cao, Weihua Wu, Qianzhi Zhou, "Research And Implementation Of A Novel Single Phase Ac/Ac Variable Frequency Technology", High Voltage Engineering, vol. 10, pp. 139-143, March 2009.
- [7] G.Zenginobuz, I.Cadirci, M.Ermis, C.Barlak, "Soft Starting Of Large Induction Motors At Constant Current With Minimized Starting Torque Pulsations", IEEE Industry Applications Conference, vol.3, pp.1593-1604, March 2000.
- [8] Xiuhua Zhang, Lin He, Guangxi Li, Zhou Zhou, "The analysis of structure strength and calculation of critical speed of a new type of high-speed permanent magnet motor", Energy Education Science and Technology Part A, vol. 32, pp. 143-148, January 2014.
- [9] Yunrui Wang, "Study on electric furnace power quality integrated control system", Energy Education Science and Technology Part A, vol. 32, pp. 895-904, February 2014.
- [10] Zhongxian Wang, Yonggeng Wei, Chunyan Li, "Study on APFC circuit in the single-phase variable frequency power supply", Energy Education Science and Technology Part A, vol. 32, pp.355-356, January 2014.

Development of Levenberg-Marquardt Method Based Iteration Square Root Cubature Kalman Filter and its applications

Jing Mu * , Changyuan Wang

School of Computer Science and Engineering

Xi'an Technological University, Xi'an, 710032, China;

* Corresponding author, Email: mujing1977@163.com;

Abstract. Levenberg-Marquardt (abbr. L-M) method based iterative square root cubature Kalman filter (ISRCKFLM) is proposed to improve the low state estimation accuracy of nonlinear state estimation due to large initial estimation error and nonlinearity of measurement equation. The measurement update of square root cubature Kalman filter (SRCKF) is transformed to the problem of nonlinear least square error, then we use L-M method to solve it and obtain the optimal state estimation and covariance, so the ISRCKFLM algorithm has the virtues of global convergence and numerical stability. We apply the ISRCKFLM algorithm to state estimation for re-entry ballistic target; the simulation results demonstrate the ISRCKFLM algorithm has better accuracy of state estimation.

Keywords: Nonlinear filtering, Cubature Kalman filter, Levenberg-Marquardt method

1. Introduction

A series of nonlinear filters have been developed to apply to state estimation for the last decades. Up to now the commonly used non-linear filtering is the extended Kalman filter (EKF) ^[1,2]. The EKF is based on first-order Taylor approximations of state transition and observation equation about the estimated state trajectory under Gaussian assumption, so EKF may introduce significant bias, or even convergence problems due to the overly crude approximation ^[3].

Recently, one type of suboptimal nonlinear filters based on numerical multi-dimensional integral were introduced in ^[4-6], such as cubature rules based cubature Kalman filter (CKF) and the interpolatory cubature Kalman filters (ICKFs), which used numerical multi-dimensional integral to approximate the recursive Bayesian estimation integrals under the Gaussian assumption. The CKF can solve high-dimensional nonlinear filtering problems with minimal computational effort and can be deemed as special case of ICKFs. Furthermore, the stability of CKF for non-linear systems with linear measurement is analyzed and the certain conditions to ensure that the estimation error of the CKF remains bounded are proved in ^[7]. On the other hand, in order to decrease the effect of initial estimation error and nonlinearity of measurement equation, Levenberg-Marquardt method based iteration cubature Kalman filter was developed on the basis of the CKF in Reference ^[8]. In fact, singular matrix occurs in the implementation of the above filters mentioned if the initial estimation is selected improperly. So the cubature rule is exploited as square root cubature information filter ^[9] and the square root cubature Kalman filter (SRCKF) was developed in order to mitigate ill effects and improve the numerical stability ^[5].

However, because of the large initial error and measurement error in the state estimation for re-entry ballistic target with unknown ballistic coefficient, which is the nonlinear dynamics system with the feature of hidden markov model, the SRCKF also shows its weakness in the robustness and estimation accuracy. As we know Levenberg-Marquardt (abbr. L-M) method has the global convergence and fast coverage. Making use of L-M method and the superiority of the SRCKF algorithm, we develop the L-M method based iterative square root cubature Kalman filter (ISRCKFLM), in which, we transform the measurement update of SRCKF to the problem of nonlinear least square error, then use L-M method to solve it and obtain the optimal state estimation and covariance to improve the low state estimation accuracy of nonlinear state estimation due to large initial estimation error and nonlinearity of measurement equation.

The rest of the paper is organized as follows. We begin in Section 2 with a description of square root cubature Kalman filter (SRCKF). The L-M method based iterative square root cubature Kalman filter (ISRCKFLM) is developed in Section 3. Then we apply the ISRCKFLM algorithm to track re-entry ballistic target (RBT) with unknown ballistic coefficient and discuss the simulation results in Section 4. Finally, Section 5 concludes the paper.

2. Square Root Cubature Kalman Filter

Consider the following nonlinear dynamics system:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1} \quad (1)$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \quad (2)$$

Where \mathbf{f} and \mathbf{h} are some known nonlinear functions; $\mathbf{x}_k \in \mathbb{R}^{n_x}$ and $\mathbf{z}_k \in \mathbb{R}^{n_z}$ is state and the measurement vector, respectively; \mathbf{w}_{k-1} and \mathbf{v}_k are process and measurement Gaussian noise sequences with zero means and covariance \mathbf{Q}_{k-1} and \mathbf{R}_k , respectively, and $\{\mathbf{w}_{k-1}\}$ and $\{\mathbf{v}_k\}$ are mutually uncorrelated.

Suppose that the state distribution is $\mathbf{x}_{k-1} \sim \mathcal{N}(\hat{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1})$, and a square root \mathbf{S}_{k-1} of \mathbf{P}_{k-1} such that $\mathbf{P}_{k-1} = \mathbf{S}_{k-1} \mathbf{S}_{k-1}^T$ is obtained. The square root cubature Kalman filter (SRCKF) algorithm is summarized as follows.

Step 1. Time Update

1) Calculate the cubature points and propagate the cubature points through the state equation

$$\mathbf{X}_{i,k-1} = \mathbf{S}_{k-1} \boldsymbol{\xi}_i + \hat{\mathbf{x}}_{k-1} \quad (3)$$

$$\mathbf{X}_{i,k}^* = \mathbf{f}(\mathbf{X}_{i,k-1}) \quad (4)$$

where $\hat{\mathbf{i}}_i = \sqrt{m/2} [1]_i$, $\omega_i = 1/m$, $i = 1, \dots, m = 2n_x$, the $[1]_i$ is a n_x dimensional vector and is generated according to the way described in [5].

2) Evaluate the predicted state and square root of the predicted covariance

$$\bar{\mathbf{x}}_k = \sum_{i=1}^m \omega_i \mathbf{X}_{i,k}^* \quad (5)$$

$$\bar{\mathbf{S}}_k = \text{Tria}([\chi_k^* \mathbf{S}_{Q,k-1}]) \quad (6)$$

here, $\mathbf{S}_{Q,k-1}$ denotes a square-root factor of \mathbf{Q}_{k-1} and $\text{Tria}()$ is denoted as a general triangularization algorithm. The matrix χ_k^* is defined as:

$$\chi_k^* = 1/\sqrt{m} [\mathbf{X}_{1,k}^* - \bar{\mathbf{x}}_k \quad \mathbf{X}_{2,k}^* - \bar{\mathbf{x}}_k, \dots, \mathbf{X}_{m,k}^* - \bar{\mathbf{x}}_k] \quad (7)$$

Step 2. Measurement Update

1) Calculate the predicted cubature points and evaluate the propagated cubature points

$$\mathbf{X}_{i,k} = \bar{\mathbf{S}}_k \xi_i + \bar{\mathbf{x}}_k \quad (8)$$

$$\mathbf{Z}_{i,k} = \mathbf{h}(\mathbf{X}_{i,k}) \quad (9)$$

2) Evaluate the predicted measurement, a square root of the innovation covariance and cross-covariance

$$\bar{\mathbf{z}}_k = \sum_{i=1}^m \omega_i \mathbf{Z}_{i,k} \quad (10)$$

$$\mathbf{S}_{zz,k} = \text{Tria}([\Upsilon_k \mathbf{S}_{R,k}]) \quad (11)$$

$$\mathbf{P}_{xz,k|k-1} = \chi_k^T \Upsilon_k^T \quad (12)$$

here

$$\Upsilon_k = 1/\sqrt{m} [\mathbf{Z}_{1,k} - \bar{\mathbf{z}}_k \quad \mathbf{Z}_{2,k} - \bar{\mathbf{z}}_k, \dots, \mathbf{Z}_{m,k} - \bar{\mathbf{z}}_k] \quad (13)$$

$$\chi_k = 1/\sqrt{m} [\mathbf{X}_{1,k} - \bar{\mathbf{x}}_k \quad \mathbf{X}_{2,k} - \bar{\mathbf{x}}_k, \dots, \mathbf{X}_{m,k} - \bar{\mathbf{x}}_k] \quad (14)$$

where $\mathbf{S}_{R,k}$ denotes a square root factor of \mathbf{R}_k .

3) Evaluate the state estimation and the square-root of the covariance at instant time.

$$\mathbf{W}_k = (\mathbf{P}_{xz,k} / \mathbf{S}_{zz,k}^T) / \mathbf{S}_{zz,k} \quad (15)$$

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \mathbf{W}_k (\mathbf{z}_k - \bar{\mathbf{z}}_k) \quad (16)$$

$$\mathbf{S}_k = \text{Tria}([\chi_k - \mathbf{W}_k \Upsilon_k \quad \mathbf{W}_k \mathbf{S}_{R,k}]) \quad (17)$$

where symbol “/” represents the matrix right divide operator.

3. Development of L-M Based Iterative Square Root Cubature Kalman Filter

3.1 L-M method based iterative measurement update

In the time update of the SRCKF algorithm, we get the predicted state $\bar{\mathbf{x}}_k$ and a square root of corresponding covariance $\bar{\mathbf{S}}_k$ [5], and we can obtain $\bar{\mathbf{P}}_k = \bar{\mathbf{S}}_k \bar{\mathbf{S}}_k^T$. Assuming $\bar{\mathbf{x}}_k \sim \mathcal{N}(\mathbf{x}_k, \bar{\mathbf{P}}_k)$, the current measurement is \mathbf{z}_k , and $\mathbf{z}_k \sim \mathcal{N}(\mathbf{h}(\mathbf{x}_k), \mathbf{R}_k)$. Defining the augmented matrix: $\mathbf{Y}_k = [\bar{\mathbf{x}}_k \ \mathbf{z}_k]^T$, and $\mathbf{H}(\mathbf{x}_k) = [\mathbf{x}_k \ \mathbf{h}(\mathbf{x}_k)]^T$.

Defining the residual function:

$$\boldsymbol{\Psi}(\mathbf{x}_k) = \mathbf{W}_k^{-1} \mathbf{V}_k \quad (18)$$

here

$$\mathbf{W}_k = \begin{bmatrix} \bar{\mathbf{P}}_k^{1/2} & \mathbf{0}_{n_x \times n_z} \\ \mathbf{0}_{n_x \times n_z} & \mathbf{R}_k^{1/2} \end{bmatrix} \quad (19)$$

$$\mathbf{C}_k = \mathbf{W}_k \mathbf{W}_k^T = \begin{bmatrix} \bar{\mathbf{P}}_k & \mathbf{0}_{n_x \times n_z} \\ \mathbf{0}_{n_x \times n_z} & \mathbf{R}_k \end{bmatrix} \quad (20)$$

$$\mathbf{V}_k = \mathbf{Y}_k - \mathbf{H}(\mathbf{x}_k) \quad (21)$$

Defining the cost function:

$$C_{LS}(\mathbf{x}_k) = 0.5 \boldsymbol{\Psi}^T(\mathbf{x}_k) \boldsymbol{\Psi}(\mathbf{x}_k) \quad (22)$$

Defining the matrix: $\tilde{\mathbf{P}}_k^{-1} = [\bar{\mathbf{P}}_k^{-1} + \mu_i \mathbf{I}]$, and using L-M method and manipulations, the iterate formula can be obtained:

$$\hat{\mathbf{x}}_k^{(i+1)} = \bar{\mathbf{x}}_k + \mathbf{L}_k^{(i)} \left\{ \mathbf{z}_k - \mathbf{h}(\mathbf{x}_k^{(i)}) - \mathbf{J}_h(\mathbf{x}_k^{(i)}) (\bar{\mathbf{x}}_k - \mathbf{x}_k^{(i)}) \right\} - \mu_i \left\{ \mathbf{I} - \mathbf{L}_k^{(i)} \mathbf{J}_h(\mathbf{x}_k^{(i)}) \right\} \tilde{\mathbf{P}}_k (\bar{\mathbf{x}}_k - \mathbf{x}_k^{(i)}) \quad (23)$$

where $\mathbf{J}_h(\hat{\mathbf{x}}_k^{(i)}) = \partial \mathbf{h}(\mathbf{x}_k) / \partial \mathbf{x}_k |_{\mathbf{x}_k = \hat{\mathbf{x}}_k^{(i)}}$, \mathbf{I} is identity matrix and μ_i is the tuning parameter, the gain $\mathbf{L}_k^{(i)}$ is defined:

$$\mathbf{L}_k^{(i)} = \tilde{\mathbf{P}}_k \mathbf{J}_h^T(\hat{\mathbf{x}}_k^{(i)}) \left[\mathbf{J}_h(\mathbf{x}_k^{(i)}) \tilde{\mathbf{P}}_k \mathbf{J}_h^T(\mathbf{x}_k^{(i)}) + \mathbf{R}_k \right]^{-1} \quad (24)$$

To obtain the covariance, we derive the equation (3) to get its extremum and get:

$$(\mathbf{x}_k - \hat{\mathbf{x}}_k^{(N)}) = \left\{ \mathbf{J}_V^T(\mathbf{x}_k^{(N)}) \mathbf{C}_k^{-1} \mathbf{J}_V(\mathbf{x}_k^{(N)}) \right\}^{-1} \mathbf{J}_V^T(\mathbf{x}_k^{(N)}) \mathbf{C}_k^{-1} \mathbf{V}_k \quad (25)$$

where $\mathbf{J}_V(\hat{\mathbf{x}}_k) = \partial \mathbf{V}(\mathbf{x}_k) / \partial \mathbf{x}_k |_{\mathbf{x}_k = \hat{\mathbf{x}}_k}$. Using the matrix inversion's lemmas, the covariance \mathbf{P}_k at k time can be obtained:

$$\mathbf{P}_k = [\mathbf{I} - \mathbf{K}_k \mathbf{J}_h(\hat{\mathbf{x}}_k^{(N)})] \bar{\mathbf{P}}_k \quad (26)$$

where the gain \mathbf{K}_k is defined as:

$$\mathbf{K}_k = \bar{\mathbf{P}}_k \mathbf{J}_h^T(\hat{\mathbf{x}}_k^{(N)}) \left\{ \mathbf{J}_h(\mathbf{x}_k^{(N)}) \bar{\mathbf{P}}_k \mathbf{J}_h^T(\mathbf{x}_k^{(N)}) + \mathbf{R}_k \right\}^{-1} \quad (27)$$

Covariance and cross-covariance are defined as:

$$\mathbf{P}_{xz} = \bar{\mathbf{P}}_k \mathbf{J}_h^T(\hat{\mathbf{x}}_k^{(N)}) \quad (28)$$

$$\mathbf{P}_{zz} = \mathbf{J}_h(\hat{\mathbf{x}}_k^{(N)}) \bar{\mathbf{P}}_k \mathbf{J}_h^T(\mathbf{x}_k^{(N)}) + \mathbf{R}_k = \begin{bmatrix} \mathbf{J}_h(\mathbf{x}_k^{(N)}) \bar{\mathbf{S}}_k & \mathbf{S}_{R,k} \end{bmatrix} \begin{bmatrix} \mathbf{J}_h(\mathbf{x}_k^{(N)}) \bar{\mathbf{S}}_k & \mathbf{S}_{R,k} \end{bmatrix}^T \quad (29)$$

Using a series of manipulations, we can obtain a square root \mathbf{S}_k of covariance \mathbf{P}_k :

$$\mathbf{S}_k = \text{Chol}(\begin{bmatrix} \bar{\mathbf{S}}_k - \mathbf{K}_k \mathbf{J}_h(\hat{\mathbf{x}}_k^{(N)}) \bar{\mathbf{S}}_k & \mathbf{K}_k \mathbf{S}_{R,k} \end{bmatrix}) \quad (30)$$

3.2 L-M based Iterative square root cubature Kalman filter

Suppose that the state distribution at k-1 time is $\mathbf{x}_{k-1} \sim \mathcal{N}(\hat{\mathbf{x}}_{k-1}, \mathbf{S}_{k-1} \mathbf{S}_{k-1}^T)$, L-M based iteration square root cubature Kalman filter (ISRCKFLM), which includes two process: time update and measurement update, is describe as follows.

1) Time Update

- (1) Calculate the cubature points and the predicted $\bar{\mathbf{x}}_k$ and $\bar{\mathbf{S}}_k$ using Equations (3)-(6) at k-1 time.
- (2) Evaluate the modified covariance:

$$\tilde{\mathbf{P}}_k = \begin{bmatrix} \mathbf{I} - \bar{\mathbf{S}}_k \bar{\mathbf{S}}_k^T \left(\bar{\mathbf{S}}_k \bar{\mathbf{S}}_k^T + \frac{1}{\mu_i} \mathbf{I} \right)^{-1} \end{bmatrix} \bar{\mathbf{S}}_k \bar{\mathbf{S}}_k^T \quad (31)$$

2) Measurement update

- (1) Set the initial value as: $\hat{\mathbf{x}}_k^{(0)} = \bar{\mathbf{x}}_k$.
- (2) Assuming the i-th iterate $\hat{\mathbf{x}}_k^{(i)}$, using equation (24) to calculate the matrix $\mathbf{L}_k^{(i)}$.
- (3) Using equation (23) to calculate the iterate $\hat{\mathbf{x}}_k^{(i+1)}$.
- (4) Calculate the iteration termination condition

$$\|\hat{\mathbf{x}}_k^{(i+1)} - \hat{\mathbf{x}}_k^{(i)}\| \leq \varepsilon \text{ or } i = N_{\max} \quad (32)$$

ε and N_{\max} are predetermined threshold and maximum iterate number, respectively. If the termination condition meets, continue to 5); otherwise: set $\hat{\mathbf{x}}_k^{(i)} = \hat{\mathbf{x}}_k^{(i+1)}$ and return to 3).

(5) Mark iterate number N when the termination condition meets and calculate the state estimation at k time instant

$$\hat{\mathbf{x}}_k = \mathbf{x}_k^{(N)} \quad (33)$$

- (6) Evaluate the cross-covariance and square root of innovation covariance at k time

$$\mathbf{P}_{xz} = \bar{\mathbf{S}}_k \bar{\mathbf{S}}_k^T \mathbf{J}_h^T(\hat{\mathbf{x}}_k^{(N)}) \quad (34)$$

$$\mathbf{S}_{zz} = \text{Chol}([\mathbf{J}_h(\hat{\mathbf{x}}_k^{(N)})\bar{\mathbf{S}}_k \quad \mathbf{S}_{R,k}]) \quad (35)$$

(7) Calculate the square root of covariance at k time

$$\mathbf{K}_k = \mathbf{P}_{xz} / \mathbf{S}_{zz}^T / \mathbf{S}_{zz} \quad (36)$$

And \mathbf{S}_k is calculated using the Equation (30).

The ISRCKFLM algorithm inherits the virtues of SRCKF which has better numerical stability. The measurement update of the ISRCKLM algorithm is transformed to the nonlinear least-square problem; the optimum state estimation and covariance are solved using L-M method with better performance. The sequences obtained have the global convergence.

4. Applications to State Estimation for Re-Entry Ballistic Target

To demonstrate the performance of the ISRCKFLM algorithm, we apply the ISRCKFLM to estimate state of re-entry ballistic target with unknown ballistic coefficient and compare its performance against the SRCKF and iterate square root cubature Kalman filter using Gauss-Newton method (ISRCKF) algorithms. All the simulations were done in MATLAB on a ThinkPad PC with an Intel (R) CORE i5 M480 processor with the 2.67GHz clock speed and 3GB physical memory.

In the simulation, the parameters and the initial state estimate are the same as in ^[10]. To demonstrate the performance of the ISRCKFLM algorithm, we use the root-mean square error (RMSE) and average accumulated mean-square root error (AMSRE) in the position, velocity and ballistic coefficient introduced in ^[8]. Figure. 1, Figure. 2 and Figure. 3 show the RMSEs for the SRCKF, ISRCKF and ISRCKFLM ($u = 10^{-10}$) in position, velocity and ballistic coefficient in an interval of 15s-58s. The AMSREs of the three filters in position, velocity and ballistic coefficient are listed in Table. 1. The iteration number selected in the ISRCKFLM and ISRCKF algorithms is 4. All performance curves and figures in this subsection were obtained by averaging over 100 independent Monte Carlo runs. All the filters are initialized with the same condition in each run.

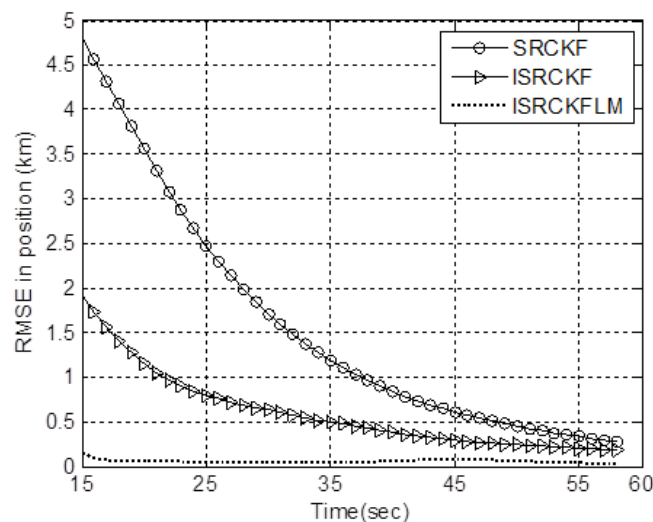


Figure.1 RMSEs in position for various filters

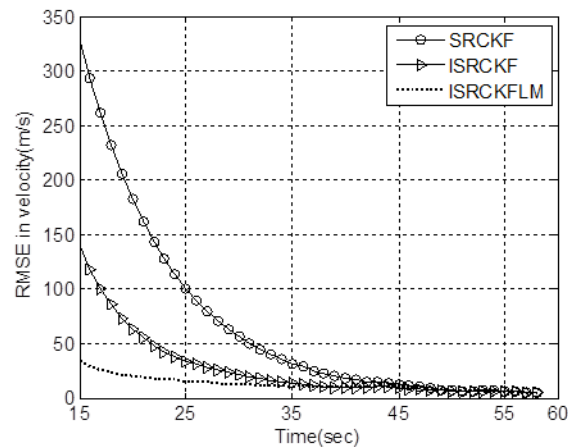


Figure.2 RMSEs in velocity for various filters

From Figure.1, we can see that the RMSE of ISRCKFLM in position is far less than that of SRCKF algorithm, and is less than that of ISRCKF algorithm. Moreover, the ISRCKFLM needs 14.5 seconds to make the RMSE in position reduce below 500 meters, the ISRCKF algorithm needs 34.6 seconds, and SRCKF algorithm needs about 47.6 seconds, so the ISRCKFLM algorithm has faster convergence rate than the SRCKF and ISRCKF algorithms. So the estimates provided by the ISRCKFLM in the position and velocity are markedly better than those of SRCKF and ISRCKF algorithms.

Observe from Figure.2, the RMSE of ISRCKFLM in velocity is far less than those of SRCKF and ISRCKF algorithm in the interval time ($t < 35s$), the ISRCKFLM still has faster convergence rate. And the RMSEs of the three filters lie at the lower level in the period ($t > 35s$).

As to the estimation of the ballistic coefficient, in the Figure.3, the RMSEs of the three filters have less improvement in the interval time ($0 < t < 35s$) because of having less effective information about it from the noisy measurement. The RMSEs of the three filters begin to decrease at about $t=37s$ because the measurements have the effective information on ballistic coefficient. In the period ($35s < t < 45s$), the RMSE of the ISRCKFLM algorithm for the ballistic coefficient decreases more rapidly than that of SRCKF, and decreases at the same rate as that of ISRCKF. At the period $45s < t < 58s$, the RMSE in the ISRCKFLM algorithm decreases most rapidly among the three algorithms. The ballistic coefficient estimate in the ISRCKFLM algorithm has the great improvement.

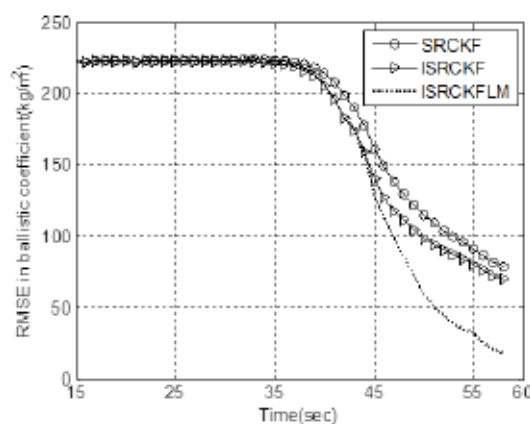


Figure .3 RMSEs in ballistic coefficient for various filters

Table 1 AMSREs in position, velocity and ballistic coefficient

Algorithms	AMSRE _p (m)	AMSRE _v (m/s)	AMSRE (kg/m ²)
SRCKF	2693.096	306.133	165.363
ISRCKF	1457.078	250.900	162.530
ISRCKFLM	856.993	220.296	160.658

According to Figure.1-Figure.3, the RMSEs of ISRCKFLM in position and velocity markedly decrease, compared with those of the SRCKF and ISRCKF algorithm. Although the RMSE of ISRCKFLM in ballistic coefficient has less improvement, its RMSE significantly reduces in the last period. So the ISRCKFLM improves the state estimation accuracy of re-entry ballistic target.

From Table.1, it is seen that, the ISRCKFLM's AMSRE in position reduces by about 68%, and its AMSRE in velocity reduces by about 28% compared to SRCKF. And compared to ISRCKF, the AMSRE of ISRCKFLM algorithm in position decreases by about 41%, and its AMSRE in velocity decreases by about 12%. Table.1 shows ISRCKFLM's AMSRE in ballistic coefficient reduces marginally, but Figure.3 shows the ISRCKFLM's RMSE is less than the other two filters in the interval of 40s-58s. Hence, the ISRCKFLM is to be preferred over the other filters in the light of AMSREs in the position, velocity and ballistic coefficient and has better performance.

Therefore, on the basis of the simulation results presented in Figure.1-Figure.3 and Table.1, one can draw a conclusion that the ISRCKFLM algorithm yields on the superior performance over the SRCKF and ISRCKF algorithms on state estimation of re-entry ballistic target.

5. Conclusion

In this study, we develop a more accurate nonlinear filtering named the L-M method based iteration square root cubature Kalman filter (ISRCKFLM). The measurement update of the ISRCKFLM algorithm is transformed to nonlinear least square problem using predicted state estimation and covariance as initial value, and then the optimal state estimation is obtained using the L-M method. The ISRCKFLM algorithm has the advantages of global convergence, fast convergence and numerical stability. The ISRCKFLM algorithm is applied to state estimation for re-entry ballistic target. Simulation results demonstrate that the performance of ISRCKFLM algorithm is superior to SRCKF and ISRCKF algorithms. So the ISRCKFLM algorithm is much more effective and improves the performance of state estimation to a marked degree.

Acknowledgments

The authors would like to thank the support of the State and Local Joint Laboratory of Advanced Network and Monitoring Control Engineering (No. GSYSJ201603).

References

- [1] Bar-Shalom, Y.; Li, X.R. Kirubarajan, T. Estimation with Applications to Tracking and Navigation. New York: John Wiley & Son, 2001.
- [2] Grewal, M.S.; Andrews, A.P. Kalman filtering: theory and practice using Matlab. New York: John Wiley & Sons, 2008.

- [3] Julier, S. J.; Uhlmann, J.K. Unscented filtering and nonlinear estimation. *Proceedings of IEEE*. 2004, 92(12): 401-422.
- [4] Arasaratnam, I.; Haykin, S.; Hurd, T.R. Cubature Kalman filtering for continuous-discrete systems: theory and simulations. *IEEE Transaction on Signal Processing*. 2010, 58(10): 4977-4993.
- [5] Arasaratnam, I.; Haykin, S. Cubature Kalman filters. *IEEE Transactions on Automatic Control*. 2009, 54 (6): 1254-1269.
- [6] Zhang, Y.G.; Huang, Y. L.; Li, N.; Zhao, L. Interpolatory cubature Kalman filters. *IET Control Theory & Applications*. 2015, 9 (11): 1731 – 1739.
- [7] Jafar, Z.; Ehsan, S. Convergence analysis of non-linear filtering based on cubature Kalman filter. *IET Science Measurement & Technology*. 2015, 9 (3):294 – 305.
- [8] Mu Jing, Cai Yuanli, Wang Changyuan. L-M Method Based Iteration Cubature Kalman Filter and Its Applications. *Journal of Xi'an Technological University*, 2013, 33(1):1-6.
- [9] Chandra, K.P.B.; Da-Wei, G.; Postlethwaite, I. Square root cubature information filter, *IEEE Sensors Journal*. 2013, 13 (2):750 – 758.
- [10] Mu. J.; Cai. Y. Likelihood-based iteration square-root cubature Kalman filter with applications to state estimation of re-entry ballistic target. *Transactions of the Institute of Measurement and Control*. 2013, 35(7): 949-958.

The Prediction of Haze Based on BP Neural Network and Matlab

Ma Limei ^{1,2,3}, Wang Fangwei ^{1,2}

¹ College Of Information Technology, Hebei Normal University,
Shijiazhuang 050024, Email: malimei@bupt.edu.cn

² Key Laboratory of Network and Information Security in Hebei Province,
Shijiazhuang 050024

³ Visiting scholar of Beijing University of Posts and
Telecommunications 100876

Abstract. In this paper, the neural network theory is used to establish the BP neural network prediction system for the occurrence of haze. The corresponding parameters are determined by MATLAB language, and the effect of the model is tested by the prediction of Shijiazhuang area. The result shows the feasibility of the predictive model. So it's valuable and has a bright future.

Keywords: BP neural network, haze, Matlab

1. Introduction

In recent years, the haze has been becoming more and more serious harm to people's daily life and health, the Beijing Tianjin Hebei region is particularly significant, the most serious area is Hebei Shijiazhuang, haze weather accounted for almost the whole winter, many factors causing haze, from the angle of the influence of climatic factors for the occurrence of haze weather, mainly includes the following 4 factors:

1) SO₂: the most common sulfur oxides. A colorless, poisonous gas with a strong irritant. One of the major pollutants in the atmosphere. The volcano emits gas when it erupts, and sulfur dioxide is produced in many industrial processes. Due to coal and oil usually contain sulfur compounds, so the combustion generates sulfur dioxide, so the car tail gas and sulfur dioxide, into the winter heating period in North China, using a large amount of coal, it will produce large amounts of sulfur dioxide, causing haze aggravated.

2) NO₂: nitrogen dioxide is a reddish brown, highly active gaseous substance, also known as nitrogen oxide. Nitrogen dioxide plays an important role in the formation of ozone. In addition to natural sources, nitrogen dioxide mainly comes from the release of high temperature combustion processes, such as motor vehicle exhaust and boiler exhaust emissions. In addition, the industrial process also produces some nitrogen dioxide. Worldwide anthropogenic pollution is estimated to be about 53 million tons of nitrogen oxides per year.

3) CO: carbon monoxide is the most widely distributed and the largest amount of pollutants in the atmosphere, and it is also one of the important pollutants produced during combustion. The main

source of CO in the atmosphere is internal combustion engine exhaust, followed by combustion of fossil fuels in boilers. Any carbon containing substance may produce carbon monoxide when combustion is incomplete.

4) O₃: the ozone layer in the atmosphere is now widely known for its protection of the living things of the earth. It absorbs most of the ultraviolet radiation released by the sun and protects animals and plants from such rays. This gives people the impression that it is protected by ozone should be better, this is not the case, if the ozone in the atmosphere, especially ozone near the ground in the atmosphere gathered too much for humans but is a scourge of high concentration ozone. Ozone is also a greenhouse gas, which can cause more serious greenhouse effect. High concentrations of ozone near the ground can stimulate and damage the mucous tissues of the eyes, respiratory system, and have a negative effect on human health.

2. Based on Neural Network Haze Forecast Principle

There are many causes of haze, as we all know, there is a close relationship between haze and weather factors, it is also affected by temperature, wind and rainfall and other factors. There is a complex interaction between the factors that affect the haze, and it is difficult to establish a precise and perfect prediction model using traditional methods. The BP neural network has good characteristics of predicting nonlinear complex systems, and can effectively describe its complex nonlinear characteristics such as uncertainty and multi input.

Prediction is to estimate the size of future unknown data by some known historical data, with time series $\{ X_i \}$, where historical data $X_n, X_{n+1}, \dots, X_{n+m}$. To predict the size of the future $n+m+k$ ($k>0$) moment, that is, to predict the size X_{n+m+k} , the method is to find the historical data, $X_n, X_{n+1}, \dots, X_{n+m}$ And X_{n+m+k} some nonlinear function relation:

$$X_{n+m+k} = F(X_n, X_{n+1}, \dots, X_{n+m})$$

Here, x_1, x_2, \dots, x_n , N represents the output of haze factors, X_{n+m+k} which means the output haze size.

Neural network is used to predict, that is, the neural network is used to fit the function $F(y)$, and the future data is obtained. Commonly used are the following three types of forecasts:

1) single step prediction, when $k=1$, network input $X_n, X_{n+1}, \dots, X_{n+m}$, m historical data, output X_{n+m+k} sizes. Such forecasts are clearly not applicable to weather haze forecasts.

2) multi step prediction, when $k>1$, that is, the network input m historical data, output $X_{n+m+1}, X_{n+m+2}, \dots, X_{n+m+k}$, The experimental results show that the prediction error of fog is larger.

3) rolling prediction, also known as iterative step by step prediction

A single step prediction is then carried out, and then the output is fed back to the input as part of the network input, as shown below (to predict the size of the next q moments):

Execution steps neural networks inputs and outputs (prediction results)

$$(1) \quad x_a, x_{a.1}, \dots, x_{a.m} \quad x_{a.m.1}$$

$$(2) \quad x_{a.1}, x_{a.2}, \dots, x_{a.m.1} \quad x_{a.m.2}$$

.....

$$q \quad x_{a.q-1}, x_{a.q}, \dots, x_{a.m.q-1} \quad x_{a.m.q}$$

Prediction of haze weather rolling forecasts, forecast of haze weather is to establish a prediction function according to the collected historical data and its impact, so the prediction results and the actual results of error as small as possible, determined between the amount and the weather factors, the future weather forecast the change trend of haze size. The fog and haze, reaches the aim of early warning, the haze hazard is reduced to the minimum.

3. Establishment of Fog and Haze Prediction Model

3.1 BP algorithm principle

BP algorithm is a common algorithm for training multilayer feedforward networks. It has three or more than three layers of neural networks, including input layer, output layer and intermediate layer. When a learning mode for the network, the neuronal activation sizes from the input layer of the middle layer (hidden layer) transmitted to the output layer, the output layer neurons corresponding to the input mode of the network response. Then, according to the principle of reducing the desired output and the actual output error, the connection rights are corrected from the output layer, the intermediate layer, and finally back to the input layer. Since this correction process takes place from the output layer to the input layer, it is called error propagation algorithm". With the error back-propagation training continuously, the network also continue to improve the correct rate of the input response pattern, as a kind of suitable for non-linear object analysis and prediction tools are widely used in various fields.

Figure 1 shows the usual BP network model with an intermediate layer (hidden layer). Where O_k represents the output unit, V_j represents the implicit unit, and ξ_j represents the input unit. The connection power from the input unit to the implied unit is w_{jk} , and the connection power from the implicit unit j to the output unit i is w_{ij} , and $\omega = (w_{jk}, w_{ij})$ represents all the connection rights.

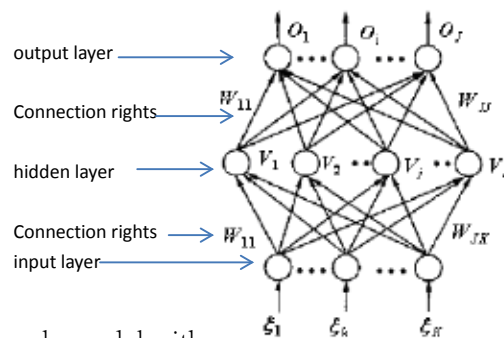


Figure 1 a BP network model with

Figure.1 BP network model with an implicit layer

3.2 BP network learning process specific algorithm steps

According to the theory of BP, three layer BP neural network input layer neuron number is L and the number of neurons in the hidden layer is M, the number of output layer neurons of N, P on the training mode, here are the specific steps of the algorithm of the BP network learning process:

1) Initialize the weights and thresholds of each neuron, and assign the random sizes (-1, +1) to $\{w_{ij}\}$, $\{w_{jk}\}$, $\{\theta_j\}$, $\{\theta_k\}$

2) A sample set of data is sampled randomly from the training sample into the network, for example, $B_n = (b_{n1}, \dots, b_{nN})$, $D_n = (d_{n1}, \dots, d_{nN})$ is provided to the network;

3) Use input mode $B_n = (b_{n1}, \dots, b_{nN})$, connection power $\{w_{ij}\}$, threshold $\{\theta_j\}$ to calculate the input of the units in the middle layer S_j ;

According to the principle of neuron model, the input S_j (activation size) of each neuron in middle layer is calculated.

$$S_j = \sum_{i=1}^L W_{ij} \cdot b_{in} - \theta_j \quad j=1,2,\dots,M \quad (1)$$

The activation size is inserted into the activation function and the output of the intermediate layer j is V_j

$$V_j = f(s_j) = \frac{1}{1 + \exp(-\sum_{i=1}^L W_{ij} b_{in} + \theta_j)} \quad n=1,2,\dots,p \quad (2)$$

4) Use the output V_j of the intermediate layer, the connection power $\{w_{jk}\}$, and the threshold $\{\theta_k\}$ to calculate the input I_k of each unit of the output layer;

According to the principle of neuron model, the input I_k (activation size) of each neuron in the output layer is calculated.

$$I_k = \sum_{j=1}^p W_{jk} \cdot V_j - \theta_k \quad j=1,2,\dots,M \quad (3)$$

When the activation size is substituted into the activation function, the response O_k of the units in the output layer can be obtained

$$O_k = f(\sum_{j=1}^p W_{jk} V_j - \theta_k) = \frac{1}{1 + \exp(-\sum_{j=1}^p W_{jk} V_j + \theta_k)} \quad k=1,2,\dots,N \quad (4)$$

5) The correction error of each node in the output layer is calculated by using the expected output mode

$D_n = (d_{n1}, \dots, d_{nN})$ and the actual output O_k of the network δ_k .

$$\delta_k = (d_k - O_k) \cdot O_k (1 - O_k) \quad k = 1, 2, \dots, N \quad (5)$$

6) The correction error of each node in the middle layer is calculated by using the connection error $\{w_{ij}\}$, the output layer's general error δ_k and the output V_j of the middle layer.

$$e_j = V_j (1 - V_j) \cdot \sum_{i=1}^L \delta_k \cdot W_j \quad j = 1, 2, \dots, M \quad (6)$$

7) Error back propagation: according to the calculation results in 5, the output of δ_k and the units of the middle layer V_j , $\{w_{ij}\}$ and $\{\theta_k\}$ are corrected to calculate the new connection weight and threshold between the next layer and the output layer.

$$w_{jk}(N+1) = w_{jk}(N) + \mu \delta_k V_j + \alpha \Delta w_{jk} \quad (7)$$

$$j = 1, 2, \dots, M \quad k = 1, 2, \dots, N \quad (0 < \alpha < 1)$$

$$\theta_k(N+1) = \theta_k(N) + \mu \cdot \delta_k \quad k = 1, 2, \dots, N \quad (8)$$

8) Error back propagation: according to the calculation results in e_j 6, the input b_n^k of each input layer of the input layer, the connection weights $\{w_{ij}\}$ and threshold $\{\theta_j\}$ are corrected, and the new connection weight and threshold between the input layer and the intermediate layer are calculated.

$$w_j(N+1) = w_j(N) + \beta e_j b_n^k + \alpha \Delta w_j \quad (9)$$

$$i = 1, 2, \dots, L \quad j = 1, 2, \dots, M \quad (0 < \alpha < 1)$$

$$\theta_j(N+1) = \theta_j(N) + \beta \cdot e_j \quad j = 1, 2, \dots, M \quad (10)$$

9) Then select a sample model randomly from the sample space and return to 3, until the training of all samples is finished.

10) Select a pattern pair from the N learning model again, and return to step 3 until the network global error E is less than a minimum size, i.e., the network convergence. The network cannot converge when the number of learning returns is greater than the pre-set size.

11) And finally, the end of study. In the above steps, 3-6 is the forward propagation process of the input learning mode, 7 and 8 are the inverse propagation processes of network errors, and 9 and 10 complete the training and convergence process.

12) The global error function of the network is defined as:

$$E = \sum_p E_p \quad (11)$$

$$E_p = \frac{1}{2} \sum_j (d_p - o_p)^2 \quad (12)$$

E_p is the error of a set. For all input modes, E is the global error of the network. The purpose of the training network is to find a set of weights to minimize the error function. From the above we can see that the amount of adjustment of each connection is proportional to the error function E_p of each learning model, which is called standard error back-propagation algorithm. Compared with the global error function, the connection weights of E should be unified after all the learning patterns are supplied to the network. This algorithm is called the cumulative error back-propagation algorithm. When the learning model is not too large, i.e., the learning mode is relatively small, the cumulative error inverse propagation algorithm converges faster than the standard error backpropagation algorithm.

The learning rule of BP nets is to achieve a gradient descent of the sum of squares and errors of E_p (or E) on the set of learning patterns rather than the gradient of the absolute error $\delta_k = (d_k - o_k)$ of a particular pattern component. Therefore, after each calibration, the network output error may also increase for some neurons, but after repeated calculations, the error should be smaller.

4. Input and Output Data Design

The number of neurons in the input and output layers of the BP network is entirely determined by the user's requirements. The number of neurons in the input layer is determined by the influence factors, consider various factors, thus produce haze, combined with the meteorological data in February 2016, the SO₂, NO₂, CO several factors and O₃ mainly influence the BP neural network model is established to determine the relationship between occurrence quantity and climatic factors of haze. According to the February 2016 air index real-time monitoring data, forecast the haze of March 2016 weather. We selected SO₂, NO₂, CO, O₃ and other four factors as the input data, so the number of input nodes is 4, the number of output nodes is 1, namely SO₂, NO₂, CO, O₃, BP neural network model is established to determine the relationship between the size and climate factors in haze, to predict the future the fog haze size.

The range of fluctuations in the data collected is large, will affect the neural network learning speed and prediction accuracy, therefore, pretreatment to the input and output data, the so-called pre-processing is normalized to the data processing, the input and output data into the parameters can receive the activation function, transform the input data in practice in order to meet the requirements of the network, the input and output of the data, before learning first to normalize the input data according to the following formula

$$y_i = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)} * 0.8 + 0.1 \quad i = 1, 2, \dots,$$

In the formula y_i is the normalized data, x_i collected real data, the maximum size of $\max(x_i)$ represents a set of data, the minimum size of $\min(x_i)$ represents a set of data, the data after normalization are located in [0.1, 0.9].

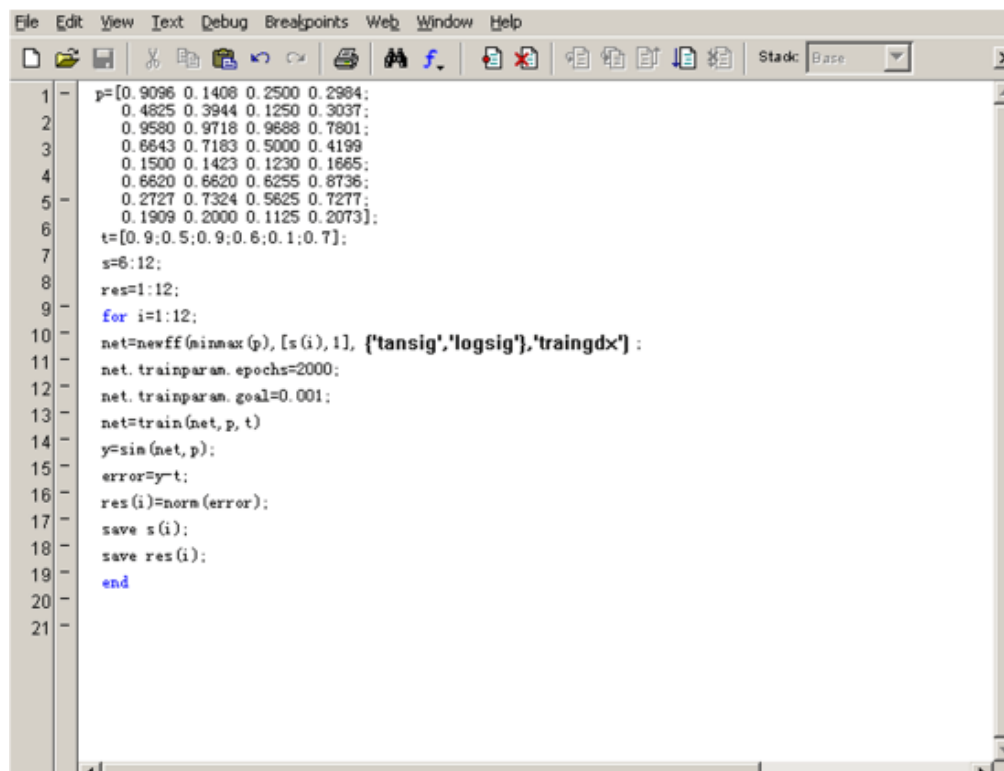
5. Using MATLAB To Realize The Determination of The Number of Hidden Layer Neurons and Transfer Function

The number of hidden layer neurons selection is a complex problem, in the application of neural network in the fact that the application of artificial neural network are often transformed into how to determine the parameters of the network structure and connection weights for each. The number of hidden layer neurons may be too little training a network or network is not strong, cannot identify previously never seen samples of fault tolerance; but the number of hidden units too much, and can make the learning time is too long, the error is not necessarily the best, so there is a proper hidden layer unit the problem of determining the number of.

General or by empirical formula to determine the number of hidden layer neurons, but this approach is not accurate, this paper used MATLAB language program to realize, the number of neurons in the input layer of the network has been determined to be 4, the number of neurons in the output layer is 1, according to the hidden layer design formula, and considering the actual situation forecast, to solve the problem of network hidden layer neuron number should be between 6~12. Therefore, we design a BP network with a variable number of neurons in the hidden layer, and determine the optimal number of neurons in the hidden layer by comparing the errors, and test the influence of the transfer function on the network performance.

5.1 Determination of the number of neurons in the hidden layer

The network design and procedures are shown in figure 2:



```

1  p=[0.9096 0.1408 0.2500 0.2984;
2     0.4825 0.3944 0.1250 0.3037;
3     0.9580 0.9718 0.9688 0.7801;
4     0.6643 0.7183 0.5000 0.4199
5     0.1500 0.1423 0.1230 0.1665;
6     0.6620 0.6620 0.6255 0.6736;
7     0.2727 0.7324 0.5825 0.7277;
8     0.1909 0.2000 0.1125 0.2073];
9  t=[0.9;0.5;0.9;0.6;0.1;0.7];
10 s=6:12;
11 res=1:12;
12 for i=1:12;
13     net=newff(minmax(p),[s(i),1],{'tansig','logsig'},'traingdx');
14     net.trainparam.epochs=2000;
15     net.trainparam.goal=0.001;
16     net=train(net,p,t);
17     y=sim(net,p);
18     error=y-t;
19     res(i)=norm(error);
20     save s(i);
21     save res(i);
22 end

```

Figure.2 The Procedure for Determining The Number of Neurons In the Hidden Layer

Thus, the transfer function of the hidden layer neurons in the network is Tansig, and the transfer function of the output layer neurons is logsig. The results of the above procedures are shown in table 1.

Table 1 The Number of Neurons and Network Errors

Numberof neurons	6	7	8	9	10	11	12
Network error	0.2642	0.6437	0.1446	0.1442	0.1448	0.1856	0.3584

Table 1 shows that, after 2000 training, the BP network with 9 neurons in the hidden layer is the best approximation to the function, because it has the least error, and the network achieves the target error after 500 training. The network errors of the hidden layer are 8 and 10, but the training time is long. Considering the training speed of network performance, the number of neurons in the network hidden layer is determined to be 9.

It can be seen from table 1, the number of neurons in the hidden layer is not the more the better performance of the network, in the program, and no error obviously with the increase of the number of hidden layer neurons decreased, when the number of neurons increased from 6 to 7, but the error increases. This phenomenon was observed from 9 to 10 and from 11 to 12.

5.2 Predictive model topology

Bp network the number of hidden units are constantly adjusted, we through the results of MATLAB programs, analysis of the hidden nodes is 9, 1 is the number of nodes in the output layer, the prediction model structure as shown in Figure 3 Bp network.

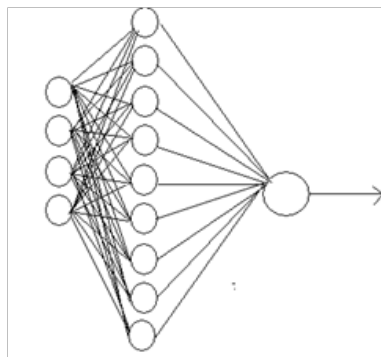


Figure.3 Topological Structure of Prediction Model

5.3 Determination of transfer function

The adoption of different transfer functions also has an impact on the performance of the network, such as convergence speed. The following uses different transfer functions to train the network and observe the results.

The transfer function of the hidden layer and the output layer is Tansig, which is trained by the gradient descent momentum method, and the learning rate is adaptive. When the number of neurons in the hidden layer is 9 and the approximation error of the network is 0.1442, the training results of the network are shown in Figure 4. From Figure 4, we can see that the network achieves the target error after 500 training.

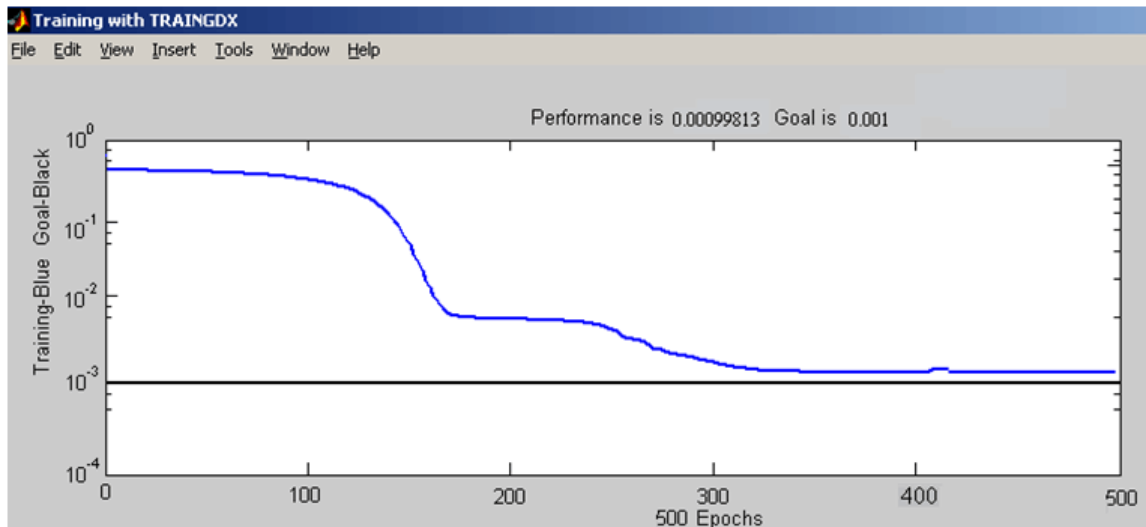


Figure.4 Training Results of Transfer Function Tansig

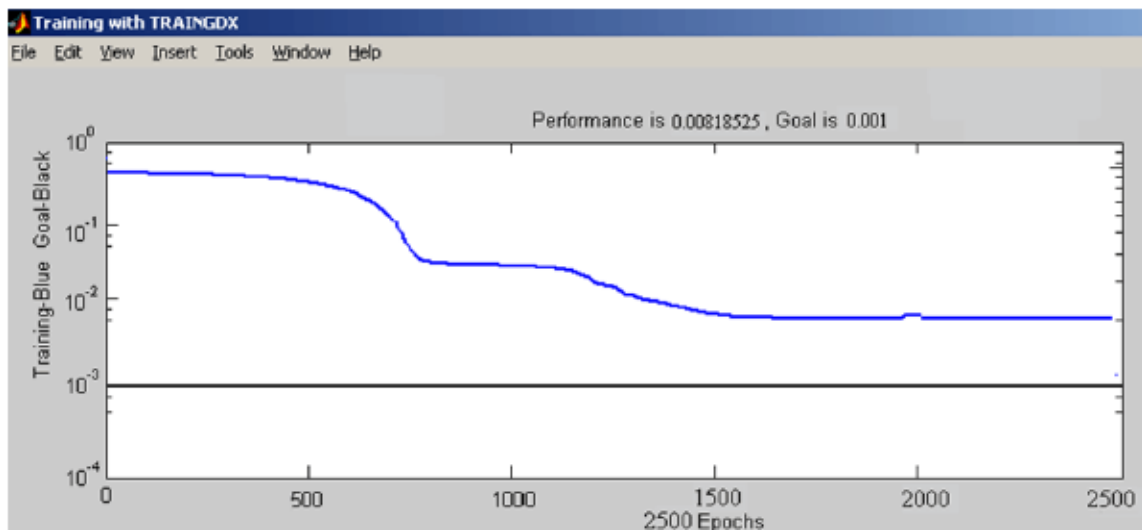


Figure.5 Training Results of Transfer Function Logsig

Finally, the transfer function of network hidden layer neurons using Tansig transfer function, the output layer neurons by logsig, with the rest of the code in Figure 1 shows that after the 2000 training, the network still did not meet the target error, but also through the training curve of the network, as shown in Figure 5, we see the training process of the network convergence very slowly, the training error of res=0.4146 is large.

At this point, we identified the final BP network structure for this prediction, as shown in table 2.

Table 2 BP network structure

Network structure	hidden layer	neuron transfer function
Single hidden layer BPnetwork	9	tansig

5.4 Determination of learning rate

After the network structure is determined, it is necessary to use the sample data to train through some learning rules, so as to improve the adaptability of the network. Learning speed is an important factor in the training process. It determines the weight change in each cycle. In general, it tends to select smaller learning rates to ensure system stability. The range of learning speed is usually between 0.01-0.07, and the learning rate is 0.05.

6. Simulation

6.1 Simulation example

In this case the data collected from the National City air quality real-time publishing platform, through 1 months of Hebei city in Shijiazhuang Province 7 haze monitoring points collected data, as shown in Table 3, due to space limitations in Table 3 for a day of data, main factors affecting the haze is SO₂, NO₂, CO, O₃, in order to deal with convenient, real-time monitoring of the data (unit: g/m³, Co mg/m³) were normalized after finishing finishing, all the data are shown in table 4.

Table 3 Data on Urban Air Quality in Real Time in China

Real-time monitoring data(ug/m2,co units mg/m2)						
site	So2	No2	co	O3	PM10	PM2.5
Great Hall of the People	50	59	2.1	58	204	118
Staff hospital	52	46	2.1	114	204	112
High-tech Zone	68	73	2.4	80	258	143
22 Zhongnan Campus	75	82	2.4	76	241	137
Century Park	56	64	2.4	69	235	151
Southwest high diocese	39	36	1.9	70	198	134
Yongsan	26	64	1.3	52	205	42

The example of a complete MATLAB program as shown in Figure 6, run the program can get the network training results as shown in Figure 7, the network can be seen after the 10 training can meet the error requirement of network output results as shown in Figure 8, QuXianru is shown in figure 9.

```

File Edit View Text Debug Breakpoints Web Window Help
p1=[0.9096 0.1408 0.2500 0.2984 0.4825 0.3944 0.1250 0.3037 0.9580 0.9718 0.9688 0.7801 0.6643 0.7183 0.5000 0.0419;
0.150 0.1423 0.1230 0.1865 0.6620 0.6620 0.0625 0.8736 0.2727 0.7324 0.5625 0.7277 0.1909 0.2000 0.1125 0.2073];
p2=[ 0.9580 1.0000 0.6875 0.0733 0.8601 0.9296 0.2812 0.3979 0.1909 0.1141 0.1563 0.4660 0.9860 1.0000 0.5625 0.4241;
0.1189 0.1127 0.6250 0.8021 0.3706 0.3521 0.0313 0.6073 0.6923 0.7324 0.3125 0.2870 0.6643 0.7324 0.0625 0.1361];
p3=[0.1350 0.1423 0.5313 0.8482 0.4266 0.5070 0.1500 0.9688 0.1490 0.1765 0.1837 0.1995 0.2587 0.3662 0.5313 1.0000;
0.7203 0.8028 0.1875 0.4346 0.9301 0.9014 0.9688 0.8691 0.3287 0.3239 0.2813 0.6602 0.6084 0.5211 0.2813 0.4346];
p=[p1 p2 p3];
%构建训练样本中的目标向量
t1=[0.9 0.5 0.9 0.6 0.1 0.7 0.3 0.1];
t2=[0.9 0.8 0.1 1.0 0.1 0.4 0.6 0.7];
t3=[0.1 0.5 0.1 0.2 0.7 1.0 0.3 0.7];
t=[t1,t2,t3];
%创建一个神经网络，隐含层有9个神经元，传递函数为'tansig'
%传递函数为'logsig',训练函数为'trainlm'
net=newff(minmax(p),[9,4],{'tansig','logsig','trainlm'});
%训练步数为50
%目标误差为0.01
net.trainparam_epochs=50;
net.trainparam_goal=0.01;
net=train(net,p,t);
%预测
p_test=[0.0490 0.2587 0.7203 0.9301 0.3287 0.6084 0.3662 0.8028 0.9014 0.3239 0.5211 0.2813
0.3287 0.5313 0.1875 0.9688 0.2813 0.2813 0.0995 1.0000 0.4346 0.8691 0.6702 0.4346];
y=sim(net,p_test);
    
```

Figure.6 MATLAB Program

Table 4 All Data After Normalization

Years	date	SO2	NO2	CO	O3	Haze size
February 2016	1	0.9096	0.1408	0.2500	0.2984	0.9
	2	0.4825	0.3944	0.1250	0.3037	0.5
	3	0.9580	0.9718	0.9688	0.7801	0.9
	4	0.6643	0.7183	0.5000	0.4199	0.6
	5	0.150	0.1423	0.1230	0.1665	0.1
	6	0.6620	0.6620	0.6255	0.8736	0.7
	7	0.2727	0.7324	0.5625	0.7277	0.3
	8	0.1909	0.2000	0.1125	0.2073	0.1
	9	0.9580	1.0000	0.6875	0.0733	0.9
	10	0.8601	0.9296	0.2812	0.3979	0.8
	11	0.1909	0.1141	0.1563	0.4660	0.1
	12	0.9860	1.0000	0.5625	0.4241	1.0
	13	0.1189	0.1127	0.6250	0.6021	0.1
	14	0.3706	0.3521	0.0313	0.6073	0.4
	15	0.6923	0.7324	0.3125	0.2670	0.6
	16	0.6643	0.7324	0.0625	0.1361	0.7
	17	0.1350	0.1423	0.5313	0.8482	0.1
	18	0.4266	0.5070	0.1250	0.8586	0.5
	19	0.1490	0.1765	0.1937	0.1995	0.1
	20	0.2587	0.3662	0.5313	1.0000	0.2
	21	0.7203	0.8028	0.1875	0.4346	0.7
	22	0.9301	0.9014	0.9688	0.8691	1.0
	23	0.3287	0.3239	0.2813	0.6702	0.3
	24	0.6084	0.5211	0.2813	0.4346	0.7

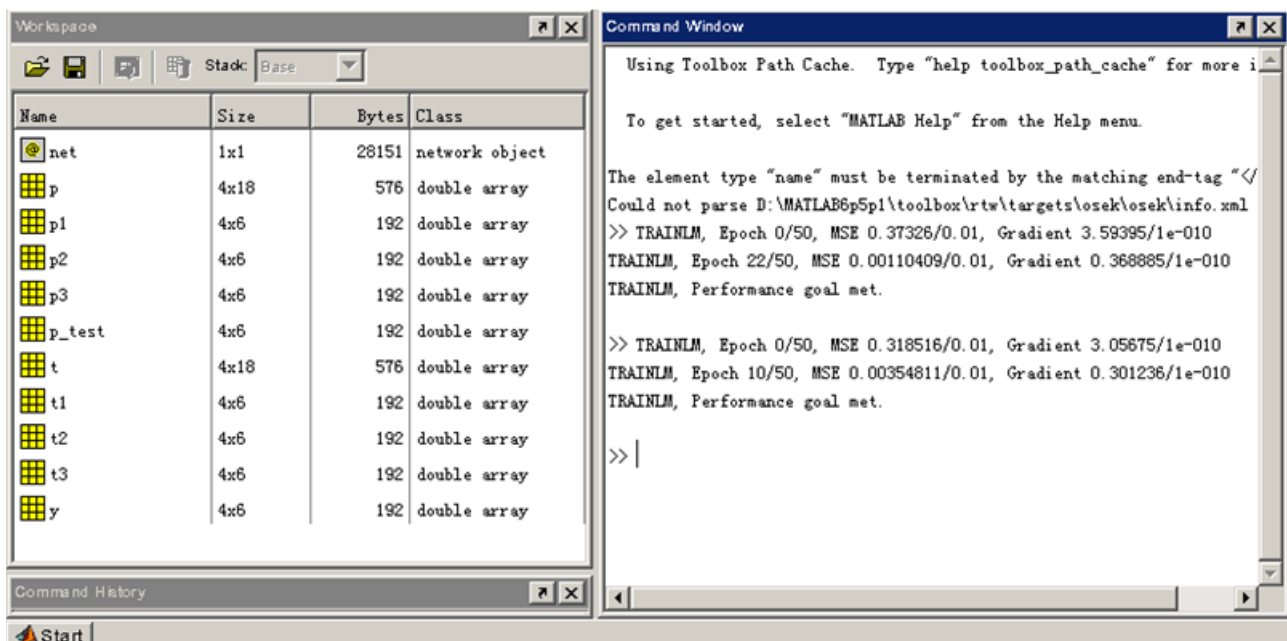


Figure.7 Training Results of the Network

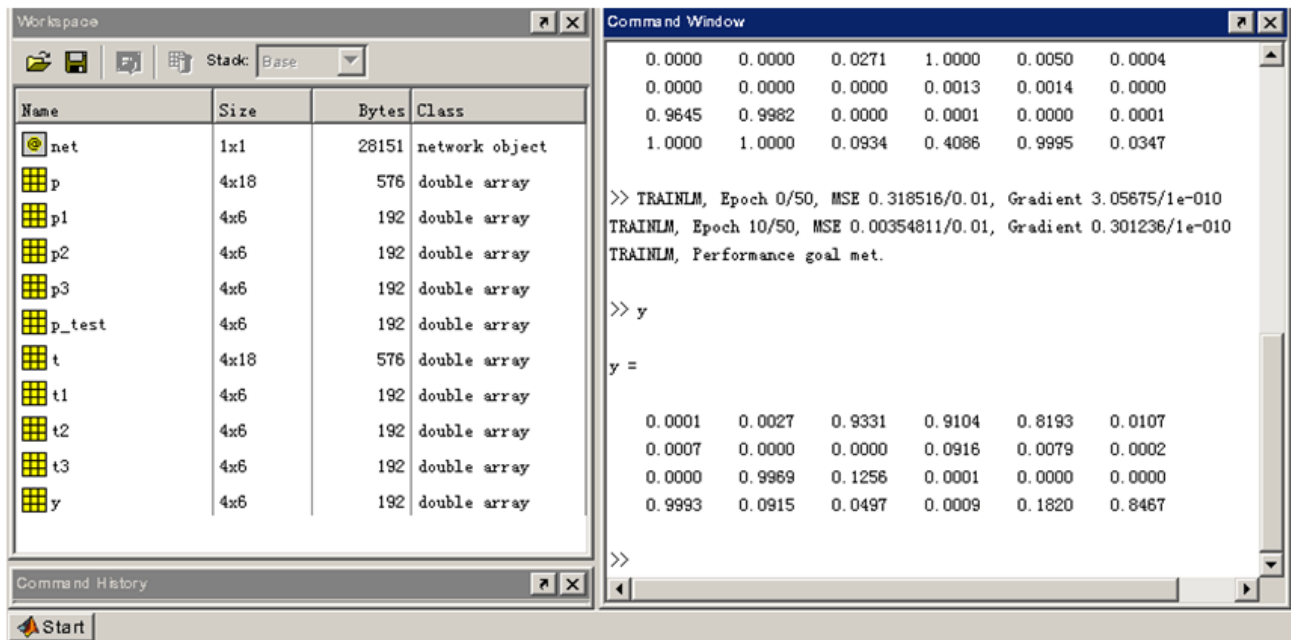


Figure.8 Output of the Program

By comparing with the actual weather, the prediction error of the network is 0.1580. Considering the small sample size, this is an acceptable result, and the forecast error curve of the network is shown in figure 10.

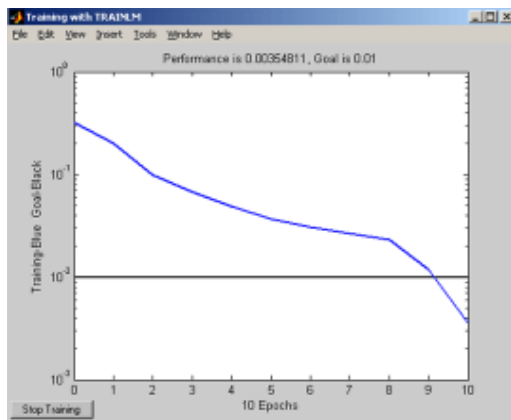


Figure.9 Training Result Curve of Program

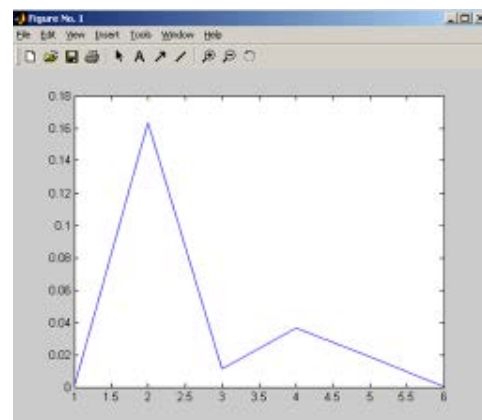


Figure.10 Prediction Error Curve Of Network

6.2 Simulation Result Analysis

With the table in February 1st -2 month 24 days from February 25th to March 2nd as the training samples for the test sample, forecast March 3rd to 10 the size of AQI, the predicted size of the input model, and the model forecasts to March 3, 2016 10, the Shijiazhuang air AQI size.

The predicted results of anti normalization, get the prediction results of the models, for the city of Shijiazhuang from March 3, 2016 to 10, the actual measurement of air pollutants AQI sizes were compared with the neural network prediction size, figure 11 is the actual measurement sizes of AQI, Figure 12 for the prediction of the size.



Figure.11 Real AQI Sizes

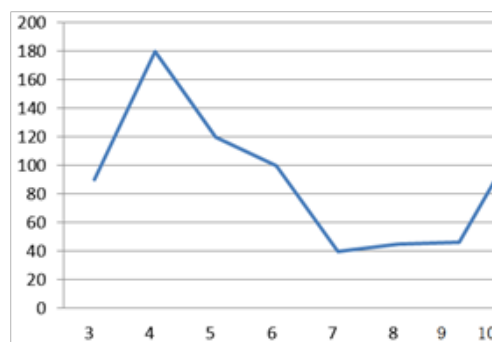


Figure.12 The Size of the Predicted AQI

7. Conclusion

The neural network model, we can see the future 10 days the AQI size and the actual size is basically the same, because the SO₂ CO₂ is one of the main causes of fog and haze, haze causes influence by temperature, wind and humidity, so with the actual monitoring data to increase the number of input nodes. And the application of neural network in the prediction of AQI, will get the desired results, more accurate.

Using MATLAB neural networks does not require tedious programming like traditional methods. He can model BP networks efficiently, accurately and quickly, and forecast the haze of the weather. Forecast of fog and haze has high application size. At the same time, the establishment of the neural network model provides a new general method, which can be used in different conditions of prediction and prediction.

Introduction: School of information technology, Hebei Normal University, associate professor, master's degree, research field: computer and application, neural network

Acknowledgment

This Project Supported by the National Natural Science Foundation of China(No.61572170), Natural Science Foundation of Hebei Province of China(No. F2016205023), Educational Commission of Hebei Province of China(No. SLRC2017042)

Reference

- [1] GaoNing. Forecast and MATLAB implementation of crop pest forecasting based on BP neural network. Anhui Agriculture University, 2003,29 (2): 191-151.
- [2] Ma Limei. Neural network stock forecasting system based on.2005 in Beijing University of Posts and Telecommunications.
- [3] Ai Hongfu, et al. Forecast of fog and haze based on Bp artificial neural network. Computer simulation, 2015 (1): 402:406

Multi Objective Optimization of Virtual Machine Migration Placement Based on Cloud Computing

Sun Hong^{1,2}, Tang Qing¹, Xu Liping¹ and Chen Shiping¹

¹University of Shanghai for Science and Technology, Shanghai
China 200093,

²Shanghai Key Laboratory of modern optical systems, Shanghai
China 200093

**Corresponding Author: Sun Hong, University of Shanghai
for Science and Technology, 200093, shanghai, China
Email:sunhong@usst.edu.cn**

Abstract. How to improve the resource utilization of the cloud computing system has been one of the key content of the research of the cloud computing. The traditional multi-objective ant colony optimization was improved, studied the virtual machine live migration framework, combined with the elimination method to solve virtual machine migration and placement of multi-objective optimization problem, the load balanced specific strategies are integrated into the framework of a dynamic migration, simulation experiments are carried out and the conclusions are made for it. The algorithm can obtain the optimal solution through the continuous updating of pheromone. The main consideration is the Service level contract violation rate(S), Resource loss(W),Power consumption (P). Experimental results show that ,compared with the traditional heuristic method and genetic algorithm, the algorithm is advantageous to the parallel computation, and it' s able to achieve the optimal tradeoff and compromise between multiple conflicting objectives. In the case of service level contract violation rate is low, system resource waste and power consumption are at the least, so it has feasibility.

Keywords: Cloud computing, Virtual machine migration, Multi objective optimization, Ant colony algorithm, Elimination method.

1. Introductiong

As a new technology, cloud computing has become a hot research topic in the field of information in recent years. As a new business computing model, business characteristics and virtualization technology is its obvious characteristics. And the task scheduling is the key technology of cloud computing, it not only affects the efficiency of the whole system, but also significantly affects the quality of service. At present, the problem of task scheduling is studied in the grid environment. Due to the diversity of user requirements in the cloud computing system, and the complexity of the task type, previous task scheduling algorithms can' t meet the requirements of the overall QoS. In the dynamic cloud computing environment, improving the efficiency of task scheduling and load balancing is an eternal problem, for users, to meet the user' s QoS requirements is the most important thing. Therefore, the research on the task scheduling algorithm with QoS expectation constraint is the key content of the

cloud computing system. Nowadays, cloud computing has become a new model to provide access and services through the Internet. If the distribution of resources is not reasonable, it will inevitably lead to waste of resources. It is of great significance to realize the multi objective optimization of virtual machine migration and placement in the present stage. Most researchers use the traditional heuristic method or genetic algorithm and other algorithms to make the virtual machine placement before; Although these algorithms can solve the problem of virtual machine migration in a certain extent, but these algorithms have their own limitations. For example, heuristic method can solve the problem of local optimal solution in virtual machine migration, but the method is short of the ability of global optimization. Although genetic algorithm has certain advantages in multi objective optimization, it can't make full use of the feedback information, so that the search is blind, and the efficiency of solving the optimal solution to a certain extent is relatively low. ant colony optimization, which also called ant algorithm, is a kind of probabilistic algorithm used to find the optimal path in the graph. Ant colony optimization is a simulated evolutionary algorithm, a preliminary study shows that the algorithm has many excellent performance. Compared with the results of genetic algorithm design, numerical simulation results show that ant colony algorithm has a new simulation evolutionary optimization method and its effectiveness and application value.

This paper introduces the management framework of the two layers of local management and global management, through this management framework is conducive to the migration of virtual machine placement and resource allocation to make better decisions. The method used in this paper is to improve the traditional ant colony algorithm, and combine with the exclusion method. It is advantageous to the parallel computation, and the efficiency is higher. To be able to obtain the optimal tradeoff between the three conflicting objectives which are service level contract violation rate(S),resource loss(W) and power consumption(P),and in the case of service level contract violation rate is low, the waste of system resources and power consumption are the least.

2. Overall Framework for Virtual Machine Migration

2.1 Virtual machine migration

Xen virtual machine is using virtualization technology which is a quasi virtualization technology, and have good performances in all kinds of architecture, it has very good performance and system isolation. Up to now, Xen is definitely the most outstanding Linux system under the open source virtual machine, Xen is now no longer originally just supported by x86, and now has wild support and even Itanium and other hardware platforms are available to it. Version 4 was released in 2010, the client can support up to 64 virtual CPU. To use a virtual machine, you should start the operating system, but Microsoft platform VMware is the first to enable the physical machine system and then start the process, and this is not the standard. After Xen started, the first step is to run the virtual machine monitor, which is Xen Hypervisor (also known as the super management program in the Xen system), then run the host operating system (or local operating system),by minimizing the connection between the super manager and the native operating system, it reduces the risk of super management program itself and the virtual machine being destroyed and information leakage.

2.2 Overall framework and optimization of virtual machine migration

The basic migration structure is implemented by four modules, including migration monitoring, execution migration, suspension and awake. As shown in Figure 1.

Monitor migration module: The primary function of the primary module is to determine the source of the migration, the start time of the migration, and the purpose of the migration. The working mode of listening and migrating module is determined by the purpose of migration. In order to ensure the load balance of each node, setting up the monitor signal in the virtual machine management program, according to the monitoring of the various nodes of the load operation to determine whether the need to migrate. Set a migration threshold, when monitor to arrive at this value, monitor migration module will send a migration signal to the source, indicates that the source machine will be migrated. Meanwhile, monitor the migration module to communicate with other nodes, look for the lower load nodes, and determine the specific location of the destination machine.

Run migration module: This module is the most important module of virtual machine migration, almost bear most of the migration work. After running the migration, this module collects the running information of the source machine, at the same time to freeze the module by sending the “frozen” signal to the source machine. This process is the key part of the migration process, directly affect the migration process downtime and migration of the total length.

Freezing module: This module is mainly responsible for how to solve the continuous service problem. It makes users feel not interruptions.

Target domain wake-up module: The function of the target domain wake-up module is to determine the time to wake up the destination machine, also ensure that the weaken target machine is consistent of the source machine, and how to ensure that the service of the target area is connected with the source area. After it shutdown, the running module will copy the remaining memory pages, then send the “weaken signal” to the weaken module which on the target machine.

Interrupt connection device is a direct consequence caused by shutdown, peripheral device cannot connect to virtual machine, this, of course, will cause the external service is not timely or appear all kinds of transmission errors. In order to improve the effectiveness of migration, reduce the time of shutdown, increase the application rate of load, we propose an optimized virtual machine dynamic migration framework, this framework is also based on Xen.

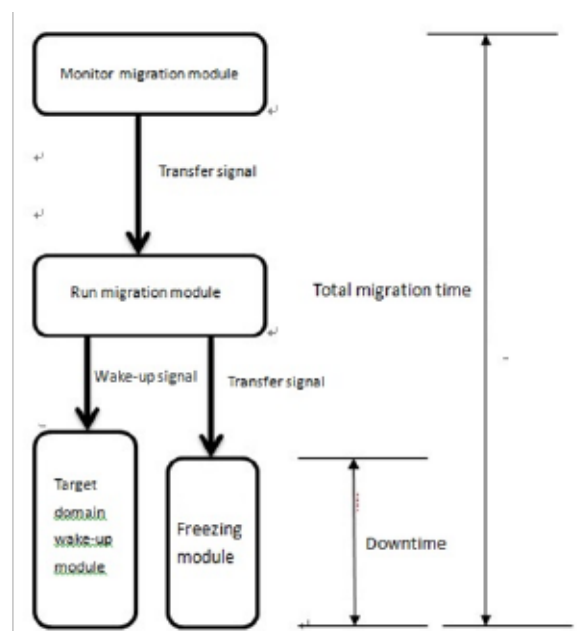


Figure.1 The virtual machine dynamic migration placement module

In order to improve the utilization of load, at the same time, it also makes the migration process more smooth and effective, optimization the design of dynamic migration of Xen virtual machine. Add two modules to achieve load balancing, one of them is the load monitor module, which add to the original monitor migration module, in order to set the identity of the current virtual machine running information, set the trigger conditions for its migration and prepare for the subsequent migration to select the appropriate load; the other one load transfer module is mainly responsible for the positioning and selection strategy of the virtual machine migration. As shown in Figure 2, three modules are marked by grey patterns.

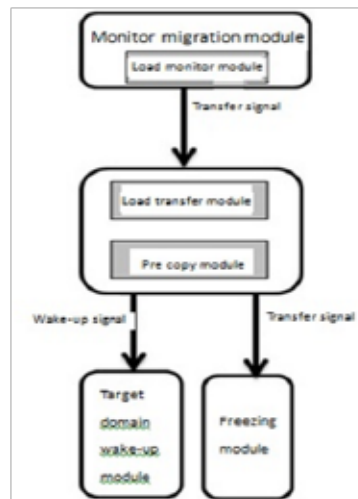


Figure.2 Dynamic migration of virtual machine placement framework optimization module

3. Virtual Machine Migration Placement Policy Model

A large number of computers will be integrated into resource pool in the cloud computing, then the computing tasks in cloud computing are distributed in the resource pool, all applications can each one takes what he needs, for example, you can have a stronger computing power to meet the needs of computing, you can set aside a larger storage space to store resources, and even more perfect online updates and other software services. In the cloud computing platform, a variety of servers to complete a specific task by way of collaboration. Because there is no unified standard for cloud computing platform application interface, the application of the various cloud environments can not be fully integrated, at the same time, the resources of the nodes in the cloud environment are different. For example, the formats provided by the same resource are different, the demands for resources in different time periods are different, there are large number of access to the same node during certain periods of time. This will lead a overload caused by excessive access on a server node in a certain period of time. While the other nodes are less load due to access relatively light, this forms the node utilization rate is not high in the whole system, causing the load is not balanced. Virtualization technology continues to mature, this is to provide a solution to this. The emergence of dynamic migration of virtual machine is an effective way to solve the load imbalance. The whole virtual machine running state can smooth and stable mutual transfer between the two physical hosts in the same cluster, of course, this is the necessary conditions for the transfers, and users don't have any feeling of stagnation. The dynamic migration of virtual machine can assist the maintenance personnel of the cloud environment, so that the nodes in the cluster can be fully used, achieve load balancing dynamically. Therefore, to improve the resource allocation in the cloud environment and to strengthen the system by designing an efficient load balancing algorithm,

this has become one of the important issues in the field of cloud computing. Virtual machines allow all computing tasks to encapsulate into the virtual machine. Because the virtual machine is one of the characteristics of isolation, so you can use the virtual machine dynamic migration technology to migrate computing tasks. The scale of cloud computing is generally relatively large, it also provide the same size of the pressure to how to adjust the distribution of the node resources. Considering the real time information of resources in cloud environment, resource scheduling must be done, this requires real-time monitoring of resources in the cloud environment, and can dynamically manage. From the point of view of the process size of the task to be migrated, cloud computing users pay more attention to the migration process in the virtual machine itself how to operate, of course, the premise is to try not to affect the user. How to provide resources to the service level agreement of the internal application of the virtual machine is a problem to be solved at present. In terms of practical ability, resource scheduling system must monitor the usage of resources and provide reference for the system itself in time, or for system management related personnel to set.

3.1 Virtual machine management framework

The number of infrastructure nodes in cloud computing is very large, which makes it very difficult to build a structure. The management framework of this paper is two layers of management, local management and global management, the details are shown in Figure 3. The management of host cloud infrastructure to enable global management to run on a host node, by monitoring the collection of various information from local management, including user service quality resource consumption and power consumption, and so on, then make decisions on the placement of the virtual machine and the allocation of resources.

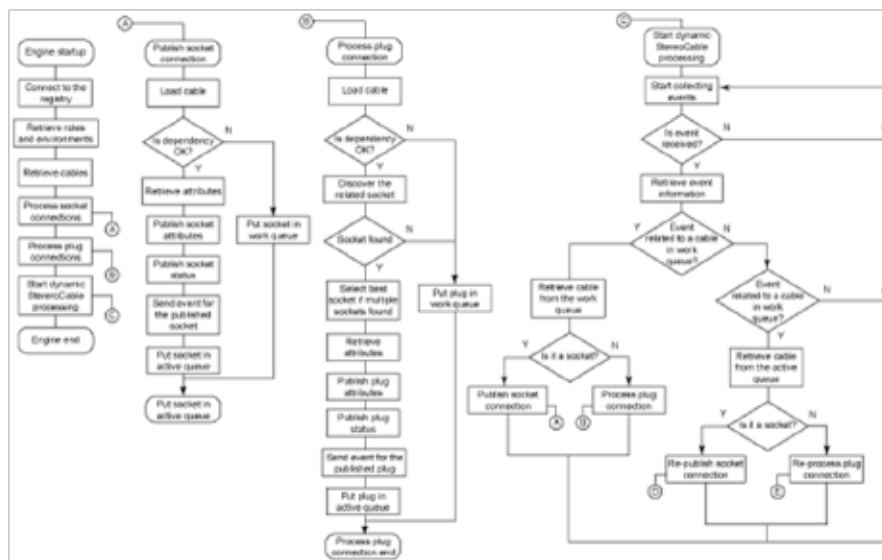


Figure.3 Virtual machine system management structure

3.2 Mathematical model of virtual machine migration

The migration and placement of virtual machine has been the focus of research in cloud computing, it is a typical bin packing problem. The literature proves that the virtual machine migration is a NP-hard problem. According to the framework of system management in Figure 3, in this paper, we need to consider the following 3 factors in the global management of virtual machine initialization: service level contract violation rate(S), resource loss(R), power consumption(P). m, n, respectively, indicating the

number of physical nodes and virtual machines, says j -th physical nodes have a corresponding resource capacity, represents the resource capacity of the request of i -th virtual machines. Its mathematical model is described as follows:

target:

$$\min \sum_{j=1}^m S_j \text{ and } \min \sum_{j=1}^m R_j \text{ and } \min \sum_{j=1}^m P_j \quad (1)$$

constraint:

$$\sum_{i=1}^n r_i^{CPU} \cdot a_{ij} < c_j^{CPU}, j = [1, \dots, m] \quad (2)$$

$$\sum_{i=1}^n r_i^{mem} \cdot a_{ij} < c_j^{mem}, j = [1, \dots, m] \quad (3)$$

$$\sum_{i=1}^n r_i^{bw} \cdot a_{ij} < c_j^{bw}, j = [1, \dots, m] \quad (4)$$

$$\sum_{j=1}^m a_{ij} = 1, i = [1, \dots, n] \quad (5)$$

The three objectives of the formula (1) are service level contract violation rate (S), resource loss (R), and power consumption (P). Formula (2-4) constrains the allocation of CPU, memory and network bandwidth resources on each physical node, which will not exceed the capacity of its own. And formula (5) constrains each virtual machine can only be assigned to a physical node.

4. Multi Objective Optimization Virtual Machine Migration Placement Strategy Based on Ant Colony Algorithm

Ant colony algorithm is a kind of technology that can be used to find the optimal solution. The algorithm is widely used in the virtual machine migration and placement problem, and has certain advantages in dealing with combinatorial optimization problems. The following is the specific design steps and process of this article.

4.1 Fitness function

The selection of the stress function is very important in the genetic algorithm. According to the formula (1), the 3 sub - suitability function is defined, the value of the range is between $[0 \sim 1]$ SLA violation rate function (f_{SLA}), resource utilization function ($f_{resource}$), power consumption function (f_{power}), Such as formula (6) - (7).

$$f_{SLA}(u_{CPU}) = \frac{1}{1 + e^{u_{CPU} - 0.9}} \quad (6)$$

$$f_{resource}(u_{CPU}, u_{mem}, u_{bw}) = u_{CPU} \times u_{mem} \times u_{bw} \quad (7)$$

$$f_{power}(u_{CPU}) = \frac{u_{CPU}}{P_{idle} + (P_{busy} - P_{idle}) \times u_{CPU}} \times P_{busy} \quad (8)$$

In the formulas, u_{CPU} 、 u_{mem} 、 u_{bw} 、 P_{busy} indicate the CPU, memory, network bandwidth utilization and multiplier factor respectively on the physical node. f_{power} reflects the amount of effective work in a certain amount of power consumed.

Taking into account the need to balance the service level contract violation rate (S), resource loss (R), power consumption (P) 3 goals. So the weight value of this paper are set to 1, and according to the experience in this definition the suitability function is

$$f(\mathbf{u}_{CPU}, \mathbf{u}_{mem}, \mathbf{u}_{bw}) = f_{SLA}(\mathbf{u}_{CPU}) + f_{resource}(\mathbf{u}_{CPU}, \mathbf{u}_{mem}, \mathbf{u}_{bw}) + f_{power}(\mathbf{u}_{CPU}) \quad (9)$$

4.2 Pheromone

pheromone update rules as shown in formula (10) - (11)

$$\gamma_{iu} = (1 - \rho) \times \gamma_{iu} + \Delta\gamma_{iu}^{best} \quad (10)$$

$$\Delta\gamma_{iu}^{best} = \begin{cases} f(S_{best}), JVM \text{ is loaded on the node } u \\ 0, \text{ others} \end{cases} \quad (11)$$

In the formula, S_{best} is the optimal solution set, ρ is the pheromone volatile coefficient, $\Delta\gamma_{iu}^{best}$ is the pheromone increment, $f(S_{best})$ is the appropriate degree function.

4.3 Probability transfer function

Probability transfer function

$$P_{iu}^k(t) = \begin{cases} \frac{\gamma_{iu}^\alpha(t) \times \eta_{iu}^\beta(t)}{\sum_{Seallowed} \gamma_{iu}^\alpha(t) \times \eta_{iu}^\beta(t)}, & i \in allowed_k \\ 0 \end{cases} \quad (12)$$

In the formula, η_{iu} is an information heuristic factor, γ_{iu} is the visibility heuristic factor.

Among them, the γ_{iu}^{CPU} , γ_{iu}^{mem} , γ_{iu}^{bw} are virtual machine I request CPU, memory and network bandwidth of the corresponding resources on the host node u ratio.

4.4 The construction of the optimal solution set

Using the exclusion method to construct the non dominated set is a common method in multi-objective genetic algorithm. In this paper, we use the rule of law and its appropriate improvements to deal with the solution of the ant colony algorithm search process, which can be used to build the Paxeto solution set, the process is as follows:

Step1: Set the solution set $D_{cycle}^* = \{D_1, D_2, \dots, D_n\}$ for a loop search, where n is the number of the solution to the search.

Step2: To evaluate each solution vector has 3 sub goals, if D_i target is better than D_j corresponding to other sub goals and sub goals, D_i and D_j were compared to non inferior, concludes that D_i dominated D_j , D_j must be removed from the current set of solutions of C, and vice versa.

Step3: And so on, will the solution D_{cycle}^* were compared with each other, to get the optimal solution set D_{cycle}^* of cycles.

Step4: The D_{cycle}^* and the global optimal solution set D_{best} are compared according to the exclusion method, and the final non dominated solution is saved to the S_{best} .

Step5: To continue the cycle, when the cycle is over, the global optimal solution set D_{best} is the Pareto optimal solution set.

5. Experimental Results and Analysis

This experiment is done on the CloudSim[6], cloudsim by Rajkumar professor Buyya team (Melbourne University) developed cloud computing simulator, Melbourne University in Australia Grid Laboratory and Gridbus project announced the launch of cloud computing simulation software.

CloudSim as a generic, scalable new simulation framework that supports seamless modeling and simulation, and can be carried out on the basis of cloud computing infrastructure and management services. This simulation framework has the following characteristics^[7]:

Simulation and example of a large-scale cloud infrastructure supporting a single physical computing node.

To provide an independent platform, the main function is to the modeling of the data center, service agent and scheduling strategy.

In a data center node to provide a virtual engine, to manage the independent virtualization services.

Flexible virtualization services can switch between shared space and shared time processing core allocation policies.

In this paper, we use ant colony algorithm for resource allocation, and some other algorithms are compared.

Experiment set 80 physical nodes, each node is configured for GB 10 memory, 1TB storage and bandwidth of 1 Gbps, while the capacity of CPU is equivalent to 1000, 2000 and 3000 MIPS. The number of requests for a virtual machine is 200, where the request for CPU is 250500750 and 1000 MIPS, 4 GB memory, 200, GB bandwidth, 200 Mbps. The power consumption of the physical node in CPU utilization is 0% and 100%, respectively. The power consumption is 175W and 250W.

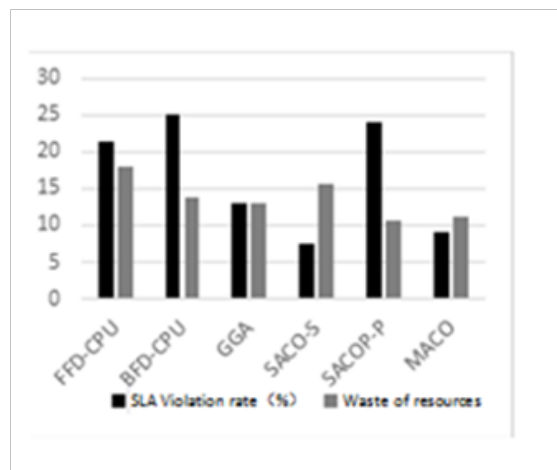


Figure.4 Comparison of SLA violation rate and resource waste in 6 placement algorithms

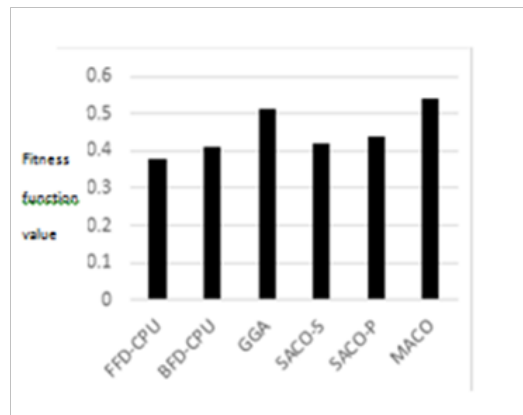


Figure.5 Comparison of the fitness function value of 6 kinds of placement algorithms

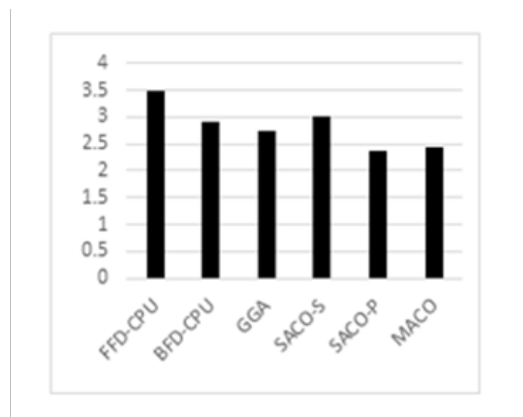


Figure.6 Comparison of 6 placement algorithms under power consumption

6. Concluding Remarks

The algorithm used in this paper is an improved ant colony algorithm for distributed multi objective optimization. This algorithm is an improvement of the traditional multi objective ant colony algorithm. Selected service level contract violation rate (S), resource consumption (W), power consumption (P) three targets. And combined with the elimination method to solve the virtual machine migration in the placement of these three objectives optimization problem. Experimental results show that compared with the traditional heuristic method and genetic algorithm, the proposed algorithm can effectively reduce the resource waste and the power consumption of the system when the service level contract violation rate is low, and it has feasibility. This paper has used the power consumption as one of the management objectives, next, we also need to consider how to take into account the data center network traffic and other aspects, so as to achieve more perfect.

References

- [1] HYEAR C, MCKEE B, GARDNER R, et al. Autonomic virtual machine placement in the data center [J]. Hewlett Packard Laboratories, Tech. Rep. HPL-2007-189, 2007:2007-189.
- [2] Li Jingchao, ChenJingyi, WuJie. Research on virtual machine placement based on improved grouping genetic algorithm [J]. Computer engineering and design, 2012, 33(5):2053-2056..
- [3] Li Yong: Analysis and Research on dynamic migration of virtual machine [Dissertation]. National Defense

Science and Technology University,2007.

[4] Li Zhiwei,WuQingbo,TanYusong. Research on dynamic migration of virtual machine based on device agent mechanism. Computer application research. Twenty-sixth volumes, April 2009.

[5] CAREY M R, JOHNSON D S. Computers and In tractability: a guide to NP-completeness [J].1979.

[6] CALHEIROS R N, RANJAN R, et al. Cloud Sim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms [J]. Software: Practice and Experience, 2011, 41(1): 23-50.

[7] Sun Hong, Zhang Huaxuan, Chen Shiping, etal. The Study of Improved FP-Growth Algorithm in Map Reduce[C].//Proc.CPCI- 1st International Workshop on cloud Computing and Information Security(CCIS) 2013 CPCI:00033591040005

Acknowledgements

This work was supported by the National natural Science Foundation of China (No.61472256, No.61170277), Innovation Program of Shanghai Municipal Education Commission(No.12zz137), and the Hujiang Foundation(C14002).

Biographies

Sun Hong: female, Han, 1964-, from Beijing, China, Master, associate professor, School of Optical-Electrical and Computer Engineering University of Shanghai for Science and Technology, master tutor, associate professor direction of research; Business schools University of Shanghai for Science and Technology doctor graduate student; the main research direction: computer network communication and clouds computing, management science and engineering, Management Information and Decision Support System . Email:sunhong@usst.edu.cn,Telephone:13916902800

Tang Qing: male, han, 1993-, master student, School of Optical-Electrical and Computer Engineering University of Shanghai for Science and Technology; the main research direction: cloud computing and management information system.Email:qingtang1993@163.com

Xu Li-ping: female, han, 1986-, master, associate professor, University of Shanghai for Science and Technology; the main research direction: cloud computing and management information system. Email:5850487@qq.com

Chen Shi-Ping: male, han, 1964-,form Zhejiang, China, professor, Ph.D., doctoral tutor Business schools University of Shanghai for Science and Technology, research direction: computer network communication and clouds computing, management science and engineering. Email:chensp@usst.edu.cn

Editor in Chief of IJANMC



Lei Yaping

Professor Lei Yaping is the Editor in Chief of IJANMC, She is the vice President of Xi'an Technological University, She got her Doctor degree form Xi'an university of technology.

Prof. Lei engaged in education teaching and scientific research more than thirty years and formed good style of teaching and scientific research quality. She is focus on Resource structure optimization and configuration. She Published 2 monographs and 2 specialized textbook compilation, She write and published more than 30 important papers and responsible for more than 20 academic research, She was awarded the teaching achievement prize from Shaanxi provincial government in 2015.



Wei Xiang

Prof. Wei Xiang is the Associate Editor-in-Chief of IJANMC, he got the Ph.D. degree in telecommunications engineering from the University of South Australia in 2004.

He is currently Foundation Professor and Head of Discipline Internet of Things at James Cook University, Australia. He is a Fellow of the IET and an elected Fellow of Engineers Australia. He received the TNQ Innovation Award in November 2016. He has been awarded several prestigious fellowship titles. He was named a Queensland International Fellow by the Queensland Government of Australia, an Endeavour Research Fellow by the Commonwealth Government of Australia, a Smart Futures Fellow by the Queensland Government of Australia, and a JSPS Invitational Fellow jointly by the Australian Academy of Science and Japanese Society for Promotion of Science. He is the Vice Chair of the IEEE Northern Australia Section.



Houbing Song

Dr. Houbing Song is the Associate Editor-in-Chief of IJANMC, He received the Ph.D. degree in electrical engineering from the University of Virginia in 2012. He is a senior member of IEEE and a member of ACM. He was the first recipient of the Golden Bear Scholar Award at West Virginia University in 2016.

He joined the Department of Electrical, Computer, Software, and Systems Engineering, Embry-Riddle Aeronautical University in 2017, where he is currently an Assistant Professor and the Doctoral tutor of the Security and Optimization for Networked Globe Laboratory (SONG Lab). He is the Founding Director of West Virginia Center of Excellence for Cyber-Physical Systems sponsored by West Virginia Higher Education Policy Commission. His research interests include cyber-physical systems, internet of things, edge computing, big data analytics, connected vehicle, and wireless communications and networking.