Research on Energy Consumption Optimal Management Method of Random Task in CloudSim Platform

Zhidan Xu

College of Teacher Education, Mudanjiang Normal University, Heilongjiang, 157000, China

Corresponding author is Zhidan Xu

Abstract
In this paper, the author researched on the optimal management method of energy consumption in cloud computing platform based on random task. The modifications enabled CloudSim to create energy-conscious provisioning policies that require real-time knowledge of
power consumption by Cloud system components. Furthermore, it allowed the accounting of total energy consumed by the system during the simulation period. To make sure that VMs are protected against the discussed threats the thesis extended the CloudSim class and implemented a security module in it. Significant contribution of the thesis to the development of the framework is the modification of the framework to ensure that VMs are protected against attacks such as replay attacks during migration.
Keywords: OPTIMAL MANAGEMENT, ENERGY CONSUMING, CLOUDSIM PLATFORM, RANDOM TASK

1. Introduction

Cloud computing can be defined as "a type of parallel and distributed system consisting of a collection of inter-connected 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 consumers. Some of the examples for emerging Cloud computing infrastructures/platforms are Microsoft Azure, Amazon EC2, Google App Engine, and Aneka.

As a new computing mode, Cloud computing, it is the development of grid computing, parallel computing and distributed computing, as well as the new technology of next generation of Internet and application. Resource Scheduling Policy in Cloud computing is an important part of cloud computing technology, it mainly focus on how to allocate compute nodes for the task submitted by users, how to carry on the dynamic extension of the compute nodes in the case of meeting the requirements of service quality from customers and taking the shortest execution time to create the highest degree of load balancing, and its efficiency directly affects the performance of the entire cloud computing environment.

The ant colony algorithm and The particle swarm optimization are two swarm intelligence algorithms in the field of computational Intelligence, the former is based on the simulation of ant colonies to collect food, and the latter simulates the process of the flock seek for food. Ant colony algorithm in the experiments of solving traveling salesman problem, assignment problem, scheduling, etc, turns out to be practical. It highlights its efficiency and superiority in solving complex problems, especially in solving discrete optimization problems. Ant colony algorithm has a

great prospect of development in the future. Particle swarm optimization is a kind of efficient parallel search algorithm, whose concept is simple and the algorithm is easy to implement. And this algorithm is good at solving continuous optimization problems.

In cloud computing environment, task scheduling is a NP-completed problem. Simulated annealing, genetic algorithm and ant colony optimization, particle swarm intelligence optimization algorithms are very suitable for solving NP problems. Genetic algorithm, particle swarm optimization algorithm and ant colony algorithm have been used to cloud computing task scheduling problems. Literature [1] shows that the effective scheduling for independent tasks is completed by using genetic algorithm in the cloud computing environment. Literature [2] presents a cloud computing task scheduling algorithm based on particle swarm optimization, which optimizes task transfer time and processing time. Literature [3] uses ant colony algorithm to schedule cloud computing task, which make the total completion time and average task completion time minimum. But any algorithm has its advantages and disadvantages, such as the genetic algorithm is fast random global searching ability. But more parameters lead that programming is more complex and the algorithm is easier to fall into local optimum. And the convergence speed and local search ability of particle swarm optimization algorithm in the early is good. The late convergence speed is slow. But compared with genetic algorithm, convergence speed and optimization performance of particle swarm optimization algorithm is superior to the genetic method. What's more the programming is easier to implement and the parameters which need to adjust are less. Although the ant colony algorithm has good optimization ability, the initial pheromone is scarce and the convergence speed is slow. On the basis of the study, this paper puts forward a kind of particle swarm optimization and ant colony optimization based on cloud computing task scheduling algorithm. This algorithm absorbs the rapid convergence of particle swarm optimization algorithm and optimization ability of ant colony algorithm. The time of the system process scheduling problem and the total task execution time are reduced, which correspondingly improves the efficiency of the cloud computing task scheduling.

2. Layered Design

Figure 1 shows the layered design of Cloud computing architecture. Physical Cloud resources along with core middleware capabilities form the basis for delivering Infrastructure as a Service (IaaS) and Platform as a service (PaaS). The user-level middleware aims at providing SaaS capabilities. The top layer focuses on application services (SaaS) by making use of services provided by the lower layer services.

PaaS/SaaS services are often developed and provided by 3rd party service providers, who are different from the IaaS providers.

Cloud Applications: This layer includes applications that are directly available to end-users. The thesis defines end-users as the active entity that utilizes the SaaS applications over the Internet. These applications may be supplied by the Cloud provider (SaaS providers) and accessed by end-users either via a subscription model or a pay-per-use basis. Alternatively; in this layer, users deploy their own applications.

In the former case, there are applications such as Salesforce.com that supply business process models on clouds (namely, customer relationship management software) and social networks. In the latter, there are e-Science and e-Research applications, and Content-Delivery Networks. User-Level Middleware: This layer includes the software frameworks such as Web 2.0 Interfaces (Ajax, IBM Workplace) that help developers in creating rich, cost-effective user-interfaces for browser-based applications. The layer also provides those programming environments and composition tools that ease the creation, deployment, and execution of applications in clouds. Finally, in this layer several frameworks that support mufti-layer applications development, such as Spring and Hibernate, can be deployed to support applications running in the upper level.

Core Middleware: This layer implements the platform level services that provide runtime environment for hosting and managing User-Level application services. Core services at this layer include Dynamic SLA Management, Accounting, Billing, Execution monitoring and management, and Pricing. The well-known examples of services operating at this layer are Amazon EC2, Google App Engine, and Aneka. The functionalities exposed by this layer are accessed by both SaaS (the services represented at the top-most layer in Figure 1) and IaaS (services shown at the bottom-most layer in Figure 1) services. Critical functionalities that need to be realized at this layer include messaging, service discovery, and load balancing. These functionalities are usually implemented by Cloud providers and offered to application developers at an additional premium. For instance, Amazon offers a load balancer and a monitoring service (Cloud watch) for the Amazon EC2 developers/consumers.

Similarly, developers building applications on Microsoft Azure clouds can use the .NET Service Bus for implementing message passing mechanism.

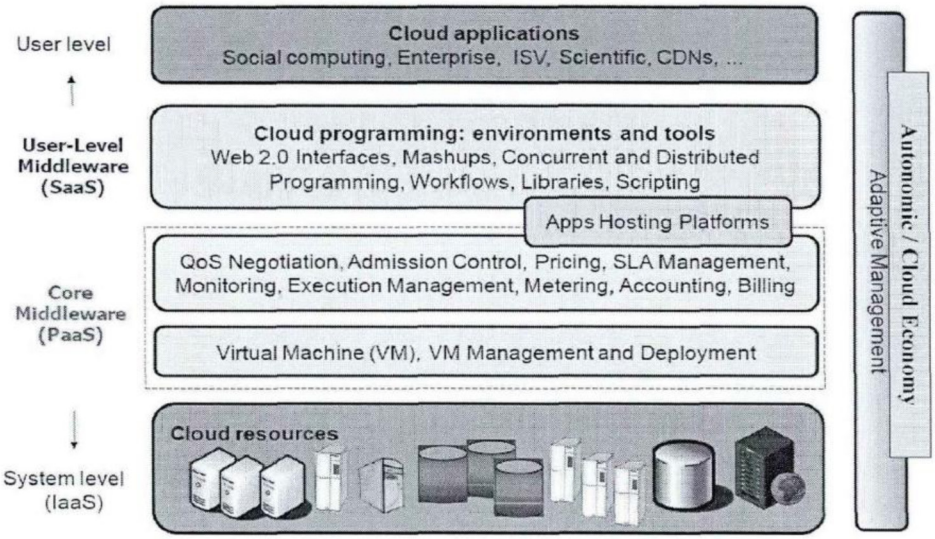

Figure 1. The layered model

System Level: The computing power in Cloud environments is supplied by a collection of data centers that are typically installed with hundreds to thousands of hosts [4-6]. At the System Level layer there exist massive physical resources (storage servers and application servers) that power the data centers. These servers are transparently managed by the higher level virtualization services and toolkits that allow sharing of their capacity among virtual instances of servers. These VMs are isolated from each other, thereby making fault tolerant behavior and isolated security context possible.

3. The Random Task Algorithm

One implication of Cloud platforms is the ability to dynamically adapt (scale-up or scale-down) the amount of resources provisioned to an application in order to attend variations in demand that are either predictable, and occur due to access patterns observed during the day and during the night; or unexpected, and occurring due to a subtle increase in the popularity of the application service. This capability of clouds is especially useful for elastic (automatically scaling of applications, such as web hosting, content delivery, and social networks that are susceptible to such behavior [7-8].

These applications often exhibit transient behavior (usage pattern) and have different Quality of Service (QoS) requirements depending on time criticality and users' interaction patterns (online/offline). A virtual machine (VM) is a software implementation of a machine (i.e. a computer) that executes programs like a physical machine. VM's are separated into two major categories, based on their use and degree of correspondence to any real machine. A system virtual machine provides a complete system platform which supports the execution of a complete operating system (OS). In contrast, a process virtual machine is designed to run a single program, which means that it supports a single process. An essential characteristic of a virtual machine is that the software running inside is limited to the resources and abstractions provided by the virtual machine-it cannot break out of its virtual world.

By the above derivation, then the basic algorithm expression is given:

$$u_{P1} = M\frac{E}{6}\sin\omega_s t + \frac{E}{3\pi}\sum_{m=1,2,\ldots}^{\infty}\sum_{n=\pm1,\pm3,\ldots}^{\pm\infty}\frac{J_n(mM\pi)}{m}\cos m(\pi-\alpha_1)\sin[(mF+n)\omega_s t] - \frac{E}{3\pi}\sum_{m=1,2,\ldots}^{\infty}\sum_{n=\pm1,\pm3,\ldots}^{\pm\infty}\frac{J_n(mM\pi)}{m}\sin m(\pi-\alpha_1)\cos[(mF+n)\omega_s t] \quad (1)$$

The initial phase angle of $u_{c2}^{'}$ is $\alpha_1 + 2\pi/3$. The initial phase angle of $u_{c3}^{'}$ is $\alpha_1 + 4\pi/3$. The double Fourier series expressions of u_{P2} and u_{P3} are given:

$$u_{P2} = M\frac{E}{6}\sin\omega_s t + \frac{E}{3\pi}\sum_{m=1,2,\ldots}^{\infty}\sum_{n=\pm1,\pm3,\ldots}^{\pm\infty}\frac{J_n(mM\pi)}{m}\cos m(\pi-\alpha_1-\frac{2\pi}{3})\sin[(mF+n)\omega_s t] - \frac{E}{3\pi}\sum_{m=1,2,\ldots}^{\infty}\sum_{n=\pm1,\pm3,\ldots}^{\pm\infty}\frac{J_n(mM\pi)}{m}\sin m(\pi-\alpha_1-\frac{2\pi}{3})\cos[(mF+n)\omega_s t] \quad (2)$$

$$u_{P3} = M\frac{E}{6}\sin\omega_s t + \frac{E}{3\pi}$$
$$\sum_{m=1,2,\ldots}^{\infty}\sum_{n=\pm1,\pm3,\ldots}^{\pm\infty}\frac{J_n(mM\pi)}{m}\cos m(\pi-\alpha_1-\frac{4\pi}{3})$$
$$\sin[(mF+\ n)\omega_s t] - \frac{E}{3\pi} \tag{3}$$
$$\sum_{m=1,2,\ldots}^{\infty}\sum_{n=\pm1,\pm3,\ldots}^{\pm\infty}\frac{J_n(mM\pi)}{m}\sin m(\pi-\alpha_1-\frac{4\pi}{3})$$
$$\cos[(mF+n)\omega_s t]$$

For

$$\sin m(\pi-\alpha_1)+\sin m(\pi-\alpha_1-2\pi/3) + \sin m(\pi-\alpha_1-4\pi/3)=0,$$

$$\cos m(\pi-\alpha_1)+\cos m(\pi-\alpha_1-2\pi/3) + \cos m(\pi-\alpha_1-4\pi/3)=\pm3 \text{ or } 0.$$

When m is the odd multiple of 3, the value is -3. When m is the even multiple of 3, the value is 3. When m is not the integral multiple of 3, the value is 0, then

$$u_A = u_{P1}+u_{P2}+u_{P3}$$
$$= M\frac{E}{2}\sin\omega_s t \pm \tag{4}$$
$$\frac{E}{\pi}\sum_{m=3,6,\ldots}^{\infty}\sum_{n=\pm1,\pm3,\ldots}^{\pm\infty}\frac{J_n(mM\pi)}{m}\sin[(mF+n)\omega_s t]$$

Although different types of PWM modulation algorithm have the difference, the output harmonic amplitude can be obtained by formula (5), (6), (7), and (8).

$$F(\omega t)=\sum_{-\infty}^{\infty}C_k e^{jk\omega t}dt \tag{5}$$

$$C_k = a_k + jb_k = \frac{1}{T}\int_{-T/2}^{T/2}F(t)e^{-jk\omega t}dt \tag{6}$$

$$F(t) = A_{00}/2 +$$
$$\sum_{n=1}^{\infty}\{A_{0n}\cos(n\omega_0 t)+B_{0n}\sin(n\omega_0 t)\}$$
$$+\sum_{n=1}^{\infty}\{A_{m0}\cos(n\omega_c t)+B_{m0}\sin(n\omega_c t)\} \tag{7}$$
$$+\sum_{m=1}^{\infty}\sum_{\substack{n=-\infty\\ n\neq0}}^{\infty}\{A_{mn}\cos(m\omega_c t+n\omega_0 t) + B_{mn}\sin(m\omega_c t+n\omega_0 t)\}$$

$$C_{mn} = A_{mn}+jB_{mn} =$$
$$\frac{1}{2\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}F(x,y)e^{j(mx+ny)}dxdy, \tag{8}$$
$$x=\omega_c t, y=\omega_0 t$$

The presented a comprehensive overview of current related work that has been done to address excessive energy consumption by data centers. A brief description of what other researchers have done and the methods they used to address the problem. While significant amount of work done provide energy efficient VM allocation, VM security has not been extensively been researched on with respect to energy consumption. The thesis strives to fill this gap that has been overlooked by other researchers. Hence, this paper is complementary to others, and offers some useful insights on server consolidation and security aspects in real world deployments. In the next section, the thesis presents solutions to previously stated problems, and it draws some research directions useful to better improve energy efficiency in data centers.

The second rule of Dynamic Round-Robin is that if a physical machine is in the "retirement" state for a sufficiently long period of time, instead of waiting for the hosting virtual machines to finish its jobs on its own, the physical machine will be forced to migrate the rest of active virtual machines to other physical machines. The machine is shut down after VM migration has been completed. This is achieved by invoking the workload controller functionality. The waiting time threshold is denoted as "retirement threshold". A physical machine which is in the retirement state but could not finish all virtual machines after the retire threshold has been exceeded, will be forced to migrate its virtual machines and shutdown. Dynamic Round-Robin method uses two basic rules in order to consolidate virtual machines deployed by the original Round-Robin method. The first rule is to avoid adding extra virtual machines to a retiring physical machine so it can be shut down. The second rule is to speed up the consolidation process and enable dynamic round-robin method to shutdown physical machines, so that it can reduce the number of physical machine used to run all virtual machines.

4. The Experiment and Data Analysis

This paper considered two sets of experiments to evaluate the proposed approach. The first set of experimental tests was done to test the approach in a two-tier (2T), three-tier (3T), and three-tier high-speed (3Ths) architectures. The second experiments that were done validate the effectiveness of energy-conscious VM provisioning technique proposed.

Proposed algorithms were evaluated through simulations using the CloudSim toolkit which offers the following novel features

(i) Support for modeling and simulation of large scale Cloud computing environments, including data centers, on a single physical computing node.

(ii) A self-contained platform for modeling Clouds, service brokers, provisioning, and allocations policies.

(iii) Support for simulation of network connections among the simulated system elements.

(iv) Facility for simulation of federated Cloud environment that inter-networks resources from both private and public domains, a feature critical for research studies related to Cloud-Bursts and automatic application scaling.

Some of the unique, compelling features that make CloudSim our chosen simulation framework are:

(i) Availability of a virtualization engine that aids in creation and management of multiple, independent, and co-hosted virtualized services on a data center node

(ii) Flexibility to switch between space-shared and time-shared allocation of processing cores to virtualized services.

The algorithm parameter Settings are as follows: in the part of particle swarm optimization (PSO) algorithm, the group scale size =100, c1 = c2 =2 and the number of iterations is 40. In the part of the ant colony algorithm, population scale size =100, α =1, β =1, ρ = 0.2 and the number of iterations is 160. The parameters of the particle swarm algorithm, ant colony algorithm are the same with the hybrid algorithm in this paper. The number of iterations is 200. And the algorithm runs repeatedly 20 times. The experimental results are shown in figure 2 and figure 3. Similarly, the execution time curve of the total task (10 recourses and 50 subtasks) is shown in figure 3 and the the execution time curve of the total task (10 recourses and 500 subtasks) is shown in figure 5. From the figure 4 and figure 5, we may get the conclusion that the number of resources does not affect the performance using different methods obviously.

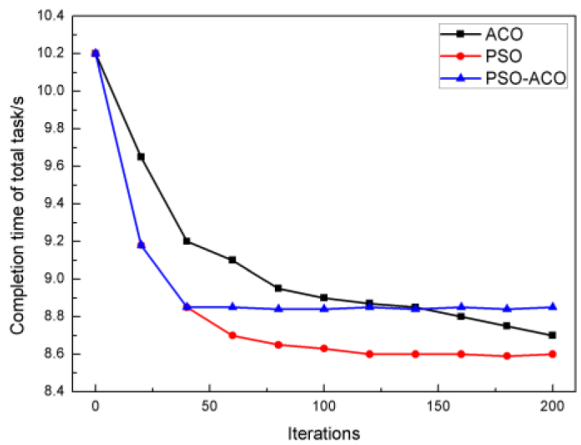

Figure 2. The execution time curve of the total task (5 recourses and 50 subtasks)

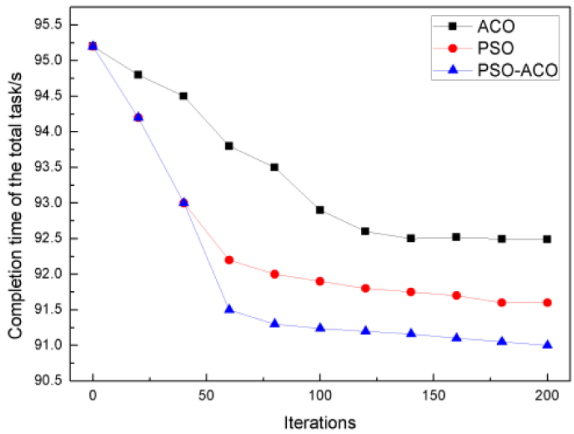

Figure 3. The execution time curve of the total task (5 recourses and 500 subtasks)

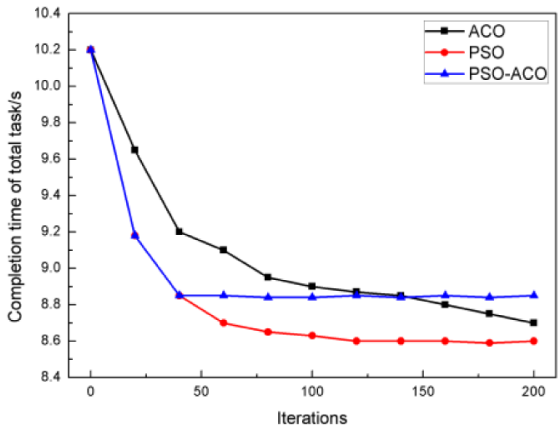

Figure 4. The execution time curve of the total task (10 recourses and 50 subtasks)

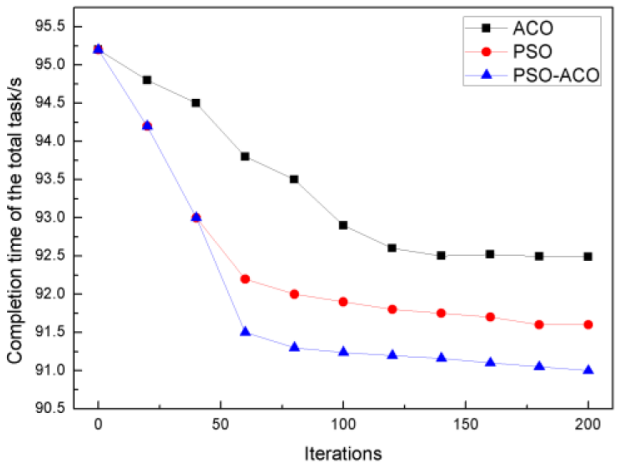

Figure 5. The execution time curve of the total task (10 recourses and 500 subtasks)

Conclusions

In this paper, the author researched on the optimal management method of energy consumption in cloud computing platform based on random task. The modifications enabled CloudSim to create energy-conscious provisioning policies that require real-time knowledge of power consumption by Cloud system

components. Although the ant colony algorithm has good optimization ability, the initial pheromone is scarce and the convergence speed is slow. On the basis of the study, this paper puts forward a kind of particle swarm optimization and ant colony optimization based on cloud computing task scheduling algorithm. This algorithm absorbs the rapid convergence of particle swarm optimization algorithm and optimization ability of ant colony algorithm. The time of the system process scheduling problem and the total task execution time are reduced, which correspondingly improves the efficiency of the cloud computing task scheduling.

References

1. Shangguang Wang, Zhipiao Liu, Qibo Sun et al. Towards an accurate evaluation of quality of cloud service in service-oriented cloud computing. Journal of Intelligent Manufacturing, 2014, 25(2), pp. 283-291.
2. Muhammad Irfan, Zhu Hong, Tauseef Qamer et al. Requirement Analysis and Design of Service Level Integration Layer for Cloud Computing Services, to Meet Service Level Agreements and Quality of Service. Journal of computational and theoretical nanoscience, 2014, 11(3), pp. 629-636.
3. Yan, G., Wen, D., Olariu, S. et al. Security Challenges in Vehicular Cloud Computing. IEEE transactions on intelligent transportation systems, 2013, 14(1), pp. 284-294.
4. Murugesan, San. Cloud Computing: The New Normal?. Computer, 2013, 46(1), pp. 77-79.
5. Abdul Nasir Khan, M.L. Mat Kiah, Sajjad A. Madani et al. Enhanced dynamic credential generation scheme for protection of user identity in mobile-cloud computing. Journal of supercomputing, 2013, 66(3), pp. 1687-1706.
6. Zou, D., Zhang, W., Qiang, W. et al. Design and implementation of a trusted monitoring framework for cloud platforms. Future generations computer systems: FGCS, 2013, 29(8), pp. 2092-2102.
7. Tao, F., Cheng, Y., Xu, L.D. et al. CCIoT-CMfg: Cloud Computing and Internet of Things-Based Cloud Manufacturing Service System. IEEE transactions on industrial informatics, 2014, 10(2), pp. 1435-1442.
8. Gutierrez-Garcia, J.O., Sim, K.M. Agent-based cloud service composition. Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies, 2013, 38(3), pp. 436-464.

Research on the Optimization Problem of the Manage Resource Distribution of the Particle Swarm Neural Network

Ji Luo, Ying Wang

School of Economic & Management, Sichuan University of Science & Engineering, Sichuan, 643000, China

Corresponding author is Ji Luo

Abstract

In this paper, the author mainly discusses the optimization problem of the manage resource distribution of the particle swarm neural network. The particle swarm optimization (PSO) is a population-based algorithm that was