Int. J. Communications, Network and System Sciences, **2011, 4, 803-811**
doi:10.4236/ijcns.2011.432098 Published Online December 2011 (http://www.SciRP.org/journal/ijcns)

IaaS Public Cloud Computing Platform Scheduling Model and Optimization Analysis

Aobing Sun[1,2,3], Tongkai Ji[1,2,3], Qiang Yue[1,2,3], Feiya Xiong[3]
[1]*Guangdong Electronic Industrial Institute, Dongguan, China*
[2]*Institute of Computing Technology of Chinese Science Academy, Beijing, China*
[3]*Sino-Cloud Science and Technology Stock Company Ltd., Dongguan, China*
E-mail: *sunaobing@gmail.com*
Received August 30, 2011; *revised September* 10, 2011; *accepted October* 14, 2011

Abstract

IaaS (*Infrastructure as a Platform*) public cloud is one mainstream service mode for public cloud computing. The design aim of one IaaS public cloud is to enlarge the hardware-usage of whole platform, optimize the virtual machine deployment and enhance the accept rate of service demand. In this paper we create one service model for IaaS public cloud, and based on the waiting-line theory to optimize the service model, the queue length and the configuration of scheduling server. And create one demand-vector based scheduling model, to filter the available host machine according to the match of demand and metadata of available resource. The scheduling model can be bonded with the virtual machine motion to reallocate the resources to guarantee the available rate of the whole platform. The feasibility of the algorithm is verified on our own IaaS public cloud computing platform.

Keywords: Cloud Computing, IaaS, Scheduling Model, Optimization Analysis

1. Introduction

Cloud Computing Technology is developed from virtualization, utility computing, IaaS (*Infrastructure as a Service*), PaaS (*Platform as a Service*), SaaS (*Software as a Service*) and etc. [1]. It puts forward one new IT business model, *i.e.* the users can acquire IT services through Internet with on-demand and expandable means. The cloud computing platform utilizes the high-speed Internet to deliver the computing, storage, software and services which are distributed all over the world, to the terminal users and make them to use the resources as electricity. The cloud computing technology brings us a new service mode to serve the users with data, application and IT resources through network [2].

Cloud computing technology is also one methodology for infrastructure, *i.e.* the cloud computing platform integrates the mass computing resources to compose one resource pool and serve the users dynamically with virtualized resources including computing, storage and service. To one user of cloud computing platform, almost everything as software, hardware, data and information service all can be rent from the cloud. The cloud computing platform can be subdivided into three layers shown as **Figure 1** [3].

***SaaS* (*Software as a Service*)** *i.e.* the software is delivered through Web browsers as a service of cloud computing platform, so the users can rent the software on demand. SaaS of cloud computing includes SaaS software and trusteed applications, e.g. Saleforce is one famous SaaS provider; it delivers ERP, SCM, CRM software and etc. through Internet with SaaS mode [3].

***PaaS* (*Platform as a Service*)** provides one platform for the users and developers with application development, test and deployment, e.g. one SaaS application. The platform includes database, middleware and development tools, and all services can be composed through Internet. For example, the Google Map platform and APP platform all are the PaaS cloud platform [2].

***IaaS* (*Infrastructure as a Service*)** is to provide the hardware infrastructure as servers, storage and hardware through Internet. The IaaS platform is created based on virtualization technology as server and storage virtualization, so virtualization, cluster and dynamic configuration software are also includes IaaS. e.g. EC2 of Amazon is one famous IaaS platform of cloud computing technology [1].

The cloud computing platforms own three types:

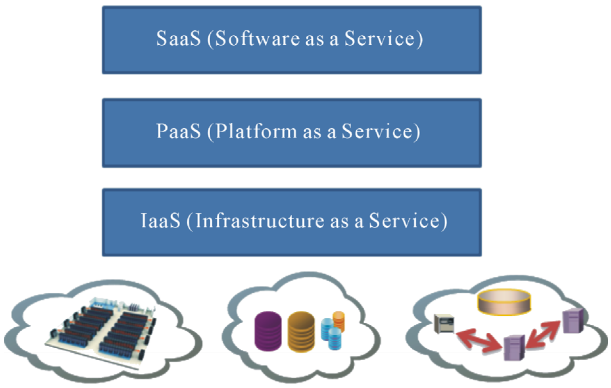

Figure 1. The layered structure of cloud computing platform.

Public Cloud serves the users that distributed all over the world across the border of enterprises and areas. Usually the public cloud platform is large-scale and composed by a few data centers in different area to provide IaaS, PaaS or SaaS service. e.g. Amazon EC2 is the IaaS public cloud, Google APP and Apple AppStore is the PaaS public cloud [3]. Public cloud serves the general users with on-demand mode, so the small enterprise users can create their IT business systems with low-cost. But the supervision of public cloud is very difficult, e.g. the resources of EC2 are used for spam mails, hack attack and Trojan attack [4].

Private Cloud only serves for one company or organization. Generally private cloud is composed by IT infrastructure of one enterprise. It contains their data center and all other devices, hardware and software linked in Internet. The private cloud is managed by the IT fellow and with high-level security. Private cloud demands the entire control of resources, and react the users with different priorities. So the users can have specific demands to resources. But generally, the pubic cloud looks the users with same priorities. The widely used private cloud includes VCloud, VSphere of VMware and XEN Cloud of Citrix [5].

Mixed Cloud owns the properties of public cloud and private cloud. It connects the resources of private clouds including its data, application and service through public cloud, e.g. private cloud connects into one public cloud and provide one access interface through one agent server. So it can guarantee the security of private cloud and support the permitted resources can be exposed into Internet. OpenNebula is one famous mixed cloud platform [3,6].

In this paper we describe our IaaS public cloud platform. The rest of the paper is organized as follow: Section 2 relates the service model of IaaS public cloud. Section 3 states our scheduling model for IaaS public cloud and its optimization means. Section 4 gives some experimental results. And Section 5 draws one conclusion and gives out future works.

2. Model of IAAS Public Cloud

IaaS public cloud is one important application mode of current cloud computing technology. With the appearance of Amazon EC2, more and more platforms come out to provide computing and storage resources. The aim of the platforms is to provide the users on demand with the virtual machines for ordered CPU frequency, quantity of core, storage space and memory size [4].

2.1. Element of IaaS Public Cloud

As shown in **Figure 2**, logically one IaaS public cloud owns three main elements as follow:

Cloud Administration Center is the access interface to Internet and also the management, scheduling and monitoring center of the resources within the cloud. The administration center of one IaaS public cloud accepts the resources request from the Internet users and create the demanded resources, e.g. virtual machine and storage resources, and configure them, then return the resources to the users.

Cloud computing Resources Center is composed by the physical computing resources. To one IaaS platform, the physical resources will be used as the host machines to be administrated by the cloud administration center. The scheduling server will select the optimal resources according to the user demands to create virtual machines. In general, multiple cloud computing resource centers access the administration center with agent servers which can also be used to support the monitoring and scheduling of the computing resources.

Cloud Storage Resources Center is composed logically by the physical storage resources. To one IaaS platform, virtual machine template, images and snapshots are also stored in the storage center which is administrated with network storage systems as NFS, S3, ISCSI and etc. The virtual machine image of users is transferred to one specific physical machine from the storage center and then is loaded into it. To the platform, the physical and virtual machines are loosely coupled. And it also is the difference of public and private cloud.

Service Workflow

We will use two operation flows to analyze the scheduling flow of cloud platform. The flow of user request to resources is as follow:

1) The registered users access the portal server and request the virtual machines with the parameters including quantity of core, frequency, memory, storage space, OS and etc.

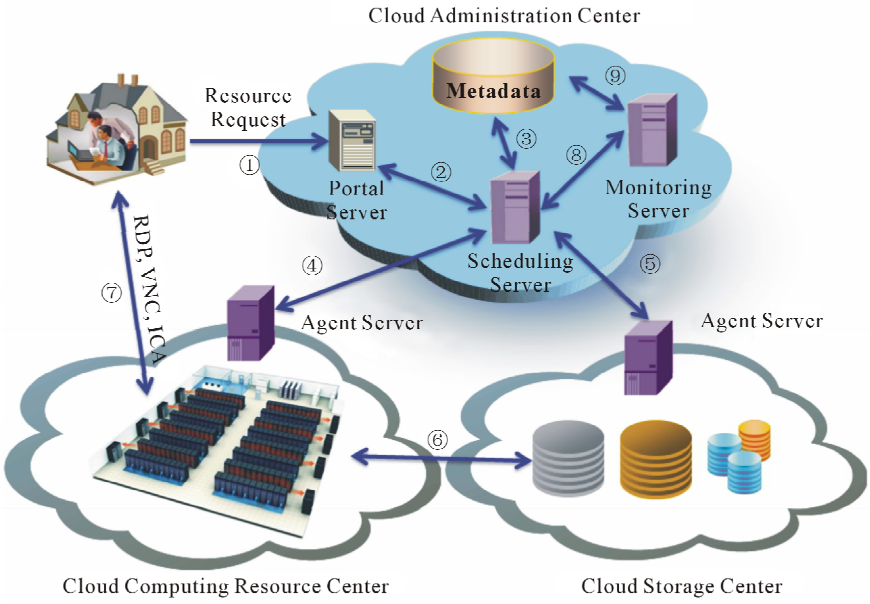

Figure 2. The elements of IaaS public cloud.

2) Portal Server sends the request to the scheduling server.

3) The scheduling server searches the physical machines to find the host to create the virtual machine according to the metadata of physical machine, which records the operation and configuration details.

4) The scheduling server chooses one optimal server and then sends the creating command of virtual machine to its agent server.

5) The scheduling sever chooses the virtual machine template form the stored templates within cloud storage administration center, and sends one request for the template to the agent server.

6) The requested virtual machine image will be sent (or mapped) to the physical sever based on the template, the scheduling sever will start the virtual machine if the image is loaded successfully. If something is wrong during (4)-(6), the scheduling server will select new virtual machine.

7) If the virtual machine starts successfully, the user can access the virtual machine through RDP, VNC, ICA or SSH.

The agent server can monitor the resources registered within the computing or storage resources management center. It will renew the metadata within the metadata database to guarantee the correctness of scheduling server to resources. The renewing of metadata follows 2 steps:

8) The monitor server will send the resource-information renew request, the scheduling server will send the request to the agent servers. If the agent severs acquire the information then send them to the monitoring server.

9) The monitoring server will renew the information within the metadata database, to guarantee its correctness, and to improve the efficiency of scheduling operations.

2.2. The Service Model of IaaS Public Cloud

According to the service flow, we can abstract one IaaS Public cloud as the model as shown in **Figure 3**. The model including three flows as follow:

1) The scheduling severs of cloud administration center, picks out the request *R* with the highest priority. The scheduling server then judges whether *R* can be meted according to the parameters of *R*, as CPU frequency, core quantity, band-width, storage, and disk space. If *R* can be met, then jump to (2). Or then judges if *R* can be met through the VM (*Virtual Machine*) motion and then release enough resources; If one motion can release enough resources, then jump to (2). Or then quit directly and report to the user that the requested resources cannot be met.

2) If the requested resources can be met, the scheduling server then choose the VM template *T* (if create one new VM) or one VM image *I* (if use existing VM) corresponding to *R*.

3) The scheduling server sends *I* to corresponding physical machine and create VM instance *V*.

There are three important problems to the model:

1) How the request-queue length is determined, and the priority of request *R* can be adjusted, to guarantee all the request can be reacted quickly.

2) How the request *R* can be parsed and then to select the optimal resource to serve users.

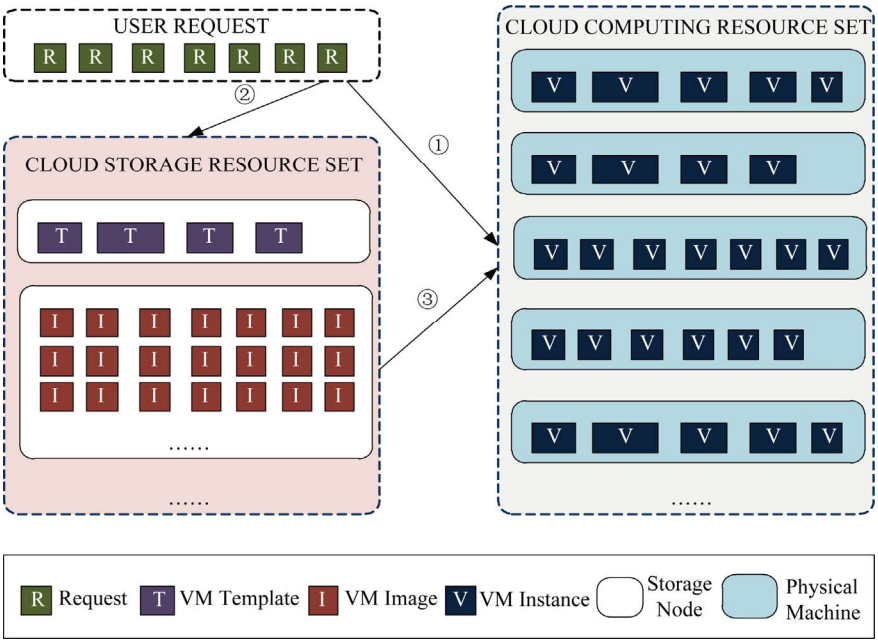

Figure 3. IaaS public cloud service model.

3) If VM motion operations are demanded, and how to guarantee the motion cost and the affection to the other VMs are minimized. To one platform the motion can affect the QoS of whole platform, so the motion of VM will be executed only when other VMs will not be disordered.

3. Scheduling Model of IAAS Public Cloud

According to the service model, we can quantize the parameters and analyze the throughout of one cloud platform, and then analyze and optimize the model [6].

3.1. Queue Model

According to the waiting-line theory, as shown in **Figure 4**, the request and react processing is one waiting-line system, the input of one waiting-line system is the request and the service counter is the scheduling server, and the output is the requested resources. One user request queue is $R=\{R_1,R_2,R_3,\cdots,R_n\}$.

We assume the request come as Poisson's ratio within one IaaS public cloud, and the service time is as exponential distribution. λ is the count of coming user request averagely in one unit of time. μ is the service efficiency (the ability of service counter). $\rho = \lambda/\mu$ is the ration that the request can be met within one unit of time, *i.e.* the service success rate. W_S is the time that one service will wait in the system, which includes the waiting time and service time. W_q is the average waiting time for user request. If there only is one counter (*i.e.* one scheduling server), so $W_S = 1/(\mu-\lambda)$, $W_q = \rho/(\mu-\lambda)$. So there are two means to increase the user request reaction speed:

1) decrease the count of user request sent to scheduling server.

2) Increase the processing speed of scheduling server.

So we can increase scheduling server within one IaaS public cloud. When there are several scheduling server, the results are as follow equations.

$$W_S = \frac{(k\rho)^k \rho}{k!(1-\rho)^2 \rho} T_0 + \frac{1}{\mu} \tag{1}$$

$$W_q = \frac{(k\rho)^k \rho}{k!(1-\rho)^2 \lambda} T_0 \tag{2}$$

$$T_0 = \left[\sum_{n=0}^{k-1} \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n + \frac{1}{k!(1-\rho)} \left(\frac{\lambda}{\mu}\right)^k \right]^{-1} \tag{3}$$

Multiple-queue and multiple-counter can be seen as several single-queue and single-counter. So based on the analysis, to control the reaction speed of system, the maximum of queue length will be fixed. When the waiting request surpass it, the requests out of the queue will be rejected. So the maximum reaction speed of user request is the whole queue processing time [7].

3.2. Model Analysis

The set of physical machine within one cloud in P, $P=\{P_1,P_2,P_3,\cdots,P_n\}$. n is the count of physical machine within P. All the parameters are as shown in **Table 1**.

Table 1. Notation of quantized scheduling model.

Notation	Presentation
P_i	Anyone physical machine within P
C_i	Sum of allocable CPU core of P_i
F_i	Sum of allocable CPU frequency of P_i
M_i	Sum of allocable memory of P_i
B_i	Sum of allocable network bandwidth of P_i
D_i	Sum of allocable disk space of P_i
V_i	*VM* set running on P_i
$F(V_i)$	Sum of used CPU frequency of V_i
$C(V_i)$	Sum of used CPU core of V_i
$M(V_i)$	Sum of used memory of V_i
$D(V_i)$	Sum of used disk space of V_i
$B(V_i)$	Sum of used network bandwidth of V_i

To user request R, the resource allocation must follow the rules which are also the necessary rules of one IaaS public cloud.

1) To one single VM, anyone of the allocated resource to v_{ij} of V_i (as frequency, core quantity, disk space and bandwidth) will be less than the total resources of P_i, *i.e.* $\forall f(v_{ij}) < F_i$, and $\forall c(v_{ij}) < C_i$, and $\forall m(v_{ij}) < M_i$, and $\forall d(v_{ij}) < D_i$, and $\forall b(v_{ij}) < B_i$. $f(v_{ij})$, $c(v_{ij})$, $m(v_{ij})$, $d(v_{ij})$ and $b(v_{ij})$ is the allocated frequency, core, memory, disk and bandwidth to VM v_{ij}.

2) The sum of the allocated resources to the virtual machines within VM set V_i will be less than the total of physical machine P_i, *i.e.* $F(V_i) < F_i$, and $C(V_i) < C_i$, and $M(V_i) < M_i$, and $D(V_i) < D_i$, and $B(V_i) < B_i$.

As shown in **Figure 5**, assuming the user request R_i can be parsed into CPU frequency request RF_i, CPU core request RC_i, disk request RD_i, memory request RM_i, network bandwidth request RB_i. The scheduling server firstly goes through the physical machines within the metadata record, and to find the physical-machine set that can meet the VM request. Then sort the physical machines according to its usage. The VM will be created on the physical machine with the lowest usage. The usage is one powered remark including CPU frequency, memory and bandwidth usage. Generally the CPU usage can be

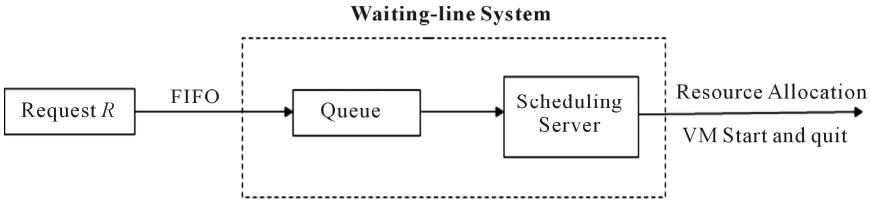

Figure 4. Waiting-line model of IaaS public cloud request.

Find Physical Computer (Request R, Computer Set P)
{empty (PM); //Empty the result set P of physical machine
for$(\vee P_i \in P)$
{if $\left(RF_i < (F_i - F(V_i))\right) \cap \left(RC_i < (C_i - C(V_i))\right) \cap$
$\left(RM_i < (M_i - FM(V_i))\right) \cap \left(RD_i < (D_i - D(V_i))\right) \cap \left(RB_i < (B_i - B(V_i))\right)$
//Remained resource of physical machine can meet the request
then P_i insert PM; //P_i insert into the queue
}
for$(\vee P_i \in$ PM$)$//To anyone physical machine in PM
{if $\left(U(P_i)\right) > \left(U(P_{i-1})\right)$
//If the usage of P_i is more than P_{i-1}
then $P\langle - \rangle P_{i-1}$; //$P_i$ and P_{i-1} exchanged within the set
}//Sort the physical machine within PM according to Usage
return PM;
}

Figure 5. Algorithm to find physical machine to host one VM.

used as the main indicator of total usage.

One public cloud platform can release resources through VM motion to meet one request. Because one VM motion will decrease the QoS of VMs within same physical machine, the platform will decrease the possible VM motions and use at the most one motion to meet the resources release demand. When the physical machine within one physical machine set all cannot meet the demand, the scheduling server firstly find two physical-machine with the lowest usage, and attempt to move the VM with the lowest usage to another physical machine to release resources. To the physical machine, if one step motion cannot release enough resources, the user request will be refused. The algorithm to release resources is as shown in **Figure 6**.

4. Experiment

G-Cloud v3.0 is one IaaS public cloud computing platform developed by GDEII (Guangdong Electronic Industrial Institute). In this paper, we use one group of experiment to verify our scheduling model. The result is as shown in **Table 2**.

The experimental platform owns 100 physical machines as the host, and one host can create 8 VM at most. We change the length of request queue and put forward the VM creation request to surpass the length to test our algorithm. We can see from **Table 2** that with the improved multi-scheduler means the minimum reaction time will not be affected by the length of the request queue with FIFO (First In First Out) Mode. But the average reaction time and the maximum reaction time will be enlarged with the increase of the queue length as shown in **Figure 7(a)**, especially when the queue length surpasses 40 s. And the maximum waiting time will surpass 120 s which is over the user-enduring time. So within our platform, the request queue length is 40. Contrasted with the general means, only with one scheduler

Motion Physical Computer (Request R, Computer Set P)
{empty (PM); //Empty the result set PM
for$\left(\vee P_i \in P\right)$//To anyone physical machine within P
{if $\left(U\left(P_i\right)\right) > T_1$)
//If the usage of P_i is less than threshold T_1
then P_i insert PM; //Insert P_i into PM
//Find the VM whose usage is less than Threshold
}
for$\left(\vee V_{ij} \in V_i\right) \cap \left(\vee P_i \in \mathrm{PM}\right)$
//To any physical machine of PM
{if $\left(U\left(V_{ij}\right)\right) > T_2$)
//If the usage of V_{ij} is less than threshold T_2
then V_{ij} insert VM; //Insert V_{ij} into VM
//Find VMs that are movable to aim physical machine
}
for$\left(\vee V_{ij} \in \mathrm{VM}\right)$//To any virtual machine of VM
{if (Motionabled$\left(V_{ij}, P_{i-1}\right)$ is True)
{//If V_{ij} is movable to one physical machine
PreMotion$\left(V_{ij}, P_{i-1}\right)$//Pre-move one VM logically
if $\left(RF_i < \left(F_i - F\left(V_i\right)\right)\right) \cap \left(RC_i < \left(C_i - C\left(V_i\right)\right)\right) \cap$
$\left(RM_i < \left(M_i - FM\left(V_i\right)\right)\right) \cap \left(RD_i < \left(D_i - D\left(V_i\right)\right)\right) \cap \left(RB_i < \left(B_i - B\left(V_i\right)\right)\right)$
//If requested resources can be met after motion
{Motion$\left(V_{ij}, P_{i-1}\right)$//Move the VM really
return true; //Return Success
}
}
return false;//Cannot find motion aim, return false
}

Figure 6. Algorithm to find VM for motion.

Table 2. Queue length and reaction tims(s) with multi-scheduler.

Queue Length	Minimum Reaction Time	Average Reaction Time	Maximum Reaction Time
10	1.4	12.3	26.4
20	1.35	17.8	30.21
30	1.41	19.1	39.11
40	1.54	26.7	42.11
50	1.3	31.8	55.11
60	2.1	46.6	65.34
70	1.3	50.2	120.12
80	1.5	60.3	165.6
90	2.1	75.9	170.4
100	2.3	99.0	220.1

and without queue length adjust algorithm the average reaction time will increase with very high speed and surpass 40 s with only 20 jobs in queue as shown in **Figure 7(b)**. So it is very important to construct more schedulers and the queue length should have one maximum and can adjust dynamically.

When the VM capacity of the whole platform surpasses 80% and if we create new VM or reconfigure one VM, the resources will not be enough to create it. So the platform will move some VM to release resource to meet the demand. The VM motion will be affected by the VM image, memory and storage size, and the backup, scheduling and motion algorithm. Out platform adopts Hot Motion means (related within **Figures 5** and **6**), *i.e.* the VM is moved when it is running. But the data coherence cannot be kept easily. After the VM image is synchronized, the VM will stop service for several seconds to synchronize the runtime memory. After image synchronization, the original VM will be closed and the new VM

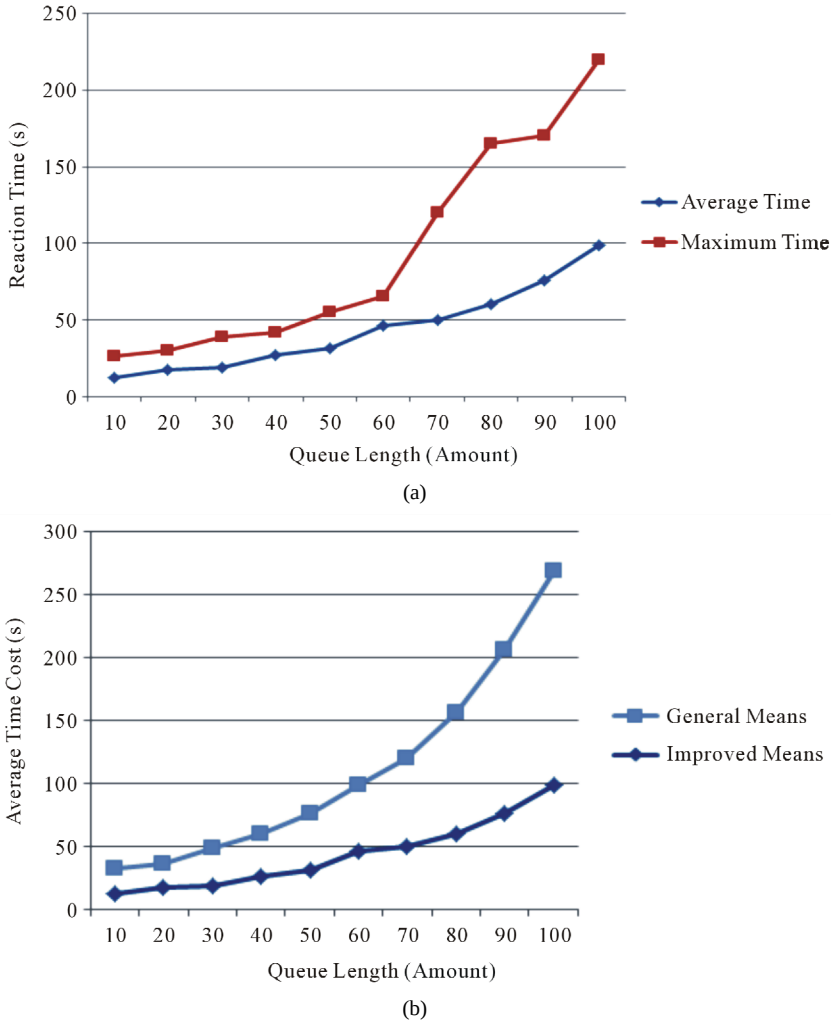

Figure 7. Queue length and reaction time. (a) Improved means with multi-scheduler; (b) Contrast of general and improved means.

will start service. Hot motion will affect the online users and the QoS. But it can keep service for online users after hot motion.

Contrasted with our algorithm, there are the cold motion and clone motion means. Cold motion means the VM is moved after it is closed. Clone Motion, *i.e.* the VM motion will only move the image files and the runtime memory file will be abandoned. So the QoS will be affected. And the Clone motion only is used when the VM will have no data renew. And the online users will be disconnected during clone motion.

We usually use one time-cost to quantity the motion cost for VM motion. It includes the cost for VM image and memory files transferring, reconfiguration and restart VM. But the service stop time is the main scale for the VM motion of one IaaS public cloud platform. We can see from **Figure 8** that the hot motion will cost more time than cold motion and clone motion. Because it need more time to avoid the service delay time and use more time to synchronize the VMs. But the hot motion will cause the minimum delay time than cold motion and clone motion as shown in **Figure 9**.

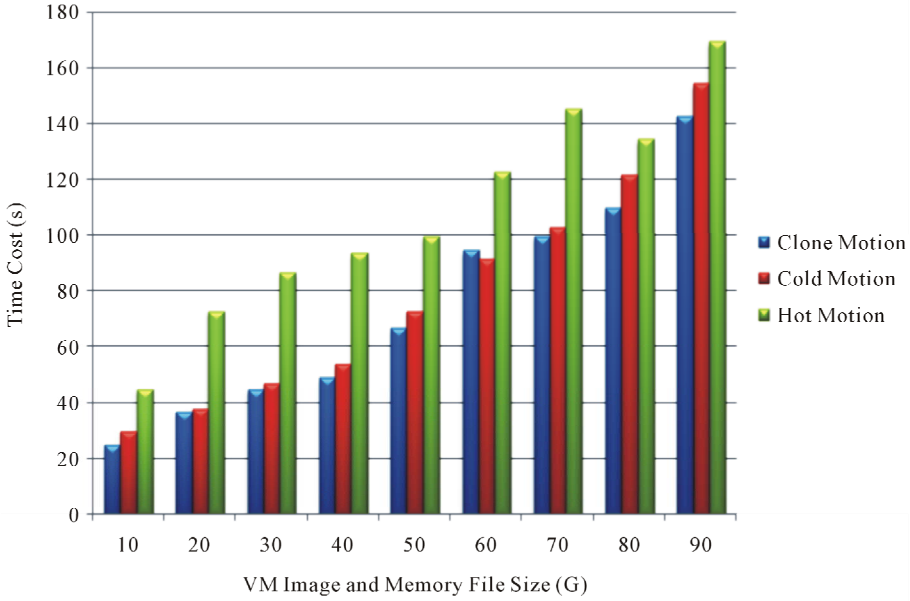

Figure 8. Total time cost for different motion mode.

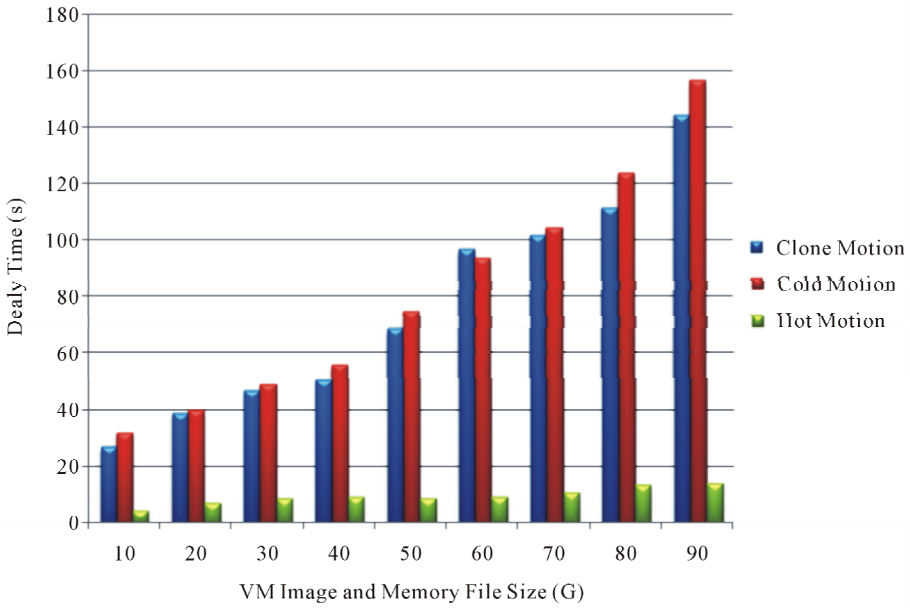

Figure 9. Service delay time for different motion mode.

5. Conclusions and Future Works

IaaS public cloud aims to provide available VMs for Internet users. In this paper, we summarize the IaaS public cloud model, and analyze the service flow according to the waiting-line theory. And aiming to maximize the platform usage and the performance of single VM, we give out one filtering algorithm based on user request to find optimal resource for user VM request [8]. The cloud administration center renews the metadata on time to support the virtual machine motion and physical machine scheduling [9]. The algorithm is verified on our G-Cloud platform of GDEII, which improves the QoS of the whole platform.

6. Acknowledgements

This work is partially supported by the Strategic Cooperation Project of Guangdong Province and Chinese Science Academy Grant # 2009A0091100002 and 2010-A090100004; Supported by Guangdong and Hong Kong invited bidding special for Dongguan Grant # 2011-20510101, 201120510106 and 201120510104; And supported by Dongguan Major Science and Technology Special Project Grant # 2009215102001.

7. References

[1] K. Chen and W. M. Zheng, "Cloud Computing: System Instance and Current State," *Journal of Software*, Vol. 20, No. 5, 2009, pp. 1337-1348. doi:10.3724/SP.J.1001.2009.03493

[2] Z. W. Xu, H. M. Liao, *et al.*, "The Classification Research of Network Computing System," *Journal of Computing Machine*, Vol. 18, No. 9, 2008, pp. 1509-1515.

[3] G. W. Zhang, R. He and Y. Liu, "The Evolution Based on Cloud Model," *Journal of Computing Machine*, Vol. 7, 2008, pp. 1233-1239.

[4] R. Buyya, Y. S. Chee and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering," 10*th IEEE International Conference on High Performance Computing and Communications*, Dalian, 25-27 September 2008, pp. 5-13.

[5] T. Berners-Lee, *et al.*, "Creating a Science of the Web," *Science*, Vol. 313, No. 5788, 2006, pp. 769-771. doi:10.1126/science.1126902

[6] A. Kemal Delic, "On Dependability of Corporate Grids," *Ubiquity*, Vol. 6, No. 45, 2005, p. 1. doi:10.1145/1113164.1113162

[7] K. Delic and M. Walker, "Architecting Enterprise Grids: Possible Inflection Points", *IADIS International Conference on Applied Computing*, Salamanca, 18-20 February 2007, pp. 113-121.

[8] I. Foster, "Service-Oriented Science," *Science*, Vol. 308, No. 6, 2005, pp. 814-817. doi:10.1126/science.1110411

[9] E. Hand, "Head in the Clouds," *Nature*, Vol. 449, No. 24, 2007, pp. 963-970. doi:10.1038/449963a

