A Novel benefit model for virtual machine resources allocation

Jiang Huowen[1,a] Zhang Huiyun[2,b] Xiong huanliang[3,c]

[1]Collage of Mathematics & Computer Science, Jiangxi Science & Technology Normal University, Nanchang 330038, China

[2]JiangXi Water Resources Institute

[3]Software College, Jiangxi Agricultural University

[a]Jhw_604@163.com [b]ashzhhy@126.com [c]xionghuanliang@126.com

Keywords: benefit model, resources allocation, management optimization

Abstract. Virtual machine technology is widely used in cloud computing. This paper put forward a benefit model aimed at optimal allocation of virtual machines of cloud data center. Differentiate service levels according to the service response time, calculate profit and fines according to the satisfaction conditions of different service levels. When a service request is responded and completed within the specified time, then the profit of the service is calculated; if not, then the fines is calculated. Evaluate comprehensively the total benefits of service based on value of service benefit and fines calculated during a certain period of time. Dynamically adjusting resource allocation with the objective of maximizing total benefits is conducive to the realization of autonomous configuration of resources and to the improvement of the overall efficiency of the service.

Introduction

To cope with the ever increasing complexity of computer systems, IBM put forward the concept of autonomic computing in 2001, inspired by the human body's complex autonomic nervous system [1]. Autonomic computing system can regulate and protect itself in the absence of intervene, and constantly adapt to the changing environment under the premise of maintaining the specified target. In recent years, autonomic computing has attracted widespread attention and made some progress [2~3].

As one of technologies of shielding the heterogeneity of software and hardware of platform computing, virtualization technology has been widely used on shared computing infrastructures [4]. It can also promote the differentiation, isolation and convenience of management of customers' applications. Because virtualization technology and virtual computing environment provides a natural support for autonomic computing systems, autonomic computing under the virtual computing environment enjoys a booming development. Literature [5] studied virtual environments in which virtual machines are connected by a virtual network under the multi-domain shared infrastructure, also designed and implemented the overall adaptive architecture of a virtual computing environment. Literature [6] studied how to allocate processing resources for different virtual machines dynamically using the autonomic computing technology when load changes. This paper studies the optimal management of virtual resources for virtual clusters, and proposes a benefit model which facilitates the optimization of the resource allocation.

System resources and service description

In the physical sense, computing resources of cloud computing center are mainly composed of server clusters. Computing resources to user experience is a virtual machine, built on server clusters by virtualization technology. Each virtual machine shares available IT resources to provide Internet services for different customers, and processes a large number of requests from different applications and customers. Virtual machines can run common operating systems and various applications, like Xen, VMWare etc.. Cloud computing center provides m ($m \geq 2$) kinds of different categories of application environment, represented respectively by $E_1, E_2, \cdots, E_m$, each of whom is served by different number of virtual machines together. For each application request from a variety of application environments,

the virtual machine provides n ($n \geq 2$) different levels of Service Level Agreement(SLA), represented respectively by $L_1, L_2, \cdots, L_n$. Different virtual service levels corresponding to different resource requirements are also different. The total number of virtual machines within a system constantly changes along with the fluctuation of service requests.

This paper's work facilitates the above virtualized service to provide a regulation mechanism of environment resource optimization, and builds a benefit model that maximizes the benefit of service provider based on the balance of resource use costs. The gross profit includes not only the profit when meets the service with quality guaranteed and fines when does not, but also the various costs of maintaining system operations.

Dynamic service description

Each virtual machin is assumed to be in an initial state at time t_0 , and to regulate virtual resources at time $t_1, t_2 \cdots, t_k, t_{k+1}, \cdots$ in order to optimize system resources. To simplify the problem, this paper only consider optimization when time intervals are equal, i.e. $(t_1 - t_0) = (t_2 - t_1) \cdots = (t_{k+1} - t_k) = \cdots$. And $tq_1, tq_2, \cdots, tq_k$ represent the resource adjustment time interval $t_0 \quad t_1, t_1 \quad t_2, \cdots, t_k \quad t_{k+1}$ respectively. The service level of request is assumed to be unchanged during the service, then the number of service requests of application environments E_i at time t_k corresponding to different levels of service can be represent as a matrix $NA(t_k)$:

$$NA(t_k) = \begin{pmatrix} na_{1,1}(t_k) & \cdots & na_{1,j}(t_k) & \cdots & na_{1,n}(t_k) \\ \vdots & \ddots & \vdots & & \vdots \\ na_{i,1}(t_k) & \cdots & na_{i,j}(t_k) & \cdots & na_{i,n}(t_k) \\ \vdots & & \vdots & \ddots & \vdots \\ na_{m,1}(t_k) & \cdots & na_{m,j}(t_k) & \cdots & na_{m,n}(t_k) \end{pmatrix}$$

where $na_{i,j}(t_k)$ and t_k represents respectively the number of and total number of the service requests of application environment E_i at time t_k corresponding to L_j service level.

The resources of application environment E_i at time t_k are provided by different number of virtual machines distributed on physical server $S_1, S_2, \cdots, S_i, \cdots S_s$. $nv_{i,j}(t_k)$ is assumed to be represented the number of virtual machine that provide resources for application environment E_i at time t_k on physical server S_j , then the distribution matrix $NV(t_k)$ of virtual machine at time t_k can be represented as follow:

$$NV(t_k) = \begin{pmatrix} nv_{1,1}(t_k) & \cdots & nv_{1,j}(t_k) & \cdots & nv_{1,s}(t_k) \\ \vdots & \ddots & \vdots & & \vdots \\ nv_{i,1}(t_k) & \cdots & nv_{i,j}(t_k) & \cdots & nv_{i,s}(t_k) \\ \vdots & & \vdots & \ddots & \vdots \\ nv_{m,1}(t_k) & \cdots & nv_{m,j}(t_k) & \cdots & nv_{m,s}(t_k) \end{pmatrix}$$

Therefore, we can draw the conclusion that $\sum_{j=1}^{s} nv_{i,j}(t_k)$ represents the number of virtual machine that provide resources for application environment E_i at time t_k , and $\sum_{i=1}^{m} nv_{i,j}(t_k)$ represents the number of virtual machine deployed on physical server S_j at time t_k .

The physical state of the server is divided into operating state and sleeping state, and $nr(t_k)$ represents the number of physical server in operating state, and $nt(t_k)$ represents the number of

physical server in sleeping state, so $nr(t_k)+nt(t_k)=s$. The virtual machines that provide service request to allocate virtual resource distributed on $nr(t_k)$ physical servers.

Virtual resources description

A variety of resources including CPU, memory, disk, network bandwidth, etc. provide service for request service. To simplify the problem, we only use virtual resources to represent a variety of resources to each service request. Obviously, the resources taken to complete the service are similar with regard to the same application environment and the same service level in a service request.

The average virtual resources taken by a service request of L_j level of service in application environment E_i is assumed to be represented as $R_{i,j}$.

The virtual resources taken by all service requests of application environment E_i at time t_k can be represented as $\sum_{j=1}^{n}(R_{i,j}\cdot na_{i,j}(t_k))$. The total virtual resources in occupancy of a system can be represented as $\sum_{i=1}^{m}\sum_{j=1}^{n}(R_{i,j}\cdot na_{i,j}(t_k))$.

Different application environments have different virtual resource needs. With regard to application environment E_i, the capacity of available virtual resource that single virtual machine can provide for different service requests are assumed to be represented as $VR_1,VR_2,\cdots,VR_i,\cdots,VR_m$ respectively. In addition to providing resources to the virtual service requests, virtual machine itself also consumes resources. The additional resources consumed by each virtual machine are are assumed to be represented as $OR_1,OR_2,\cdots,OR_i,\cdots,OR_m$, then the capacity of virtual resources with regard to application environment E_i at time t_k can be represented as $\sum_{j=1}^{s}(VR_i\cdot nv_{i,j}(t_k))$. Obviously, $\sum_{j=1}^{n}(R_{i,j}\cdot na_{i,j}(t_k))\le\sum_{j=1}^{s}(VR_i\cdot nv_{i,j}(t_k))$. The resources occupied by application environment E_i at time t_k can be represented as $\sum_{j=1}^{s}((VR_i+OR_i)\cdot nv_{i,j}(t_k))$, and the total virtual machine resources in occupancy on physical server S_j can be represented as $\sum_{i=1}^{m}((VR_i+OR_i)\cdot nv_{i,j}(t_k))$.

The capacity of resources provided by each physical server is are assumed to be represented as TR, then the formula $\sum_{i=1}^{m}((VR_i+OR_i)\cdot nv_{i,j}(t_k))\le TR$ holds at any given time t_k. The total number of resource at time t_k can be represented as $r(t_k)=TR\cdot na(t_k)$. Obviouly, here $r(t_k)=TR\cdot na(t_k)=\sum_{i=1}^{m}\sum_{j=1}^{s}((VR_i+OR_i)\cdot nv_{i,j}(t_k)$ and $\sum_{i=1}^{m}\sum_{j=1}^{n}(R_{i,j}\cdot na_{i,j}(t_k))\le r(t_k)$.

Benefit model for resources allocation

To simplify the problem, the service level is assumed to be only related to the service response time. With regard to L_i level of service, its service response time threshold is represented as RT_{L_i}. For each service request finished within the grade of service, its service response time is represented as rt and always have the benefit PRO_{L_i}, then we can conclude the following formula:

$$PRO_{L_i}=\begin{cases} BEN_{L_i} & rt\le RT_{L_i} \\ BEN_{L_i}-PEN_{L_i} & rt>RT_{L_i} \end{cases},\quad BEN_{L_i}\le PEN_{L_i}$$

where BEN_{L_i} and PEN_{L_i} represent the benefit of finishing service and fines for response time exceeding the threshold respectively.

From section 2.2, we know that the resources occupied by service request of L_j service level of application environment E_i is represented as $R_{i,j}$, and its benefit is represented as PRO_{L_j}. The ratio of benefit and resources is assumed to be represented as θ, and $\theta_{i,j}^{+}$ for the benefit when $rt \le RT_{L_j}$, and $\theta_{i,j}^{-}$ for the benefit when $rt > RT_{L_j}$. Obviously, here $\mu_{i,j}^{+} = BEN_{L_j} \big/ R_{i,j}$, $\theta_{i,j}^{-} = \left(BEN_{L_j} - PEN_{L_j} \right) \big/ R_{i,j}$.

We cannot use the difference between the number of service requests at time t_{k+1} and that of at time t_k due to the following three reasons: (1) service requests arrive dynamically; (2) the length of service time is not certain; (3) the finishing time of service is not certain. $nqa_{i,j}\left(tq_k\right)$ is assumed to be represents the number of L_j level of service requests of application environment E_i within tq_k, then the total number of finished L_j level of service requests can be represented as . Therefore, the total service benefit obtained within tq_k can be represented as $\sum_{i=1}^{m} nqa_{i,j}(tq_k)$.

Considering the required cost of operation management, the operating cost for each single physical server when regulating resources during tq_k is represented as $oqs\left(tq_k\right)$, which includes the energy charge, the sharing cooling cost of air conditioning, the sharing management costs and other costs. The cost of single physical server in sleeping state is represented as $oqd\left(tq_k\right)$. Obviously, here $oqd\left(tq_k\right)$ $oqs\left(tq_k\right)$ .Then the cost within tq_k is represented as $cos\left(tq_k\right) = oqs\left(tq_k\right) \cdot nr\left(t_k\right) + oqd\left(tq_k\right) \cdot nt\left(t_k\right)$. Therefore, the total benefit within tq_k is represented as $v\left(tq_k\right) = pro\left(tq_k\right) - cos\left(tq_k\right)$. The objective of the virtual resources regulation mechanism within tq_k can be represented as $\max\left(v\left(tq_k\right)\right)$. System needs to meet as many service requests that conform to service level agreement as possiple， and at the same time to decrease the number of physical server in operating state as much as possible.

Simulation experiment

We can easily design resource allocation method based on benefits maximum principle through building the benefit model for resources allocation of virtual machines. In order to verify the usefulness of the benefit model proposed, we carry on an experiment according to the resources management method put forward by literature[7]. Experimental environment referring to literature[7] is as follows: operating system kernel-Linux2.6.14, 100M Ethernet connection，using client to simulate the user request and 6 PC machine as a test client.

Our experiment obtains the variation of response latency of two different service categories C_1 , C_2， who have different quality of service requirements, as shown in Figure 1.

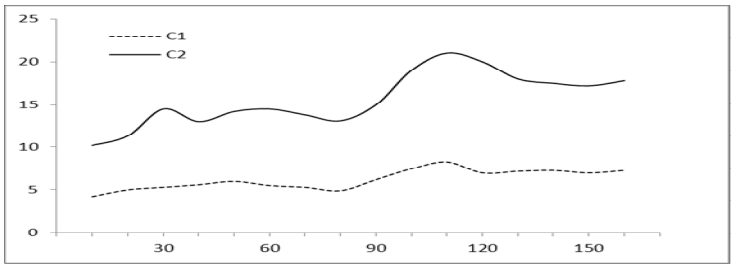

Figure.1 Response latency comparison between two service categories

Through the experiment we can see that the benefit model proposed by us that designs allocation method and allocates resources according to service quality is able to complete the service control self-management better.

Acknowledgements

This research work was supported by the Project of the Natural Science Foundation of Jiangxi Province under grant No. 20151BAB207040. the Subject of Teaching Reform in Universities of Jiangxi Provincial Education Department under grant No. JXJG-14-10-9.

References

[1] Kephart JO, Chess DM. The Vision of Autonomic Computing. Computer, 2003(36):41-50.

[2] Eymann T. The infrastructures of Autonomic Computing. Knowledge Engineering Review, 2009(21):189-194.

[3] Hariri S, Khargharia B, Chen H, et al. The autonomic computing paradigm. Cluster Computing, 2006(9): 5-17.

[4] Figueiredo R, Dinda P, Fortes J. A Case for Grid Computing on Virtual Machines. In: Proc. of 23th IEEE International Conference on Distributed Computing Systems, Rhode Island USA, May 2013, 550-559.

[5] Ruth P, Rhee J, Xu D, et al. Autonomic Live Adaptation of Virtual Computational Environments in a Multi-Domain Infrastructure. In: Proc. of 3rd IEEE International Conference on Autonomic Computing, Dublin Ireland, June 2006, 5-14.

[6] Menasce DA, Bennani MN. Autonomic Virtualized Environments. In: Proc. of 2nd IEEE International Conference on Autonomic and Autonomous Systems, Silicon Valley USA, July 2006, 271-280.

[7] Liu T, Zeng G.S. Self-management Model for Control Differentiated QoS in Web Clusters.Computer Engineering(in chinese),Vol.33(24):32-34.