Study on the Cost Optimization of Visual Machine Based on Multi-dimensional QoS Cloud Resource Scheduling Algorithm

Miaochan Zhao*, Shuling Gao

**E-mail: miaochanzhao@163.com*
School of Mathematics and Statistics, Zhoukou Normal University,
Zhoukou, Henan, 466001, China

Abstract

In this paper, the author mainly discusses the cost optimization of visual machine (VM) based on multi-dimensional QoS cloud resource scheduling algorithm. The author proposes a dynamic approach for live migration to support an Infrastructure-as-a-Service (IaaS) and avoid physical machine overload. Although the scalability limitation may exist, the overall scalability of the system is significantly improved to make comparing to existing completely centralized and manual solution for the project. Management of resource utilization and power can be also achieved by exploiting the shared memory among VMs running on the same machine. Precisely speaking, some limitations that are innate to the cloud resource platform requires a continued contributions in related areas like scheduling, server consolidation, other monitoring and measuring tools for analyzing system performance with respect to SLA violation and QoS.

Keywords: COST OPTIMIZATION, VISUAL MACHINE, MULTI-DIMENSIONL QOS CLOUD RESOURCE SCHEDULING ALGORITHM

1. Introduction

Cloud computing has revolutionized the Information and Communication Technology (ICT) industry that attracts significant attention from public and private sectors nowadays. The pay-as-you-use model for cloud computing infrastructures, which enable convenient on-demand provisioning of elastic computing resource has attracted a diverse interest from individual users as well as organizations.

The number one technology priority for 2011 was cloud computing, based on the survey that included responses from more than 2000 CIOs worldwide [1-2]. Governments also plan to invest into cloud computing, Chinese government is expected to be the main customer of the city-size cloud computing data center, which is predicted to be finished in 2016 and is built by a local company in collaboration with IBM [3].

The organizations can achieve their computational needs in the cloud, and avoid the burden of high cost investment incurred private computing infrastructure, software, personal need to run enterprise services, shared technology solutions, deployment of customized and continuously costs for maintenance and upgrade, or costs to improve the physical resources. In contrast, they can reduce the amount of rented resources for the deflation organizations. Major cloud service providers have driven development of many clouds such as Amazon, Rackspace, Sales force. Cloud computing providers deliver their services according to several fundamental models such as Infrastructure as a Service (IaaS) which is the focus of this study, Platform as a Service (PaaS), and Software as a Service (SaaS).

Infrastructure as a service is a provision model in which an organization can use the physical resources to support operations, including storage, hardware, servers and networking components. The cloud computing provider owns the equipment and is responsible for housing, running and maintaining it [4]. The client typically pays on a per-use basis. Amazon and Rack Space are probably the most well-known to provide this service, and the most famous among the open source cloud projects are C1oudStack, Eucalyptus, and Open Nebula [5]. The open-source cloud platforms provide the ability to deploy private infrastructure as a service clouds and compatible.

Application Programming Interfaces (API) with public cloud, which improves the flexibility and usability of the private clouds. The cloud computing providers introduce the cloud service on two way 1-guaranteed service class (reserve the physical resource for VM demand) or 2- best-effort class (share the cloud platform and guaranteed the effort) [6]. The cloud gives guaranteed service class high priority. In the first case, the customer is guaranteed the performance and the delivery of services because the cloud providers reserve the physical resource for VM demand, this service is expensive but more secure and has high quality-of-service. In the second case, the cloud provider guarantee the effort in sharing cloud platform, this service is of lower cost than the first one. However, cloud computing in terms of QoS and service level agreement still presents significant challenges concerning performance, availability, energy consumption and economical costs of the cloud. The existing public clouds provide very little guarantee in terms of performance and dependability. The service provider needs to manage the physical resource tightly to guarantee a provision of high QoS without SLA violation.

In infrastructure as a service, overloaded physical machine and aggressive consolidation VMs on the physical machine cause the SLA violation that need a fit methodology to manage the resource in a dynamic way and guarantee QoS for the customers. We believe that a differentiating element between cloud computing environments will be the QoS and the SLA provided by the cloud without violation, whereas will increase the trust between customers and cloud computing providers. This work focused to avoid the overload on physical machines in cloud computing at a suitable time and maintaining the required quality of service without a violation. For the purpose of knowing the scope of this research and its importance, we would start to explain the technique of cloud computing, and then define the main problem of the research, as well as the questions posed about the problem.

2. The Live Migration

Virtualization can provide significant benefits in cloud computing by enabling virtual machine migration to manage the VM guests [7]. Migration is the process of movement of VMs OS with all application running on it as one part between source hosts to destination hosts to achieve the goals of load balancing, energy saving, failure recovery, and system maintenance [8]. The entire migration operating system for guests with all applications as one unit helped us to avoid a lot of difficulties.

The most benefit for the migration in cloud computing is balancing and redistributing the workload by accomplishing migration for the VMs out of the overloaded servers to decrease load servers [8]. Migrating an instance operation system between the physical hosts is useful tools for administrators to facilities the management load balance, fault tolerance, and low-level system maintenance. Migration also is useful to rearrange OS guest load across machines to relieve the load on congested hosts, or avoid an overload on that machines. On the other hand, migration is complicated by the need to consider multiple resources CPU, network, and memory for each application and physical server.

The majority of migration can achieve by two ways live and suspend migration. In the suspend migration, the OS guest will suspend and the user will know the connect is down for some time. While in the live migration, the OS guest will running during migration and the user will not know there is migration operation. The administrator tries to move the OS instance between physical hosts with a short downtime during migration. When we achieve the migration, the a key challenge in managing the migration of OS instances is migrating the resource that are provided from physical machines to the guest VMs such as memory, disks and network interface with near-zero downtime to make real live migration.

There is two important factors to achieve real live migration, downtime and total migration time, which should be minimize as we can. The downtime is the period during no currently executing instance of guests where the service is unavailable. This period will be visible to clients of the VM as service delay. The total migration time is the duration between the migration initiation and when the operating system finished moving OS instance guest between source host and destination host.

The migration process consists of transferring the memory image from the source host to the destination host. The hypervisor pre-copies memory pages of the

VM to the destination without interrupting the OS, or any of its applications. The page copying process is repeated in multiple rounds on which dirty pages are continuously transferred. Normally, there is a set of pages that are modified so often that the VM must be stopping for a moment until this set is fully transferring to the destination. The VM can be resumed in the new server; but the bad effects of migration were acceptable or negligible in contrast to its potential benefits to the system fault tolerance. The live migration process is illustrated in the Figure 1 that explain the live migration on run time for VM instances.

There are seven core components of Visual machine Compute, Object Storage, Identity, Dashboard, Block Storage, Network and Image service.

Compute (codenamed "Nova"): this is the core of the Visual machine which originated as a project at NASA Ames research laboratory and started as open-source software in early 2010.

Object Store (codenamed "Swift"): which provide object storage but cannot mount directories like fileserver. It allow to store and retrieve files Image (codenamed "Glance"): which provide a repository for virtual disk images which most commonly use in Visual machine compute.

Dashboard (codenamed "Horizon"): is web front ends which provides a modular web-based GUI user interface for all the Visual machine services.

Identity (codenamed "Keystone"): provides authentication and authorization for all the Visual machine services. It also provides a service catalog of services within a particular Visual machine cloud.

Network (codenamed "Quantum"): provides network connectivity as a service between interface devices managed by other Visual machine services (most likely Nova)

Block Storage (codenamed "Cinder") provides persistent block storage to guest VMs. Figure 2 depict the high level interaction between Visual machine service.

3. The Cost Optimization of Visual Machine

To carry out dynamic live migration for avoid overload physical machine to prevent SLA violation, we consider the distributed system model same as shown in Figure 3. The target is IaaS environment, the system models consist of some parts distributed on the environment equipment such as controllers and compute node. The end-user submits requests for provisioning WVIs characterized by requirements to some CPU performance like MIPS (Million Instruction per second), RAM, and memory and network

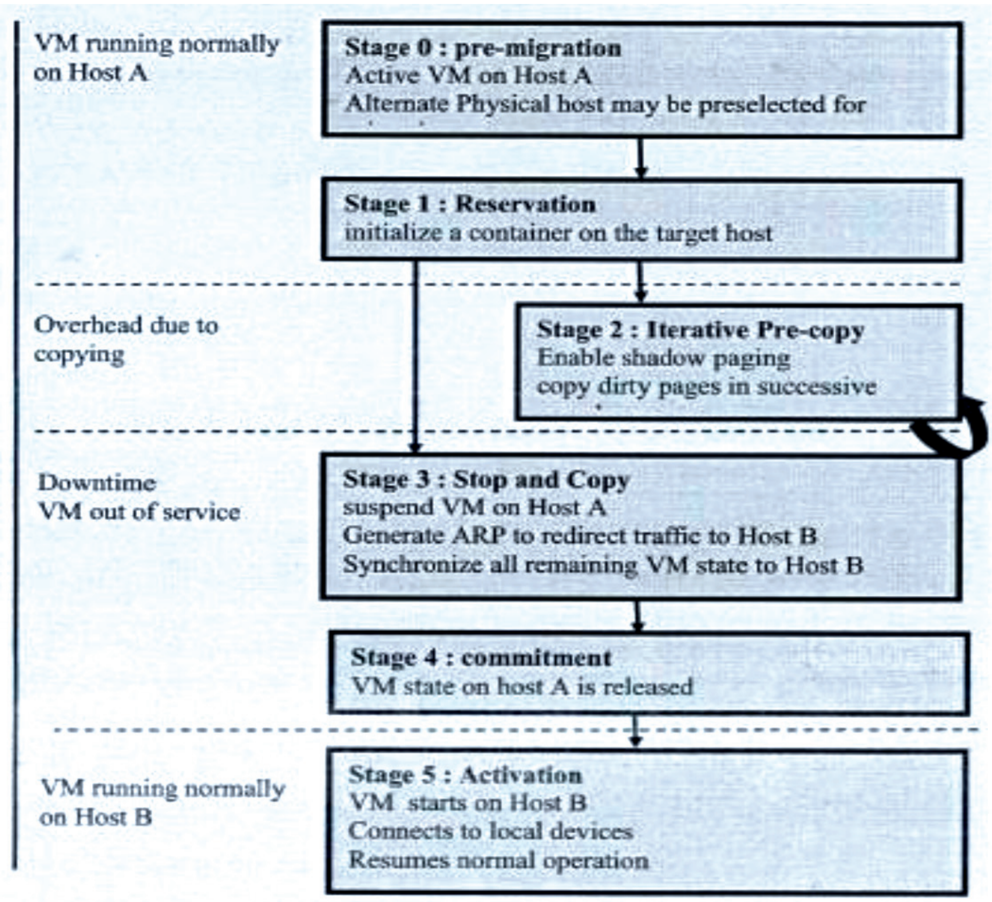

Figure 1. Live migration model

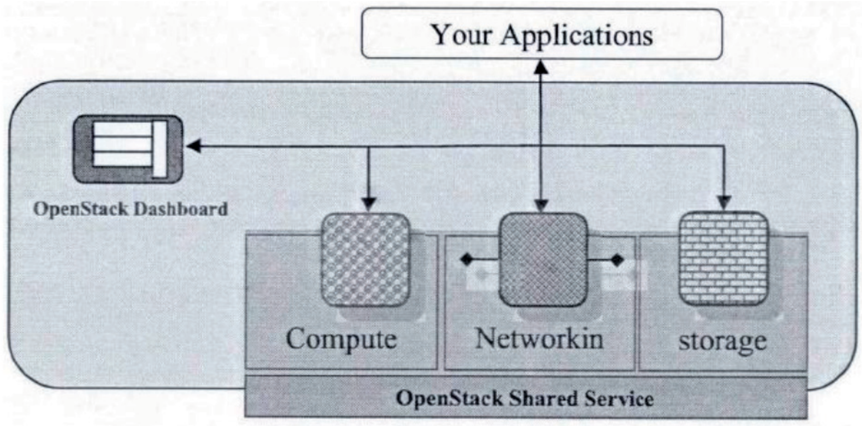

Figure 2. Live migration model

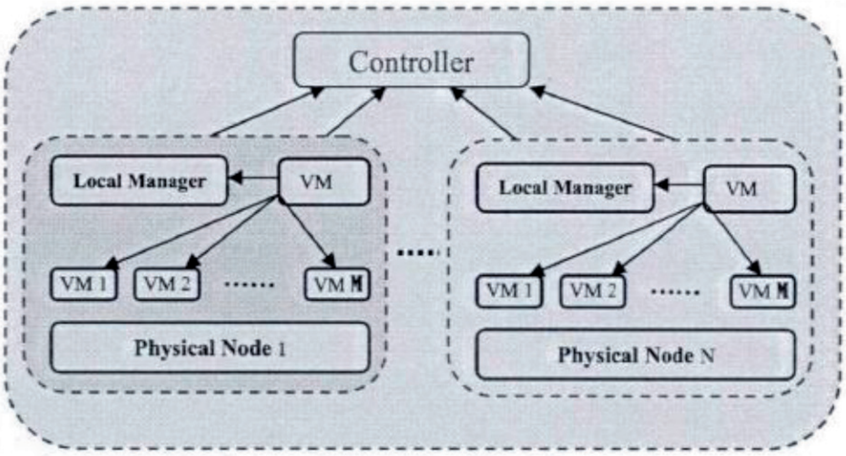

Figure 3. Distribute system model

bandwidth. The compute node servers do not have direct-attached storage, while the storage is provided by shared storage such as a Network Attached Storage (NAS), or Storage Area Network (SAN) or other available technology to enable VM live migration. The multiple VMs are running on a single physical node is shared and managed by independent users using the isolation technique provided by virtualization cloud comp using. The mixed workload is formed by various apes of applications, such as web-application and HPC (High Performance) which utilize the resources at the same time. Consequently, the provider pays to the user in case of SLA isolations. The SL: As establish between customer and service provider to provide the resource for the customer to formalize the QoS.

In this model, the software layer of the system is distributed to local and global mangers. The local managers which are part of VM monitor on each compute node, that are responsible for continuously monitor the physical machine load, and decide when and which a VMs should to be migrated from physical node to avoid the overload machine. The local manager objective is monitoring of the node's CPU utilization, saves the historical monitoring, and reports the global manager about the state of machine, then send migration requests. The global manager resides on the control node, and receives the information with migration requests from local manager to initiate the VMs migration process to enhance the VM placement.

We used distributed model for the cloud Visual machine software to reach our aims. The dynamic migration implementation carry out to avoid overloaded physical machine to deliver good QoS for the customer by apply the SLA without violation. The proposed model for the cloud Visual machine software depended on the previous model, and composed of the three main components:

1. Global manager (controller): a component that is deployed on the controller host, and makes the global management decisions, such as mapping VM instance on host ,and initiating VM migration.

2. Local manager (compute hosts): a component that deployed on every compute host, and makes local decisions, such as deciding that the host is overloaded, and selecting VMs to migrate to other hosts. (Algorithms overload deciding and VMs selecting).

3. VMM (virtual machine monitoring): a component that is deployed on every compute host, and responsible for collecting data about the resource usage by VM instance, and store these data locally, which employ by algorithms.

4. Multi-dimensional QoS Cloud Resource Scheduling Algorithm

Online migration of VMs means transferring a VM between hypervisors on physical nodes without suspension or long downtime. However, live migration increase the performance for the source node and impact on the performance of VM and applications running on it during a migration in physical machine. Therefore, the performance degradation and downtime depend on the application's behavior such as how many memory pages the application updates during its execution. This is investigated in voorsluys et al study, which studies the value of this impact and finds a way to model it. Voorsluys took the average performance degradation for web-applications like example with the downtime and estimated as approximately 1 Q% of CPU utilization. This means migration cause some SL.A violation. So, the minimize number of VM migration is crucial. Therefore, we should try to minimize the number of VM need for migration and try to choose the most fit VM which take the load down under the overload threshold. In the same time the selected VM should be the light one fit the condition because the length of live migration depends on the total amount of memory provided to VM during running time and available network bandwidth between the source and destination.

The algorithm can be expressed as following equation (1-2):

$$C^e = C_v + C_s = \int_{v_e} \mu_1 N^T N dV + \int_{v_e} B^T \mu_2 B dV$$
$$\rho_1 = \rho - \rho_0 \quad (1)$$

$$EpIp = \frac{d^4 u_p}{dz^4} = K(u_p - u_s) \quad (2)$$

The we get:

$$m(x,y) = \sqrt{\begin{matrix}(L(x+1),y) - L(x-1,y)^2 \\ +(L(x,y+1) - L(x,y-1))^2\end{matrix}} \quad (3)$$

$$W_\psi^k(j,i_1,i_2) = \frac{1}{\sqrt{I_1 I_2}} \sum_{y=0}^{I_2-1} \sum_{x=0}^{I_1-1} f(x,y) \psi_{j,i_1,i_2}^k(x,y),$$

$$Y = \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \sum_{j_3=1}^{J_3} g_{j_1 j_2 j_3} a_{j_1}^{(1)} \circ a_{j_2}^{(2)} \circ a_{j_3}^{(3)} + E$$
$$= G \times \{A\} + E = \hat{Y} + E \quad (4)$$

In which,

$$\alpha^2 = \frac{\rho_0 \omega^2}{C_{11}^0},$$
$$\alpha^2 = \frac{\rho_0 \omega^2}{C_{66}^0}, \beta_\perp^2 = \frac{\rho_0 \omega^2}{C'_{44}},$$
$$C'_{44} = C_{44}^0 + \frac{(e_{15}^0)^2}{\eta_{11}^0} \quad (5)$$

Rewrite again Eq. (4) as

$$\hat{f}_H^\alpha(x) = \frac{1}{\Gamma(1+\alpha)} \int_{-\infty}^{\infty} \frac{f(t)}{(t-x)^\alpha} (dt)^\alpha$$
$$= \frac{1}{\Gamma(1+\alpha)} \int_{-\infty}^{\infty} f(t) g(x-t)(dt)^\alpha \quad (6)$$
$$= f(x) * g(x),$$

$$\partial_j (C_{ijkl} \partial_k u_l + e_{kij} \partial_k \varphi) - \rho \ddot{u}_i = 0 \quad (7)$$

$$\partial_j (e_{ijkl} \partial_k u_l - \eta_{kij} \partial_k \varphi) = 0 \quad (8)$$

The linear equation can be expressed into the following simplified forms:

$$L(\nabla, \omega) f(x, \omega) = 0$$
$$L(\nabla, \omega) = T(\nabla) + \omega^2 \rho J \quad (9)$$

In which,

$$D_F(Y \| G, \{A\}) = \frac{1}{2} \| Y - \hat{Y} \|^2 \quad (10)$$

$$Z_t | F_{t-1} \sim C(u_{1t}, u_{2t}, ..., u_{nt}; \alpha_1, \alpha_2, ..., \alpha_N) \quad (11)$$

$$E(Z_t Z_t^{'} | F_{t-1}) = \sum_t = (\sigma_{ij,t}), \sigma_{ij,t} = 1 \quad (12)$$

The above equation will help us to calculate the cost of migration which would help us later to calculate the SLA violation. Minimizing the number of migrated VM would keep the performance degradation low as we can. The idea here to look for the favorable consuming CPU, that suitable to offload the load down the threshold, and spend short time to move between source and destination PM. Live migration should achieve by dynamic way to make the decision on suitable time depending on monitoring data.

Conclusions

In this paper, the author mainly discusses the cost optimization of visual machine (VM) based on multi-dimensional QoS cloud resource scheduling algorithm. The author proposes a dynamic approach for live migration to support an Infrastructure-as-a-Service (IaaS) and avoid physical machine overload. We believe that a differentiating element between cloud computing environments will be the QoS and the SLA provided by the cloud without violation, whereas will increase the trust between customers and cloud computing providers.

The local manager objective is monitoring of the node's CPU utilization, saves the historical monitoring, and reports the global manager about the state of machine, then send migration requests. Although the scalability limitation may exist, the overall scalability of the system is significantly improved compared to existing completely centralized and manual solution for the project. This work focused to avoid the overload on physical machines in cloud computing at a suitable time and maintaining the required quality of service without a violation. For the purpose of knowing the scope of this research and its importance, we would start to explain the technique of cloud computing, and then define the main problem of the research, as well as the questions posed about the problem.

References

1. M. Tamer Ayvaz, Alper Elçi. A groundwater management tool for solving the pumping cost minimization problem for the Tahtali watershed (Izmir-Turkey) using hybrid HS-Solver optimization algorithm. Journal of Hydrology, 2013, PP. 478-490.
2. Sha Liu, Ran Tao, Chi Ming Tam. Optimizing cost and CO 2 emission for construction projects using particle swarm optimization. Habitat International, 2013, PP. 37-51.
3. Seong-Rin Lim, Yoo Ri Kim, Seung H. Woo, Donghee Park, Jong Moon Park. System optimization for eco-design by using monetization of environmental impacts: a strategy to convert bi-objective to single-objective problems. Journal of Cleaner Production, 2013, pp. 39-49.
4. Souma Chowdhury, Jie Zhang, Achille Messac, Luciano Castillo. Optimizing the arrangement and the selection of turbines for wind farms subject to varying wind conditions. Renewable Energy, 2013, pp. 52-62.
5. Motaz Amer, A. Namaane, N.K. M'Sirdi. Optimization of Hybrid Renewable Energy Systems (HRES) Using PSO for Cost Reduction. Energy Procedia, 2013, pp. 42-51.
6. T. Ashuri, M.B. Zaaijer, J.R.R.A. Martins, G.J.W. van Bussel, G.A.M. van Kuik. Multidisciplinary design optimization of offshore wind turbines for minimum levelized cost of energy. Renewable Energy, 2014, pp. 68-79.
7. Jason Ng Cheng Hin, Radu Zmeureanu. Optimization of a residential solar combisystem for minimum life cycle cost, energy use and exergy destroyed. Solar Energy, 2014, pp. 100-117.
8. João P. Ribau, Carla M. Silva, João M.C. Sousa. Efficiency, cost and life cycle CO 2 optimization of fuel cell hybrid and plug-in hybrid urban buses. Applied Energy, 2014, pp. 129-138.

Miniaturization Design of Two-Component Grouting Pump with Small Grouting Fluctuation and Automatic Cleaning Function

SUN Zhi-jia[1], GUO Ming-qi[2], GUO Jin-zhan[3]

[1] College of Mechano-Electronic, Lanzhou University of Technology, Lanzhou 730050, China;
[2] Harbin Institute of Technolgy, Weihai, 264209, China;
[3]Beite Mining Equipment Co. Ltd., Jiaozuo 45400, China.

Corresponding author is SUN Zhi-jia