SLA-Aware Adaptive Provisioning Method for Hybrid Workload Application on Cloud Computing Platform

Seunghwan Yoo, and Sungchun Kim

***Abstract*— Resource provisioning is a general technique for handling the resource allocation in cloud environment. Monitoring the system performance and the user request is crucial for efficient cloud resource management. Also in cloud environments, issues such as cost and resource provisioning based on QoS constraints are yet to be addressed. In this paper, we present a SLA(Service Level Agreement) - Aware Adaptive (SAA) provisioning method for hybrid workload that employ a flexible determining model. We present advanced cloud infrastructure which maintain proper virtual machine numbers by optimizing resources allocation. Our experiments show that our adaptive model minimize the total number of virtual machines while satisfying user average response time constraint and the request arrival rate constraint. And the extra cost can be effectively reduced.**

***Index Terms*—Provisioning, Service Level Agreement, Adaptive, Cloud Computing Platform**

I. Introduction

Virtualization technologies have enabled various cloud computing services[1]. Cloud computing [2,3] are provided as three kinds of services type. Such as Infrastructure as a Service(IaaS), Platform as a Service(PaaS), and Software as a Service(SaaS). We consider mainly IaaS, which provide computing resources or storage as a service to users. The main purpose of management cloud computing infrastructure is to ensure the quality and cost-effectiveness. The online users can get services by sending their requests to service provider. Also, the cloud services should satisfy Service Level Agreements (SLAs). SLA is agreement about quality of cloud services signed between cloud service providers and consumers. SLA specifies concretely expected performance metric and charging model, which include response time, throughput, availability, reward and penalty, and so forth. Current cloud computing paradigms are not easily to meet users' purpose, especially facing the requirement of diverse applications from different users.

In order to meet the constraint of SLA and provisioning existing virtualized resources optimally. At the same time, cloud computing service providers make the profits by providing high-quality services through efficiently allocating the resources on demand. We present a SLA-based framework handling resource allocation on a cloud computing platform. This would support to accomplish two goals simultaneously: minimized user response time and minimized resource usage cost.

The activities to accomplish both goals may conflict with each other. For example, user response time can be reduced by assigning more resources while the cost may be lowered by allocating less resources. Since the workload of an application service usually varies with time, this is a great challenge for resource allocation optimally. So refined provisioning method would need to achieve the goal.

For this reason, this paper proposes a SLA-Aware Adaptive resource provisioning method(SAA provisioning Mechanisms). SAA provisioning method provides scalable processing power with dynamic resource provisioning mechanisms, where the number of virtual machine used is dynamically adapted to the time-varying incoming request workload. To evaluate our framework and method, we applied GridSim [18] to simulate the cloud environment. In the simulation, the workload estimation for hybrid workload is investigated comprehensively. To evaluate the performance of SAA provisioning method, we compare it with QuID [14], [16], a dynamic resource provisioning approach proposed recently.

The remainder of the paper is organized as follows. Section 2 presents survey related to our work. Section 3 describes our SAA provisioning method on the cloud computing platform. Section 4 presents our adaptive resource provisioning algorithm and its performance evaluation. Section 5 concludes the paper and points out some future research directions.

Manuscript received January 8, 2013; revised January 30, 2013. This research was all-supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2012R1A1A2009558), and part-supported by Technology Innovation Development Program funded by Small & Medium Business Administration(S2057016)

Seunghwan Yoo is doing Ph.d Course in the Sogang University, Seoul, Korea (e-mail : ossforever.hpcs@ gmail.com).

Sungchun Kim is a Full Professor in the Sogang University, Seoul, Korea, (e-mail:ksc@sogang.ac.kr)

ISBN: 978-988-19251-8-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

II. Related Work

Nowadays, some researches have focused on the issue of resource management and performance control in cloud computing platform[5, 6]. However, new challenges are introduced while service providers benefit from the planning flexibility in technical and economic aspects. Some challenges and opportunities of automated control in cloud computing is discussed in [7]. And other researchers work to improve the resource utilization, such as resource virtualization [8,16], on-demand resource provisioning management based on virtual machines [9, 10], and QoS management of virtual machine [11].

Also, many researchers [12, 13] focus on improving resource utilization as well as guaranteeing quality of the hosted services via on-demand local resource scheduling models or algorithms within a physical server. However, most of them could not be good solutions to tradeoff between resource utilization and SLA. For example, [12] present a novel system-level application resource demand phase analysis and prediction prototype to support on-demand resource provisioning. The process takes into consideration application's resource consumption patterns, pricing schedules defined by the resource provider, and penalties associated with SLA violations. The authors in [13] improve resource utilization and performance of some services by hugely reducing performance of others. How to improve resource utilization, as well as guarantee SLA, is a challenge in a VM-based cloud data center

In the context of the dynamic resource provisioning, the author in [16] introduce three mechanisms for web clusters. The first mechanism, QuID [14], optimizes the performance within a cluster by dynamically allocating servers on-demand. The second, WARD [15], is a request redirection mechanism across the clusters. The third one is a cluster decision algorithm that selects QuID or WARD under different workload conditions.

For multi-tier internet applications, the modeling is proposed that a provisioning technique which employs two methods that operate at two different time scales : predictive provisioning at the time-scale of hours or days, and reactive provisioning at time scales of minutes to respond to a peak load[17]. This modeling is a multi-tier application as a network of queues where each queue at a tier represents a server, and the queues from a tier feed into the next tier. Given the request arrival rate and per-tier response time, the number of servers needed at each tier is computed individually by the proposed algorithm. While the above techniques are aimed for multi-tier web applications, our work in this paper targets at interactive workflow applications.

III. Proposed Scheme

A. SAA Framework

This section presents a scalable framework for interactive workload applications on the cloud computing platform. The framework deals with the scenario that hosted on a cloud computing platform, handle many virtual machines simultaneously according to the incoming user requests. Since the amount of incoming requests changes with time and the cloud platform is a pay-per use service, the application has to dynamically assign the resources it uses to maintain guaranteed response time and reduce the total owner cost under various workloads. In the framework, server pool, combining a distinct computing server, is capable of processing multiple hybrid workload requests.

To efficiently utilize resources, there are two key issues considered in the framework. The first is finding the least loaded resource for dispatching incoming requests. The second issue deals with SAA provisioning for adaptively handling dynamic workloads. With resource state monitoring, each workflow enactment request will be sent to the least loaded resource for service. The effectiveness of least load dispatching largely depends on how to accurately capture the computing load on each resource.

Fig. 1 shows an overview of the framework in handling user requests for virtualized application execution environments (VAEEs). The architecture consists of four main components that Monitor, Analyzer, Resource scheduler, and Virtualized Application Executor (VAE) control loops architecture. The goal is to meet the user requirements while adapting cloud architecture to workload variations. Usually, each request requires the execution of virtualized application allocated on the VM of each physical server. A cloud computing resource amount enables multiple virtualized applications may be increased when workload increases and reduced when workload reduces. This dynamic resource provisioning allows flexible response time in a VAEE where peak workload is much greater than the normal steady state.

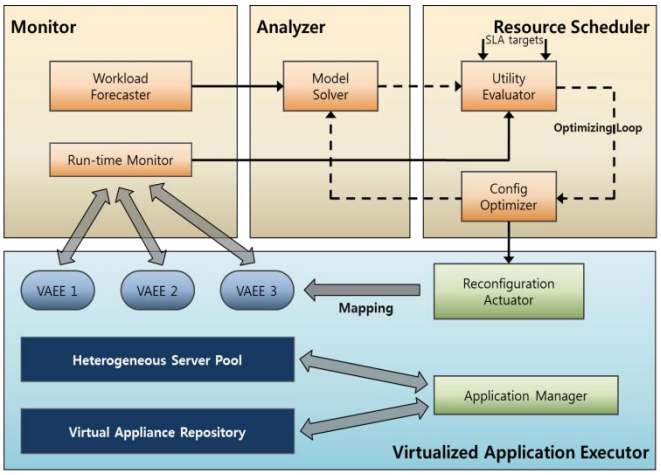

Fig. 1 Proposed SAA-Provisioning Framework

ISBN: 978-988-19251-8-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Fig 1 provides a high-level dynamic resource provision architecture for cloud computing platform, which shows relationships between heterogeneous server resources pool and self-management function. Server pool contains physical resources and virtualized resources. A lot of VMs hold several VAEEs sharing the capacity of physical resources and can isolate multiple applications from the underlying hardware. VMs of a virtualized application may correspond to a physical machine.

Self-management function means mechanisms to automate the VMs of configuring and tuning the virtualized application so as to maintain the guaranteed response time for requirements of the diverse users. As previously stated, four main components more detail explanation are as follows:

① Monitor: Collects the workload and the performance metric of all running VAEEs, such as the response time, the request arrival rate, the average service time, and the CPU utilization, etc.

② Analyzer:

Receives and analyzes the logged parameter from the monitor to estimate next state workload. It also receives the response times of different users

③ Resource Scheduler:

Sets up updated configuration metric for each VAEE, and uses its optimizer with the optimization model to determine resource provisioning according to these workload estimates and response time constrains of different users such that the resource requirements of the overall VAEE is minimized.

④ Virtualized Application Executor:

Assigns the virtual machine configuration, and then runs the VAEEs to satisfy the resource requirements of the different customers according to the optimized decision.

In conclusion, Fig. 1 is represented the dynamic resource provisioning method. Our research is a great help of on the improved design of resource scheduler for hybrid workload. The goal is to minimize the using of resources under a workload while satisfying different users for the guaranteed response time.

B. Proposed SAA provisioning Algorithms

In this section, we propose an auto-control algorithm denoted as SAA provisioning method (SLA Aware Adaptive) to dynamically provide an adequate amount of resources to virtualized application. To maintain acceptable response time and cost efficiency, it would find the configuration value which the Sum of each VAEEs profits is maximized. Considering all of virtual machine system parameters observed by monitor, especially response time and usage cost, we compute the profit value of each VAEEs. Through equation (1), our method calculates the optimized next step setting value. Resource scheduler receives the modified configuration parameter. Then it reflects the value next schedule period.

$$\begin{aligned}\text{Profit(E)} = \alpha \times (\text{Arrival Rate} \times \text{SLA satisfied Rate} - \text{VM failed Rate}) \\ -\beta \times (\text{Active VM maintain Cost} + \text{Idle VM maintain Cost})\end{aligned} \quad (1)$$

After each VAEE Profit is calculated, it is added up to find Global Maximum Profit.

$$\max\{\ Profit_{global} = g\ (p_1, p_2, \ldots\ldots p_n\ )\ \} \quad (2)$$

Our SAA provisioning algorithms would find SLA-guaranteed response time and low maintenance cost.

IV. PERFORMANCE ANALYSIS

Our simulation environment for the following experiments is based on GridSim [18]. GridSim is a discrete event simulator built on top of the simulation package SimJava [19] and can be used to model and simulate various entities in parallel and distributed computing environments. GridSim controls all the entities, delivers the events, and advances the simulation time. We also include random selection and the Round-Robin load balancing algorithm in the experiments for performance comparison.

Table.1 Simulation parameters

Number of tasks	10 ~ 20
Mean task execution time	5 sec
Mean user thinking time	7 sec
Maximum degree of a task	3
Input file size	100 byte
Output file size	100 byte
Request arrival interval (Poisson distribution)	2.2
Arrival rate measurement interval	40
Response time measurement interval	30
Computing speed	100MIPS*20

ISBN: 978-988-19251-8-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Experimental results are summarized in Table 2. For SAA provisioning method, the workload limit on each resource is set to 11 tasks in the very beginning. The value 11 was determined by simulation studies on the performance of various values for the workload limit. The results indicate that SAA method can outperform the others. For comparison with QuID, in experiment 1, where at most 16 resources are available, SAA method has shorter response time than QuID by 31%, but 0.23 more in resource usage.

In experiment 2, SAA method outperforms QuID in both response time and resource usage. Comparing with the static provisioning approach, let the target response time be 8.2 seconds, SAA method requires roughly 15.93 resources in average while static provisioning requires 16 resources. From another aspect, let the resources usage be 12 for economic reasons, SAA method provides an average response time of 9.01 seconds and static provisioning provides 26.67 seconds. SAA method is almost three times quicker.

Table.2 Simulation result

	Static provisioning		Dynamic resource provisioning			
			Exp 1 : 4~16		Exp 2 : 4~50	
	Fixed 16	Fixed 12	SAA	QuID	SAA	Quid
Avg response time	8.2	26.67	11	15.76	9.01	17.32
Avg resource usage	16	12	13.9	13.66	14.09	15.93

In addition, utilization rate is used in QuID for measuring the workload on each resource. One potential drawback of utilization rate is that when utilization rate reaches 100%, it cannot effectively calculate the amount of resources to increase. Moreover, utilization rate is a time-interval based measurement, such as arrival rate and average response time. Therefore, it is a crucial issue to determine an appropriate measurement interval. However, this is difficult since response time and resource usage are not monotonically increasing or decreasing with the measurement intervals.

V. Conclusion

In this paper, it is argued that dynamic provisioning of virtualized applications environment raises new challenges not addressed by prior work on provisioning technique for cloud computing platform. We presented an optimal autonomic virtual machine provisioning architecture. We proposed a novel dynamic provisioning technique, which was a algorithms for hybrid workload in cloud computing platform. Hence the efficiency and flexibility for resource provisioning were improved in cloud environment.

Currently many server applications adjust the amount of resources at runtime manually. The framework in this paper allows applications to automatically manage the amount of resources according to the system workload. It offers application providers the benefits of maintaining QoS-satisfied response time under time-varying workload at the minimum cost of resource usage. Also, we adopt Service Level Agreement (SLA) based negotiation of prioritized applications to determine the costs and penalties by the achieved performance level. If the entire request cannot be satisfied, some virtualized applications will be affected by their increased execution time, increased waiting time, or increased rejection rate.

The framework mainly deals with the issue: resource provisioning. For dynamic resource provisioning, SAA propsed method is as a feedback controller to automate resource provision by taking information of the characteristics of hybrid workload. Experimental results show that our method outperforms static provisioning and the utilization rate based QuID approach in both average response time and resource usage.

References

[1] D. Reed, I. Pratt, and P. Menage, et al, “Xenoservers: Accountable execution of untrusted programs”, The Seventh Workshop on Hot Topics in Operating Systems, Rio Rico, Arizona, 1999.

[2] M. Armbrust, A. Fox, and R. Griffith, et al, “Above the clouds: A Berkeley view of cloud computing”, Technical Report No. UCB/EECS-2009-28, University of California Berkley, USA, Feb. 10, 2009.

[3] R. Buyya, C.S. Yeo, and S. Venugopal, et al, “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility”, Future generation computer systems, Elsevier science, Amsterdam, the Netherlands, 2009, 25(6), pp. 599-616.

[4] D. Gupta, S. Lee, and M. Vrable, et al, “Difference engine: harnessing memory redundancy in virtual machines”, The 8th USENIX Symposium on Operating Systems Design and Implementation, 2008, pp. 309-322

[5] E. Kalyvianaki, T. Charalambous, and S. Hand, “Self-adaptive and self-configured CPU resource provisioning for virtualized servers using kalman filters”, Proceedings of the 6th international conference on Autonomic computing, Barcelona, Spain, June 15-19, 2009.

[6] W. E. Walsh, G. Tesauro, and J. O. Kephart, “Utility functions in autonomic systems”, Proceedings of the First IEEE International Conference on Autonomic Computing, New York, NY, USA, May 17-18, 2004

[7] E. H. Miller Lim, H., Babu, S., Chase, J., Parekh, S.: Automated Control in Cloud Computing: Challenges and Opportunities. In: 1st Workshop on Automated Control for Datacenters and Clouds, 2009.

[8] P. Barham, B. Dragovic, and K. Fraser, et al, “Xen and the art of virtualization”, Proceedings of the 19th ACM Symposium on Operating Systems Principles, Bolton Landing, NY, USA, 2003, pp. 164-177.

[9] Y. Song, Y. Li, and H. Wang, et al, “A service-oriented priority based resource scheduling scheme for virtualized utility computing”, Proceedings of the 9th IEEE International Symposium on Cluster

Computing and the Grid, 2009, pp. 148-155.

[10] J. Zhang, M. Yousif, and R. Carpenter, et al, "Application resource demand phase analysis and prediction in support of dynamic resource provisioning", Proceedings of the 4th International Conference on Autonomic Computing, 2007.

[11] X.Y. Wang, Z.H. Du, and Y.N. Chen, et al, "Virtualization based autonomic resource management for multi-tier Web applications in shared data center", The Journal of Systems and Software, 2008, 81(9), pp. 1591-1608

[12] J. Zhang, M. Yousif, and R. Carpenter, et al, "Application resource demand phase analysis and prediction in support of dynamic resource provisioning", Proceedings of the 4th International Conference on Autonomic Computing, 2007, pp. 12-12.

[13] P. Padala, X. Y. Zhu, M. Uysal, et al, "Adaptive control of virtualized resources in utility computing environments", EuroSys, 2007, pp. 289-302.

[14] Ranjan, S., Rolia, J., Fu, H., Knightly, R.: QoS-Driven Server Migration for Internet Data Centers. In: The Tenth International Workshop on Quality of Service, Miami, FL, 2002

[15] Ranjan, S., Karrer, R., Knightly, E.: Wide Area Redirection of Dynamic Content in Internet Data Centers. In: The IEEE INFOCOM, HongKong, 2004

[16] Ranjan, S., Knightly, E.: High-Performance Resource Allocation and Request Redirection Algorithms for Web Clusters. IEEE Transactions on Parallel And Distributed Systems 19(9), 2008.

[17] Urgaonkar, B., Shenoy, P., Chandra, A., Goyal, P., Agile, T.W.: Dynamic Provisioning of Multi-Tier Internet Applications. ACM Transactions on Autonomous and Adaptive Systems 3(1), 2008.

[18] Sulistio, A., Cibej, U., Venugopal, S., Robic, B., Buyya, R.: A Toolkit for Modelling and Simulating Data Grids: An Extension to GridSim. Concurrency and Computation: Practice and Experience (CCPE) 20(13), 1591–1609, 2008.

[19] Howell, F., McNab, R.: Simjava: a Discrete Event Simulation Package for Java with Applications in Computer Systems Modelling. In: First International Conference on Webbased Modelling and Simulation, Society for Computer Simulation, San Diego CA, 1998.

ISBN: 978-988-19251-8-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)