Available online at www.sciencedirect.com

ScienceDirect

AASRI Procedia 5 (2013) 235 – 242

AASRI Procedia

www.elsevier.com/locate/procedia

2013 AASRI Conference on Parallel and Distributed Computing Systems

A User Preference and Service Time Mix-aware Resource Provisioning Strategy for Multi-tier Cloud Services

Dandan Hu, Ningjiang Chen*, Shilong Dong, Yimin Wan

College of Computer, Electronic, and Information, Guangxi University, Nanning, Guangxi 530004, China

Abstract

As the majority of cloud services deploy a multi-tier architecture, it poses a real challenge to resource provisioning. Because of the cross-tier dependencies of multi-tier cloud services, the traditional tailor-made resource provisioning methods for single-tier applications can't be adopted or extended directly. Resource provisioning in the cloud is usually driven by performance predictions, so it's important to characterize workload fluctuations accurately in order to understand how to allocate resources. And user preference also contributes to workload variants. Therefore, we propose a dynamic resource provisioning strategy for multi-tier cloud services that employs (i) a user preference and service time mix-aware workload prediction method to be used as a foundation of resource provisioning, and (ii) a dynamic resource provisioning strategy based on the queuing theory to determine how many resources to be provisioned to each tier. Experimental results demonstrate that our workload prediction method is accurate, and our provisioning strategy is able to improve the accuracy of resource provisioning, reduce the allocated resources and SLAs violations.

© 2013 The Authors. Published by Elsevier B.V. Open access under CC BY-NC-ND license.
Selection and/or peer review under responsibility of American Applied Science Research Institute

Keywords: resource provisioning, workload prediction, queuing theory, multi-tier cloud services

* * Corresponding author. Tel.: +86 -771-3236389;
E-mail address:chnj@gxu.edu.cn.

2212-6716 © 2013 The Authors. Published by Elsevier B.V. Open access under CC BY-NC-ND license.
Selection and/or peer review under responsibility of American Applied Science Research Institute
doi:10.1016/j.aasri.2013.10.084

1. Introduction

Cloud computing provides a capability that business applications are exposed as sophisticated services that can be accessed over network [1]. And many IT giants provide computing and storage resources in a pay-as-you-go manner. Thus, more and more services are deployed in the cloud in order to save infrastructure cost or make a fortune. Most of cloud services deploy a multi-tier architecture, [2] demonstrates using variants of single-tier provisioning methods for multi-tier cloud services has limitations. Resource provisioning in the cloud is usually driven by performance predictions [3-9]. In order to improve prediction precision, it's important to characterize workload fluctuations precisely. The majority of previous studies only use request volume as a workload metric. [10] proves that request service time mix is also a very important metric to predict workload variants. What's more, tracking of user preference is important to predict workload variants, too. For instance, in an e-commerce website, some users tend to browser, which will put pressure on the front-end web servers, while, some users tend to order, which will put pressure on the back-end database servers. We propose a dynamic resource provisioning strategy for multi-tier cloud services which aims to improve the precision of workload precision, reduce the allocated resources and the SLAs violations. Firstly, we introduce a workload prediction method bases on user preference and service time mix. By classifying and clustering requests into fine-grained sets according to user-preference and service time respectively, it can be much easier and more precise to predict workload variants. And then, we propose a resource provision strategy based on the queuing network to determine how many resources to be provisioned to each tier. Experimental results demonstrate that the workload prediction method is accurate, and the resource provisioning method is effective which can reduce allocated resources and SLAs violations.

2. The user preference and service time mix-aware workload prediction method

Workload prediction act as a basis for resource provisioning. In order to predict workload fluctuations accurately, we need to characterize workload in a fine-grained manner, especially in the multi-tier cloud services that different tiers have different workload features. In this paper, we divide requests into different sets according to user preference and service time, and then predict each set's workload variants.

The prediction method comprises of three steps: classification, clustering and workload prediction. Classification focuses on user preference, and clustering focuses on service time. The detailed steps are as follows.

- User preference-based classification

A user running an application tends to do specific operations more frequently than others. We define this tendency as preference. Therefore, in order to reflect user preference, we collect user behaviour information and then analyse it to grasp user preference.

We detect user operations and then map them to three specific resource-related classes: CPU-preference class, I/O-preference class and memory-preference class.

Once the user preference based classification is done, we use service time as a metric to cluster requests of each class into several sets.

- Service time-based clustering

We use k-means clustering algorithm to cluster each class into different sets. The clustering metric is service time. Though other clustering algorithms could also be used, k-means has the strength that it's easy to understand and widely studied. And it can converge to an optimum quickly. The traditional k-means clustering algorithm requires k, the number of sets, as an input. Whereas k is not priori known in this paper, the algorithm needs to be modified a little. The detailed steps of the clustering are as follows.

Step 1- Determine k and clustering: As the traditional k-means algorithm requires k as an input, we need to

compute k first. We use iterations to determine k. We choose an initial k, and adjust it continuously until we get the best result. Unfortunately, the appropriate k may be far from the initial k.

After clustering, each request in the same set belongs to the same service time interval defined by a lower and a higher service time bound. As we use service time of requests as inputs, each request belongs to a service time bound set.

Step 2- Calculate set means: The mean of a set is the average service demand of requests falling in the same set. Let $\mathbf{T}=\{t_1, \ldots, t_m\}$ denote the service time set, and $\mathbf{F}=\{f_1, \ldots, f_m\}$ denote the accordingly frequency set. The mean of a set can be calculated by the following formula.

$$mean = \frac{\sum_{i=1}^{m} t_i f_i}{\sum_{j=1}^{m} f_j} \quad (1)$$

Step 3- Recalculate set means: As user preference, service time and frequencies change over time, the set means change accordingly and need to be recalculated.

When the request division including classifying and clustering is done, requests can be divided into fine-grained sets, which makes it easier for us to capture workload variants and predict workload fluctuations.

- Workload prediction

After division, requests are divided into different sets. Each set has a predictor used to predict request arrival rate λ_i. We use time series-based method to predict. The total arrival rate of requests belong to the same class can be calculated by the following formula.

$$\lambda_r = \sum_{i=1}^{k} \lambda_i \quad (2)$$

The prediction result is used as a basis for resource provisioning. The multi-tier service model is responsible for calculating the required resources for each tier, and the resource allocator is responsible for making provision for each tier. Figure 1 illustrates the provision architecture of our system.

3. Dynamic resource provisioning for multi-tier cloud services

Dynamic resource provisioning methods need to provision appropriate resources reacting to workload variants in order to maintain performance, so as to avoid SLAs violations. As each tier of multi-tier cloud services has different resource demands, our strategy takes each tier as an independent entity to characterize. We use the workload prediction method proposed in Section 2 to predict each tier's workload fluctuations, and then compute each tier's required resource.

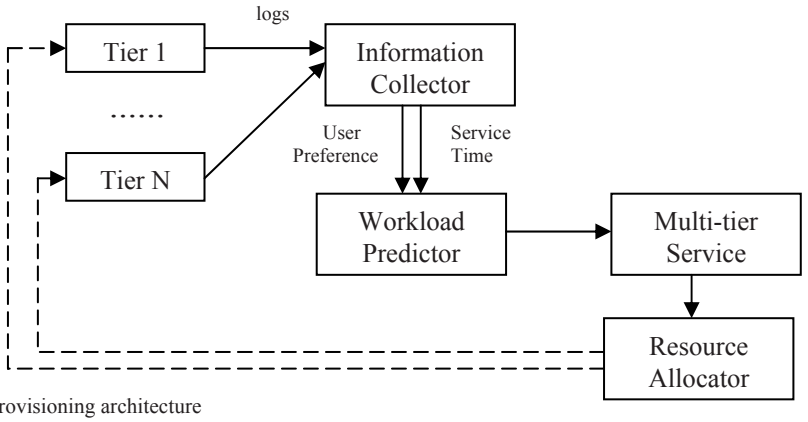

Fig. 1. The resource provisioning architecture

We model a multi-tier cloud service as a queuing network, and specify SLA as 95th percentile response time guarantee. Compared to average response time, 95th percentile response time has the benefits of easy to reason about and to capture the user's perception of service performance [11]. The detailed resource provisioning steps are as follows.

Step 1- Building the queuing network model: Assuming a multi-tier cloud service comprises *M* tiers, we model the application as a queue network comprises of *M* G/G/1 queues. One G/G/1 queue represents one tier. We use the G/G/1 queue because it can capture arbitrary request arrival distribution and service time distribution. As a result, it can capture different tiers' behaviors. In the G/G/1 queue, the service time is assumed to meet a known, fixed distribution. Requests are serviced in a FCFS order.

If a SLA requires 95th percentile response time should be less than *d* seconds, the maximum request arrival rate needs to meet the following constrain.

$$\lambda < \left[s + \frac{\sigma_a^2 + \sigma_b^2}{2(d/3 - s)} \right]^{-1} \tag{3}$$

$$s = \frac{\sum_{j=1}^{3} \sum_{i=1}^{k} \lambda_i m_i}{\sum_{j=1}^{3} \sum_{i=1}^{k} \lambda_i} \tag{4}$$

In Equation 4, s is average service time, λ_i is the arrival rate in set i which can be obtained in the workload prediction method, m_i is the set mean of set i. In Equation 3, σ_a^2 is the variance of inter-arrival time, and σ_b^2 is the variance of service time. As d is known, σ_a^2 and σ_b^2 can be monitored online, by substituting these values into Equation 1, λ is obtained. λ represents the maximum request rate that a benchmark capacity server can handle meeting the 95th percentile response time constraint. Then we can calculate the ratio m of the required resource to the benchmark capacity.

$$m = \left\lceil \frac{\lambda_r}{\lambda} \right\rceil \quad (5)$$

Step 2- Resource provisioning: Our provisioning logic is triggered in four forms: periodic, triggered by request volume variants, triggered by service time mix variants, and triggered by user preference variants. Once the provisioning logic is triggered, the required CPU, I/O and memory capacity of each tier are calculated, and then resource allocator begin to allocate or recollect resource.

4. Experimental Evaluation

To evaluate the effective of our proposed strategy, we use TPC-W benchmark to build our testbed. Because the three inherent workload mixes of TPC-W can't meet our experimental requirement, we use LoadRunner to generate different workload mixes.

We build a small cloud using 4 physical machines. One of them is specified to run the resource provisional strategy, and the other three are used for running different components of TPC-W. The configuration of all the machines is 4 Intel Xeon(TM) CPU 3.60GHz processors, 2GB RAM, 70GB disk, and Linux kernel 2.6.26.

We compare our strategy with the service time mix-aware strategy [10]. Both strategies are provided with the same workload comprises of 1000 sessions which are formed by 3 different interactions of TPC-W, which are "Product Detail", "Search Request", and "Shopping Cart". "Product Detail" and "Search Request" belong to browsing interactions which will pose pressure on the CPU, while "Shopping Cart" belongs to ordering interaction which will consume more I/O resources. The measured service times of the three requests are 2.0ms, 4.4ms and 4.5ms, respectively.

In the first 40 minutes, the percentage of "Product Detail" will decrease while the percentage of "Search Request" will increase every ten minutes. In the next 40 minutes, "Search Request" will be replaced by "Shopping cart". In the last 40 minutes, "Shopping cart" will be replaced by "Search Request" in adverse. Figure 2(a) shows the request mix sent to the TPC-W. Figure 2(b) shows the average service time. Figure 2(c) shows the ratio of the allocated resource to the benchmark capacity, and from which**,** we can see that our resource provision strategy can allocate resource more precisely and reduce the total allocated resource capacity. Figure 2(d) shows the 95^{th} percentile response time of the service, and indicates our strategy can meet 95% of the SLA, while the service time mix-aware strategy violates most of the SLAs at t=70 minutes.

Compare to [10], in the aspect of workload prediction, we consider the impact of user preference to workload variants additionally. By classifying and clustering requests according to user preference and service time respectively, we can divide requests into sets, and then use time series method to predict each set's workload, which makes prediction more accurate. In the aspect of resource provisioning, based on the workload prediction method, we can predict requirements of different resources precisely, and allocate resource with a fine-grained and targeted manner. What's more, the total allocated resource capacity and SLA violations are reduced.

5. Conclusion

We mainly study the dynamic resource provisioning strategy of multi-tier cloud services in the virtual data center. We analyze the workload prediction strategy, and propose a novel prediction method based on user preference and service time mix. By classifying and clustering requests into fine-grained sets, the precision of prediction providing a basis for resource provisioning can be improved. Then we propose our resource provisioning methods. We model a multi-tier cloud service as a queuing network, and compute each tier's required resources. Compared to the service time mix-aware method, the experimental result shows our

method can improve the precision of workload prediction, reduce the allocated resources, and SLAs violations.

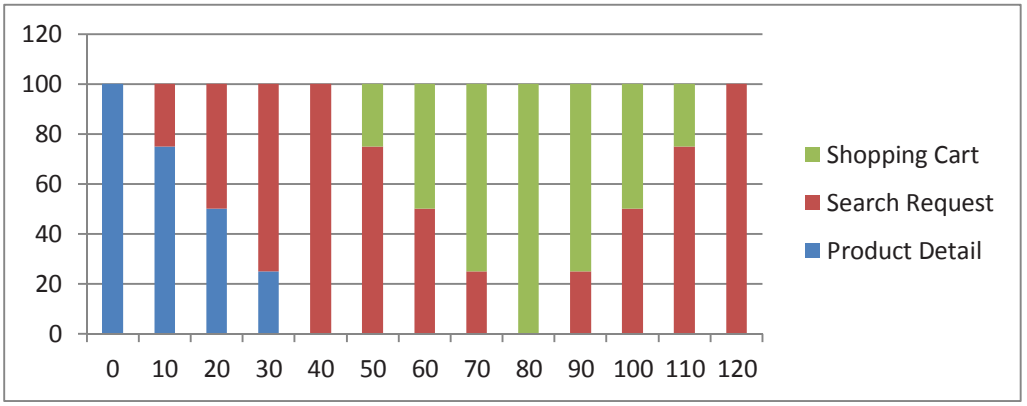

Fig. 2(a) mix of interaction in the workload

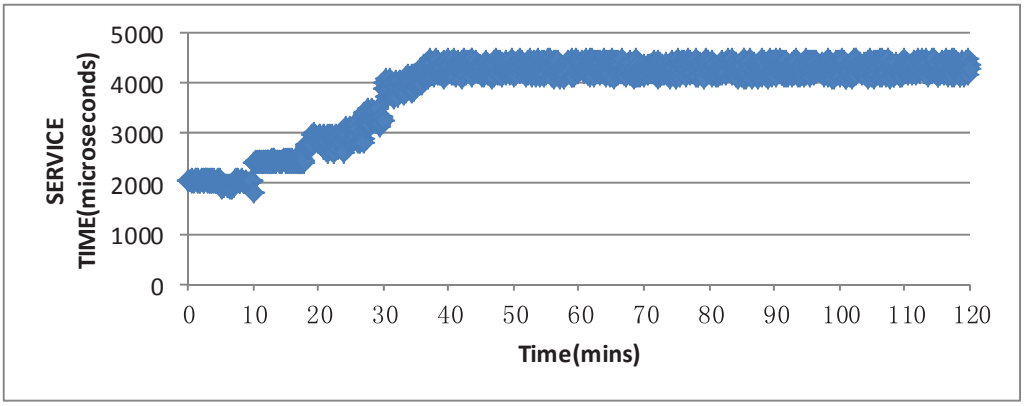

Fig. 2(b) average service time

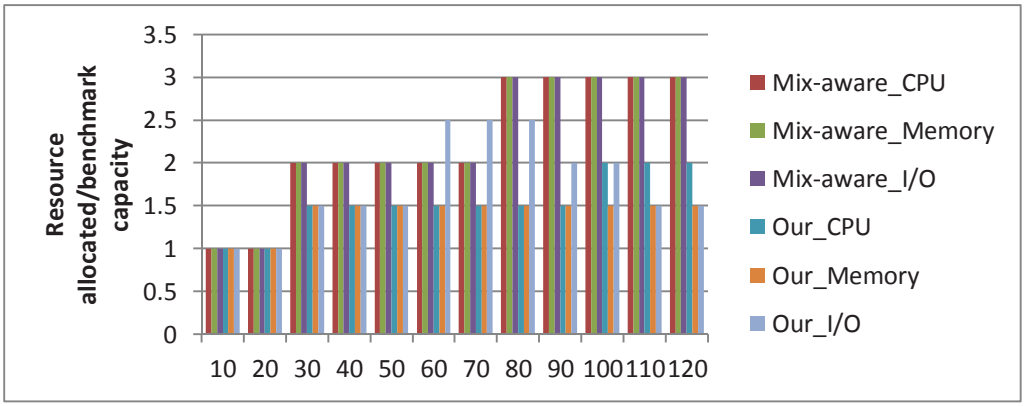

Fig. 2(c) ratio of the allocated resource to the benchmark

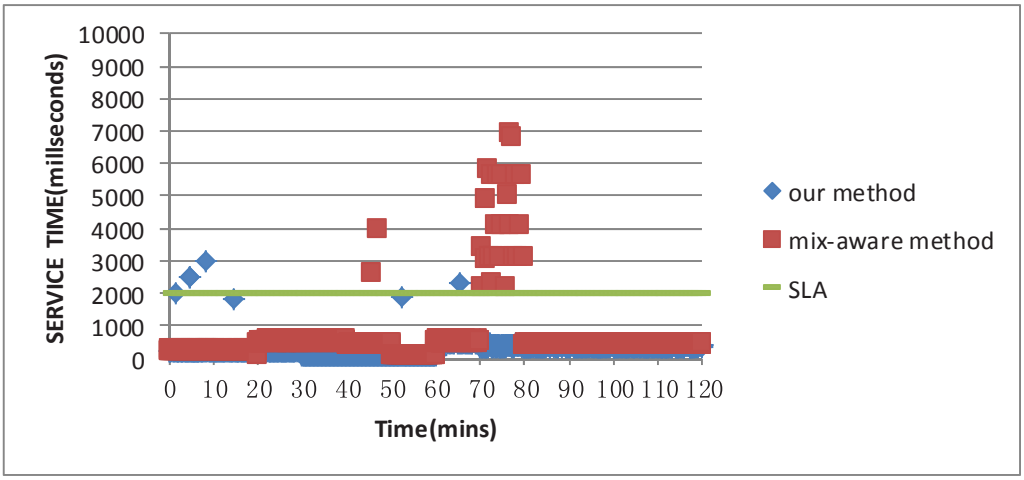

Fig. 2(d) 95th percentage response time

Acknowledgements

This work is supported by the Natural Science Foundation of China(No. 61063012), Guangxi Natural Science Fund (No. 2012GXNSFAA053222), Guangxi university talents support project(No. [2011]40), Guangxi provincial scientific research projects([2010]10).

References

[1]R. Buyya, C. S. Yeo, and S. Venugopal. Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities. Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications, 2008, 5-13
[2]B. Urgaonkar, P. Shenoy, A. Chandra, et al. Agile Dynamic provisioning of multi-tier internet applications. ACM Transactions on Autonomous and Adaptive Systems, 2008, 1-39
[3]Jing Bi, Zhiliang Zhu, Ruixiong Tian, et al. Dynamic provisioning modeling for virtualized multi-tier applications in cloud. Proceedings of 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), 2010, 370 - 377
[4]W. Iqbala, M. N. Dailey, D. Carrera, et al. Adaptive resource provisioning for read intensive multi-tier applications in the cloud. Future Generation Computer Systems Journal, 2011; 27: 871-879
[5]E. Kalyvianaki, T. Charalambous, and S Hand. Resource provisioning for multi-tier virtualized server applications. Computer Measurement Group (CMG) Journal, 2010; 126: 6-17
[6]H. Goudarzi and M. Pedram. Multi-dimensional SLA-based resource allocation for multi-tier cloud computing systems. Proceedings of 2011 IEEE International Conference on Cloud Computing (CLOUD), 2011, 324 – 331
[7]K. Joshi, M. Hiltunen, and G. Jung. Performance aware regeneration in virtualized multitier applications. Workshop on Proactive Failure Avoidance Recovery and Maintenance, 2009.
[8]P. Lama, and Xiaobo Zhou. Efficient server provisioning with control for end-to-end response time guarantee on multitier clusters. IEEE Transactions on Parallel and Distributed Systems, 2012; 23: 78-86
[9]W. Lloyd, S. Pallickara, O. David, et al. Performance implications of multi-tier application deployments on Infrastructure-as-a-Service clouds: Towards performance modelling. Future Generation Computer Systems,

2013; 29: 1254–1264
[10]R. Singh, U. Sharma, E. Cecchet, et al. Autonomic mix-aware provisioning for non-stationary data center workloads. Proceedings of the 7th international conference on Autonomic computing, 2010, 21-30
[11]M. Welsh and D. Culler. Adaptive overload control for busy internet servers. Proceedings of USENIX Symposium Internet Technologies and Systems (USITS), 2003, 43-57