Cost-Efficient Resource Allocation for Multi-tier Web Applications in a Cloud Environment

Adnan Ashraf
Department of Information Technologies, Åbo Akademi University, Turku, Finland.
Email: adnan.ashraf@abo.fi
Turku Centre for Computer Science (TUCS), Turku, Finland.
Department of Software Engineering, International Islamic University, Islamabad, Pakistan.

***Abstract*—In this research work, our focus is on deployment platforms to ensure scalability of web applications from really low to really high loads. We study dynamic resource allocation approaches to autonomously deploy and scale multiple web applications on a given Infrastructure as a Service cloud. The proposed approach provides automatic deployment and scaling of web applications and cost-efficient resource allocation for the application server and the database server tiers.**

I. Introduction

Web applications are often deployed in a three-tier computer architecture that consists of client, application, and database tiers. Both the application and the database tiers are implemented using a computer cluster to be able to process many user requests simultaneously. Traditionally, these clusters are composed of a fixed number of computers and are dimensioned to serve a predetermined maximum number of concurrent users. However, Infrastructure as a Service (IaaS) clouds currently offer computing resources on demand, such as storage and virtual machines (VMs), which can be used to create a dynamically scalable server tier consisting of a varying number of VMs.

Determining the number of VMs to provision for a cluster is an important problem. The exact number of VMs needed at a specific time depends upon the user load and the Quality of Service (QoS) requirements. Allocating too little resources will lead to subpar service, allocating too much resources will lead to increased operation costs.

II. Problem Statement

Our primary focus in this research work is on designing a cost-efficient resource allocation approach for multi-tier web applications in a cloud environment. However, the existing literature on admission control [1], [2] and server consolidation [3], [4] suggest that, for cost-efficiency, it is necessary to augment resource allocation with admission control and server consolidation mechanisms.

A. Cost-Efficient Resource Allocation

The main objective is to design dynamically scalable server tiers to deploy and scale a large number of web applications of varying resource needs. Thus, in addition to dynamic scaling of the application server and the database server tiers, the proposed dynamic resource allocation approach should also provide automatic deployment and scaling of web applications.

With the contemporary IaaS offerings, provisioning of a VM takes a considerable amount of time. Due to this inevitable delay, handling of a sudden peak load becomes very challenging. Therefore, the resource allocation approach should be augmented with a mechanism to handle VM provisioning delay.

The cost-efficiency can be further improved if the resource allocation approach supports deployment of multiple simultaneous applications on a single VM. Thus, at any given time, an application may be deployed in zero, one or more VMs. Popular applications would often be deployed in many VMs, while sporadically used applications would not be deployed at all in order to save resources.

B. Server Consolidation

When deploying and scaling a large number of applications in a shared hosting environment, one of the main challenges is to ensure that any underutilized VMs are consolidated together in order to reduce the number of required VMs. VM consolidation should migrate all active sessions from the least loaded underutilized VMs to other existing VMs and then terminate the least loaded VMs. This is achievable with the support for live migration [4]. However, the main challenge is to augment resource allocation with a server consolidation mechanism [3], [4], which uses minimum number of VMs with minimum number of migrations.

C. Admission Control

Resource allocation alone does not prevent servers from becoming overloaded [1], [2]. One of the main reason is that the load balancer may always direct some new sessions to an already overloaded server. Another reason is the VM provisioning delay, which may lead to over-admission. Therefore for preventing servers from becoming overloaded, resource allocation should be augmented with an admission control mechanism. For improved QoS, the proposed admission control mechanism should provide adaptive session-based admission control [2].

III. Related Work

The existing works on dynamic resource allocation can be classified into two main categories: Plan-based approaches and control theoretic approaches. Ardagna et al. [5], TwoSpot [6], Hu et al. [7], Chieu et al. [8], and Iqbal et al. [9] are plan-based approaches, while Dutreilh et al. [10], Pan et al. [11], and Patikirikorala et al. [12] are control theoretic approaches. One common characteristic of all of these existing works is that they use a dedicated hosting environment, where each VM is used to deploy one particular application. Another common characteristic is that they do not provide a separate mechanism for scaling of individual web applications.

Server consolidation approaches, such as [3], [4], dynamically reallocate VMs to physical nodes with the aim of minimizing the total number of nodes. However, in our context, we require VM consolidation, which should migrate all active sessions from the least loaded underutilized VMs to other existing VMs, thus releasing the least loaded VMs for termination.

Admission control approaches, such as [1], [2], aim to prevent server overloading under high load situations. One common characteristic of these traditional approaches is that they make decisions only on acceptance or rejection of incoming user load. In the context of cloud computing, it may also be possible to allow incoming load to wait until new VMs are provisioned or existing VMs become less loaded. Therefore, there is an opportunity to develop an admission control mechanism, which may choose between using an existing VM or provisioning a new VM for each new user session.

IV. Proposed Approach and Evaluation Plan

We propose a dynamic resource allocation approach that provides cost-efficient resource allocation for the application server and the database server tiers. It also provides cost-efficient resource allocation for multiple web applications, while deploying and scaling multiple simultaneous applications on each VM. Moreover, for preventing servers from becoming overloaded, the resource allocation is augmented with an adaptive session-based admission control mechanism. Similarly, the underutilization of VMs is minimized by providing a VM consolidation mechanism.

The proposed approach will be first validated by creating discrete-event simulations. Once validated, it will be implemented in a prototype, which will be deployed on an IaaS cloud.

V. Expected Contributions of the PhD Research

The main expected contribution is a dynamic resource allocation approach to create scalable server tiers for deploying and scaling multiple simultaneous web applications. Some more specific contributions include dynamic resource allocation and deallocation algorithms, an adaptive session-based admission control approach, an approach for automatic load generation on web applications, and a VM consolidation algorithm.

VI. Obtained Results

The first outcome of this work is an approach for automatic load generation for performance and scalability testing of web applications called ASTORIA [13]. Dynamic resource allocation with integrated admission control and server consolidation mechanisms is an ongoing research project. We have been currently working on an approach to create dynamically scalable application server tiers that deploy and scale multiple web applications per VM [14].

Acknowledgements

The author wants to thank Professor Ivan Porres at Åbo Akademi University for his guidance and encouragement.

References

[1] S. Muppala and X. Zhou, "Coordinated session-based admission control with statistical learning for multi-tier internet applications," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 20 – 29, 2011.

[2] L. Cherkasova and P. Phaal, "Session-based admission control: a mechanism for peak load management of commercial web sites," *Computers, IEEE Transactions on*, vol. 51, no. 6, pp. 669 –685, jun 2002.

[3] W. Vogels, "Beyond server consolidation," *ACM Queue*, vol. 6, no. 1, pp. 20–26, Jan. 2008.

[4] A. Murtazaev and S. Oh, "Sercon: Server consolidation algorithm using live migration of virtual machines for green computing," *IETE Technical Review*, vol. 28, no. 3, pp. 212–231, 2011.

[5] D. Ardagna, C. Ghezzi, B. Panicucci, and M. Trubian, "Service provisioning on the cloud: Distributed algorithms for joint capacity allocation and admission control," in *Towards a Service-Based Internet*, ser. Lecture Notes in Computer Science, E. Di Nitto and R. Yahyapour, Eds. Springer Berlin / Heidelberg, 2010, vol. 6481, pp. 1–12.

[6] A. Wolke and G. Meixner, "TwoSpot: A cloud platform for scaling out web applications dynamically," in *Towards a Service-Based Internet*, ser. Lecture Notes in Computer Science, E. Di Nitto and R. Yahyapour, Eds. Springer Berlin / Heidelberg, 2010, vol. 6481, pp. 13–24.

[7] Y. Hu, J. Wong, G. Iszlai, and M. Litoiu, "Resource provisioning for cloud computing," in *Proceedings of the 2009 Conference of the Center for Advanced Studies on Collaborative Research*, ser. CASCON '09. New York, NY, USA: ACM, 2009, pp. 101–111.

[8] T. Chieu, A. Mohindra, A. Karve, and A. Segal, "Dynamic scaling of web applications in a virtualized cloud computing environment," in *e-Business Engineering, 2009. ICEBE '09. IEEE International Conference on*, oct. 2009, pp. 281 –286.

[9] W. Iqbal, M. N. Dailey, D. Carrera, and P. Janecek, "Adaptive resource provisioning for read intensive multi-tier applications in the cloud," *Future Generation Computer Systems*, vol. 27, no. 6, pp. 871 – 879, 2011.

[10] X. Dutreilh, N. Rivierre, A. Moreau, J. Malenfant, and I. Truck, "From data center resource allocation to control theory and back," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, july 2010, pp. 410 –417.

[11] W. Pan, D. Mu, H. Wu, and L. Yao, "Feedback control-based QoS guarantees in web application servers," in *High Performance Computing and Communications, 2008. HPCC '08. 10th IEEE International Conference on*, sept. 2008, pp. 328 –334.

[12] T. Patikirikorala, A. Colman, J. Han, and L. Wang, "A multi-model framework to implement self-managing control systems for QoS management," in *Proceedings of the 6th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, ser. SEAMS '11. New York, NY, USA: ACM, 2011, pp. 218–227.

[13] N. Snellman, A. Ashraf, and I. Porres, "Towards automatic performance and scalability testing of rich internet applications in the cloud," *Software Engineering and Advanced Applications, Euromicro Conference*, vol. 0, pp. 161–169, 2011.

[14] T. Aho, A. Ashraf, M. Englund, J. Katajamäki, J. Koskinen, J. Lautamäki, A. Nieminen, I. Porres, and I. Turunen, "Designing IDE as a service," *Communications of Cloud Software*, vol. 1, no. 1, December 2011.