Simulation Modelling Practice and Theory 70 (2017) 83–100

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

Cloud autoscaling simulation based on queueing network model

T. Vondra*, J. Šedivý

Department of Cybernetics, Czech Technical University, Faculty of Electrical Engineering, Technická 2, Prague 6, 166 27, Czechia

ARTICLE INFO

Article history:
Received 24 February 2016
Revised 8 August 2016
Accepted 20 October 2016

Keywords:
Cloud computing
Simulation
CloudSim
Automatic scaling
Queueing networks
PDQ

ABSTRACT

For the development of a predictive autoscaler for private clouds, an evaluation method was needed. A survey of available tools was made, but none were found suitable. The CloudAnalyst distribution of CloudSim was examined, but it had accuracy and speed issues. Therefore, a new method of simulation of a cloud autoscaler was devised, with a queueing network model at the core. This method's outputs match those of a load test experiment. It is then evaluated with basic threshold-based algorithms on traces from e-commerce websites taken during Christmas. Algorithms based on utilization, latency, and queue length are assessed and compared, and two more algorithms combining these metrics are proposed. The combination of scaling up based on latency and down based on utilization is found to be very stable and cost-efficient. The next step will be the implementation of predictive methods into the autoscaler, which were already evaluated in the same R language environment.

1. Introduction

The cloud is becoming ubiquitous. It is considered as a deployment option with nearly any software project nowadays. Nevertheless, the workloads on the cloud are very often legacy static applications.

The properties of cloud, mainly the elasticity of resources used by an application and the agility in the reuse of physical resources between different applications and even different users, which were not present in server virtualization products before the cloud age, can only be used if applications are engineered with them in mind.

Elasticity can be exploited only if there is an autoscaling element in the applications, which monitors their resource usage and changes the amount of used resources accordingly. It is also known as the self-adaptive load balancer. We think there is not enough knowledge of cloud performance engineering among application developers and deployers, so they cannot set up the autoscalers effectively. In this work, we present a way to simulate the autoscaler with different algorithms and settings, which should help alleviate fears associated with its use and aid with estimation of cloud service costs.

If there is an autoscaler on every interactive workload in a cloud (we are mostly concerned with private cloud clusters), resources will get properly released in periods of low usage. This will allow the provider to shut down inactive compute resources, which is a fundamental element of Green Computing, which we define as a drive towards consolidating workload to as few machines as possible while maintaining SLA (Service Level Agreements).

* Corresponding author.
E-mail addresses: vondrto6@fel.cvut.cz (T. Vondra), sedivja2@fel.cvut.cz (J. Šedivý).

http://dx.doi.org/10.1016/j.simpat.2016.10.005
1569-190X/

Green computing is a part of the provider's view on the cloud. Embracing the concept of IaaS (Infrastructure as a Service), i.e. offering computing power as a service between subjects (be it two companies or just departments inside the same one), optimization can be done either on the side of the provider or the customer. The provider of cloud services wants to minimize the running costs of the system while maintaining GoS (Grade of Service) for the customer, which at the level of IaaS means availability of memory, CPU cycles and network and disk bandwidth in agreed upon amount and quality.

Leaving compute resources active even in periods of low loads enables a data center to achieve good values of PUE (Power Usage Effectiveness), which is a measure of data center efficiency comparing the total power draw to the power draw of only the computing components (excluding mainly cooling and power supply power usage). When servers are shut down at night, this indicator will become worse, but the total energy usage should be lower, which is what matters the most.

In our previous work [1], we proposed to exploit cloud agility between interactive and batch computations. We think there needs to be a service, which forecasts the usage of resources of the interactive tasks so that longer running grid-type tasks can be run without the risk of being terminated because the interactive workloads needs more computing power.

The forecasting service will use data about resources used on the cluster by interactive traffic (taken from the autoscalers), do a prediction on that, and fill the unused resources by batch jobs, yielding a cluster that is highly utilized all the time, but with minimal job preemptions or terminations (depending on the batch queue used). The simulation method presented here is a part of this research. In the article mentioned, we have demonstrated the feasibility of forecasting the load of several different purpose web services using statistical methods. The result can be utilized not only on the provider's side to turn computers off at night or schedule batch jobs in periods of low activity but also on the client's side for predictive autoscaling, decreasing their cloud service costs. It answers the question: "How many slots for VMs will be used for the next X h with probability P?"

From conversations with multiple companies, the most desired place to apply optimization is currently the customer side of IaaS. It also correlates with the lower proliferation of private cloud, particularly in the Czech Republic. The companies are mostly clients of public clouds and would like advice in the form of performance models or autoscaler deployment settings that would let them save on cloud costs. Moreover, the client-side optimization can also be employed in a private cloud, once they build it. On the side of the cloud client, who does not see the infrastructure, there is space for optimization inside the autoscaler, which is where the proposed simulation platform comes in. The goal is to minimize the cost the customer pays for cloud resources while maintaining GoS for end users. The end user GoS is defined as the response time distribution of the web application.

We think that performance prediction will also make a significant contribution to autoscaler quality and will be implemented in the simulation framework in the next release. If its benefit is verified, it will be applied either in our autoscaling software ScaleGuru [1] or as a module for OpenStack. Current autoscalers available for both private and public clouds are reactive, meaning that they can add resources after a threshold of some monitored performance variable is breached. If the autoscaler contained some prediction mechanism, it could add resources before the threshold was breached. That would allow the thresholds to be set higher without affecting GoS.

The prediction methods for entire data centers and single applications may not be the same and may require different parameter settings. However, datacenter-scale data is not easy to obtain. We tried negotiating with Czech companies offering cloud computing, such as TC Pisek, Odorik, Master Internet, Forpsi, and PonyCloud. The problems we encountered were mostly that their systems are too small and are not ready for autoscaling or virtual machine migration. We did not get any response from other European cloud companies except one, which offers Cloud Foundry, which in its open-source version lacks any monitoring functions, which are a prerequisite for autoscaling. With emerging global cloud providers such as Mega and Digital Ocean, there was already a problem with data privacy concerns. Our only source of data remains at the Masaryk University in Brno, which operates the Czech national grid Metacentrum and, as an associated service, a large OpenNebula cluster MetaCloud[1]. The problem with this data is that, in contrast with a business cloud, a scientific cloud lacks any interactive services (only about 1% of the virtual machine traces have daily seasonality, which indicates human interactive use).

Popular demand and data availability led us to focus our project on the client side on IaaS and to develop the hereby presented simulation method. The proposed forecasting algorithms from our previous work also need to be tested and tuned in a simulation environment before being implemented in a real scaling application.

Due to numerous citations (See Section 2), the CloudSim event-based simulator was the first candidate for a simulation platform. Another, yet unpublished, article of ours [2], presents our work on implementing a reactive autoscaling policy inside of CloudSim, which required the writing of methods to add and remove VMs at run time and to provide improved statistics gathering. However, the predictive autoscaler was not implemented in the simulator due to poor accuracy observed in a load test experiment. Efforts were redirected to finding a suitable replacement simulation platform.

After performing a load test, we suspected that the problems in the simulator were due to erroneous queueing logic. We have rewritten that, and also the traffic generation code, getting a version of CloudSim that is capable of reproducing a load test experiment with reasonable accuracy and in a manner consistent with queueing theory. The rewrite is the main focus

[1] http://www.metacentrum.cz/en/cloud.