ElastMan: Autonomic Elasticity Manager for Cloud-Based Key-Value Stores

Ahmad Al-Shishtawy
KTH Royal Institute of Technology
Stockholm, Sweden
ahmadas@kth.se

Vladimir Vlassov
KTH Royal Institute of Technology
Stockholm, Sweden
vladv@kth.se

ABSTRACT

The increasing spread of elastic Cloud services, together with the pay-as-you-go pricing model of Cloud computing, has led to the need of an elasticity controller. The controller automatically resizes an elastic service in response to changes in workload, in order to meet Service Level Objectives (SLOs) at a reduced cost. However, variable performance of Cloud virtual machines and nonlinearities in Cloud services complicates the controller design. We present the design and evaluation of ElastMan, an elasticity controller for Cloud-based elastic key-value stores. ElastMan combines feedforward and feedback control. Feedforward control is used to respond to spikes in the workload by quickly resizing the service to meet SLOs at a minimal cost. Feedback control is used to correct modeling errors and to handle diurnal workload. We have implemented and evaluated ElastMan using the Voldemort key-value store running in a Cloud environment based on OpenStack. Our evaluation shows the feasibility and effectiveness of our approach to automation of Cloud service elasticity.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Control theory*

Keywords

Cloud Computing; Elasticity Controller; Key-Value Store

1. INTRODUCTION

Cloud computing [3], with its pay-as-you-go pricing model, provides an attractive environment to provision elastic services as the running cost of such services becomes proportional to the amount of resources needed to handle the current workload.

However, sharing the physical resources among Virtual Machines (VMs) running different applications makes it challenging to model and predict the performance of the VMs [5]. Managing the resources for Web 2.0 applications, in order to guarantee acceptable performance, is challenging because of the highly dynamic workload that is composed of both gradual (diurnal) and sudden (spikes) variations [2].

HPDC'13, June 17–21, 2013, New York, NY, USA.
ACM 978-1-4503-1910-2/13/06.

The pay-as-you-go pricing model, elasticity, and dynamic workload of Web 2.0 applications, call for the need for an elasticity controller that automates the provisioning of Cloud resources. The elasticity controller leverages the horizontal scalability of elastic services by provisioning more resources under high workloads in order to meet required SLOs. The pricing model provides an incentive for the elasticity controller to release extra resources once the workload decreases.

2. ELASTICITY CONTROLLER

The objective of ElastMan is to regulate the performance of key-value stores according to a predefined SLO expressed as the 99th percentile of read operations latency over a fixed period of time (R99p thereafter). To address the challenges of controlling a noisy signal and variable performance of VMs, ElastMan consists of two components, a feedforward controller and a feedback controller. ElastMan relies on the feedforward controller to handle rapid large changes in the workload (e.g., spikes). This enables ElastMan to smooth the noisy 99th percentile signal and use the PI feedback controller to correct errors in the feedforward system model in order to accurately bring the 99th percentile of read operations to the desired SLO value. In other words, the feedforward control is used to quickly bring the performance of the system near the desired value and then the feedback control is used to fine tune the performance.

Due to the nonlinearities in elastic Cloud services, resulting from the diminishing reward of adding a service instance (VM) with increasing the scale, we propose a scale-independent model used to design the feedback controller. This enables the feedback controller to operate at various scales of the service without the need to use techniques such as gain scheduling. To achieve this, our design leverages the near-linear scalability of elastic service.

In the design of the feedback controller, we propose to model the target store using the average throughput per server as the control input. Although we cannot control the total throughput on the system, we can indirectly control the average throughput of a server by adding/removing servers. Adding servers reduces the average throughput per server under the same load, whereas removing servers increases the average throughput per server. Thus, the controller decisions become independent of the number of service instances. The major advantage of our proposed approach to model the store is that the model remains valid as we scale the store, and it does not depend on the number of severs.

For the feedforward controller we use a binary classifier built using logistic regression as proposed in [6]. The model

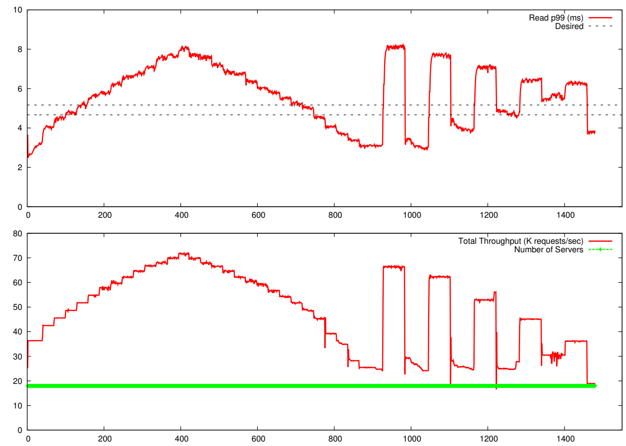

Figure 1: Performance of Voldemont without ElastMan with fixed number of servers (18 virtual servers) under gradual diurnal workload (0-900 min) and under workload with spikes (900-1500 min)

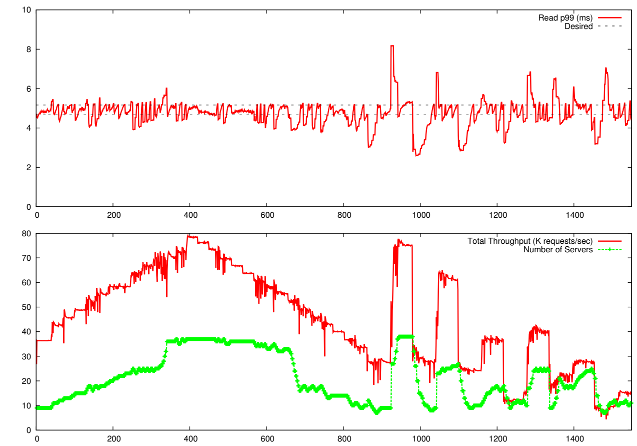

Figure 2: Performance of Voldemort with ElastMan under gradual diurnal workload (0-900 min) and under workload with spikes (900-1500 min)

is trained offline by varying the average intensity and the ratio of read/write operations per server. The final model is a line that splits the plane into two regions. In the region on and below the line, the SLO is met; whereas in the region above the line, SLO is violated. Ideally, the average measured throughput should be on the line, which means that the SLO is met with the minimal number of servers.

3. EVALUATION RESULTS

We have implemented ElastMan in order to evaluate our proposed approach to automation of Cloud service elasticity. In order to evaluate ElastMan, we have chosen the Voldemort (version 0.91) Key-Value Store [4] which is used in production in many applications such as LinkedIn. We run our experiments on a cluster of 11 servers each with two Intel Xeon X5660 processors (24 HW threads), and 44 GB of memory. The cluster runs Ubuntu 11.10. We setup a private Cloud using OpenStack Diablo release [1].

We have tested ElastMan controller with both gradual diurnal workload and sudden changes (spikes) in workload. The goal of ElastMan controller is to keep R99p at a value specified in the service SLO. In our experiments we choose the value to be 5 ms in 1 min period. Fig. 1 depicts the performance of Voldemort without ElastMan, i.e., with a fixed number of servers. Results shows that the store can not meet required SLO most of the time. Fig. 2 depicts the performance of Voldemort with ElastMan. Results show that ElastMan is able to keep the R99p within the desired region most of the time under a gradual workload (0-900 min) as well as under workload with spikes (900-1500 min).

4. CONCLUSIONS

ElastMan combines and leverages the advantages of both feedback and feedforward control. The feedforward control is used to quickly respond to rapid changes in workload. This enables us to smooth the noisy signal of the 99th percentile of read latency and thus use feedback control. The feedback control is used to handle gradual workload and to correct errors in the feedforward control due to the noise that is caused mainly by the variable performance of Cloud VMs. The feedback controller uses a scale-independent model by indirectly controlling the number of servers (VMs) by controlling the average workload per server. This enables the controller, given the near-linear scalability of key-value stores, to operate at various scales of the store.

5. ACKNOWLEDGMENTS

This research has been funded by the Complex System Engineering project in the ICT-TNG Strategic Research Areas initiative at KTH; the End-to-End Clouds project funded by the Swedish Foundation for Strategic Research; and the FP7 project Clommunity funded by the European Commission.

6. REFERENCES

[1] Openstack: Open source software for building private and public clouds. `http://openstack.org/`.

[2] M. Arlitt and T. Jin. A workload characterization study of the 1998 world cup web site. *Network, IEEE*, 14(3):30 –37, May/June 2000.

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. A view of cloud computing. *Commun. ACM*, 53(4):50–58, Apr. 2010.

[4] R. Sumbaly, J. Kreps, L. Gao, A. Feinberg, C. Soman, and S. Shah. Serving large-scale batch computed data with project voldemort. In *The 10th USENIX Conference on File and Storage Technologies (FAST'12)*, February 2012.

[5] O. Tickoo, R. Iyer, R. Illikkal, and D. Newell. Modeling virtual machine performance: challenges and approaches. *SIGMETRICS Perform. Eval. Rev.*, 37(3):55–60, Jan. 2010.

[6] B. Trushkowsky, P. Bodík, A. Fox, M. J. Franklin, M. I. Jordan, and D. A. Patterson. The SCADS director: scaling a distributed storage system under stringent performance requirements. In *Proceedings of the 9th USENIX Conference on File and Storage Technologies*, FAST'11, pages 12–12, USA, 2011.