Introducing Service-level Awareness in the Cloud

Cristian Klein[1], Martina Maggio[2], Karl-Erik Årzén[2], Francisco Hernández-Rodriguez[1]
[1] Umeå University, Sweden, [2] Lund University, Sweden
{cristian.klein,francisco}@cs.umu.se, {martina.maggio,karlerik}@control.lth.se

Managing the resources of a virtualized data-center is a key issue in cloud computing [1]. Existing research mostly assumes that applications are either allocated the required resources or fail [2–15]. Combined with the fact that most cloud applications have dynamic resource requirements [16], this imposes a fundamental limitation to cloud computing: To guarantee on-demand resource allocations, the data-center needs large spare capacity, leading to inefficient resource utilization.

This is especially problematic when dealing with unexpected events, such as flash crowds [17], hardware failures [18] and performance inference among applications [19]. These phenomena are well-known and software is readily written to cope with them, as long as resource provisioning is sufficient [14, 18, 20]. However, given the short duration and large magnitude of such events, provisioning enough capacity is often economically unfeasible. Hence, the data-center may become overloaded, rendering hosted applications unresponsive.

To efficiently and robustly deal with unexpected events, we introduce **service-level awareness** in the cloud. Application are augmented with a dynamic parameter, the **service-level (SL)**, that monotonically affects both their resource requirements and their delivered end-user experience. For example, online shops offer end-users recommendations of similar products. No doubt, such recommender engines greatly increase usability, however, they are highly resource demanding [21]. By selectively deactivating the corresponding code, resource requirements can be controlled at the expense of end-user experience. In case of unexpected events, the infrastructure can simply ask applications to temporarily reduce their requirements. Consequently, end-user experience is downgraded, but the user is at least provided with partial content in a timely manner.

SOCC '13, Oct 1–3 2013, Santa Clara, CA, USA
ACM 978-1-4503-2428-1/13/10.
http://dx.doi.org/10.1145/2523616.2525936

We built the necessary software to add SL-awareness to clouds, with contributions both on application-side as well as infrastructure-side.

On the application-side, we proposed a model for SL-aware cloud applications. It is applicable to any application featuring optional code. Execution of such code is desirable, as it improves user experience, but not necessary to satisfy the user's request. The optional code is activated or deactivated for each request, with a probability given by a dynamic parameter, the SL. More precisely, the model relates the SL of the application to its response time, which is used as an indicator for overload. Next, we synthesized a controller for the SL so as to keep the maximum response time around a given setpoint. The resulting cloud application self-adapts its SL to the input load and capacity available to it.

On the infrastructure-side, we implemented a resource manager that runs on each physical machine to complement the existing cloud stack. It allocates capacity among applications based on performance data they send, called *matching value*, thus hiding application internals from the infrastructure. This separation between SL choice, implemented by the application, and capacity allocations, implemented by the resource manager, allows the former to customize its definition of SL and the latter to run with a complexity linear in the number of applications. Using game theory, we proved that the resulting system converges to fair capacity allocations.

Finally, we evaluated the resulting framework experimentally, testing peak load handling and resource capacity distribution [22]. To show the applicability of our approach, we extended two well-known cloud benchmark applications, RUBiS [23] and RUBBoS [24], with SL-aware recommender engines, adding less than 170 lines of code. The results show that our proposition enables cloud infrastructures to more robustly deal with unexpected load peaks or unexpected failures, without requiring costly spare capacity. If capacity is abundant, applications execute at maximum service level, whereas during capacity shortages, applications gradually reduce their service level to maintain response time.

We believe that service-level awareness opens up a new level of flexibility in cloud computing, whose full potentials need to be further studied. Therefore, to foster further research, but also to make our results reproducible, we released all source code:
https://github.com/cristiklein/cloudish

References

[1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic. "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility". In: *Future Generation Computer Systems* 25.6 (2009).

[2] J. Xu and J. Fortes. "A multi-objective approach to virtual machine management in datacenters". In: *ICAC*. 2011.

[3] X. Meng, C. Isci, J. Kephart, L. Zhang, E. Bouillet, and D. Pendarakis. "Efficient resource provisioning in compute clouds via VM multiplexing". In: *ICAC*. 2010.

[4] X. Zhu, D. Young, B. J. Watson, Z. Wang, J. Rolia, S. Singhal, B. McKee, C. Hyser, D. Gmach, R. Gardner, T. Christian, and L. Cherkasova. "1000 Islands: Integrated Capacity and Workload Management for the Next Generation Data Center". In: *ICAC*. 2008.

[5] Q. Zhang, L. Cherkasova, and E. Smirni. "A Regression-Based Analytic Model for Dynamic Resource Provisioning of Multi-Tier Applications". In: *ICAC*. 2007.

[6] U. Sharma, P. Shenoy, S. Sahu, and A. Shaikh. "A Cost-Aware Elasticity Provisioning System for the Cloud". In: *ICDCS*. 2011.

[7] E. Feller, L. Rilling, and C. Morin. "Snooze: A Scalable and Autonomic Virtual Machine Management Framework for Private Clouds". In: *CCGrid*. 2012.

[8] A. Gulati, G. Shanmuganathan, A. Holler, and I. Ahmad. "Cloud-scale resource management: challenges and techniques". In: *HotCloud*. 2011.

[9] Z. Gong, X. Gu, and J. Wilkes. "PRESS: PRedictive Elastic ReSource Scaling for cloud systems". In: *CNSM*. 2010.

[10] R. Ghosh and V. K. Naik. "Biting Off Safely More Than You Can Chew: Predictive Analytics for Resource Over-Commit in IaaS Cloud". In: *CLOUD*. 2012.

[11] I. Hwang and M. Pedram. "Portfolio Theory-Based Resource Assignment in a Cloud Computing System". In: *CLOUD*. 2012.

[12] L. Wang, R. Hosn, and C. Tang. "Remediating Overload in Over-Subscribed Computing Environments". In: *CLOUD*. 2012.

[13] K. Mills, J. Filliben, and C. Dabrowski. "Comparing VM-Placement Algorithms for On-Demand Clouds". In: *CLOUDCOM*. 2011.

[14] A. J. Ferrer, F. Hernández, J. Tordsson, E. Elmroth, A. Ali-Eldin, C. Zsigri, R. Sirvent, J. Guitart, R. M. Badia, K. Djemame, W. Ziegler, T. Dimitrakos, S. K. Nair, G. Kousiouris, K. Konstanteli, T. Varvarigou, B. Hudzia, A. Kipp, S. Wesner, M. Corrales, N. Forgó, T. Sharif, and C. Sheridan. "OPTIMIS: A holistic approach to cloud service provisioning". In: *Future Generation Computer Systems* 28.1 (2012), pp. 66–77.

[15] S. Vijayakumar, Q. Zhu, and G. Agrawal. "Automated and dynamic application accuracy management and resource provisioning in a cloud environment". In: *GRID*. 2010.

[16] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch. "Heterogeneity and Dynamicity of Clouds at Scale: Google Trace Analysis". In: *SOCC*. 2012.

[17] P. Bodik, A. Fox, M. J. Franklin, M. I. Jordan, and D. A. Patterson. "Characterizing, modeling, and generating workload spikes for stateful services". In: *SOCC*. 2010.

[18] L. A. Barroso and U. Hölzle. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Morgan & Claypool, 2009.

[19] J. Mars, L. Tang, R. Hundt, K. Skadron, and M. L. Soffa. "Bubble-Up: increasing utilization in modern warehouse scale computers via sensible colocations". In: *MICRO*. 2011.

[20] J. Hamilton. "On designing and deploying internet-scale services". In: *Proceedings of the 21st conference on Large Installation System Administration Conference*. LISA'07. USENIX Association, 2007, 18:1–18:12. ISBN: 978-1-59327-152-7.

[21] J. A. Konstan and J. Riedl. "Recommended to you". In: *IEEE Spectum* (Oct. 2012).

[22] C. Klein, M. Maggio, K.-E. Årzén, and F. Hernández-Rodriguez. *Introducing Service-level Awareness in the Cloud*. Tech. rep. ISRN LUTFD2/TFRT-7641-SE. Lund University, July 2013. URL: `http://lup.lub.lu.se/record/3917495/file/3917498.pdf`.

[23] *Rice University Bidding System*. URL: `http://rubis.ow2.org`.

[24] *Bulletin Board Benchmark*. URL: `http://jmob.ow2.org/rubbos.html`.