Automated Heterogeneous Service Management in Cloud Systems

Christopher C. Lamb, Pramod A. Jamkhedkar,
Gregory L. Heileman, Chaouki T. Abdallah
University of New Mexico
Department of Electrical and Computer Engineering
Albuquerque, NM 87131-0001
{cclamb, pramod54, heileman, chaouki}@ece.unm.edu

May 25, 2011

Abstract

In this paper, we examine the problem of a single provider offering multiple types of service level agreements, the implications thereof, and finally simulate scenarios of a hypothetical provider managing these agreements under various stressful situations. We use usage management techniques in tandem with control-theoretic constructs to provide automated control and management of simulated system resources in the scenarios in question. In order to effectively engage with these types of techniques, we first analyze the system from both a usage management and control theoretic perspective, extracting service level objectives and indicators that are then profiled and managed.

1 Introduction

The past few years have witnessed unprecedented expansion of commercial computing operations as the idea of cloud computing has become more mainstream and widely adopted by forward thinking technical organizational leadership. This rate of adoption promises to increase in the near future as well. With this expansion has come opportunity as well as risk, embodied by recent major service outages at leading cloud providers like Amazon. These issues promise to become more difficult to control as managed infrastructure expands. This expansion will simply not be possible without large amounts of automation in all aspects of cloud computing systems.

The current state of the art in cloud systems is poorly differentiated and not as customer-focused as it could be. Current providers place the responsibility of monitoring performance and proving outages on the consumer rather than providing more transparent and monitorable infrastructure [REF NEEDED]. Furthermore, providers as a whole only provide one type of service level agreement (SLA) in a loosely-defined one-size-fits-all type of arrangement [REF NEEDED]. This provides strong differentiating opportunities for smaller, second generation cloud system providers who have established the technology required to scalably manage multiple, competing SLAs on the same infrastructure in tandem with clear customer system visibility.

These second generation providers will rely on automated infrastructure management in order to scale, and one key area to automate is resource provisioning and performance management.

Herein, we will elaborate the idea of applying usage management to single system governed by multiple different types of SLAs. In doing so we will apply common system design principles and standards [1], [2], [3], usage control ideas [4, 5, 6], computing control theory [7], [8], [9], [10], [11], [12], and

interoperability [13], [14], [15]. These ideas will all contribute toward a simple proof-of-concept system providing primitive usage management over a single autonomously controlled cloud provider service.

In Section **??** this paper first addresses how to create a controllable system with feedback suitable for system evaluation from the perspective of a single provider. Here, we will address the constraints and advantages of such an approach and how providers could begin to offer these kinds of services. In this first example, we will focus on QoS data specifically. Next in Section **??** we will extend our single provider system to provide control over attributes more specific to the usage management domain, with examples and associated analysis. Finally in Section **??** we extend this single provider model to a system deployed to multiple cloud providers in a realistic system-of-systems scenario.

1.1 Previous Work

Cloud computing is emerging as the future of utility systems hosting for consumer-facing applications. In these kinds of systems, components, applications, and hardware are provided as utilities over the Internet with associated pricing schemes pegged by system demand. Users accept specific QoS guidelines that providers use to provision and eventually allocate resources. These guidelines become the basis over which providers charge for services.

Over the past few years multiple service-based paradigms like web services, cluster computing and grid computing have contributed to the development of what we now call cloud computing [16]. Cloud computing distinctly differentiates itself from other service-based computing paradigms via a collective set of distinguishing characteristics: market orientation, virtualization, dynamic provisioning of resources, and service composition via multiple service providers [**?**]. This implies that in cloud computing, a cloud-service consumer's data and ap-

plications reside inside that cloud provider's infrastructure for a finite amount of time. Partitions of this data can in fact be handled by multiple cloud services, and these partitions may be stored, processed and routed through geographically distributed cloud infrastructures. These activities occur within a cloud, giving the cloud consumer an impression of a single virtual system. These operational characteristics of cloud computing can raise concerns regarding the manner in which cloud consumer's data and applications are managed within a given cloud. Unlike other computing paradigms with a specific computing task focus, cloud systems enable cloud consumers to host entire applications on the cloud (i.e. Software as a Service) or to compose services from different providers to build a single system. As consumers aggressively start exploiting these advantages to transition IT services to external utility computing systems, the manner in which data and applications are handled within those systems by various cloud services will become a matter of serious concern.

A growing body of research has begun to appear over the past two years applying control theory to tuning computer systems. These range from controlling network infrastructure [8] to controlling virtualized infrastructure and specific computer systems [9], [10] to exploring feedforward solutions based on predictive modeling [11]. Significant open questions remain to research within this field [7], [12].

2 Cloud System Models

Current cloud systems do not ignore SLA restrictions; rather, they are designed from the ground up to support a single type of SLA. That SLA generally encompasses total system uptime and some kind of response time metric [**?**, **?**]. If for some reason the cloud provider can no longer adhere to the terms outlined, some kind of compensation strategy applies to affected customers. Future cloud

providers can very well use the ability to support multiple SLAs as a way to differentiate available products from competitors.

2.1 Current Model

Current systems like Amazon's EC2 or Rackspace products are designed around high availability, and this is reflected in the focus of their supplied SLAs. This common design focus is also evident in the artifacts generated by other vendors [**?**]. Furthermore, Amazon offers clear guidance on how to develop systems that take advantage of their robust architecture as well as services that provide some measure of automatic scaling [**?**, **?**]. This combination of market leading position and products and the extensive supplied guidance make Amazon a clear choice to examine when reflecting on the current state-of-the-art.

Amazon's Cloud Watch products used in tandem with Auto Scaling provide the ability to control the number of deployed instances in response to specific system loads [**?**, **?**]. Cloud Watch gives customers the ability to monitor various system performance metrics for their virtual machines, including but not limited to latency, processor use, and request counts. Furthermore, users can set resource levels at which additional EC2 instances are created or destroyed. This provides some level of personalized management and control over deployed systems within Amazon's cloud infrastructure.

2.2 Future Reference Model

3 Service Level Agreements Defined

4 Controlling with Service Level Agreements

4.1 Computational Complexity

4.2 Space Complexity

4.3 Verification v. Solution

4.4 Approximation and other techniques

5 Control Theoretic Modeling

5.1 Results and Analysis

6 Conclusions and Future Works

6.1 Conclusions

6.2 Future Works

[17]

References

[1] M. S. Blumenthal and D. D. Clark, "Rethinking the design of the Internet: The end-to-end arguments vs. the brave new world," *ACM Transactions on Internet Technology*, vol. 1, no. 1, pp. 70–109, Aug. 2001.

[2] D. D. Clark, "The design philosophy of the DARPA internet protocols," in *ACM SIGCOMM*, Stanford, CA, Aug. 1988, pp. 106–114.

[3] D. D. Clark, J. Wroclawski, K. R. Sollins, and R. Braden, "Tussle in cyberspace: Defining tomorrow's internet," in *SIGCOMM*, Pittsburg, Pennsylvania, USA, Aug. 2002, pp. 347–356.

[4] P. A. Jamkhedkar and G. L. Heileman, "A formal conceptual model for rights," in *Proceedings of the Eighth ACM Workshop on Digital Rights Management*, Alexandria, VA, Nov. 2008.

[5] P. A. Jamkhedkar, G. L. Heileman, and C. Lamb, "An interoperable usage management framework," in *Proceedings of the Tenth ACM Workshop on Digital Rights Management*, Chicago, Oct. 2010.

[6] J. Park and R. Sandhu, "The UCON$_{ABC}$ usage control model," *ACM Trans. Inf. Syst. Secur.*, vol. 7, no. 1, pp. 128–174, 2004.

[7] X. Zhu, M. Uysal, Z. Wang, S. Singhal, A. Merchant, P. Padala, and K. Shin, "What does control theory bring to systems research?" *SIGOPS Oper. Syst. Rev.*, vol. 43, pp. 62–69, January 2009. [Online]. Available: http://doi.acm.org/10.1145/1496909.1496922

[8] Y. Ariba, F. Gouaisbaut, and Y. Labit, "Feedback control for router management and tcp/ip network stability," *Network and Service Management, IEEE Transactions on*, vol. 6, no. 4, pp. 255 –266, 2009.

[9] Z. Wang, Y. Chen, D. Gmach, S. Singhal, B. Watson, W. Rivera, X. Zhu, and C. Hyser, "Appraise: application-level performance management in virtualized server environments," *Network and Service Management, IEEE Transactions on*, vol. 6, no. 4, pp. 240 –254, 2009.

[10] M. Kjaer, M. Kihl, and A. Robertsson, "Resource allocation and disturbance rejection in web servers using slas and virtualized servers," *Network and Service Management, IEEE Transactions on*, vol. 6, no. 4, pp. 226 –239, 2009.

[11] S. Abdelwahed, J. Bai, R. Su, and N. Kandasamy, "On the application of predictive control techniques for adaptive performance management of computing systems," *Network and Service Management, IEEE Transactions on*, vol. 6, no. 4, pp. 212 –225, 2009.

[12] J. Hellerstein, S. Singhal, and Q. Wang, "Research challenges in control engineering of computing systems," *Network and Service Management, IEEE Transactions on*, vol. 6, no. 4, pp. 206 –211, 2009.

[13] P. A. Jamkhedkar and G. L. Heileman, "DRM as a layered system," in *Proceedings of the Fourth ACM Workshop on Digital Rights Management*, Washington, DC, Oct. 2004, pp. 11–21.

[14] G. L. Heileman and P. A. Jamkhedkar, "DRM interoperability analysis from the perspective of a layered framework," in *Proceedings of the Fifth ACM Workshop on Digital Rights Management*, Alexandria, VA, Nov. 2005, pp. 17–26.

[15] R. H. Koenen, J. Lacy, M. MacKay, and S. Mitchell, "The long march to interoperable digital rights management." *Proceedings of the IEEE*, vol. 92, no. 6, pp. 883–897, 2004.

[16] R. Buyya, "Market-oriented cloud computing: Vision, hype, and reality of delivering computing as the 5th utility," in *Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, ser. CCGRID '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 1–. [Online]. Available: http://dx.doi.org/10.1109/CCGRID.2009.97

[17] Y. Chen, D. Gmach, C. Hyser, Z. Wang, C. Bash, C. Hoover, and S. Singhal, "Integrated management of application performance, power and cooling in data centers," in *Network Operations and Management Symposium (NOMS), 2010 IEEE*, 2010, pp. 615 –622.