Engineering Proprioception in SLA Management for Cloud Architectures

Funmilade Faniyi, Rami Bahsoon
University of Birmingham
Edgbaston, Birmingham
B15 2TT, UK
f.faniyi@gmail.com, r.bahsoon@cs.bham.ac.uk

***Abstract*—With the wide adoption of the Cloud, there remains an open challenge to provide more dependable, transparent, and trustworthy provision of services. Service terms are typically defined in the Service Level Agreement (SLA) binding both service providers and users. For the service user, there is a need to ensure that s/he is enjoying the agreed level of service and any violations are reported accordingly. For the service provider, there is a need to manage a resilient infrastructure capable of meeting SLA terms and inform strategies for maximising profit and resource utilisation. The massive size, dynamism and unpredictability of Cloud architectures makes these goals difficult to accomplish using classic Service Level Management (SLM) approaches. In this paper, we motivate the need for novel dynamic and decentralised approaches for the design of SLM. Requirements and key design decisions for the new SLM are described. Also, a conceptual architecture for realising these requirements is presented. We roadmap and discuss research directions, which can benefit from the new SLM.**

***Keywords*-SLA, Cloud Computing, Cloud Architectures**

I. Introduction

Traditionally, service level agreements (SLAs) have been used as instruments to formalise the roles of parties to a contract and to clearly state expectations, penalties for violations and other contractual terms (E.g. legal and pricing). In the service economy (Grid computing, clustering, P2P, Cloud Computing etc), the need to automate SLAs has been well motivated due to cost-effectiveness, transparency, accountability and improved monitoring of provisioned services [1], [11]. Patterns and metric measurements from such SLAs could be used as input into more sophisticated mechanisms for resource management and price optimisation [2], [17], [18].

For the purpose of this paper, we refer to SLA as the contract composed among parties to a service (consumers and providers) consisting of well-defined expectations (for both functional and non-functional requirements) of the electronic service excluding legal issues and other computationally intractable terms. We describe the closely related term Service Level Management (SLM) as the overall infrastructure for negotiating, creating, managing and enforcing terms of an SLA.

The challenge for SLM in the Cloud is of particular interest because of the inevitable loss of control associated with migrating existing services to the Cloud or provisioning new services which were designed and deployed in the Cloud. We argue that the dependability of the Cloud relies heavily on the effectiveness of the SLM. Such infrastructure should provide guarantees for mitigating risks that may affect the Cloud users from both functional and non-functional dimensions. From a business perspective, the Cloud is perhaps the most cost-effective option for most startup small and medium enterprises (SMEs). Also, the Cloud holds promises for corporations willing to outsource aspects of their IT services which are outside their core competencies. Given the state-of-the-art in Cloud SLA specifications as exemplified by prominent Cloud providers like Amazon and Salesforce; the SLA is often entirely composed by the service provider and usually makes no provision for compliance monitoring. Hence, monitoring Cloud SLAs poses a major problem to decision makers who are faced with the question of whether to adopt Cloud Computing or not. Thus, there is a need to explore mechanisms for providing trustworthy monitoring and enforcing SLAs as agreed by participating parties.

Many of the existing research in the area of automated SLA have explored diverse approaches mostly tailored to environments where the assumptions fail to meet the requirements of the Cloud [7]. Examples of such assumptions include: (i) The existence of a substantial amount of time for negotiating SLA (possibly offline) prior to actual resource provisioning, (ii) The number of resources allocated to a job are fixed and require significant notice period should there be a need to increase such resources, and (iii) Entrusting SLA management function to an intelligent centralised controller with absolute knowledge about nodes in the network; this controller is often responsible for receiving SLA violation alerts and triggering re-provisioning actions to ensure SLAs are not violated.

We argue that the traditional centralised approaches to SLA monitoring fall short for the case of Cloud architectures. This is because of the dynamic topology, ultra-large scale, elasticity, and unpredictability underlying Cloud architectures [10]. We urge the need for a novel decentralised approach which relies on the dual concepts of *self-awareness* and *self-expression*[1] in engineering proprioception [9], [12] into ultra-large scale architectures as it is the case of Cloud. By self-awareness, we mean the ability of each node in the Cloud infrastructure to monitor the level of compliance to SLAs associated with the tasks under its control. By self-expression, we mean the ability of the node to trigger an alternative execution plan based on feedback from the environment about the extent to which it meets the task's SLA and reasoning about its own current state. These ideas aim to contribute to the foundational concepts required to incorporate self-awareness and self-expression properties into Cloud architectures to achieve more transparent SLA monitoring, improved resilience to events that could result to SLA violations and trustworthy compliance reporting.

The rest of this paper is structured as follows: section II gives an overview of related work. Our contribution to emerging requirements for Cloud SLM and its design model is the focus of section III, while section IV describes our approach for a conceptual architecture to realise these requirements. The impact assessment of this work and possible evaluation strategies are detailed in section V. The paper concludes in section VI with pointers to future work.

II. Related Work

Many of the previous approaches have assumed a fixed and well-defined external interaction between the service user and provider. In particular, they assume that there is a substantial period of time to negotiate the SLA, deploy it and then it remains fixed throughout the lifetime of the service. Many of these assumptions were motivated from research in Grid computing [7] in which the notion of dynamism is not as emphasised and impactful as evident in Cloud architectures. A number of previous works have outlined SLA requirements for electronic services in general [15] and Cloud Computing in particular [14]. While those requirements are fundamental to any automated SLA, they do not incorporate the dynamism inherent in Cloud Computing as a core requirement in their proposals.

Extensive research have be carried out in the area of formalising automated SLA with capabilities for efficient creation, monitoring, enforcement and storage for audit purposes. Most of these (E.g. SLAng [15], WSLA [8]) agree on the notion that any SLA formalism should be extensible and capable of incorporating as many unanticipated service parameters as possible. However, the real-time composition of SLA as stated in the Cloud Computing requirement (see section III) is not shared by all such formalism. Another setback is that many of these formalisms have focused more on representation of availability and performance metrics in the SLA, while the representation of security requirements (e.g. access control, authentication etc) in a way that can be easily monitored by the SLM framework is yet to be fully researched.

Concepts from the area of autonomic computing have also been adopted to facilitate self-management of the service infrastructure towards meeting service level objectives (SLO) [1], [5], [13], [16]. While such ideas are promising for dynamic provisioning, some of the underlying assumptions about the Cloud environment do not necessarily hold. For example, in [16], the assumption of a fixed number of nodes and a global controller possessing full knowledge of all the nodes in the network does not hold true for all Cloud environments. Research work in autonomic computing based SLA have also explored resource allocation policies [2], pricing models [17] and profit optimisation [18] for electronic services. While some of our ideas are similar to work in this area, we pursue a more robust and resilient architectural approach to the design of a SLM framework as discussed in section IV.

III. Cloud SLM: New Requirements and Key Design Decisions

This section contributes to the requirements and design model of a Cloud SLM which exhibits self-awareness and self-expression properties. We then describe the key design decisions.

A. Requirements

Here, we describe a set of emerging requirements for engineering a new generation of SLM framework for the Cloud as follows:

1) A negotiation mechanism with support for SLA customisation based on flexible definition of terms and conditions.
2) An efficient and transparent monitoring mechanism following SLA deployment.
3) Accurate detection and localisation of SLA violations.
4) Trustworthy SLA compliance and violation reporting.
5) Efficient re-provisioning/adjustment mechanism following violation of SLA.
6) Capability to re-negotiate service terms at any point during the service lifetime and rapidly readapt to meet newly negotiated terms.
7) A SLA specification language for formalising the functional and non-functional terms of the agreement in a way that Cloud users, providers and other affected parties can easily relate with.

Of particular importance is the need for re-negotiation of SLA parameters during active provisioning of the service. Cloud service providers must be able to accommodate such changes and respond to them in real-time. For example, the case of Animoto (as report by [3]) stated that the service provider experienced a load surge from 50 to 3500 servers within a three days period.

[1] These terms are the focus of a new on-going EU FP7 research project: Engineering Proprioception in Computer Systems (EPiCS), in which the authors are actively involved.

Using a classic centralised SLM approach, requirements 2-6 are difficult to accomplish within a massively scalable and distributed Cloud infrastructure with heterogeneous components. Hence, we propose moving the intelligence required to meet requirements 2-6 to the nodes within a fully- or semi-decentralised architecture. Consequently, self-awareness will inform decisions of whether to cooperate with other nodes to meet a service term or offload the tasks to other nodes due to anticipated failure. Each node utilises its self-expression properties to gather feedback about its behaviour and improve its strategies towards meeting service terms.

The novelty of this SLM paradigm is in the area of designing these self-adaptive mechanisms at the node-level to cope with unanticipated changes resulting from the Cloud environment, component failures and SLA re-negotiation. Furthermore, since Cloud architectures incorporating the proposed SLM paradigm will either be fully- or semi-decentralised; failure of critical nodes may not degrade the quality of service severely when compared with centralised architectures.

This new paradigm of reasoning about SLM in the Cloud would be of benefit to both users and service providers. This is evident because the Cloud users will be assured of more transparent monitoring and capable of enforcing penalties should the service provider fail to meet SLA terms. Similarly, service providers will benefit from reduced operational cost and time spent managing the Cloud infrastructure, since the components in the architecture possess self-adaptive properties with which they are capable of taking appropriate risk mitigation actions autonomously.

B. *Modelling Key Design Decisions*

We pursue the goal of designing novel SLM architectures to address the requirements outlined above. While previous work have largely assumed a centralised approach in the architecture of SLM frameworks, we seek a decentralised approach to realise these requirements. The motivation for our approach is encompassed in the following arguments:

- The Cloud is fundamentally dynamic and it is hard to ensure a robust SLA event monitoring and re-provisioning mechanism at any central point within the large distributed Cloud architecture.
- The rapid fluctuation in demand/supply for Cloud services, dynamic resource management, outsourcing of workload among Cloud providers and the inherent unpredictable failures of components within the Cloud architecture makes our approach more resilient to avoid violating SLAs and effectively localising faults in the event of component failures.
- A decentralised SLM framework where self-awareness and self-expression are exhibited at the nodes (e.g. server, cluster, VM, software component) is more likely to rapidly respond to changes in SLA resulting from re-negotiation of service terms.

The notion of what constitutes a node could differ based on various considerations. For example, in an Infrastrusture-as-a-Service (IaaS) environment (E.g. Amazon EC2), a node could be a VM or hypervisor. Thus, the distributed knowledge of the service terms for a particular class of applications/tasks may be encoded in the hypervisor which by itself is capable of learning about the level of compliance to the application SLA and if possible trigger an alternative provisioning decision should it be unable to meet the SLA at some point during the lifetime of the service.

The mechanism for intelligent decision making at the node could be reactive or anticipatory. Initially, the node may utilise on-line learning algorithms to capture the properties of the task it is executing, the behaviour of the environment and actions of other nodes within its neighbourhood. After sufficiently learning about these features, the node could anticipate mitigation strategies to avoid violating service terms before those events occurs. The virtualised nature of Cloud resources could lead to competitive resource contentions, hence we account for such scenarios within the design of our framework.

To ensure trustworthiness of the SLA monitoring and compliance process, we hope to explore lessons from trusted computing research. In particular, one approach is to exploit a virtualised Trusted Platform Module (vTPM) [4] for the nodes responsible for monitoring SLA within the Cloud infrastructure. A service user could ascertain the trustworthiness of such nodes via attestation protocols and consequently get assurance that the SLA reports are indeed valid.

IV. Architecture for Dynamic SLM

A candidate conceptual architecture for realising the requirements outlined in section III is presented in Figure 1.

At each layer of the Cloud infrastructure, nodes exhibit self-awareness and self-expression properties which makes them capable of taking decisions which were not anticipated at design time or encoded in policy specifications. In addition, each layer of the architecture is transparent to the others, hence, control could be transferred from a higher layer to a lower one and vice versa.

The lifecycle commences when a Cloud user negotiates an SLA with the service provider and both parties commit to it prior to provisioning the service. Once the Admission Controller in the Cloud infrastructure receives the job request, it makes a decision to allocate the job to one of the Resource Managers (RMs) based on its scheduling policies and self-awareness of the current state of the system. The RMs dispatches the execution of the job amongst a set of nodes (e.g. server, cluster, VM, software component).

In the event of an action that may lead to SLA violation, components at each layer of the architecture are capable of initiating mitigation strategies by cooperating

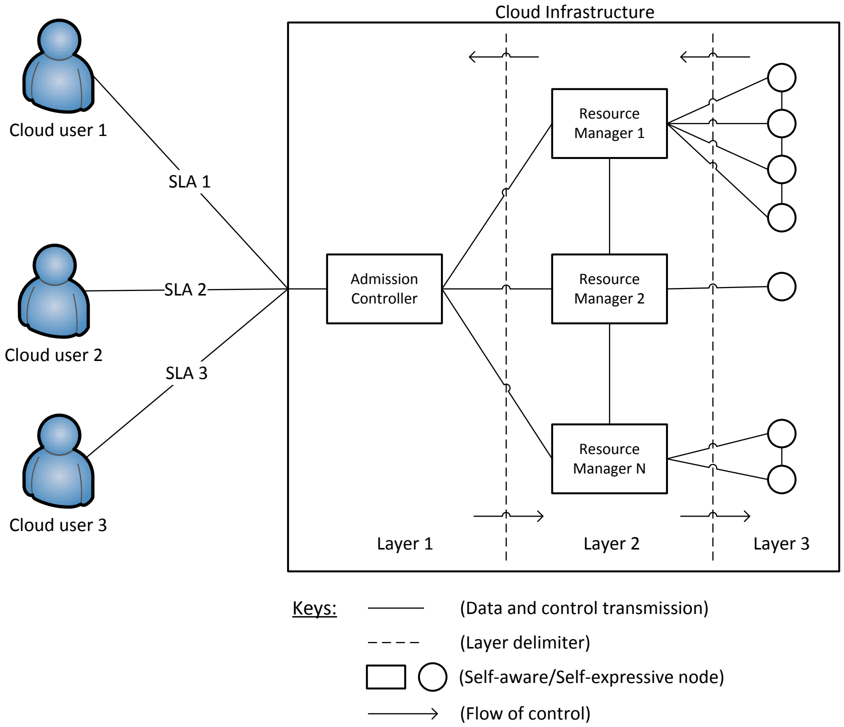

Figure 1. Candidate Conceptual Architecture for Cloud SLM

with other components on the same layer based on their self-awareness/self-expression properties. Only in the event that a service violation event that could not be mitigated at the layer where it was generated will components at the immediate higher layer be contacted to mitigate the risks of violating the SLA. In addition, each node is equipped with a trust function (E.g. vTPM) which serves the purpose of signing compliance and violation measurements before reporting to SLA parties.

V. Impact Assessment & Evaluation Strategies

Our approach to the design of SLM frameworks is beneficial to both the research and business communities. For example, trust and accountability via SLA is capable of improving Cloud adoption and also provides a basis for reputation management among Cloud providers with respect to their abilities to adhere to SLA terms. In addition, given that our approach does not assume any specific utility function, we see opportunities for leveraging on our SLM framework to address fundamental challenges in the Cloud via composition of utility functions tailored to different objectives. In particular, dynamic pricing models, resource management, power-aware Cloud strategies and investigative transparency are areas that could benefit from our ideas.

To realise these benefits, first, we seek answers to the following research questions:

- Given the various Cloud topologies and architecture-styles, what are the limits of a decentralised SLM framework?
- How efficient is a decentralised SLM framework when compared to a centralised approach with respect to SLA fulfillment, violation detection and adaptivity to unanticipated workload situations?
- To what extent can adaptivity benefit different components in the Cloud architecture?
- In case of scarce resources, what are the implications of different mechanisms for reasoning about job priorities?
- In the event of unavoidable failures, how can nodes make decisions about neglecting certain jobs in a way that will least impact the SLA objectives of the system as a whole?

VI. Conclusion

In this paper, we have outlined the challenges of monitoring and responding to service level agreement (SLA) in Cloud Computing. Thus, we motivated emerging requirements for a service level management (SLM) framework capable of addressing these challenges. We argued that the dynamism and unpredictability underlying Cloud users to service providers interactions and resource management mechanisms in the Cloud makes classic approaches incapable of meeting these requirements. Consequently, we proposed a generic architecture for a novel decentralised Cloud SLM framework in which nodes exhibit self-awareness and self-expression properties towards meeting SLAs. Important research questions were highlighted to stimulate future research work focused on enriching the proposed conceptual architecture.

Currently, we are exploring the idea of using market-based mechanisms [6] for designing Cloud architectures with the objective of reducing SLA violations. We shall investigate promising market mechanisms using simulations and at a later stage via empirical case studies of different Cloud scenarios. We shall carry out comparative studies of our results against other dynamic SLA management approaches using metrics such as number of SLA violations, efficiency of detection mechanism, violation localisation and adaptivity to unanticipated workloads.

Acknowledgment

The research leading to these results has received funding from the European Union Seventh Framework Programme under grant agreement n° 257906, Engineering Proprioception in Computer Systems (EPiCS), in which the authors are actively involved.

References

[1] B. Addis, D. Ardagna, B. Panicucci, and L. Zhang. Autonomic Management of Cloud Service Centers with Availability Guarantees. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pages 220 –227, jul. 2010.

[2] D. Ardagna, M. Trubian, and L. Zhang. SLA Based Resource Allocation Policies in Autonomic Environments. *J. Parallel Distrib. Comput.*, 67:259–270, Mar. 2007.

[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*, 53:50–58, April 2010.

[4] S. Berger, R. Caceres, K. A. Goldman, R. Perez, R. Sailer, and L. van Doorn. vTPM: Virtualizing the Trusted Platform Module. In *Proceedings of the 15th USENIX Security Symposium*, pages 305–320. USENIX, Aug. 2006.

[5] I. Brandic. Towards Self-Manageable Cloud Services. In *Computer Software and Applications Conference, 2009. COMPSAC '09. 33rd Annual IEEE International*, volume 2, pages 128 –133, jul. 2009.

[6] S. H. Clearwater, editor. *Market-based Control: A Paradigm for Distributed Resource Allocation*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1996.

[7] K. Czajkowski, I. Foster, and C. Kesselman. Agreement-Based Resource Management. *Proceedings of the IEEE*, 93(3):631 –643, Mar. 2005.

[8] A. Dan, D. Davis, R. Kearney, A. Keller, R. King, D. Kuebler, H. Ludwig, M. Polan, M. Spreitzer, and A. Youssef. Web services on demand: Wsla-driven automated management. *IBM Syst. J.*, 43:136–158, January 2004.

[9] EPiCS. Engineering Proprioception in Computing Systems. http://www.epics-project.eu/index.html [Last Accessed: 20-Mar-2011].

[10] F. Faniyi, R. Bahsoon, A. Evans, and R. Kazman. Evaluating Security Properties of Architectures in Unpredictable Environments: A Case for Cloud. (to appear). In *Proceedings of the 9th Working IEEE/IFIP Conference on Software Architecture, WICSA*, June 2011.

[11] H. Li, G. Casale, and T. Ellahi. SLA-Driven Planning and Optimization of Enterprise Applications. In *Proceedings of the first joint WOSP/SIPEW international conference on Performance engineering*, WOSP/SIPEW '10, pages 117–128, New York, NY, USA, 2010. ACM.

[12] S. Parsons, R. Bahsoon, P. R. Lewis, and X. Yao. Towards a Better Understanding of Self-Awareness and Self-Expression within Software Systems. Technical Report CSR-11-03, School of Computer Science, University of Birmingham, UK, 2011.

[13] P. Rubach and M. Sobolewski. Autonomic SLA Management in Federated Computing Environments. In *Proceedings of the 2009 International Conference on Parallel Processing Workshops*, ICPPW '09, pages 314–321, Washington, DC, USA, 2009. IEEE Computer Society.

[14] Z. Shu and S. Meina. An Architecture Design of Life Cycle based SLA Management. In *Advanced Communication Technology (ICACT), 2010 The 12th International Conference on*, volume 2, pages 1351 –1355, Feb. 2010.

[15] J. Skene, F. Raimondi, and W. Emmerich. Service-Level Agreements for Electronic Services. *IEEE Transactions on Software Engineering*, 99(RapidPosts):288–304, 2009.

[16] H. N. Van, F. Tran, and J.-M. Menaud. SLA-Aware Virtual Resource Management for Cloud Infrastructures. In *Computer and Information Technology, 2009. CIT '09. Ninth IEEE International Conference on*, volume 1, pages 357 –362, oct. 2009.

[17] C. S. Yeo, S. Venugopal, X. Chu, and R. Buyya. Autonomic Metered Pricing for a Utility Computing Service. *Future Gener. Comput. Syst.*, 26:1368–1380, oct. 2010.

[18] L. Zhang and D. Ardagna. SLA based Profit Optimization in Autonomic Computing Systems. In *Proceedings of the 2nd international conference on Service oriented computing*, ICSOC '04, pages 173–182, New York, NY, USA, 2004. ACM.