Automation of The SLA Life Cycle in Cloud Computing

(Extended Abstract)

Waheed Aslam Ghumman
Institut für Systemarchitektur, Fakultät Informatik
Technische Universität Dresden

Cloud computing has become a prominent paradigm to offer on-demand services for softwares, infrastructures and platforms. Cloud services are contracted by a service level agreement (SLA) between a cloud service provider (CSP) and a cloud service user (CSU) which contains service definitions, quality of service (QoS) parameters, guarantees and obligations. Cloud service providers mostly offer SLAs in descriptive format which is not directly consumable by a machine or a system. The SLA written in a natural language may impede the utility of rapid elasticity in a cloud service. Manual management of SLAs with growing usage of cloud services can be a challenging, erroneous and tedious task especially for the CSUs acquiring multiple cloud services. The necessity of automating the complete SLA life cycle (which includes SLA description in machine readable format, negotiation, monitoring and management) becomes imminent due to complex requirements for the precise measurement of the QoS parameters. Current approaches toward automating the complete SLA life cycle, lack in standardization, completeness and applicability to the cloud services. Automation of different phases of the SLA life cycle (e.g. negotiation, monitoring and management) is dependent on the availability of a machine readable SLA. In this work, a structural specification for the SLAs in cloud computing (S3LACC in short) is presented which is designed specifically for the cloud services, covers complete SLA life cycle and conforms with the available standards. Additionally, S3LACC defines a single SLA structure to be used as an SLA template and as a final agreement as well.

An SLA may include many negotiable QoS parameters with each QoS parameter having multiple acceptable values. Different combinations of the acceptable values for all of the QoS parameters grow exponentially as number of the QoS parameters and their selectable values increase. Manually negotiating for all of the QoS parameters considering their various options can be a specious and diligent task with a lower utility. In this work, a negotiation technique (*flip-flop* negotiation) is also presented (based on the S3LACC) for the SLAs in cloud services. The *flip-flop* negotiation strategy is based on a time dependent 3D utility function and on a dynamic concession strategy using the polynomial extrapolation of opponent's concession. This negotiation strategy is useful for the time-critical cloud services as it operates with a motive of reaching an SLA agreement at an earlier time. The *flip-flop* negotiation strategy estimates the expected final offer from the opponent and tries to reach at the same offer value earlier in time by increasing the concession in a safe mode to mitigate the possible utility loss due to increase in concession. Moreover, a multi-provider concurrent negotiation method is presented in this work using the *flip-flop* negotiation strategy. Apart from the presented negotiation strategy, the S3LACC is capable to include any custom (static or dynamic) negotiation strategy by modifying the negotiation parameters.

After successful SLA negotiation process, next leading task in the SLA life cycle is to monitor the cloud services for ensuring the quality of service according to the agreed SLA. Manually monitoring QoS parameters as agreed in the SLA can be error-prone, time consuming and laborious task resulting in consumption of human/technical resources. A distributed monitoring approach for the cloud SLAs is presented in this work, which is suitable for services being used at single or multiple locations. The proposed approach reduces the number of communications of SLA violations to a monitoring coordinator by eliminating the unnecessary communications. The decoupled nature of the proposed monitoring approach makes it adaptable for multiple locations without any inter-location effect. A monitoring simulation program is also presented to validate the proposed monitoring method with the help of extensive experiments.

This work includes a detailed overview of state-of-the-art approaches for the complete SLA life cycle and for the individual phases of the SLA life cycle as well. An implementation of the S3LACC, a prototype (covering the proposed methods) and a comparative analysis of presented work with state-of-the-art are given in this work. An illustrative use case is also presented to show the practicability of the S3LACC for the complete SLA life cycle.