SYBL+MELA: Specifying, Monitoring, and Controlling Elasticity of Cloud Services*

Georgiana Copil, Daniel Moldovan, Hong-Linh Truong, and Schahram Dustdar

Distributed Systems Group, Vienna University of Technology
{e.copil,d.moldovan,truong,dustdar}@dsg.tuwien.ac.at

Abstract. One of the major challenges in cloud computing is to simplify the monitoring and control of elasticity. On the one hand, the user should be able to specify complex elasticity requirements in a simple way and to monitor and analyze elasticity behavior based on his/her requirements. On the other hand, supporting tools for controlling and monitoring elasticity must be able to capture and control complex factors influencing the elasticity behavior of cloud services. To date, we lack tools supporting the specification and control of elasticity at multiple levels of cloud services and multiple elasticity metrics. In this demonstration, we will showcase a system facilitating the multi-level and cross-layer monitoring, analysis and control of cloud service elasticity.

1 Motivation

Simplifying the requirements specification and providing rich features for controlling and monitoring elasticity is crucial for several stakeholders, e.g. software service developers and providers, for exploiting the benefits of cloud systems. So far, existing frameworks or tools demonstrate limited, still complex, elasticity specification and controls, e.g., considering cost, or quality when deciding for control actions for cloud services [1–3], without considering the multi-dimensional nature of elasticity, and the fact that the cloud service developer or provider would be interested in describing requirements for different parts of his/her service, and at a high level, without worrying about virtual machine level information. Moreover, there is a lack of elasticity monitoring tools that support cross-layered, multi-dimensional elasticity of complex cloud services. In our work, we overcome some of the above-mentioned limitations by developing a system that performs multi-level monitoring data aggregation, analysis, and control of cloud services, paving the way for truly elastic cloud services.

By demonstrating our system, we could show that, on the one hand, the service providers or developers of the cloud services can easily monitor and specify elasticity requirements for their services at different levels and perspectives. On the other hand, we can discuss with other researchers how our control service takes automatic decisions for adapting the cloud service, in order to meet specified

* This work was supported by the European Commission in terms of the CELAR FP7 project (FP7-ICT-2011-8 #317790).

S. Basu et al. (Eds.): ICSOC 2013, LNCS 8274, pp. 679–682, 2013.

elasticity requirements. Such a demonstration, therefore, will be useful for us to understand the complexity to develop novel solutions to simplify the specification and support rich features for elasticity monitoring and control.

2 System Overview

The system to be demonstrated supports an integrated environment for specifying, controlling and monitoring elasticity of cloud services. Depicted in Figure 1, our system receives the cloud service description together with its elasticity requirements which are specified in SYBL [4]. When the cloud service is executed, the *Elasticity Control Service* manages an internal representation of the cloud service, which decomposes the cloud service into service topologies (parts of the cloud service) which can be further decomposed into service units. Each cloud service entity (e.g. service unit, service topology, virtual machines) is associated with various runtime elasticity metrics and elasticity requirements. The elasticity metrics are provided by the *MELA Service* which monitors the elasticity behavior of service entities and conducts elasticity analysis. Based on service structure, elasticity metrics and elasticity requirements, the *Elasticity Control Service* decides suitable elasticity control actions in order to fulfill the requirements.

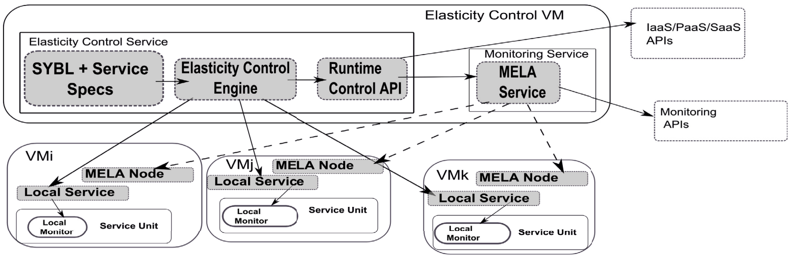

Fig. 1. Monitoring and controlling elastic cloud services

Controlling Elasticity of Cloud Services: For the specification of elasticity requirements we use SYBL language [4], which is a directive-based language enabling the description of high-level elasticity requirements and for the control of elasticity we use the control mechanism detailed in [5]. SYBL has three types of directives: `monitoring`, `constraints` and `strategies`, which can target different levels of the cloud service. The *Elasticity Control Service* analyses the elasticity requirements which can be conflicting or even contradicting, and produces new elasticity requirements on the basis of which it generates action plans for elasticity control of cloud services. For example, elasticity requirements at service topology level could be `Co1: CONSTRAINT costPerClientPerHour < 5 euro` or `Co2: CONSTRAINT responseTime < 0.5 s`. The *Elasticity Control Service* would evaluate these requirements, and in case one of them, let's say `Co1`, is violated, it would generate a new action plan consisting of a `scaleout` for a service unit inside the service topology, and a `reconfigure(''highPerformance'')` for the service topology. For evaluating whether or not the requirements are fulfilled and what would be the result of enforcing an action the *Elasticity Control Service* uses information coming from the *MELA Service*.

Monitoring and Analyzing Elasticity: Our system collects monitoring data from existing monitoring sources for different types of metrics, from virtual machine level metrics like memory, CPU, to application level metrics like response time or throughput. It aggregates the monitored metrics into higher level metrics, composing metrics associated with different levels of the cloud service (e.g. cloud service, service topology or service unit). A service unit can be deployed over several virtual machine instances, and therefore metrics targeting this level have be aggregated from metrics at virtual machine level. In the same manner, a cloud service is composed of several service topologies which in turn are composed of several service units and therefore metrics at cloud service level need to be aggregated from metrics at service topology level, which are in turn aggregated from service unit level. Moreover, we can have complex metrics as is the case for the metric targeted in constraint `Co1` which is aggregated by dividing cost per hour at service topology level and number of clients for the service topology. The cost at service level is the sum of the cost at service unit level, which in turn can be aggregated from, e.g. cost per virtual machine, cost for I/O operations or cost for network interface. The monitored metrics are analyzed for detecting whether or not the cloud service is in elastic behavior (i.e. fulfills all elasticity requirements), what could be the trend for the evolution of the metrics and what is the correlation among them. All this information is fed into the *Elasticity Control Service* for generating elasticity control action plans in case the cloud service, or parts of the cloud service do not expose the expected behavior, defined through elasticity requirements.

3 Demonstrating Cloud Services

We use a pilot, but realistic, Data-as-a-Service in an M2M (Machine-to-Machine) cloud platform as the cloud service in our demonstration. Figure 2 shows a snapshot of our demonstration[1] in which detailed service cloud structures together with their runtime elasticity behaviors and elasticity control actions are analyzed and presented to the developer/provider. Via a rich interface, the demonstration will show how elasticity requirements are described, how the cloud service provider can view in real-time complex high level monitoring metrics understanding their values and impact upon the cloud service and how the cloud service is automatically controlled for elasticity using all this information.

The user can specify SYBL elasticity requirements at different levels, e.g. in source code using Java annotations for service units, or in XML for service topologies. After being started, the *Elasticity Control Service* processes information from the monitoring service and takes elasticity control actions when the

[1] The demo video can be found at `dsg.tuwien.ac.at/research/viecom/prototypes/demo/syblmelaicsoc.wmv`

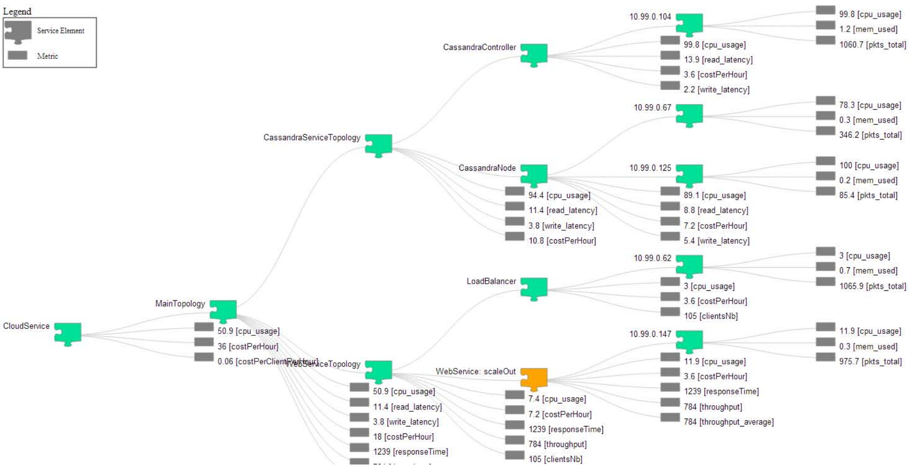

Fig. 2. Example of monitoring and controlling elasticity

requirements are not fulfilled. The user (e.g. cloud service provider) can view the monitoring interface during the application execution, for checking if the behavior is the expected one, for seeing how metrics are aggregated from lower to higher levels, and for learning how the application behaves under different circumstances (e.g. how does adding more resources impact quality?).

References

1. Kazhamiakin, R., Wetzstein, B., Karastoyanova, D., Pistore, M., Leymann, F.: Adaptation of service-based applications based on process quality factor analysis. In: ICSOC/ServiceWave 2009. LNCS, vol. 6275, pp. 395–404. Springer, Heidelberg (2010)
2. Guinea, S., Kecskemeti, G., Marconi, A., Wetzstein, B.: Multi-layered monitoring and adaptation. In: Kappel, G., Maamar, Z., Motahari-Nezhad, H.R. (eds.) ICSOC 2011. LNCS, vol. 7084, pp. 359–373. Springer, Heidelberg (2011)
3. Tsoumakos, D., Konstantinou, I., Boumpouka, C., Sioutas, S., Koziris, N.: Automated, Elastic Resource Provisioning for NoSQL Clusters Using TIRAMOLA. In: 2013 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), pp. 34–41. IEEE Computer Society (2013)
4. Copil, G., Moldovan, D., Truong, H.L., Dustdar, S.: SYBL: an Extensible Language for Controlling Elasticity in Cloud Applications. In: 2013 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), pp. 112–119. IEEE Computer Society (2013)
5. Copil, G., Moldovan, D., Truong, H.-L., Dustdar, S.: Multi-level Elasticity Control of Cloud Services. In: Basu, S., Pautasso, C., Zhang, L., Fu, X. (eds.) ICSOC 2013. LNCS, vol. 8274, pp. 429–436. Springer, Heidelberg (2013)