Future Generation Computer Systems 78 (2018) 142–154

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Managing cloud via Smart Cloud Engine and Knowledge Base

Pierfrancesco Bellini [a,b], Ivan Bruno [a,b], Daniele Cenni [a,b], Paolo Nesi [a,b,*]

[a] *Distributed Systems and Internet Technology Lab, DISIT lab, University of Florence, Florence, Italy*[1]

[b] *Department of Information Engineering, University of Florence, Florence, Italy*[2,3]

HIGHLIGHTS

- Knowledge based Smart cloud engine is proposed.
- Low and high level metrics for cloud service level agreement are computed.
- Flexible and scalable smart cloud engine.

ARTICLE INFO

Article history:
Received 31 March 2016
Received in revised form
10 September 2016
Accepted 6 October 2016
Available online 24 October 2016

Keywords:
Knwoledge base
Smart cloud
Cloud computing
Service level agreement

ABSTRACT

Complexity of cloud infrastructures needs models and tools for process management, configuration, scaling, elastic computing and cloud resource health control. This paper presents a Smart Cloud Engine and solution based on a Knowledge Base, KB, with the aim of modeling cloud resources, Service Level Agreements and their evolutions, and enabling the reasoning on structures by implementing strategies of efficient smart cloud management and intelligence. The solution proposed provides formal verification and intelligence tools for cloud control. It can be easily integrated with a large range of cloud configuration manager, cloud orchestrator, and monitoring tools, since the connections with these tools are performed by using REST calls and XML files. The proposed solution has been validated in the context of large ICARO Cloud project and in the cloud facility of a national cloud service provider. Some data resulting from the validation phases have been reported and are referring to the dynamic management of real ECLAP social network http://www.eclap.eu.

1. Introduction and related work

Any relevant software infrastructure is presently deployed on cloud to manage resources in an efficient manner. The resource management is becoming a relevant hot topic for solutions that need to provide high availability and quality of service. Therefore, specific solutions for cloud monitoring, analyzing and dynamically changing configurations and services in the cloud are becoming mandatory to increase resilience and reliability. To this end, the modeling and formalization of cloud resources and information are becoming more relevant to manage different aspects of a cloud at its different levels (i.e., IaaS, PaaS, SaaS), and towards specific resources: hosts, virtual machines (VMs), networks, memory, storage, processes, services, applications, etc., and their relationships and technical tools. Cloud infrastructures are becoming every year more complex to be manually managed, especially for the presence of process configuration and reconfiguration tools, for the needs of dynamic scaling and elastic computing, and for monitoring resources health. The resources to be managed are obviously related to elements on cloud such as: hosts, VM, services, storages, processes, software applications, networks, etc., and thus on their corresponding composition and Service Level Agreements (SLA). The SLA can be regarded as the contract associated

* Corresponding author. Fax: +39-055-2758570.
E-mail addresses: pierfrancesco.bellini@unifi.it (P. Bellini), ivan.bruno@unifi.it (I. Bruno), daniele.cenni@unifi.it (D. Cenni), paolo.nesi@unifi.it (P. Nesi).

1 http://www.disit.dinfo.unifi.it

2 http://www.dinfo.unifi.it

3 http://www.unifi.it

http://dx.doi.org/10.1016/j.future.2016.10.006

with a set of cloud resources and their configuration, contributing to a given service and fixing some parameters to service control. For example, a SLA signed by the customer with a Cloud Service Provider (CSP) describes its requests and requirements with respect to cloud services configuration and quality of service (e.g., 5 hosts, 4 CPUs at 5000 MHz minimum, 3 VMs in a certain configuration, minimum network speed of 50 Mbps, availability greater than the 99.99% of time). Thus, with the SLA the CSP declares the service level guaranteed and the constraints associated with the usage of cloud services. To this end, a number of metrics are defined and assessed for computing the business costs on cloud on ''as a Service'' basis, and are used to verify the SLA conformity. A review about SLAs models and types offered by commercial Cloud Providers can be obtained from Wu and Buyya, 2011 [1].

As a general consideration, the automation of cloud management may imply to cope with a number of activities also with respect to the SLA, such as: (i) formal verification and validation of cloud configurations in terms of resources, their relationships and matches (this action, can be performed before or after the resource's deployment; when it is performed before, it can be regarded as a sort of simulation with respect to the available resources); (ii) verification and reasoning about cloud security, taking into account networking and storage aspects; (iii) facilitating interoperability among public and private clouds, and/or among different cloud segments managed by different cloud orchestrators or managers in the same cloud infrastructure; (iv) discovering and brokering services and resources; (v) reasoning about cloud workload conditions, maybe via simulation; (vi) computing capability for horizontal and/or vertical scaling, thus elastic computing. In the literature, these aspects are addressed in several different manners. Some of them into SLA brokers such as in Cuomo et al., 2012 [2], and in Pengcheng et al., 2011 [3], while for the minimal monitoring you can see Ward and Barker, 2014 [4].

Therefore, reasoning tools about constraints and configurations on cloud are needed, and they have to model the infrastructure, the resources and the rules to manage all of them, with respect to the real data collected on the cloud via some monitoring solution. To this end, a data model representing the cloud complexity is mandatory.

In a seminal work of Youseff et al., 2008 [5], an approach to create a cloud ontology has been proposed, decomposing cloud modeling problems into five layers: applications, software environments, software infrastructure, software kernel, and hardware. Moreover, Zhang et al., 2012 [6] proposed a solution to make easier the searching services and resources into the cloud by presenting the CoCoOn ontology, also integrating other QoSOnt ontology (Dobson et al., 2005 [7]), for the description of service discovering and parameters. For the description at level of IaaS, the INDL (Infrastructure and Network Description Language) ontology defines nodes connected via links and interfaces, as described in Ghijsen et al., 2012 [8]. Virtual Nodes are used to model VMs in execution on former nodes. Node Components are used to represent resources as memory, storage, CPU, while many details are missing as the real network addresses, and the information describing physical aspects of VMs and Hosts. In mOSAIC EC project (Moscato et al., 2011 [9]), the cloud knowledge modeling has been addressed with the aim of creating a common model to cope with the heterogeneity of different clouds vendors, and with systems with different terminologies. The mOSAIC ontology allows describing aspects of hosts and VMs, while presenting some lacks on modeling connections. For modeling the main cloud entities some generic ontologies could be used. For example, https://geni-orca.renci.org/owl/owl-test/compute.rdf# ontology has been defined into ORCA project https://geni-orca.renci.org, which is mainly focused on modeling the hierarchical aspects of networking, while it presents limited capabilities in modeling cloud entities and applications at all cloud levels of NIST.

For the description of general cloud services, Linked-USDL (Unified Service Description Language) in Pedrinaci et al., 2014 [10] provides a set of ontological models for describing services, SLA, security, prices and intellectual property. Linked USDL is a remodeling of USDL language and reused other RDF(S) vocabularies such as: GoodRelations (http://www.heppnetz.de/projects/goodrelations/), Minimal Service Model (http://iserve.kmi.open.ac.uk/wiki/index.php/IServe_vocabulary), and FOAF (http://www.foaf-project.org/). At application level of the cloud, the standard OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA) of Binz et al., 2012 [11], and in Binz et al., 2014 [12], allows to describe via XML the application components, their dependencies, and the plan (workflow) for provisioning on the infrastructure; thus formalizing the work of the Orchestrators. In the work of Bernstein et al., 2010 [13], the ontological model is used for describing intercloud architectures adopting an ontology for describing cloud services. This approach is at the basis of the IEEE P2302 standard (https://standards.ieee.org/develop/project/2302.html) and exploits the mOSAIC model for the services and resources with the related limitations. Currently, there are a few efforts in building smart cloud solutions grounded on an ontology on cloud computing, Androcec et al., 2012 [14].

As a more lightly related work, the modeling of services via OWL-S has been proposed for describing web services with their service profile and the WSDL formalism. For the SLA of Web Services, WSLA has been proposed via an XML schema, and allows composing metrics on specific services, see Ludwig et al. 2003 [15]. In this context, WS-Agreement has been developed by ''*Grid Resource Allocation Agreement Protocol Working Group*'' (GRAAP-WG) with the aim of describing SLA among distributed entities in the grid (Andrieux et al. 2007 [16]). On the basis of the WS-Agreement, in Oldham et al., 2006 [17], an ontology has been defined. All these results have not been open to the public and are strongly focused on web services and related to the quality of service.

The work presented in this paper describes a Smart Cloud Engine (SCE) and solution based on modeling cloud resources and information via a Knowledge Base (KB). The proposed KB and tools also cope with SLAs, detailed cloud resource descriptions, and monitors information associated with resources; thus allowing the monitoring of the SLA evolution, managing complex configurations, according to the SLA and related strategies for dynamic scaling and elastic computing. In this context, the SLA is modeled and addressed as the service agreement between the CSP and the cloud customer and not at level of web service, or network service. The adoption of a KB to model the cloud knowledge grounded on a cloud ontology and data instances enables the reasoning on cloud structures and their evolutions. And thus, it is suitable for implementing strategies of smart cloud management and intelligence. Moreover, the proposed Smart Cloud Engine and solution can be exploited in connection with a large rangeof cloud management tools such as configurators, orchestrators, and monitoring tools. The solution proposed in this paper has been developed in the context of the ICARO Cloud project. It has been validated with respect to the cloud infrastructure of Computer Gross, a CSP providing cloud services at different levels (i.e., IaaS, PaaS and SaaS), in which allocated applications at SaaS level are provided by several different vendors. The tools allocated on the cloud of Computer Gross belong to categories of multitier solutions for CRM (Customer Relationship Management), ERP (Enterprise Resource Planner), workflow, marketing, business intelligence, cultural heritage, social media, etc. A large variety of solutions on cloud increases the complexity of cloud management, that is configuration and dynamic management. Moreover, these problems motivate the needs of a flexible smart cloud engine as that presented in this paper. Therefore, the validation phase of the proposed Smart Cloud Engine has been mainly focused on assessing