Future Generation Computer Systems 50 (2015) 49–61

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

An autonomic approach to manage elasticity of business processes in the Cloud

Mohamed Mohamed [a,*], Mourad Amziani [a,b], Djamel Belaïd [a], Samir Tata [a], Tarek Melliti [b]

[a] Institut Mines-Telecom, Telecom SudParis, UMR CNRS Samovar, 9 rue Charles Fourier 91011 EVRY Cedex, France
[b] University of Evry, IBISC, 23 Boulevard de France 91037 EVRY Cedex, France

HIGHLIGHTS

- We propose an autonomic management model for Cloud resources.
- We extend Open Cloud Computing Interface to describe our autonomic model.
- We illustrate our autonomic model with an approach for Business Processes elasticity.
- We propose a formal approach for Business Processes elasticity.
- We show the efficiency of our approach on a realistic Cloud environment.

ARTICLE INFO

Article history:
Received 15 May 2014
Received in revised form
1 October 2014
Accepted 8 October 2014
Available online 22 October 2014

Keywords:
Cloud computing
Autonomic computing
QoS
Service business process
Elasticity
OCCI

ABSTRACT

Cloud Computing is gaining more and more importance in the Information Technologies (IT) scope. One of the major assets of this paradigm is its economic model based on *pay-as-you-go* model. Cloud Computing gets more attention from IT users when it fits their required QoS and reduces their expenses. This task cannot be done without increasing the autonomy of the provisioned Cloud resources. In this paper, we propose a holistic approach that allows to dynamically adding autonomic management facilities to Cloud resources even if they were designed without these facilities. Based on the Open Cloud Computing Interface (OCCI) standard, we propose a generic model that allows describing the needed resources to render autonomic a given Cloud resource independently of the service level (Infrastructure, Platform or Software). Herein, we define new OCCI Resources, Links and Mixins that allow provisioning autonomic Cloud Resources. In order to illustrate our approach, we propose a use case that specializes our autonomic infrastructure to ensure the elasticity of Service-based Business Processes (SBPs). The elasticity approach that we are using is based on a formal model that features duplication/consolidation mechanisms and a generic Controller that defines and evaluates elasticity strategies. To validate our proposal, we present an end to end scenario of provisioning an elastic SBP on a public PaaS. Evaluation of our approach on a realistic situation shows its efficiency.

1. Introduction

Cloud Computing is a challenging area involving different aspects of IT (Information Technologies) services. Its adoption is increasing due its economic model based on *pay-as-you-go* model. A survey conducted by the Cloud Industry Forum [1] in 2012 involving 300 companies shows that 53% of the companies are currently adopting the Cloud. The same survey showed that 73% of them are planning to increase their adoption of Cloud services in the next 12 months. The adoption of Cloud could increase rapidly if providers prove their ability to continuously ensure the required Quality of Service (QoS). Indeed, this is a challenging task because Cloud resources are exposed to dynamic evolution during their life-cycle. This evolution is due to the environment dynamic. In order to preserve the QoS in such environment, an autonomic infrastructure should be offered by the Cloud provider. This infrastructure has to enable dynamic monitoring of resources and their adaptation with minimal cost and performance degradation.

Autonomic management [2] is the ability to manage computing resources automatically and dynamically to respond to the

* Corresponding author.
E-mail addresses: mohamed.mohamed@telecom-sudparis.eu (M. Mohamed), Mourad.Amziani@telecom-sudparis.eu (M. Amziani), Djamel.Belaid@telecom-sudparis.eu (D. Belaïd), Samir.Tata@telecom-sudparis.eu (S. Tata), tmelliti@ibisc.fr (T. Melliti).

http://dx.doi.org/10.1016/j.future.2014.10.017
0167-739X/

requirements of the business based on Service Level Agreement (SLA). Autonomic computing is usually represented by an autonomic MAPE (Monitor, Analyze, Plan and Execute) loop. This loop consists on harvesting monitoring data, analyzing them and generating reconfiguration actions to correct violations (self-healing and self-protecting) or to target a new state of the system (self-configuring and self-optimizing). We advocate that coupling Autonomic management with Cloud computing allows to maintain the QoS of Cloud resources and reduces their costs. However, this coupling remains critical and challenging since Cloud environments are composed of heterogeneous resources [3].

Cloud resources provisioning relies on SLA that includes non-functional requirements of services specified as QoS [4]. QoS is an important concept that informs users about their consumed services [5,6]. Obviously, using autonomic management in the Cloud will enhance the QoS of resources. In this paper, we will focus on the use case of the elasticity. We advocate that elasticity is a critical facility that preserves a high level of QoS. The principle of elasticity is to ensure the provisioning of necessary and sufficient resources such that a Cloud service continues running while ensuring the needed QoS even as the number or quantity of its use scales up or down, thereby avoiding under-utilization and over-utilization of resources [7].

In this paper, we propose a generic model that allows adding autonomic management facilities to any Cloud resource. This model covers all the functions defined within the autonomic MAPE loop. To support our generic model, we propose an extension for Open Cloud Computing Interface (OCCI) standard to add the needed mechanisms. Our extension entails the definition of new OCCI Entities and Mixins that could be dynamically used to establish an autonomic loop around any Cloud resource. To illustrate the usage of our approach, we propose to specialize the autonomic infrastructure to manage the elasticity of Service-based Business Process (SBPs). This kind of business processes are being increasingly deployed and executed in Cloud environments. Since the elasticity of platforms, process engines and service containers is not sufficient to ensure the elasticity of the deployed business processes [8], we advocate that these latter should be provided with their own elasticity mechanisms to allow them to dynamically adapt to the workload changes. The main challenges in this direction are to define the needed mechanisms to perform elasticity of SBPs, and a way to couple these mechanisms with a Controller in order to ensure SBPs elasticity.

The remaining part of this paper is organized as follows. In Section 2, we propose our generic infrastructure to support autonomic management of Cloud resources. Then, in Section 3, we present our extension of OCCI standard to describe autonomic management resources. The extension entails all OCCI Entities and Mixins that allow the establishment of an autonomic loop around a given Cloud resource. To give more understanding of our approach, in Section 4, we propose to specialize our infrastructure in order to support elasticity of SBPs as a specific kind of Cloud resources. We chain up with the validation of our approach in Section 5, which contains the different aspects of our implementations and some experiment results. Further, we present some related works in Section 6 before listing our conclusions and perspectives.

2. Autonomic management for Cloud resources

IBM [2] defines Autonomic Computing as the ability to manage computing resources that automatically and dynamically respond to the requirements of the business based on SLA. In our work, we are interested in Cloud environments. Such environments are composed of an increasing number of heterogeneous resources. The management of these resources is becoming more and more complex. We advocate that management tasks like monitoring, configuration, protection, optimization are not the main functional objective of most resources, but if they are not properly addressed, these resources cannot accomplish their tasks. The challenge, then, is to enable autonomic management to take control of all these non functional tasks, letting the developers focus on the main functional goals of the resource. In order to really free developers from the burden of autonomic management features on their applications, there must exists a way to dynamically add these features to Cloud resources.

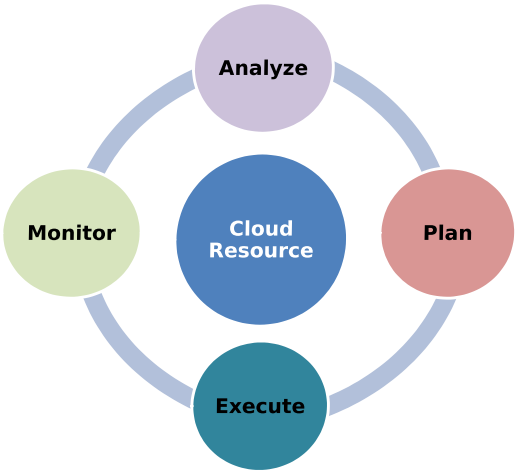

Fig. 1. Autonomic control loop for a Cloud resource.

Autonomic management is usually presented as a Monitor, Analyze, Plane and Execute (MAPE) loop. In the following we will detail each function of this loop.

2.1. MAPE: monitor, analyze, plan and execute

Autonomic management features for Cloud resources are presented in Fig. 1. In this autonomic loop, the central element represents any Cloud resource for which we want to exhibit an autonomic behavior. For this loop we need monitoring mechanisms to collect information about the managed Resource. We need also configuration mechanisms to carry out changes to the managed Resource.

The different functions of the autonomic control loop are defined as: (1) Monitor function that provides the mechanisms to collect, aggregate, filter and report monitoring data collected from a managed Resource; (2) Analyze function that provides the mechanisms that correlate and model complex situations and allow the autonomic manager to interpret the environment, interpret the current state of the system and predict future situations; (3) Plan function that provides the mechanisms that construct a plan of actions needed to achieve a certain goal, usually according to some guiding strategies; and (4) Execute function that provides the mechanisms to control the execution of the plan over the managed Resources.

2.2. MAPE for Cloud resources

The use of autonomic capabilities in conjunction with Cloud computing provides an evolutionary approach in which autonomic computing capabilities anticipate runtime resource requirements and resolve problems with minimal human intervention. To add autonomic management features for Cloud resources, we propose to dynamically add the needed resources and extensions to ensure the MAPE (Monitor, Analyze, Plan and Execute) functions. It is obvious that Monitor and Execute functions are tightly coupled with the resource itself. Consequently, we propose to dynamically