



INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume3, Issue3)

Available online at www.ijariit.com

Study of Tosca of Cloud Application with Its Services

Snehal V. Raut

Dr. Rajendra Gode Institute of Technology and Research,
Amravati
snehalraut5002@yahoo.com

Dr. H. R. Deshmukh

Dr. Rajendra Gode Institute of Technology and Research,
Amravati
hrdphd@rediffmail.com

Abstract: With the advent of Cloud Computing, organizations are increasingly migrating their information and communication technology (ICT) resources to the cloud. Cloud computing is driving formidable change in the technology industry and transforming how to do business in around the world. [4] Using TOSCA, the cloud providers are able to define the interoperable description of services and their relationships and to enable the portability and automated management across cloud platforms and infrastructures. However, the verification of the cloud orchestration design with TOSCA is still crucial to ensure and alert when the safety properties of the cloud design are violated. In this paper, formal verification of cloud orchestration design is described and also presents the different services of TOSCA.

Keywords: TOSCA-Topology and Orchestration Specification for Cloud Applications

1. INTRODUCTION

Topology and Orchestration Specification for Cloud Applications (TOSCA) [1] is a standard for cloud-based web services design and the processes to manage them. TOSCA is developed and approved by the Organization for the Advancement of Structured Information Standard (OASIS). It helps construct agility and reduce restriction with responsibility in resource management. Using the TOSCA standard, the cloud service providers are capable of describing their cloud computing infrastructure and architecture and manage the implementation of the cloud resources. There are several cloud orchestration tools providing the automation of the deployment of implementation.

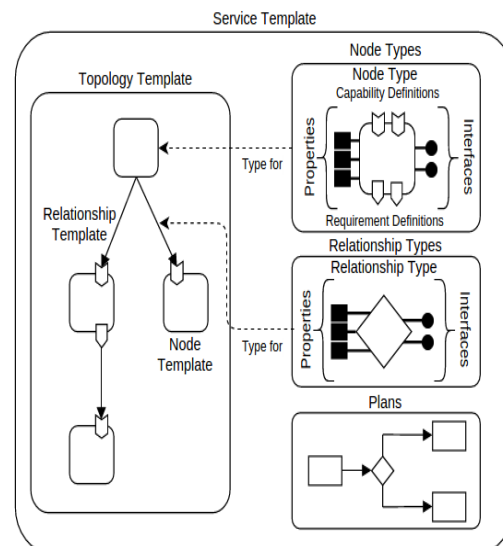


Figure 1: Structural elements of a service template and their relations [11]

2. LITERATURE REVIEW

Karn Yongsiriwit et. al. [1] proposed a semantic framework tackling this heterogeneity issue. We develop a set of ontologies to semantically represent cloud resources by looking at three cloud resource description standards: Topology and Orchestration Specification for Cloud Applications (TOSCA), Open Cloud Computing Interface (OCCI), and Cloud Infrastructure Management Interface (CIMI). Hence, our framework promotes the creation of a common semantic knowledge base of cloud resources described using these different standards. This knowledge base allows a seamless translation of cloud providers' resource descriptions. We developed an application to validate our approach as a proof of concept. We also evaluated the feasibility and completeness of our semantic framework on use cases obtained from standard specifications.

Warun Chareonsuk et. al. [2] proposed an alternative mean to do the formal verification of the cloud orchestration design by superimposing the relevant BPEL of web services over the existing TOSCA description of a cloud orchestration. The resulting formal model of the superimposition between BPEL and TOSCA defines not only the orchestration of the web services but also their service interfaces and the corresponding high-level behaviors of the services. In this paper, the safety properties of the cloud orchestration are focused only and defined using the linear temporal logic formula. Our formal model is correctly written in Promela and formally verified using model checker SPIN.

Rawaa Qasha et. al. [3] shows how TOSCA, a new standard for cloud service management, can be used to systematically specify the components and lifecycle management of scientific workflows by mapping the basic elements of a real workflow onto entities specified by TOSCA. Ultimately, this will enable workflow definitions that are portable across clouds, resulting in the greater reusability and reproducibility of workflows.

Beniamino Di Martino et. al. [4] this paper we propose an overview of two solutions for cloud services description and orchestration, TOSCA (Topology and Orchestration Specification for Cloud Applications) and HOT (Heat Orchestration Template), a comparison among this two solution and examples of how these two solutions correlate with cloud patterns.

Bisera Ivanovska et. al. [5] uses the P-TOSCA model for other issues that are also very important in virtualized data centers and cloud computing, that is, to enlarge/extend the energy efficient management system. A prototype application that dynamically creates a target virtual machine on utilized physical compute node, ports the application(s) from a virtual machine hosted on an underutilized physical server to the target virtual machine in Eucalyptus cloud is presented, which is specified with P-TOSCA. After migration, the prototype application will shut down the underutilized empty physical node.

3. TOSCA OVERVIEW

The Topology and Orchestration Specification for Cloud Applications (TOSCA) is an open-source language used to describe the relationships and dependencies between services and applications that reside on a cloud computing platform. TOSCA can describe a cloud computing service and its components and document the way those components are organized and the orchestration process needed to use or modify those components and services. This provides administrators with a common way to manage cloud applications and services so that those applications and services can be portable across different cloud vendors' platforms.

TOSCA is a highly extensible language, allowing developers to add vendor- or domain-specific mechanisms, as needed, to accommodate specific use cases. For example, a cloud provider could use TOSCA to define and compose a specific cloud service. TOSCA facilitates simpler application deployment to any cloud platform, cloud bursting and support for multicolored environments. TOSCA can also facilitate the standardization of cloud-based services, allowing cloud providers to offer ubiquitous services that users can map to their respective infrastructures.

4. CLOUD APPLICATION DEPLOYMENT WITH TOSCA

A cloud application in TOSCA consists of a number of connected nodes called a service template. The nodes can represent horizontally distributed functions, vertical stacks of software/middleware dependencies or both. Nodes are the building blocks of a TOSCA cloud model, shaping the way an application is designed and the way in which imperative scripts are linked to applications and resources. You can visualize TOSCA templates as a series of connected stacks of nodes, each representing an application element.

5. THE ROLE OF TOSCA PLANS

More complex application deployment and lifecycle management are done through TOSCA plans, which also use the interfaces specified by the nodes. TOSCA plans are workflows, or sequential steps, and are often authored in a business process language, such as BPEL. Plans must be able to handle exceptions that could arise, and be able to undo what they have already done to roll back to a prior state.

All plans have access to the full-service template, meaning they can operate across the entire cloud application deployment. They can also operate on a single application node. This allows the TOSCA cloud standard to support both local lifecycle processes, such as restarting an application in a VM or container, and global processes, such as spinning up a new set of components for load balancing.

Plans use node data to commit resources, make connections, install components and parameterize them. They also use the exposed management interfaces to control resource elements. If nodal template information and interfaces are standardized across the implementation, then TOSCA service templates are portable throughout the entire cloud environment.

6. SERVICES OF TOSCA

A. TOSCA2Chef runtime environment

The core entity of the proposed architecture is the TOSCA2Chef runtime environment: a set of components and services that parse a TOSCA document extracts the information regarding the described application and triggers the execution of the necessary command towards the Knife client. The TOSCA2Chef execution environment incorporates two basic operations: (a) the parsing of the TOSCA document in order to perform the deployment of a given topology, and (b) the execution of the deployment (or unemployment) TOSCA plans. The TOSCA plans are defined as process models, i.e. a workflow of one or more steps. TOSCA specification relies on existing languages like BPMN or BPEL in order to capture such plans. Therefore, in our experimentation, the plan execution operation is realized using the Apache ODE BPEL engine [REF]. On the other hand, the parsing of the TOSCA document and execution of the necessary commands towards the test-bed is being managed through the TOSCA Container web service.

B. Modeling the use case with TOSCA

For the realization of this proof of concept project, a specific multi-tier application service topology had to be designed as the use case through which we would evaluate the related technologies. The selected application is a basic 3-tier web application, comprising a Load Balancer component implemented through HAProxy open source software package [23] and two application servers implemented through Apache Tomcat [24], hosting a Demo Web Application for demonstrating the operation. On the data layer, we introduced a database server through a MySQL implementation that keeps some dummy data for the demonstration.

C. Use case un-/deployment process

After the long lasting state-of-the-art analysis, design phase, and development of the orchestration framework, this PoC project dealt with the evaluation of the implemented use case scenario. During that phase two business processes were defined within the TOSCA document, responsible for the deployment and un-deployment actions. The processes were defined in BPEL XML notation within the plans section of the TOSCA document. Both BPEL processes could only be modeled with the knowledge of the TOSCA2Chef web service interfaces. Either BPEL process interacts with the web service to deploy or un-deploy nodes from the use case application topology.

CONCLUSION

In this paper, formal verification of cloud orchestration design is described and also presents the different services of TOSCA. Cloud Computing environment, ways to define machine-readable standards for the description of Cloud services and their orchestration are highly desired features. The capability of application packaging which enables the reusability and portability of applications or part of them is an important prerequisite to obtaining full benefits from Cloud Computing. With the cloud, resource pools can use virtualization to abstract away the heterogeneity and regional distribution of manufacturing resources.

REFERENCES

- [1] Karn Yongsiriwit, Mohamed Sellami, Walid Gaaloul, "A Semantic Framework Supporting Cloud Resource Descriptions Interoperability", IEEE 9th International Conference on Cloud Computing, 2016.
- [2] Warun Chareonsuk, Wiwat Vatanawood, "Formal Verification of Cloud Orchestration Design with TOSCA and BPEL", IEEE, 978-1-4673-9749-0/16/\$31.00, 2016.
- [3] Rawaa Qasha, Jacek Cala, Paul Watson, "Towards Automated Workflow Deployment in the Cloud using TOSCA", IEEE, 8th International Conference on Cloud Computing, 2015.
- [4] Beniamino Di Martino, Giuseppina Cretella, Antonio Esposito, "Defining Cloud Services Workflow: a Comparison between TOSCA and OpenStack Hot", IEEE, Ninth International Conference on Complex, Intelligent, and Software Intensive Systems, 2015.
- [5] Bisera Ivanovska, Sasko Ristov, Magdalena Kostoska, Marjan Gusev, "Using the P-TOSCA Model for Energy Efficient Cloud", IEEE, MIPRO, Opatija, Croatia, 25-29 May 2015.
- [6] Florian Haupt, Frank Leymann, Alexander Nowak, Sebastian Wagner, "Lego4TOSCA: Composable Building Blocks for Cloud Applications", IEEE, International Conference on Cloud Computing, 2014.
- [7] Christos Tsigkanos, Timo Kehrer, "On Formalizing and Identifying Patterns in Cloud Workload Specifications", 13th Working IEEE/IFIP Conference on Software Architecture, 2016.
- [8] Soheil Qanbari, Fei Li, and Schahram Dust, "Toward Portable Cloud Manufacturing Services", IEEE Computer Society, November/DECEMBER 2014.
- [9] Abdul Razaq, Huaglory Tianfield, Peter Barrie, Hong Yue, "Service Broker Based on Cloud Service Description The language", IEEE, 15th International Symposium on Parallel and Distributed Computing, 2016.

- [10] Alexandru-Florian Antonescu, Philip Robinson, Torsten Braun, "Dynamic Topology Orchestration for Distributed Cloud-Based Applications", IEEE, Second Symposium on Network Cloud Computing and Applications, 2012
- [11] Shin Nakajima, Lightweight formal analysis of Web service flows. In Progress in Informatics Journal, 2005