TITLE:

Mixing Multi-tenancy and Elasticity of Business Processes in Cloud Environments

SUPERVISORS:

Tarek MELLITI, UEVE/IBISC/COSMO
Samir TATA**,** TSP/Samovar/ACMES

DOCTORANT:

MOTS CLÉS:

Cloud Computing, Business process management, Elasticity, Multi-tenancy

CONTEXT & PROBLEM STATEMENT:

Cloud computing is now recognized as an effective paradigm for developing and delivering of services over the Internet. As it has been the case with other technologies, the availability of Business Process Management (BPM) in the cloud allows imagining new usage scenarios (in smart cities, administrations, agencies, etc). Nevertheless, the entry cost of effective Business Process Management is too high and too complex even if it provides quick return on investment. Therefore, business processes should be provided with elasticity and multi-tenancy mechanisms allowing adapting to the workload changes while ensuring the desired functional and non-functional properties and taking into account preferences of tenants (holders). The existing approaches for mutli-tenancy provide a very low level degree of resource sharing or require high development complexity. In addition, approaches for elasticity are intrusive (require changes in existing application servers) or do not support predictive and reactive elasticity strategies at the same time.

OBJECTIVES:

In this thesis proposal, we aim at using configurable business processes to support multi-tenancy and elasticity mechanisms in an integrated way. Actually, we propose to use configurable languages not only for modeling purpose, as it is the case in the state of the art, but also for supporting multi-tenancy and elasticity at the runtime. To this end, we will (1) define an operational semantics for a particular configurable language (C-WF-nets) to support multi-tenancy, (2) provide business processes with elasticity mechanisms using a formal model to define and verify elasticity and their strategies and (3) develop control mechanisms to decide about when and where elasticity is applied while taking into account the multi-tenanacy property.

DETAILED DESCRIPTION

1 Context

Cloud computing is now recognized as an effective paradigm for developing and delivering of services over the Internet. It typically involves provisioning of dynamically scalable and often virtualized resources. It delivers services at the levels of infrastructure, platform and software.

As it has been the case with other technologies, the availability of Business Process Management (BPM) in the cloud allows imagining new usage scenarios. Typically, these scenarios include the execution of thousands of processes (set of coordinated activities) during a very short period of time requiring temporarily a very important amount of resources and targeting different preferences of tenants (holders).

Further than getting new usage scenarios, Cloud users can take benefit from cloud infrastructures to execute innovative solution in scenario like political campaign management or crisis management where the need for quick adaptation is essential but at the cost of very sophisticated development.

In fact, the entry cost of effective applications, services or Business Process Management (BPM) for SMEs is too high and too complex even if it provides quick return on investment. Novel and innovative approaches for modeling, deploying and enactment of business processes should be developed to allow supporting the scenario cited above and others in a safer and cost effective way.

2 Objective

Business processes should be provided with elasticity and multi-tenancy mechanisms allowing adapting to the workload changes while ensuring the desired functional and non-functional properties and different preferences of tenants (holders). Targeting this objective mainly raises the following questions:

- How to formally characterize and model the elasticity and multi-tenancy properties?
- How to formally describe and enforce strategies for elasticity and multi-tenancy?
- How to handle the interaction between the elasticity and multi-tenancy in a consistent and non intrusive way?

3 Background and State of the art

3.1 Multi-tenancy and configurable business processes

Cloud computing is particularly interesting in situations where many organizations need to support similar processes. For example, smart cities, administrations, agencies, etc. all need to support similar processes but they also need local and controlled variations of these similar processes. Therefore, cloud platforms should provision mechanisms such that business processes can be customized to tenants while allowing them to share resources. These are called multi-tenancy mechanisms.

There are mainly three approaches developed for multi-tenancy in cloud environments[8] . These approaches require varying the degree of resource sharing and the development complexity:

Approach 1. sharing a single business process instance among all tenants,
Approach 2. running tenant specific process instances on a shared process engine,
Approach 3. running tenant specific process instances on a dedicated process engine.

Approaches 2 and 3 are based on deployment techniques. **They provide a very low level degree of resource sharing**. In approach 1, all tenants share the OS, the process engine and a single instance of the business process. This is accomplished by parameterizing a single instance of a business process with a tenant identification parameter. However, **the drawback of this approach is its high development complexity which could prevent democratization of such technique.**

3.2 Elasticity of business processes

The principle of elasticity is to ensure the provisioning of necessary and sufficient resources in such a way that a cloud service continues running smoothly even as the number or quantity of its use scales up or down. Thus, the elasticity property allows to avoid under-utilization and over-utilization of resources [5] [10] .

Provisioning of resources can be made using two approaches: vertical and horizontal elasticity [6] . Vertical elasticity increases or decreases the resources of a specific cloud service. Actually, vertical elasticity requires extending process engines. We have established that the existing service containers are not elastic and we have developed a new elastic engine [2] . **It should be noted that vertical elasticity mechanism are rather intrusive**.

The horizontal elasticity replicates or removes instances of cloud services [7] . On one hand, to scale up a business process, the elasticity mechanisms have to create, as many copies as necessary, of over-loaded business services (part of the considered business process). On the other hand, to scale down a business process, the elasticity mechanisms have to remove unnecessary copies of under-loaded services. While, the development of vertical elasticity is feasible, the development of horizontal elasticity for business processes is challenging. In fact, performing horizontal elasticity operations should follow an elasticity strategy that uses load information of each business service, *e.g.,* in terms of number of current invocations, to make elasticity decisions (when and how to add/remove copies of services). Some of these strategies are reactive (based on minimal and maximal thresholds) and some others are predictive, *e.g.,* based on prediction of invocations. In [3] , Galante and de Bona presents a survey of cloud computing elasticity. **None of the 28 studied approaches can handle horizontal elasticity of business processes (applications) while considering both predictive and reactive strategies**.

4 Approach

We aim in this project at developing models and mechanisms to support multi-tenant business processes provided with elasticity facilities. To do so, we propose an approach for:

1) providing a formal model for horizontal elasticity of processes which allow predictive and reactive strategies,
2) adapting configurability to support multi-tenancy of business processes which does not need high development complexity,
3) mixing elasticity and multi-tenancy in a consistent and non intrusive way.

4.1 Elasticity of configurable business processes

We have conducted a primary investigation of two elasticity approaches for business processes. The first approach investigated consists in developing a general controller to enforce elasticity of business processes [1] modeled in Petri nets (note that several process modeling languages such as BPEL and BPMN can be mapped to Petri nets). This approach does not request any effort from the designer. Nevertheless, we realized that it suffers from the problem of single point of failure, *i.e.*, the central controller. The second approach investigated [9] consists in composing a model of a business process with models of mechanisms for elasticity making the whole business process elastic. According to this approach there should be as many controllers as elastic services. These services should be identified by the designer who is aware about business and technical characteristics of business process and consequently s/he will be able to make such identification.

We are now convinced that this second approach is more suitable for elasticity of business processes. We plan to develop this approach using a formal model (*i.e.,* Petri net) in order to formally define and verify/validate both elasticity and strategies. The idea consists in developing a parameterized Petri net model to control elasticity of one business service. When the generic controller of one service is composed with the model of a given business process, the strategy (predictive or reactive) and the elasticity mechanisms are defined. Reactive strategies will be described based on the marking of places (pre-condition) of elastic services while the predictive strategies will be described based on the marking of places of previous services (wherever they are elastic or not). Not that this later information can be retrieved from the structure of the Petri net.

4.2 Using configurable business processes for multi-tenancy

Multi-tenancy allows variability across organizations. In fact, it is not realistic to enforce "one size fits all" as tenants may have different needs and preferences. In this proposal, we focus on the process perspective and suggest using the so-called configurable process models to support variability. A configurable process model represents a family of process models, that is, a model that can be customized, for a particular setting, by configuration. It contains a core part that is common or shared and several variant parts dedicated to the different variations. Configuration is used to select a desired behavior and is achieved by hiding (*i.e.*, bypassing) or blocking (*i.e.*, inhibiting) some fragments of the configurable process model.

Various existing process languages have been extended to allow configurability (*e.g.,* C-EPCs, C-iEPCs, C-WF-nets, C-SAP, C-BPEL). However, only few of them are actually supported by modeling software. In addition, **they are only used at the design time.**

Here, we propose to use configurable languages **not only for modeling purpose but also for supporting multi-tenancy at runtime.** To this end, we will define operational semantics for C-WF-nets (extension of Petri nets). This choice will help in mixing elasticity and multi-tenancy mechanisms. Using Petri nets for describing multi-tenancy and elasticity will help us in formally describing these properties and verifying them separated or mixed[1].

4.3 Integration of elasticity and multi-tenancy

Figure 1 illustrates our vision of using configurable process models to handle multi-tenancy and elasticity of business processes in the Cloud.

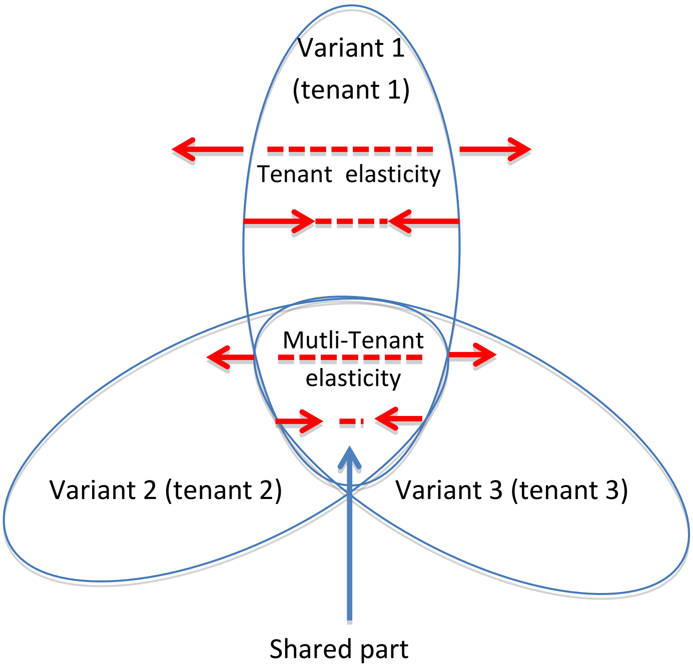

Figure 1. Configurable business process

When applying elasticity mechanisms, we will develop control mechanisms to decide about when and where the elasticity mechanisms are applied while multi-tenancy is enforced. This can involve several cases:

[1] Note that we have a solid background in using Petri nets for the description of the verification of service-oriented application and business processes.

1. The scope of elasticity covers the whole set or a sub-set of all tenants (see for example scale-up and scale-down operations called in Figure 1). We call such operation a multi-tenant elasticity.
2. The scope of elasticity covers one specific tenant (see for example scale-up and scale-down operations in Figure 1). We call such operation tenant elasticity.

Identifying these cases for a given business process will identify parts (or services) that are elastic, multi-tenant, elastic and multi-tenant or mono-tenant and non elastic.

Regarding the running of elastic and multi-tenant business processes, we plan to develop two solutions:

1) The first solution will consists in defining of a set of rules in order to transform elastic and multitenant business process in a set of connected business services which are elastic, multi-tenant, elastic and multi-tenant or mono-tenant and non elastic. Each service will be deployed on any application server (*e.g.,* our developed micro-container [2]). This approach is not intrusive since it does not require any change in application servers.
2) The second approach is intrusive. It will be developed in collaboration with Prof. Schahram Dustdar (Distributed Systems Group, Technical University of Vienna)[2]. It will consist in extending a process engine, Vienna Platform for Elastic Processes, to support the enactment of configurable business processes.

A comparative study on these two approaches will be conducted at the end of this work. This study will consider, among others, the usability and performance management.

5 References

[1] M. Amziani, T. Melliti, and S. Tata. A generic framework for service-based business process elasticity in the cloud. In BPM, volume 7481 of LNCS. Springer, 2012.

[2] S. Yangui, M. Mohamed, S. Tata, and S. Moalla. Scalable service containers. In IEEE CloudCom, 2011.

[3] G. Galante and L. C. E. de Bona. A survey on cloud computing elasticity. In IEEE/ACM International Conferenceon Utility and Cloud Computing, UCC ’12, 2012.

[4] L.-S. Lê, H.-L. Truong, A. Ghose, and S. Dustdar. On elasticity and constrainedness of business services provisioning. In IEEE International Conference on Services Computing, 2012.

[5] J. Geelan, M. Klems, R. Cohen, J. Kaplan, D. Gourlay, P. Gaw, D. Edwards, B. de Haaff, B. Kepes, K. Sheynkman, O. Sultan, K. Hartig, J. Pritzker, T. Doerksen, T. von Eicken, P. Wallis, M. Sheehan, D. Dodge, A. Ricadela, B. Martin, B. Kepes, and I. W. Berger. Twenty-One Experts Define Cloud Computing. http://cloudcomputing.sys-con.com/node/612375

[6] D. Sanderson. Programming Google App Engine: Build and Run Scalable Web Apps on Google’s Infrastructure. O’Reilly Media, Inc., 1st edition, 2009.

[7] P. Kranas, V. Anagnostopoulos, A. Menychtas, and T. A. Varvarigou. ElaaS: An Innovative Elasticity as a Service Framework for Dynamic Management across the Cloud Stack Layers. In CISIS, pages 1042–1049. IEEE, 2012.

[8] C. Osipov G. Goldszmidt M. Taylor and I. Poddar. Develop and Deploy Multitenant Web-Delivered Solutions Using IBM Middleware, Part 1: Challenges and Architectural Patterns. White Paper, IBM, 2009.

[9] K.Klai and S. Tata. Formal Modeling of Elastic Service-based Business Processes. 0th International Conference on Services Computing (IEEE SCC 2013, Application & Experience Track), IEEE Computer Society, June 27-July 2, 2013, Santa Clara Marriott, CA, USA

[2] Note that we have already a strong collaboration with Prof. Schahram Dustdar (several common published papers) on business process management but this work will enlarge this collaboration to the cloud computing field.

[10] M. Amziani, T. Melliti and S. Tata Formal Modeling and Evaluation of Stateful Service-based Business Process Elasticity in the Cloud, The 21st International Conference on Cooperative Information Systems (OTM/CoopIS 2013), September 9-13, 2013, Graz, Austria