Challenges for Automatic Multi-Cloud Configuration

Gustavo Sousa, Walter Rudametkin, Laurence Duchien

Challenges for Automatic Multi-Cloud Configuration

Gustavo Sousa, Walter Rudametkin, Laurence Duchien

University Lille 1, France
LIFL UMR CNRS 8022
Inria Lille - Nord Europe
{gustavo.sousa, walter.rudametkin, laurence.duchien}@inria.fr

ABSTRACT. *Multi-cloud computing enables customers to exploit benefits of different cloud providers to optimize reliability, performance and costs. Meanwhile, using multiple cloud providers reduces the risk of vendor lock-in as customers reduce their reliance on provider specific features. However, the large number of available cloud provider offerings and the differences among them makes it very complex to choose the best combination of cloud providers to deploy an application. Feature models from Software Product Line Engineering have been used to describe variability in cloud provider offerings and automatically generate valid cloud configurations. In this paper we explore the challenges that must be faced to extend the use of feature models to automatically configure multi-cloud environments.*

RÉSUMÉ. *La multitude des offres de nuages permet aux clients d'exploiter les avantages de chaque fournisseur pour optimiser la fiabilité, la performance et les coûts des logiciels déployés. En même temps, l'usage de fournisseurs multiples de nuages réduit le risque d'être dépendant des caractéristiques spécifiques d'un fournisseur. Néanmoins, le grand nombre d'offres de fournisseurs de nuages, et leurs différences, rendent très difficile le choix d'une combinaison optimale de fournisseurs pour deployer une application. Les modèles de caractéristiques issus de l'ingénierie des lignes de produits logiciels ont déjà été utilisés pour décrire la variabilité parmi les offres des fournisseurs de nuage et pour générer automatiquement des configurations valides. Dans cet article, nous explorons les défis qui doivent être abordés pour étendre cette approche en vue de configurer automatiquement des environements de type multi-nuages.*

KEYWORDS: *cloud computing, multi-cloud, software product lines, feature models, domain-specific modeling languages, model-driven engineering*

MOTS-CLÉS : *informatique dans les nuages, multi-nuages, lignes de produits logiciels, modèles de caractéristiques, langages de modélisation spécifique au domaine, ingénierie dirigée par les modèles*

1. Introduction

Multi-cloud computing has the potential to allow customers to exploit the specific advantages of each cloud provider, while reducing their reliance on specific vendors. When choosing an appropriate multi-cloud configuration, customers need to consider multiple factors simultaneously such as available features, pricing policy, data center location, availability, etc. Finding a balance between these often conflicting factors is a very complex task. Besides that, there is still the matter of individually configuring each selected cloud provider. As each cloud provider uses different concepts and management interfaces, substantial effort may be required to properly set them up.

Several research projects and commercial tools have been proposed to automate cloud computing management tasks. In the European projects PaaSage [JEF 13], mOSAIC [DIM 11] and MODACLOUDS [ARD 12] there is substantial work on the design, deployment and management of multi-cloud environments. However, there are still no tools that enable developers to automatically find an optimal combination of cloud providers and set them up to support application requirements.

Recent work on software product lines propose using feature models and ontologies to automate cloud provider selection and configuration for single cloud configurations [QUI 14]. In this paper, we identify challenges in order to use software product line principles to automatically select and configure multi-cloud environments.

2. Motivation

To better understand the challenges that arise when configuring multi-cloud computing environments, lets consider an example e-commerce web application. As illustrated in Figure 1, our example application is composed of six different services: four application services and two databases.

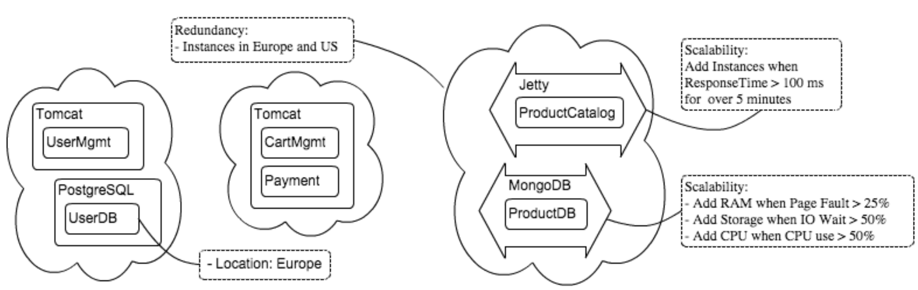

Figure 1. *Example multi-cloud e-commerce web application.*

`UserMgmt` deals with user account management and accesses data stored in the relational database `UserDB`. Meanwhile, `ProductCatalog` allows users to browse products, accessing data from the NoSQL database `ProductDB`. Due to tight communication requirements between the services and their respective databases, developers want to assure that they are placed within the same cloud to reduce latency and communication costs. In addition, `UserDB` must be located in Europe to comply with local regulations. `ProductCatalog` should have redundant instances in North America and Europe to reduce latency to end users and increase reliability. `ProductCatalog` and `ProductDB` also need to automatically scale to handle changes in workload.

From this simple example scenario we notice that setting up a multi-cloud environment can be difficult. As more restrictions and requirements are included, it becomes more difficult to manually configure the environment. To tame this complexity, we need an automated approach that enables developers to configure a multi-cloud environment in a provider-agnostic way. To achieve this vision, we have identified the following challenges:

$\mathcal{C}_1$: *Describing multi-cloud configuration requirements and restrictions.* The challenge is to define a language that allows developers to express their multi-cloud configuration in a provider-independent way. This language should be high-level to shield developers from provider-specific configuration details but still have enough expressive power to describe all relevant multi-cloud requirements. Specifically, it should enable developers to express requirements such as scalability, redundancy and location requirements.

$\mathcal{C}_2$: *Finding an optimal combination of cloud providers that meet requirements.* The challenge is to automatically find a set of cloud providers that supports the multi-cloud configuration while considering concerns such as cost and performance. We consider that configuration variability in a given cloud provider is described by means of an extended feature model [QUI 13]. However, we still need a method that takes into account multiple feature models, each one from a different provider, and evaluate them against requirements to identify which features should be selected in each selected provider to obtain an optimal multi-cloud configuration.

$\mathcal{C}_3$: *Automatically configuring selected providers.* This challenge is about handling cloud heterogeneity, including the configuration of each of the selected cloud providers according to the requirements for the *to-be-deployed* services. This includes abstracting differences between cloud provider APIs, automatically provisioning resources and deploying required software packages.

3. Proposal

To face the challenges identified in the previous section, we propose a model-driven approach which relies on a domain-specific modeling language, feature models and model transformation. A multi-cloud configuration is defined by means of a high-level model that specifies software and hardware requirements for services, how they are distributed, as well as their scalability, redundancy and location requirements.

From this model we extract the application's required features and match them against feature models that describe variability in cloud provider offerings. As a result of this process we obtain a suitable combination of cloud providers that, together with the multi-cloud configuration model, is used to generate individual cloud-specific configuration models. These configuration models are transformed into provider-specific artifacts that provision resources, setup auto-scaling mechanisms and deploy required software packages. The following subsections describe in more detail how we plan to realize this approach.

3.1. *Multi-Cloud Configuration Model*

As discussed before, setting up a multi-cloud environment without tool support is an error-prone task that is subject to many accidental complexities. To ease this task we propose the use of high-level multi-cloud configuration models that help cloud developers to design their multi-cloud configuration while abstracting low-level and provider-specific details.

We envision a modeling language that embodies knowledge from cloud computing patterns [FEH 14], providing the means to reuse proven design solutions using familiar terminology for cloud practitioners. This language should allow cloud developers to describe their service requirements in terms of software and hardware requirements, as well as the following attributes:

Scalability. Scalability can be specified in relation to application servers, databases and infrastructure. Different scalability strategies based on predictive or reactive algorithms may be chosen. According to the chosen strategy, different monitoring features and metrics may be required from providers.

Redundancy. Redundancy can be used to increase the availability of services by having multiple instances of a given service, possibly distributed among multiple cloud providers, reducing the chance of service unavailability due to software faults or cloud outages. Redundancy can be specified within a given cloud provider, across multiple cloud providers and relative to geographical regions.

Location. Organizational policies and government regulations may require data to be located in a given country or geographical region. In addition, organizations may want to keep some sensitive services in their private cloud while other services can be deployed in data centers that are cheaper or closer to end-users. Location constraints can be associated to services, specifying the country, geographical area or data center where the service will be deployed. These constraints also influence the choice of cloud providers by excluding those without data centers in the desired region.

3.2. *Feature Models for Multi-Cloud Configurations*

To find multi-cloud configurations we employ SALOON [QUI 14], a tool that enables developers to verify which cloud providers satisfy a set of requirements. SALOON relies on feature models to describe variability in cloud provider offerings and ontologies to map requirements to cloud feature models. We propose to extend SALOON to support multi-cloud requirements. To do so we need to investigate how feature models from different providers can be consistently combined into multi-cloud feature models and how to evaluate them against a set of required features from the application. As cloud feature models involve cardinalities, attributes and functional constraints [QUI 13], we also need to evaluate how evolution in individual cloud feature models affects multi-cloud feature models.

3.3. *Automatic Cloud Configuration*

Once the cloud providers have been selected, we transform the multi-cloud configuration model into many cloud-specific configuration models. Each cloud-specific model will be used to provision the required resources and deploy the selected features in a given cloud provider.

In this paper we have identified challenges to automatically configure multi-cloud environments, and we have proposed possible directions to move forward. We intend on investigating how to extend feature models tooling to calculate multi-cloud configurations while considering scalability, redundancy and location of services.

4. References

[ARD 12] ARDAGNA D., DI NITTO E., MOHAGHEGHI P., MOSSER S., BALLAGNY C., D'ANDRIA F., CASALE G., MATTHEWS P., NECHIFOR C.-S., PETCU D. et al., "Modaclouds: A model-driven approach for the design and execution of applications on multiple clouds", *Modeling in Soft. Engin. (MISE), 2012 ICSE Workshop*, IEEE, 2012, p. 50–56.

[DIM 11] DI MARTINO B., PETCU D., COSSU R., GONCALVES P., MÁHR T., LOICHATE M., "Building a mosaic of clouds", *Euro-Par 2010 Parallel Processing Workshops*, Springer, 2011, p. 571–578.

[FEH 14] FEHLING C., LEYMANN F., RETTER R., SCHUPECK W., ARBITTER P., *Cloud Computing Patterns: Fundamentals to Design, Build, and Manage Cloud Applications*, Springer Publishing Company, Incorporated, 2014.

[JEF 13] JEFFERY K., HORN G., SCHUBERT L., "A Vision for Better Cloud Applications", *Proceedings of the 2013 International Workshop on Multi-cloud Applications and Federated Clouds*, MultiCloud '13, New York, NY, USA, 2013, ACM, p. 7–12.

[QUI 13] QUINTON C., ROMERO D., DUCHIEN L., "Cardinality-based feature models with constraints: a pragmatic approach", *Proceedings of the 17th International Software Product Line Conference*, ACM, 2013, p. 162–166.

[QUI 14] QUINTON C., ROMERO D., DUCHIEN L. et al., "Automated Selection and Configuration of Cloud Environments Using Software Product Lines Principles", *IEEE CLOUD 2014*, 2014.