

MDE Opportunities in Multi-Tenant Cloud Applications

Mohammad Abu Matar¹ and Jon Whittle²

¹ Etisalat British Telecom Innovation Center
Khalifa University of Science, Technology and Research
Abu Dhabi, United Arab Emirates
mohammad.abu-matar@kustar.ac.ae

² School of Computing and Communications
InfoLab21, Lancaster University
Lancaster LA1 4WA United Kingdom
j.n.whittle@lancaster.ac.uk

Abstract. Cloud computing promotes economies of scale by sharing software and hardware resources across multiple tenants. To date, there has been relatively little research on how MDE can best support multi-tenant cloud applications, where there is a need to separate the logic and data of multiple tenants. In this position paper, we sketch out five key research opportunities for applying MDE to multi-tenant cloud applications.

1 Introduction

Cloud computing is a widespread model for sharing computing resources that promotes economies of scale by hosting software applications on a network of remote servers shared across multiple customers [1]. Generally speaking, there are two models of cloud computing: single-tenant and multi-tenant. In the single-tenant model, each customer buys a separate instance of a software application which runs on a logically isolated hardware environment. In the multi-tenant model, all customers use the same instance of the software and hardware infrastructure. In this case, the cloud provider must take care only to reveal part of an application appropriate to each customer. Whilst the single-tenant model is simpler both technically and conceptually, the multi-tenant model is preferred because it allows cloud providers to minimize resource requirements as the number of customers increases. By maintaining a single instance for multiple tenants, the provider can significantly reduce the costs of hardware provision, software licenses and software maintenance.

As has been noted elsewhere (e.g., [3]), cloud computing is an appealing application area for model-driven engineering (MDE). Cloud computing and MDE can be related in two ways [3]: (1) MDE for the Cloud, where MDE technologies are used to develop cloud applications; and (2) MDE in the Cloud, where the cloud is used to offer modeling technologies as a service (also referred to as Modeling as a Service or MaaS [3]). Multi-tenant applications bring additional challenges when compared to single-tenant applications: although a multi-tenant approach is financially advantageous for a cloud provider, maintaining separation of logic and data from different

clients is complex. In this paper, we argue that this complexity of multi-tenant applications makes them a good target for MDE.

This position paper presents five key research opportunities for applying ‘MDE for the Cloud’ specifically in the multi-tenant case: (1) MDE as a way to deal with “everywhere variability”; (2) Runtime modification of multi-tenant cloud applications; (3) A Domain-Specific Language (DSL) for the cloud; (4) MDE to support the economics of the cloud; and (5) MDE for enabling new business opportunities in the cloud.

2 “Everywhere Variability” in Multi-tenant Cloud Applications

Multi-tenant cloud applications exhibit a huge number of variation points at many different levels. As one example, data segmentation between tenants can be implemented in different ways: a dedicated database for each tenant, a single database for all tenants but with a separate schema per tenant, or a single database and schema/tables for all tenants [6]. As another example, each tenant may demand a customized business process workflow.

Indeed, the need for multiple tenants to share resources introduces potential variabilities at all levels of the cloud computing stack – that is, at the so-called Software, Platform, and Infrastructure-as-a-Service levels [1]. Different tenants may be offered a different hardware configuration, may have access to different APIs from the cloud provider, and will demand different configurations of an application. The “pay as you go” model of cloud computing actively encourages tenants to pay only for what they need; but this necessitates the maintenance of potentially many thousands of different configurations sharing the same underlying resources.

We categorize the different levels of variability in cloud systems as follows: (1) *Application Variability* – Software as a Service (SaaS) tenants have varying functional requirements [2]; (2) *Data Variability*; (3) *Business Process Variability* where the business workflow may vary; (4) *Platform Variability* – Operating systems, Programming languages, Frameworks, and solution stacks, i.e. Platform as a Service (PaaS); (5) *Provisioning Variability* – Hardware, Networking, Virtual Machines, and Elasticity, i.e. Infrastructure as a Service (IaaS); (6) *Deployment Variability* – Public, Private, Community, and Hybrid clouds [1]; (7) *Provider Variability*, e.g., Amazon Web Services (AWS), Google Application Engine (GAE), or Salesforce.

One of the potential strengths of MDE is the ability to abstract and manage variability; this has been demonstrated in many papers that integrate MDE and software product lines (SPLs) [8,11]. Hence, a combination of MDE and SPLs is an obvious potential solution. Schmid and Rummel [10] discussed how to exploit software product line (SPL) techniques for runtime customization of cloud-based systems. Similarly, a case for applying SPLs to cloud-based development is made in [11] where feature models are adapted to cater for specific cloud computing concerns like provisioning, customization, and price calculation. Schroeter et al. [12] propose to extend the component model (CCM) with tenant configuration and constraints. However, many of these works are at an early stage and there remain many challenges in applying MDE/SPLs to variability in cloud-based systems. A key point to note is that variabil-

ity management in the cloud is fundamentally different even than dynamic SPL approaches [10]: dynamic SPLs adapt a configuration at runtime but there is only a single instance of the product family running at any given time; in cloud-based systems, there will be multiple variations running, one for each tenant.

3 Runtime Modification of Multi-tenant Cloud Applications

A cornerstone of cloud computing is dynamic provisioning and resource allocation to achieve the desired performance and reliability as systems scale. This is especially true of multi-tenant applications where, for example, data access has to be carefully managed so that one tenant's requests do not dominate over others. In the worst case, a cloud application may have to be re-architected at runtime to, for example, move a greedy tenant to a separate server. In current practice, these runtime reconfigurations are handled by dashboards that allow performance monitoring and APIs that allow reconfigurations. However, MDE potentially supports more sophisticated dynamic reconfigurations resulting in radical architectural changes: these might be needed, for example, if an initial estimate on the number of tenants turns out to be wildly off, or if the security requirements of new tenants differ radically from the initial set of tenants.

There have been some attempts to manage dynamic cloud reconfigurations using advanced software development techniques. Jegadeesan and Balasubramanian [4] employ aspect-oriented programming to design service variations for multi-tenant SaaS systems. Almorsy et al. [5] apply MDE to generate security aspects for different tenants and then inject them into tenants' code. Abu-Matar et al. [8] show that MDE can play a central role in the runtime modification of multi-tenant systems by using a shared feature model to manage different tenant requirements at runtime.

Clearly, however, more research is needed in this area. For example, a model based runtime environment could be developed to manipulate tenant configurations by having a representation of the tenants' models at runtime. Another opportunity could be the development of a model-driven platform-as-a-service (PaaS) where multitenant applications can be deployed automatically.

4 Domain-Specific Languages (DSLs) for the Cloud

Multi-tenant cloud applications are specialist applications with their own set of concerns, such as partitioning for different tenants, extensibility to support new tenants, provisioning, testability of a single code base used by multiple tenants, and customization of a single code base for multiple tenants with different requirements [7]. In essence, these concerns define a domain of interest, which could be encapsulated into a DSL for generating and/or maintaining cloud implementations.

There has been some initial work in this direction. For example, CloudML [9] is a DSL to model the provisioning of multi-cloud applications, that is, applications that could run on multiple cloud providers. CloudML models the provisioning of these applications on IaaS clouds and it provides runtime support to deploy the modeled apps. Additionally, the DSL has an associated IDE, namely MODACloudML [9].

However, CloudML does not provide support for multi-tenant cloud software applications, nor it does provide support for tenants' applications evolution at runtime.

A DSL for multitenant cloud applications would include concepts like: Tenant, Tenant Configuration, Cloud Provider, Tenant Database, Tenant Schema, Tenant Table, and Deployment Type. Our previous work on SPL support for cloud architectures included a service oriented cloud meta-model that incorporates some of these concepts [8]. Transformation rules could then be used to model the evolution of multitenant applications like: Tenant Onboarding and Tenant Customization.

5 MDE to Support the Economics of the Cloud

Service providers using the cloud must think carefully about how they charge their customers. Since cloud applications typically run on a "pay as you go" model, service providers must decide if their customers will be charged on a subscription basis or per transaction. In both cases, the service provider must reconcile these charges with the amounts they pay to the cloud provider. Hence, the issue of pricing in multi-tenant cloud applications can be complex, especially when coupled with software maintenance costs which vary widely depending on the architecture chosen. MDE could be used in a novel way to manage these economics. For example, there could be a tool that lets service providers model their applications where the financials are evaluated on the model; hence, service providers can continuously refine their applications based on the predicted financials of the model.

Most current cloud pricing models are based on the usage and/or lease of virtual infrastructure resources where consumers pay for computing, storage, and network resources either on subscription or on-demand basis. For SaaS tenant-based consumption, pricing has to be based on which software features a tenant uses. This would necessitate that SaaS applications are built in a modular manner where features are associated with pricing units. Thus, MDE could be used to generate different metered versions of the same cloud application to suit consumers' needs. MDE could also be used as an enabler for self-service pricing decision support systems.

6 Enabling New Business Opportunities in the Cloud

Whittle et al. [14] argue that MDE is more likely to be a success if it enables new business opportunities rather than simply bringing productivity gains to existing business opportunities. There is a strong case that MDE can bring such new business opportunities to the cloud. In multi-tenant applications, customizations for each tenant are required; however, current approaches typically only allow somewhat simple customizations because of the software maintenance costs that would be incurred by allowing more complex customizations. For example Salesforce [13], the largest SaaS provider, handles tenants' customizations through metadata that allows customers to modify mainly the database.

However, cloud providers typically stop short of allowing radical customizations, such as wildly different business logic or architecture. This is because of the costly

maintenance and evolution costs that would be associated with this. MDE, however, potentially allows these maintenance costs to be reduced, hence enabling service providers to open up completely new business avenues by allowing tenants to add new business functionality that distinguish them from other tenants in contrast to mere database or simple interface customization.

7 Conclusion

Cloud computing is rapidly becoming the favored computing paradigm for the IT industry. New technological innovations are needed to make multi-tenant cloud computing a sustainable mainstream. To that end, we believe that MDE has the potential to create new disruptive opportunities for multi-tenant applications. In this paper, we have highlighted some of these opportunities and provided samples of ongoing research in this field.

8 References

1. P. Mell and T. Grance, "The NIST Definition of Cloud Computing." National Institute of Standards and Technology, Special Publication 800-145, Bethesda, Maryland, September 2011.
2. K. Schmid and A. Rummler, "Cloud-based software product lines," in Proceedings of the 16th International Software Product Line Conference, 164–170. New York, NY, 2012,
3. H. Brunelière, et al, "Combining Model-Driven Engineering and Cloud Computing," in 4th Workshop on Modeling, Design, and Analysis for the Service Cloud, Paris, France, June 2010.
4. H. Jegadeesan and S. Balasubramaniam, "A Method to Support Variability of Enterprise Services on the Cloud," IEEE International Conference on Cloud Computing, 117-124, 2009.
5. M. Almorsy, J. Grundy, and A. S. Ibrahim, "Adaptable, model-driven security engineering for SaaS cloud-based applications," Automated Software Eng. Journal, vol. 29, 2013, 1–38.
6. D. Betts et al., "Developing Multi-tenant Applications for the Cloud on Windows Azure," Microsoft Patterns & Practices, March 2013
7. S. Walraven, et al, "Efficient Customization of Multi-tenant Software-as-a-Service Applications with Service Lines", The Journal of Systems & Software, vol. 91, 48-62, 2014.
8. M. Abu-Matar, et al, "Towards Software Product Lines Based Cloud Architectures," Proceedings of the IEEE International Conference on Cloud Engineering (IC2E), 2013.
9. Nicolas Ferry, et al, "Towards Bridging the Gap Between Scalability and Elasticity," 4th International Conference on Cloud Computing and Services Science, 746-751, 2014.
10. K. Schmid and A. Rummler, "Cloud-based Software Product Lines," Proceedings of 16th International Software Product Line Conference (SPLC), 164-170, 2012.
11. E. Cavalcante, et al, "Exploiting Software Product Lines to Develop Cloud Computing Applications," the 16th International Software Product Line Conference, 179-187, 2012.
12. J. Schroeter, et al., "Towards Modeling a Variable Architecture for Multi-Tenant SaaS-Applications," VaMoS 2012, 111-120, ACM, New York, NY.
13. https://developer.salesforce.com/page/Multi_Tenant_Architecture, June 8, 2014.
14. J. Whittle, et al, "The State of Practice in Model-Driven Engineering." IEEE Software 31(3), 79-85, 2014.