Modelling Cloud Computing Infrastructure

Marianne Hickey and Maher Rahmouni,

HP Labs, Long Down Avenue, Bristol, BS34 8QZ, UK
{Marianne.Hickey, Maher.Rahmouni}@hp.com

Abstract. We present a modelling approach for an adaptive cloud infrastructure consisting of secure pools of virtualized resources, in order to facilitate automated management tasks and interaction with the system by a human administrator, or programmatically by a higher level service. The topology of such a system is rapidly changing as, for example, it has the abilities to create, modify or destroy pools of virtual resources according to customer demand, as well as dynamically modify the mapping of virtual to physical resources. It is also highly distributed and management data needs to be compiled from disparate sources. Our modelling approach, based on the semantic web, allows us to represent complex topologies, model incomplete or erroneous systems and perform operations such as query, while still allowing validation of the models against system invariants and policies. It also supports distributed modelling, allowing sub-models to be combined, data merging, and shared vocabularies.

Keywords: Modelling, Cloud Computing, RDF, Ontology, Rules, Validation

1 Introduction

There is currently a shift towards cloud computing, which changes the model of provision and consumption of information technology (IT) services, and separates the IT consumer or customer organisation from much of the direct cost and management of IT provision. Rather than an organization managing IT services on their own computing infrastructure, cloud computing takes the approach of meeting an organisation's IT needs, partly or wholly, by IT services available on the internet.

The infrastructure to support cloud computing needs to be highly adaptive and distributed. The topology is rapidly changing, as the platform on which applications run will have the abilities to create and destroy pools of virtualized resources, to manage dynamic resource allocation across the population of resources, and to detect and recover from failures. Information about the infrastructure needs to be assimilated from, and integrated into, a variety of management sources. Throughout all this change, topological constraints and management policies will need to be applied.

Our aim is to model a platform for cloud computing, to enable interaction programmatically by other management systems and higher level services, as well as by human administrators. This paper discusses the requirements of modelling a cloud computing infrastructure (Section 2) and presents a solution based on semantic web [1] technologies (Section 3). Sections 4 and 5 discuss related work and conclusions.

2   Issues in Modelling a Cloud Computing Platform

Fragmentation of service data is one issue that increases the difficulty of IT management. With virtualization, the fragmentation of data accelerates. A cloud computing infrastructure is highly distributed; for example, virtual machines in a particular customer's resource pool may be spread across a heterogeneous physical infrastructure, spanning inside and outside the enterprise. The system model itself may be distributed amongst hosts but may need to be merged, for example for a complete view of the system for validation. The infrastructure model needs to interoperate with a variety of data and management agents. Information about the infrastructure needs to be assimilated from a range of sources, including discovery and a range of other management sources, from within and beyond an organization's boundaries. Likewise, information from the system model may need to be merged with other data in a larger management system.

The topology of an infrastructure to support the demands of cloud computing obviously needs to be highly dynamic, for example: physical hosts may fail or need to be taken down for servicing; virtual machines may be migrated between physical hosts because of this, or as a result of dynamic resource allocation that aims to reduce cost by limiting the number of physical hosts used and/or power consumption; customers may request a new resource pool at any time or modify existing ones to meet their demand. While all this change is going on, we need to be able to validate that the system adheres to topological constraints and policy requirements, and if it does not, query the model to investigate why.

With an adaptive infrastructure, we need to model complex topologies and dependencies. For example, many systems have topologies where partonomies, based on part-whole relationships, are as important as subsumption hierarchies that are based on sub-class relationships.

The platform will need to support a variety of different customers and keep pools of resources owned by different customers secure. We need to represent system invariants and policies, and these need to be validated; for example, a policy might require that a virtual machine cannot belong to two different pools of resources.

3   Our Solution

Our approach to a programmatic way to model, validate, query and compare a cloud computing platform is based on semantic web technologies, see Fig. 1, as they provide several useful features for modelling a distributed, rapidly changing, heterogeneous environment, such as: a standard, open representation and set of tools; ability to model complex topologies; data merging; shared vocabularies; ability to model incomplete/erroneous data; inference; ability to represent policy and topology constraints. The approach uses the Resource Description Framework (RDF), [2], to represent instances of an adaptive infrastructure, and an ontology and system topology rules to model the domain. For validation, it uses layers of generic and domain specific validation rules.

The topologies we need to model require relationships to be expressed between

classes, such as cardinality restrictions, and we therefore need to use OWL [3], rather than RDF-Schema (RDFS) [4]. Further, we need to model partonomies as well as subsumption hierarchies, and other relationships that are not expressible in OWL [5]. We use a rules language to model such relationships. Rules are also used to represent system invariants and policy, and these rules can be run in a specific validation phase. Our approach separates the description of the domain and validation criteria, which enables the same underlying model to be used with different criteria. We can model a system that does not conform to system invariants, and still be able to process and query the model.
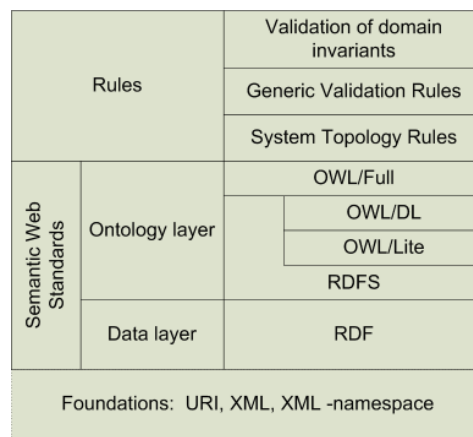
Rules		Validation of domain invariants
		Generic Validation Rules
		System Topology Rules
Semantic Web Standards	Ontology layer	OWL/Full
		OWL/DL
		OWL/Lite
		RDFS
	Data layer	RDF
Foundations: URI, XML, XML -namespace		

Fig. 1. Modelling with the Semantic Web Technology Stack

Using the Jena toolkit [6], we have developed a proof of concept demonstrator, based on an OWL ontology, Jena rules [7] to support topology that cannot be expressed in OWL and Jena validation rules to detect and report minimum and maximum cardinality violations, and other violations of domain invariants.

4   Related Work

Management systems often have proprietary representation, querying and data manipulation methods. Using RDF/OWL for modelling, with SPARQL [6] for querying the models, is a promising approach to eventually facilitate interoperability between management systems and data integration between management data repositories. In the context of cloud computing platforms, this is more relevant than ever before.

There has been work on standardizing systems modelling languages. The most prominent of these is the Service Modelling Language (SML) [8], which is based on profiles of XML Schema (for defining structural aspects of SML models) and the ISO Schematron language (for defining constraints), with some additional support for inter-document references. A key difference between an RDF/OWL approach and an

XML-Schema approach is that it deals directly with a model of the domain, whereas XML-Schemas make statements about XML element structures for describing the domain. With the latter approach, syntactic considerations can get in the way of modelling and operations such as query and transformation [9]. RDF/OWL also has good support for semi-structured data.

The issue of transferring characteristics across a partonomy is addressed in [10], which uses a different Rules language, SWRL, and primarily considers domains such as medical terminologies and web services.

5   Conclusions

With the shift towards cloud computing, IT systems models are needed more than ever, to provide an administrator or automated management tools good visibility and control in order to better manage IT services. However, cloud computing itself presents significant challenges to modelling as the management data is potentially more and more fragmented and topologies are rapidly changing to meet customer demands and resource allocation decisions. Models from disparate management sources need to interoperate, and data needs to merge transparently. Throughout all this change, topological constraints and management policies need to be applied.

This paper presents a modelling approach to attempt to meet these challenges. The approach is based on the semantic web technology stack, and thus uses a standard, open representation and tools, as well as providing features such as: support for distributed models, including data merging and shared vocabularies; inference; the ability to represent incomplete systems, complex topologies, topology constraints and policy rules.

6   References

1. W3C Semantic Web Activity, http://www.w3.org/2001/sw/
2. Manola, F., Miller, E., McBride, B. (eds.): RDF Primer, W3C Recommendation, http://www.w3.org/TR/2004/REC-rdf-primer-20040210/ (2004)
3. Bechhoffer, S. et al: OWL Web Ontology Language Reference, W3C Recommendation, http://www.w3.org/TR/2004/REC-owl-ref-20040210/ (2004)
4. Brickley, D., Guha, R.V., McBride, B. (eds.): RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation, http://www.w3.org/TR/2004/REC-rdf-schema-20040210/ (2004)
5. Horrocks, I.: OWL Rules, OK?: W3C Workshop on Rule Languages for Interoperability, 27-28 April 2005, Washington, D.C. (2005)
6. Jena – A Semantic Web Framework for Java, http://jena.sourceforge.net/
7. Jena 2 Inference support, http://jena.sourceforge.net/inference/index.html
8. Service Modeling Language, Version 1.1, W3C Recommendation, http://www.w3.org/TR/2009/REC-sml-20090512/ (2009)
9. Reynolds, D., Thompson, C., Mukerji, J., Coleman, D.: An Assessment of RDF/OWL Modelling, HP Labs Technical Report, HPL-2005-189 (2005)
10. Horrocks, I., Patel-Schneider, P.F., Bechhofer, S., Tsarkov, D.: OWL Rules: A Proposal and Prototype Implementation, Journal of Web Semantics, vol. 3, no. 1, pp. 23--40 (2005)