Proceedings Paper:
Armstrong, D, Djemame, K, Nair, S et al. (2 more authors) (2011) Towards a contextualization solution for cloud platform services. In: Lambrinoudakis, C, Rizomiliotis, P and Wlodarczyk, TW, (eds.) Proceedings - 2011 3rd IEEE International Conference on Cloud Computing Technology and Science. Proceedings - 2011 3rd IEEE International Conference on Cloud Computing Technology and Science, CloudCom , 29 Nov - 01 Dec 2011, Athens, Greece. IEEE , 328 - 331. ISBN 9780769546223

https://doi.org/10.1109/CloudCom.2011.51

Towards a Contextualization Solution for Cloud Platform Services

Django Armstrong*, Karim Djemame*, Srijith Nair†, Johan Tordsson‡ and Wolfgang Ziegler§
*School of Computing, University of Leeds, Leeds, UK LS2 9JT
Email: {een4dja,scskd}@leeds.ac.uk
†Security Futures Practice, BT, Office Minerva, Herikerbergweg 2, Amsterdam 1101CM, Netherlands
Email: srijith.nair@bt.com
‡Department of Computing Science, Umeå University, SE-901 87 Umeå, Sweden
Email: tordsson@cs.umu.se
§Department of Bioinformatics, Fraunhofer Institute SCAI, 53754 Sankt Augustin, Germany
Email: Wolfgang.Ziegler@scai.fraunhofer.de

***Abstract*—We propose a cloud contextualization mechanism which operates in two stages, contextualization of VM images prior to service deployment (PaaS level) and self-contextualization of VM instances created from the image (IaaS level). The contextualization tools are implemented as part of the OPTIMIS Toolkit, a set of software components for simplified management of cloud services and infrastructures. We present the architecture of our contextualization tools and the feasibility of our contextualization mechanism is demonstrated in a three-tier web application scenario. Preliminary performance results suggest acceptable performance and scalability of our prototype.**

I. Introduction

Modern virtualization technologies enable rapid provisioning of Virtual Machines (VMs) and thus allow cloud services to scale up and down on-demand. This elasticity [6], [8] comes with a new set of challenges for dynamic service configuration. Contextualization is a set of processes and mechanisms that enable a service to scale elastically alongside the resources and software that support it through the orchestration of these dependencies toward the common goals of the service. We focus on horizontal elasticity where scaling is achieved by adding or removing VMs to a service during its operation. The related case of vertical elasticity, i.e. application scaling through VM resizing, is much easier from a contextualization perspective. For horizontal elasticity scenarios, predefined yet flexible contextualization mechanisms enable the VMs of a service to self-discover and communicate.

We identify three main challenges in enabling elasticity through contextualization where VMs are added and removed during service operation: i) contextualization support offered by the Platform as a Service (PaaS) layer, to replace the traditional approach that requires complex and time consuming manipulation of VM images as part of development of each Software as a Service (SaaS) solution; ii) contextualization of services that are deployed across multiple Infrastructure as a Service (IaaS) providers; and iii) contextualization with support for functional requirements such as secure network overlays and the incorporation of licence-protected software in services.

Our contextualization tools are implemented as part of the OPTIMIS Toolkit, a set of software components for simplified management of cloud services and infrastructures. The feasibility of our contextualization mechanisms is demonstrated in a three-tier web application scenario, where the database and application server tiers both are replicated and scaled elastically. The contextualization mechanisms enable elasticity for this application such that instances of database and application server can be transparently added and removed from the service during operation. We also provide results that demonstrate the scalability and performance of our mechanism in a multi-user cloud environment.

The remainder of this paper is structured as follows; Section 2 discusses the challenges within contextualization and motivates our work. Section 3 describes our proposed contextualization architecture for OPTIMIS, an illustrative use case and preliminary performance results. Section 4 discusses related work and finally Section 5 presents our conclusions and future work.

II. Contextualization Challenges

We argue that contextualization in cloud computing remains a highly pervasive key technological requirement of any cloud service, where elastic resource management is critical to the on-demand scalability of a service. The holistic nature of the services deployed on clouds makes it difficult to provide flexible generic and open PaaS tools without limiting heterogeneity of supported services. We identify three inherent challenges to providing elasticity through contextualization where VMs are added and removed during service operation.

The first challenge to overcome is the complete contextualization of cloud services across all classifications [14] within the Cloud ecosystem: SaaS, PaaS, and IaaS. This pertains to low-level contextualization of virtual resources, as found in IaaS providers, where virtual devices require context to enable VMs to be bootstrapped to existing virtual infrastructures. This approach has been partially explored by Reservoir [13]. In addition, the contextualization of software dependencies that supports a deployed service in a PaaS provider needs to be scalable. Finally, the service itself must be developed in such a fashion that enables scalability.

The second challenge to overcome is contextualization across multiple IaaS domains for reasons of interoperability. Many IaaS providers, such as Amazon Web Services [1], offer PaaS services that are not interoperable with those of other providers. In these PaaS services, contextualization is performed as part of service development, which makes the process customized to a single provider only. This presents an opportunity for contextualization at service deployment time only, thus enabling interoperability between IaaS providers by not having to rely on these platform services. Contextualization at deployment time only, incurs additional challenges in relation to the re-contextualization of resources during runtime for the accommodating of service elasticity and on-demand scalability, whereby platform services must be self-discovering and autonomous. This adds complexity to the contextualization mechanism that must be used to support the software dependencies of a service at the platform level.

The third challenge pertains to a set of functional requirements for real world clouds and their impact on contextualization. Notable among these are end-to-end security through contextualization mechanisms that support a Virtual Private Network (VPN) overlay and software license management systems. Both of these have unique contextualization requirements: contextualization must be secure with no VPN keys stored unless in use; and contextualization that is able to accommodate license protected software and licensing tokens.

III. The Contextualization Tools

A. Contextualization Architecture

The contextualization tools are comprised of a core component, the VM Contextualizer and a repository of scripts. This component provides an interface for consuming service context data, such as: security certificates, VPN hostnames, VPN DNS and Gateway IP addresses, mount points for network data stores, monitoring manager hostnames, offline software license tokens and a list of software dependencies. The VM Contextualizer provides two capabilities. The first is a bootstrap mechanism to prepare a VM image for receipt of context agnostic of the operating system type used. The second capability provides a mechanism for creating ISO CD ROM images that contain context data and a data processing script for the manipulation of the data into a format suitable for consumption at runtime when the ISO image is mounted. The VM Contextualizer mounts a VM image and modifies it to include an assortment of bespoke scripts that interact with the guest operating system, service and service dependencies at boot time, preparing the image to receive context in a reusable fashion. At run time these scripts access contextualization data held within this ISO image, as per the OVF recommendation [11], giving a VM its identity.

The inclusion of the ISO image as a mechanism to store contextualization information provides a facility to separate the contextualization data from the VM image. This removes the time consuming need to create multiple unique VM images for each VM that is required to be contextualized, while also improving the security of the contextualization process as security certificates are not stored in the VM image itself but instead stored in the ISO image when it is dynamically generated. The inclusion of a script to process the contextualization data provides an approach to store the data agnostic of the operating system, service and service dependencies used.

The VM Contextualizer uses parts of QEMU [12], a generic and open source machine emulator and virtualizer to manipulate images. QEMU provides a tool named "qemu-img" that enables the conversion of virtual machine images. In addition, Linux system tools (such as "mount", "iosetup" and "kpartx") are used to mount VM images as loop devices for write manipulation. For the creation of ISO images, the Linux system tool "mkisofs" is used in addition to the previously mentioned tools to create and modify ISO images.

Our contextualization research is performed in the context of the OPTIMIS Toolkit [4], a set of software components aimed to simplify and optimize the construction, deployment, and operation of services (at the SaaS and PaaS level) as well as the operation of virtualized hardware needed to deliver these services (IaaS level). Although OPTIMIS targets scenarios and capabilities somewhat different to the ones provided by current PaaS and IaaS providers, the OPTIMIS contextualization requirements are general and the solution should be applicable in a wide range of cloud providers.

B. Service and Toolkit Component Contextualization

Figure 1 illustrates the instance-level contextualization process of a VM at the beginning of its execution. During the boot sequence of a VM the contextualization tools mount an ISO CD image that contains contextualization data and a script to process this data into a useable form. This script communicates through a known interface to the OS specific contextualisation scripts embedded in the VM image. These OS specific scripts manipulate configuration files of associated OPTIMIS components and software dependencies that support the end user's service(s), setting their context. The scripts can remain in a daemon-like mode for a component or software dependency that requires continuous updates to its context, enabling dynamic reconfiguration and instant discovery of elastic resources, if needed.

There are several problems to solve regarding contextualization of OPTIMIS system level components, such as those associated with license management and cloud security. The following subsections outline these issues:

1) License Management: Access to licenses for authorizing the execution of an application in a Cloud beyond the administrative domain of the site running the license server usually leads to applications aborting during start-up because of unreachable license servers, e.g. due to firewall issues. In OPTIMIS we use a prototype for software licensing developed in the European project SmartLM [2], which provides licensing technology for location independent application execution. Separation of authorization for license usage and authorization for application execution on the one

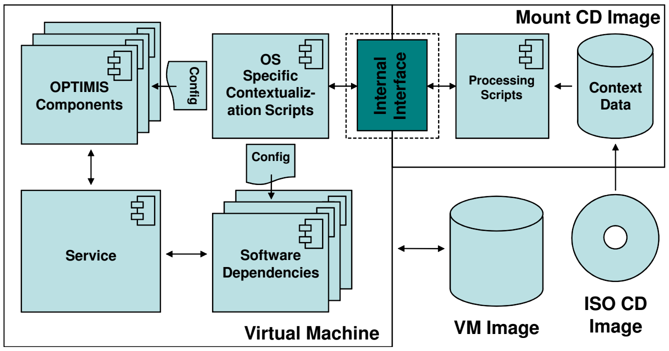


Fig. 1. Interaction between VM image and ISO Image at run time.

hand and software tokens that carry the authorization information on the other hand provide the necessary flexibility for licenses following applications into Clouds. It is one of the tasks of the VM Contextualizer to retrieve and embed a license token into the VM hosting the application. In case of multiple applications in a VM, required e.g. for a workflow, the VM Contextualizer assures that all required licenses are in place when the applications start up. No communication between the application and the license server that issued the token is required at runtime. Additional approaches will be implemented enhancing the SmartLM solution: (i) dynamic deployment of a trusted instance managing a number of tokens for one or multiple applications and (ii) dynamic deployment of a full license service with a subset of the licenses available at the home organisation of the user. The configuration of the dynamically deployed license service will be managed by the VM Contextualizer. This approach is especially useful when the same Cloud resources are used over a longer period of time for running license protected applications. In the first approach the contextualizer tools are responsible for configuring and deploying the trusted instance for the respective network environment and to transfer tokens.

2) Cloud Security: Each instance of a VM requires specific security customisations based on the service it provides and its threat profile. For example, the firewall rules specific to a web server VM is different from that of a database VM and these variations are handled by the contextualization tools. In addition, the OPTIMIS Data Manager provides a means of provisioning secure encrypted storage devices for VMs, where the decryption keys are stored outside an IP. The specifics of these secure device configurations are different across various VMs and are set by the contextualization steps. Various Identity and Access Management (IdAM) components that need to be installed, along with policies specified at the VM endpoints, are also set by this component. If required by the end user or SP, other security mechanisms like Intrusion Prevention Software can also be instantiated and customised by similar mechanisms.

C. A Contextualization Use Case

At the IP level, predefined context from a SP is applied to a VM as it is brought online. This does not require

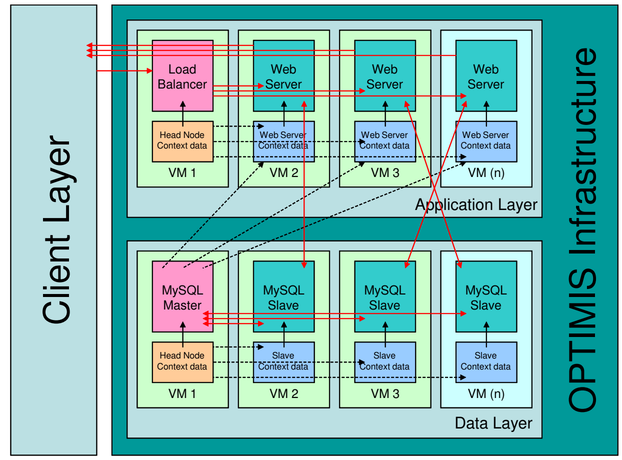


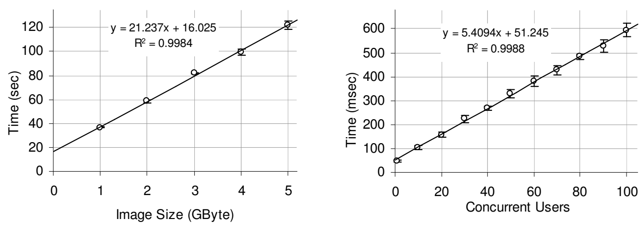
Fig. 2. Contextualization in a three-tier web application.

communication with any IP level component and addresses problematic re-contextualization of VMs at runtime. A three tier web application demonstrates the OPTIMIS toolkit in operation and the complexities involved in the contextualization of VMs. Figure 2 illustrates the relationship between the contextualization data and a VM. From the figure it can be seen that each layer of the service forms a cluster of cooperating resources that rely on a head node to provide information on the state of VMs and balance load. Each new VM brought online contains information about the head node to which it is to register for active duty in addition to other VMs to which it must communicate to perform its role. The information is stored within the contextualization data of the VM, which includes a subset of the contextualization data from the head node and can reference other sources of information. These other sources of information can update the contextualization data continuously during runtime if needed. This enables VMs to be taken offline without disturbing the operation of a service.

To confirm the validity of our contextualisation approach in such a use case we have created and tested a prototype of our contextualization tools on our cloud testbed using a Dual CPU (Intel Xeon E5630) server with 16 GB of RAM and 1 TByte WD SATA 7200 rpm HDD. Figure 3a and 3b provide evidence on the potential performance of our approach for contextualization with regards to preparing VM Image sizes in the range of 1-5 GByte in increments of 1 Gbyte and with varying numbers of concurrent user requests from 10-100 in increments of 10, to create ISO CD Images containing 1 Mbyte context data. The results show adequate scalability and response time over 10 iterations of the experiment with minimal variance, as shown by the error bars on the graphs.

IV. Related Work

Approaches to contextualization vary considerably depending on the nature of the application or virtual appliance. In the Grid community, research into the effective use of Cloud computing for academic use and the implications on



(a) (b)

Fig. 3. Left: Time to prepare a VM image. Right: response time of concurrent user requests to generate ISO images.

contextualization have been explored [7], [9]. The approach to contextualization by the Nimbus Project [10] in [7] is to integrate heavily with the Globus Toolkit [5], limiting the general applicability of the contextualization solution to a small number of use cases. In addition, re-contextualization of resources is considered to be a necessity unlike the approach outlined in this paper. In [9], details of a mechanism for the contextualization of scientific virtual appliance are discussed for the purpose of replicating Grid services. An approach is proposed that exposes users to a high level declarative language in XML for performing typical steps in the deployment of a scientific application. The goal of this contextualization mechanism differs from our own in that we seek to enable elasticity rather than fault tolerance of Cloud resources and again the proposed solution is specific to a single use case from the academic community.

Finally in [3], the challenges and techniques of contextualization are presented for a cluster environment at the Clemson University. The paper describes low level issues of contextualization and makes some recommendations on contextualization from practical experience. Image-level contextualization and the problems associated with its automation and complexity are discussed in detail but no solution is provided. We differentiate our work through the application of contextualization in several cloud scenarios from the OPTIMIS project providing viable solution to these problems.

V. Conclusion and Future Work

In this paper we have discussed the nature of contextualization in the light of the OPTIMIS Project and presented details of the architecture and implementation of our tool for the contextualization of platform level services as well as virtual infrastructure. We have discussed the implication of contextualization in clouds, the motivation behind our work and suggested a landscape for the evolution of contextualization tools across all classes of Clouds within the ecosystem of the future. We contribute to both the image and instance level contextualization of VM's and illustrate the potential effectiveness of our tool through a simple use case and performance results. We describe the need for contextualization of platform tools from the OPTIMIS toolkit and the problems we overcome to support them. Finally some related work on alternative contextualization mechanisms are discussed and differentiated from our own.

As part of our future work we plan to extend our existing architecture to enable support for a greater number of image formats and hypervisors. In addition we plan to integrate further with other core components of the OPTIMIS toolkit such as the tools responsible for VM level monitoring. Finally we plan to implement advanced caching mechanisms that improve the ability of the tools to reuse existing contextualized images, increasing the general applicability of our solution to a wider number of applications while reducing the time to market for a Cloud SaaS solution.

Acknowledgments

This work has been partially supported by the EU within the 7th Framework Programme under contract ICT-257115 - Optimized Infrastructure Services (OPTIMIS).

References

[1] "Amazon Web Services," http://aws.amazon.com. [Online]. Available: http://aws.amazon.com

[2] C. Cacciari, F. D'Andria, B. Hagemeier, D. Mallmann, J. Martrat, D. Garc'a PerŽz, A. Rumpl, W. Ziegler, and C. Zsigri, "Software licenses as mobile objects in distributed computing environments," in *Euro-Par 2010 workshops*. Springer, Berlin, 2010.

[3] M. Fenn, S. Goasguen, and J. Lauret, "Contextualization in Practice: The Clemson Experience," in *Proceedings of the 13th International Workshop on Advanced Computing and Analysis Techniques in Physics Research (ACAT '10)*, Jaipur, India, 2010.

[4] A. Ferrer, F. Hernández, J. Tordsson, E. Elmroth, A. Ali-Eldin, C. Zsigri, R. Sirvent, J. Guitart, R. Badia, K. Djemame, W. Ziegler, T. Dimitrakos, S. Nair, G. Kousiouris, K. Konstanteli, T. Varvarigou, B. Hudzia, A. Kipp, S. Wesner, M. Corrales, N. Forgó, T. Sharif, and C. Sheridan, "OPTIMIS: a Holistic Approach to Cloud Service Provisioning," *Future Generation Computer Systems*, 2011, doi:10.1016/j.future.2011.05.022.

[5] "The Globus Toolkit," http://www.globus.org/toolkit. [Online]. Available: http://www.globus.org/toolkit

[6] Z. Gong, X. Gu, and J. Wilkes, "PRESS: PRedictive Elastic ReSource Scaling for cloud systems," in *Proceedings of the 6th International Conference on Network and Service Management (CNSM '10)*, Piscataway, NJ, USA, 2010, pp. 9 – 16.

[7] K. Keahey and T. Freeman, "Contextualization: Providing One-Click Virtual Clusters," in *Proceedings of the 4th IEEE International Conference on eScience (ESCIENCE '08)*, Washington, DC, USA, 2008, pp. 301 – 308.

[8] S. Krishnan and J. Counio, "Pepper: An Elastic Web Server Farm for Cloud Based on Hadoop," in *Proceedings of the 2nd IEEE International Conference on Cloud Computing Technology and Science (CloudCom '10)*, Los Alamitos, CA, USA, 2010, pp. 741 – 747.

[9] G. Moltó and V. Hernández, "On Demand Replication of WSRD-based Grid Services via Cloud Computing," in *Proceedings of the 9th International Meeting on High Performance Computing for Computational Science (VECPAR '10)*, 2010, pp. 9 – 16.

[10] "Nimbus - An open source toolkit for Infrastructure-as-a-Service Cloud clusters," http://workspace.globus.org/. [Online]. Available: http://workspace.globus.org/

[11] "Open Virtualization Format (OVF) - A standard from the Distributed Management Task Force," http://www.dmtf.org/standards/ovf. [Online]. Available: http://www.dmtf.org/standards/ovf

[12] "QEMU - An open source machine emulator and virtualizer," http://www.qemu.org. [Online]. Available: http://www.qemu.org

[13] B. Rochwerger, J. Caceres, R. Montero, D. Breitgand, E. Elmroth, A. Galis, E. Levy, I. Llorente, K. Nagin, and Y. Wolfsthal, "The Reservoir model and architecture for open federated cloud computing," *IBM Journal of Research and Development*, vol. 53, no. 4, p. Online, 2009.

[14] L. M. Vaquero, L. Rodero Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 50 – 55, 2009.