

# ConPaaS: an Integrated Runtime Environment for Elastic Cloud Applications

Guillaume Pierre<sup>1</sup>  
Guillaume.Pierre@irisa.fr

Kaveh Razavi<sup>2,\*</sup>  
K.Razavi@vu.nl

Renato Figueiredo<sup>2,6</sup>  
renato@acis.ufl.edu

Ana-Maria Oprescu<sup>3</sup>  
A.M.Oprescu@uva.nl

Matej Artač<sup>5</sup>  
Matej.Artac@xlab.si

Thilo Kielmann<sup>2</sup>  
Thilo.Kielmann@vu.nl

Bert IJff<sup>2</sup>  
Bert.IJff@vu.nl

Alexandru Uta<sup>2,\*</sup>  
A.Uta@vu.nl

Thorsten Schütt<sup>4</sup>  
schuett@zib.de

Aleš Cernivec<sup>5</sup>  
Ales.Cernivec@xlab.si

Emanuele Rocca<sup>2</sup>  
ema@linux.it

Hector Fernandez<sup>2</sup>  
Hector.Fernandez@vu.nl

Alexandra Vintila<sup>2,\*</sup>  
A.A.Vintila@vu.nl

Michael Berlin<sup>4,\*</sup>  
berlin@zib.de

- 1 IRISA/Université de Rennes 1, France
  - 2 Vrije Universiteit, Amsterdam, the Netherlands
  - 3 University of Amsterdam, the Netherlands
  - 4 Zuse Institute Berlin, Germany
  - 5 XLAB d.o.o., Ljubljana, Slovenia
  - 6 University of Florida, USA
- \* student author

## 1. INTRODUCTION

Most cloud applications are re-enactments of traditional applications such as Web applications, content delivery and high-throughput computing. The advantages of the cloud are well-known: access to a near-infinite number of resources, ability to scale an application on demand, pay-as-you-go pricing model. However, cloud application developers need to pay attention to new topics: building custom VM images, making applications elastic and scalable, controlling performance, fault-tolerance, etc.

ConPaaS is an open-source, platform-as-a-service middleware which aims at simplifying the deployment of applications in the cloud [4]. In ConPaaS, an application is defined as a composition of one or more *services*. Each service is an elastic component dedicated to the hosting of a particular type of functionality. Examples of services include a Web hosting service [6] supporting PHP and Servlets, a MySQL database service, and a task-farming service [3].

Each ConPaaS service is self-managed and elastic: it can deploy itself on the cloud, monitor its own performance, and increase or decrease its processing capacity by dynamically (de-)provisioning instances in the cloud. Services are designed to be composable: an application can, for example,

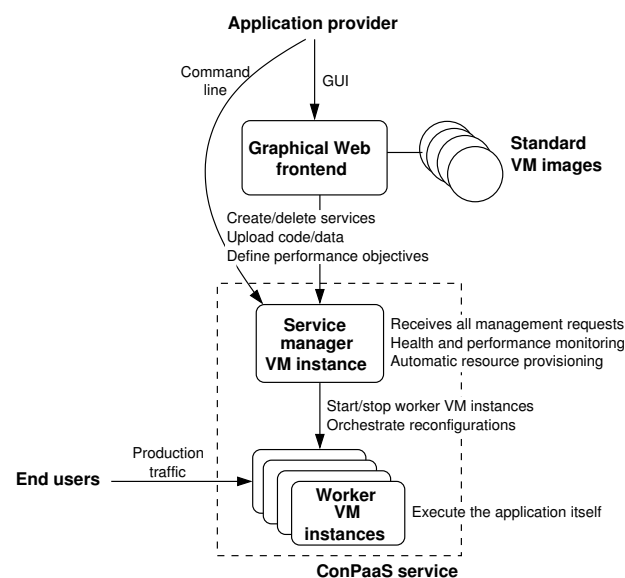


Figure 1: Organization of one ConPaaS service

use a Web hosting service, a database service for internal application state, a file storage service to store access logs, and a MapReduce service to periodically compute statistics from these logs.

## 2. ARCHITECTURE

Application providers interact with ConPaaS using a Web GUI or a command-line interface. The interface allows users to create and delete services, upload application code and data to them, monitor their health and performance, and control their resource usage by changing the number of virtual machines used by each service.

As shown in Figure 1, each service is under the control of one “manager” VM. This VM does not run the application itself. It is in charge of executing all management requests, centralizing health and performance monitoring data, and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

HPDC 2013, June 2013, New York, NY, USA.

Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...\$15.00.

controlling the allocation of resources assigned to this service. The actual application traffic is not addressed to the manager VM. Requests from end users willing to access the application are directed directly to the worker VMs.

A service manager is in charge of requesting or releasing VMs from the underlying cloud infrastructure, uploading the service code or data to the new VMs, and coordinating the reconfiguration among all its VMs. Reconfigurations are carefully organized such that the application remains available during the entire reconfiguration process. Service managers also make sure to keep the first worker VM active at all times so that applications remain bound to a stable address, even after multiple reconfigurations.

Depending on the nature of the service, worker VMs can be fully symmetric or further specialized into a number of roles. For example, all nodes of a task farming service are identical, and can process any request indifferently. Conversely, a Web hosting service is decomposed into a load balancer (running in the first VM acting as the external contact address of the service), and any number of Web servers serving static content and application servers processing dynamic document generation requests.

### 3. RESEARCH CHALLENGES

The work on ConPaaS makes us address a number of difficult research challenges. Among them are the following.

#### 3.1 Complex application composition

A ConPaaS application can be composed of several independent services. Deploying it automatically in the cloud means that a different set of machines is used every time. However, each part of the application must be able to invoke other components at a well-known address. Rather than updating the client-side addresses at runtime, ConPaaS dynamically establishes and maintains a VPN between all virtual machines involved in a composed application. Each ConPaaS service can thus choose a local DNS name which gets resolved within the VPN. The VPN is dynamically updated each time a machine instance is added or removed from the application. Besides facilitating service composition, this organization also helps securing the network traffic internal to the application.

#### 3.2 Performance guarantees

A cloud environment allows application programmers to design elastic applications capable of dynamically varying their processing capacity to follow workload variations. For example, a Web application may add resources when load is high, and later release some of these resources to save costs. However, in a composed application a new issue arises: which individual element should be (de-)provisioned such that performance is guaranteed at minimum cost?

Current cloud platforms (Amazon's AutoScale, RedHat's OpenShift etc.) allow one to give a performance objective to each service (web server tier, database tier, etc.), and provision each service individually. However, a better approach consists of giving a performance objective only to the front-end services, and making the other components *negotiate* the most efficient use of resources so that the global performance is maintained at minimum cost. ConPaaS will exploit its global knowledge of each application's service composition to guarantee performance using as few resources as possible.

### 3.3 Cloud heterogeneity

The performance of virtual instances provided by current clouds is largely heterogeneous, even when requesting the exact same type of instance each time [1]. This has a strong impact on resource provisioning strategies: each time ConPaaS requests a new virtual instance from the cloud, it must measure the individual performance profile of this particular instance before being able to decide what is the most efficient use of this resource.

### 3.4 Extensibility

ConPaaS currently comprises seven services: (i) Web hosting; (ii) MapReduce; (iii) Task Farming [3]; (iv) MySQL; (v) Scalaris, a strongly consistent NoSQL database [5]; (vi) XtremFS, a large-scale distributed file system [2], and (vii) a Selenium functional testing service. Although these services are general enough to support a wide range of applications, we aim to make the platform easily extensible to new types of services such as workflows, data streaming, etc.

## 4. SOFTWARE AVAILABILITY

ConPaaS is open source. Current releases are available from [www.conpaas.eu](http://www.conpaas.eu). The site also provides an open testbed (*ConPaaS online*) where potential new users can test the technology without having to install any software.

## 5. DEMO

We intend to demonstrate ConPaaS, in particular the task farming service. Requirements for the demo are minimal: a table for putting a laptop on top, and a stable network connection providing small amounts traffic (for use by a Web browser).

## Acknowledgements

This work is partially funded by the FP7 Programme of the European Commission in the context of the Conrail project under Grant Agreement FP7-ICT-257438.

## 6. REFERENCES

- [1] J. Dejun, G. Pierre, and C.-H. Chi. EC2 performance analysis for resource provisioning of service-oriented applications. In *Proc. NFPSLAM-SOC*, 2009.
- [2] F. Hupfeld, T. Cortes, B. Kolbeck, E. Focht, M. Hess, J. Malo, J. Marti, J. Stender, and E. Cesario. XtremFS – a case for object-based file systems in grids. *CCPE*, 20(8), 2008.
- [3] A.-M. Oprescu and T. Kielmann. Bag-of-Tasks Scheduling under Budget Constraints. In *Proc. 2nd IEEE International Conference on cloud Computing Technology and Science (CloudCom 2010)*, Indianapolis, USA, Nov/Dec 2010.
- [4] G. Pierre and C. Stratan. ConPaaS: a Platform for Hosting Elastic Cloud Applications. *IEEE Internet Computing*, 16(5):88–92, September–October 2012.
- [5] F. Schintke, A. Reinefeld, S. Haridi, and T. Schütt. Enhanced paxos commit for transactions on DHTs. In *Proc. CCGrid*, 2010.
- [6] Z. Wei, G. Pierre, and C.-H. Chi. CloudTPS: Scalable Transactions for Web Applications in the Cloud. *IEEE Transactions on Services Computing*, 5(4):525–539, Oct-Dec 2012.