Contrail: Distributed Application Deployment under SLA in Federated Heterogeneous Clouds*

Roberto G. Cascella[1], Lorenzo Blasi[2], Yvon Jegou[1], Massimo Coppola[3], and Christine Morin[1]

[1] Inria, Campus Universitaire de Beaulieu, 35042 Rennes, France
{roberto.cascella,yvon.jegou,christine.morin}@inria.fr,
[2] HP Italy Innovation Center, via Grandi 4, 20063 Cernusco Sul Naviglio (MI), Italy
lorenzo.blasi@hp.com
[3] CNR/ISTI "A.Faedo", via G. Moruzzi 1, 56124 Pisa, Italy
massimo.coppola@isti.cnr.it

Abstract. Cloud computing market is in rapid expansion due to the opportunities to dynamically allocate a large amount of resources when needed and to pay only for their effective usage. However, many challenges, in terms of interoperability, performance guarantee, and dependability, should still be addressed to make cloud computing the right solution for companies. In this chapter we first discuss these challenges and then we present three components developed in the framework of the Contrail project: Contrail federation; SLA manager; and Virtual Execution Platform (VEP). These components provide solutions to guarantee interoperability in a cloud federation and to deploy distributed applications over a federation of heterogeneous cloud providers. The key to success of our solutions is the possibility to negotiate performance and security guarantees for an application and then map them on the physical resources.

Keywords: Cloud computing, federation, SLA, QoS, standards, resource management, interoperability, distributed application deployment.

1 Introduction

After decades in which companies used to host their entire IT infrastructures in-house, a major shift is occurring where these infrastructures are outsourced to external operators such as data centers and computing clouds. The growth of interest toward computing clouds is facilitated by virtualization technologies which offer several advantages over traditional data center approach to computing. On the one hand, companies can move their applications to the cloud freeing themselves from the control and management of the infrastructure so that they can focus on the deployed services. On the other hand, companies can rent resources of cloud providers only when needed according to a pay-as-you-go pricing model reducing the cost of the infrastructure. In a nutshell, this paradigm represents a new opportunity for companies and organisations to rely on highly dynamic distributed infrastructures to run applications and services.

* Invited Paper.

A. Galis and A. Gavras (Eds.): FIA 2013, LNCS 7858, pp. 91–103, 2013.

The cloud computing market is in rapid expansion and many cloud providers are flourishing in Europe to offer Infrastructure as a Service (IaaS) services to contrast big players, like Amazon, that have so far dominated. The market opportunities are quite challenging for new IaaS cloud providers, which might have limited resources to offer. They play in a competitive service market where the organisations and companies they want to attract are looking for large pool of resources, as well as for guarantees in terms of the reliability and availability for their services. However the growth of cloud computing may soon be hindered by other factors such as the customers' concerns to be locked-in within a single commercial offer (which reduces the necessary competition between many infrastructure providers), ownership and privacy issues of the data stored in the cloud, and the lack of performance predictability of current clouds. Other major issues are legal requirements for data: they cannot be stored anywhere for legal jurisdiction implication or need to have specific privacy requirements to stick with company or country legislation. These concerns can be summarised in the lack of trust on the clouds with customers being sceptical in the cloud model and in services offered by a cloud provider if no guarantees exist.

A first step to address the users' concerns is to avoid vendor lock-in giving the opportunity to select the most convenient cloud provider based on the application requirements or price of the offer. Interoperability among cloud providers is the only way to challenge vendor lock-in and open the way toward a more competitive market. Moreover, interoperability is a need for small players to enter a market dominated by big cloud providers which can count on a huge number of resources. As such, interoperability becomes even more handy and needed in a multi provider scenario, where customers can protect their investment by counting on a wider number of options to offer their services on top of cloud systems. At the same time, they take full advantage of the elasticity and pay-as-you-go concepts. One way to achieve interoperability is via the adoption of cloud standards or a middleware service that adapts the application to a specific cloud provider. A more comprehensive way to address interoperability is the cloud federation: it can help in hiding the complexity of managing heterogeneous resources and using a multitude of cloud providers at the same time.

However, on top of these federated cloud providers it is of utmost importance to ensure the availability of the computing resources and to provide strict guarantees in terms of quality of service (QoS) and quality of protection (QoP). Users and organizations should have the opportunity to specify these features in a negotiated Service Level Agreement (SLA) and to monitor them at runtime. Deploying applications and services under a SLA will make cloud computing a valid alternative to private data centers responding to the users requirements in terms of availability, reliable application execution, and security.

Few approaches exist so far and they focus more on brokering among cloud providers, ranking them and selecting one based on an objective function [17]. Other approaches focus on creating a level of abstraction to present the resources of cloud providers in a transparent way to the user and then orchestrating the deployment over different cloud providers [4,14,19]. However, to the best of our

knowledge, there is no previous work providing a complete solution that tackles interoperability issues, security, and performance guarantees.

Contrail [1] is a European project addressing all these issues to allow a federation of heterogenous clouds to deploy distributed applications under QoS and QoP constraints. Contrail is an open source integrated approach to virtualization, which aims at offering Infrastructure as a Service services (IaaS), services for federating IaaS clouds, and Contrail Platform as a Service services (ConPaaS) on top of federated clouds. In Contrail, the user is relieved from managing the access to individual cloud providers and can focus on specifying the service or application to be automatically deployed over a multitude of heterogeneous providers. The providers can rely on different cloud technologies, exploit different hardware, or offer different types of guarantees.

Contrail offers performance (QoS) and security (QoP) guarantees via SLA enforcement by monitoring the execution of the application, and a scalable management of the computing resources via an interoperable federation. The federation service is the interface with the user, who needs to submit the description of the distributed application to be deployed in the cloud, along with its runtime configuration, and specify the requirements in terms of OVF (Open Virtualization Format specification) [7] and SLA documents respectively. Then, the federation ensures that the providers' resources are utilized as needed for offering a dependable and trustworthy cloud service to customers.

The application is deployed via the Virtual Execution Platform (VEP), a provider-level service supporting federation-level interoperability. The use of VEP allows to deploy a distributed application under the terms of a SLA over the resources of any of the supported IaaS providers, regardless e.g. of the underlying cloud management system. Elasticity of the application is also ensured by monitoring the usage of the resources stated in a negotiated SLA, both within the cloud provider infrastructure and at the federation level.

In this chapter, we present the Contrail federation service, SLA management (negotiation and enforcement) in federated heterogeneous clouds, and the Virtual Execution Platform. The remainder of this chapter is organized as follows. Section 2 highlights the architecture of Contrail software stack and discusses the main services offered. Section 3 presents the cloud federation, a relatively new concept, and the Contrail view and implementation. Section 4 discusses the Contrail project achievements in terms of managing and negotiating Service Level Agreements (SLAs). Section 5 presents the Virtual Execution Platform services and the management of a distributed application. Section 6 concludes this chapter and draws future directions.

2 Contrail Architecture

Fig. 1 depicts the Contrail architecture. The federation layer is the entry-point for users, who register and authenticate to use the Contrail services. The way the Contrail federation is conceived enables seamless access to the resources of multiple cloud providers, avoiding potential vendor lock-in for the end users. It reaches

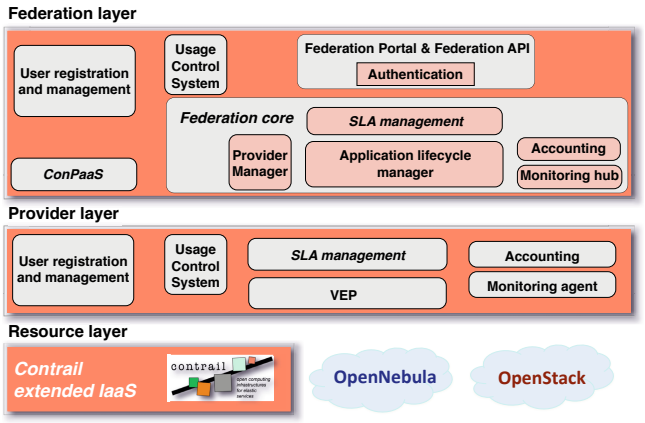

Fig. 1. Simplified vision of the Contrail architecture

a high degree of interoperability by managing private or public cloud providers' resources regardless of the technology implemented or underlying hardware. The Contrail federation enables users to deploy distributed applications on demand on different cloud providers by only interacting with this single component. The federation incorporates in the *Federation core* the necessary functionalities to negotiate SLAs and monitor their enforcement while the application is running. The Federation core, together with its interfaces, is deployed at each *Federation Access Point* (FAP). A Contrail federation is thus made up of distributed, interconnected instances of FAPs and providers. The user submits the description of the application based on the OVF [7] standard format and negotiates the SLAs terms, then the federation selects the most suitable cloud providers based on the resources available, the expressed SLA terms for QoS and QoP, and the reputation of the providers, i.e., matching the level of performance and trustworthiness required by the application. Hence, the federation proceeds to negotiate proper SLA terms with each provider in a transparent way for the users. Contrail technology is able to satisfy the user needs for the deployment of elastic and scalable applications by adding or removing resources in order to satisfy the SLA terms without the need of a new SLA negotiation. Monitoring and auditing are performed during application execution, to ensure that there is no violation of the SLA. Proper authorization and security mechanisms are enforced primarily at the federation layer and then at the other layers to guarantee quality of protection (QoP).

The provider layer implements the business part of a cloud provider: (i) negotiation with the federation and enforcement of the SLA; (ii) resource reservation and application management; (iii) monitoring and accounting. The resource layer is in charge of managing the physical resources of a cloud provider. Contrail

does not implement a new IaaS, but leverages the existing ones[1] by adding those functionalities required to provide performance and security guarantees for an application.

In Contrail, each cloud provider runs a copy of the Virtual Execution Platform (VEP) software which in turn seamlessly integrates the provider resources within the Contrail federation. VEP is an open source technology implementing standards that exploits resource virtualization to provide virtualized distributed infrastructures for the deployment of end-user applications independently from the underlying platform: Contrail extended IaaS, OpenNebula or OpenStack[4]. It offers a reliable application deployment platform, which is resilient to operational failures and which ensures that an application is deployed respecting QoS requirements. The degree of interoperability and features that the federation can exploit on each single cloud provider depend on the specific functionalities implemented at the cloud provider level. Interoperability is achieved through the VEP component.

Other relevant services developed in Contrail but not detailed in this chapter are: (i) the Virtual Infrastructure Network (VIN) service which assures the internetworking between Virtual Machines (VMs) of an application and with the public Internet, providing bandwidth reservation capabilities within a data center and isolated environments for an application; (ii) the Global Autonomous File System (GAFS) which guarantees a reliable and highly available storage service for VM images and system logs, and scalable Storage as a Service to cloud users and applications; (iii) self-managed, elastic, and scalable ConPaaS (Contrail PaaS) services [16] which can deploy themselves on the cloud, monitor their own performance, and increase or decrease their processing capacity by dynamically (de-)provisioning instances of the ConPaaS service in the cloud.

The following sections focus on the Contrail components that enable the deployment of distributed applications under the terms of a SLA over a federation of heterogeneous cloud providers. These are (i) the Contrail federation integrating under a common umbrella the resources of different cloud providers relieving the user from the negotiation of the application with each provider; (ii) the SLA component running within the federation core and at the provider layer offering SLA negotiation, management, and enforcement services thanks to the monitoring services; and (iii) the VEP component managing the resources of a cloud provider and offering elastic application deployment within the constraints expressed in a SLA.

3 Federation Concept and Service

A cloud federation is a platform where multiple cloud providers interoperate with each other, creating a service marketplace for users to dynamically match

[1] At the time of writing this paper, Contrail extends and supports only OpenNebula whereas OpenStack is future work. The support of non *Contrail extended IaaS* limits the level of control of the resources, thus the type of guarantees that could be offered to a customer.

their needs with the offers from a subset of providers. In addition to the normal service of a federation, the Contrail federation provides state-of-the-art SLA management of QoS and QoP. Contrail also removes most basic user lock-in constraints offering a uniform approach to actor identification, management policies, costs and application description. In the following, we will explain how a federation differs from other interoperation-oriented approaches, and discuss the main features of the Contrail federation.

In the last few years, the cloud market has grown in terms of its IT market penetration, of the number of players in cloud service provisioning and in terms of differentiation of services between the providers. This variegated cloud offer is an opportunity for companies that want to use public clouds for their applications, but also forces to deal with a non trivial comparison and selection problem.

As the cloud market was still taking shape, different proprietary protocols to describe and rent services did imply a certain degree of user lock-in. While on the one hand we now have much more options to choose from (at the IaaS as well as the PaaS and SaaS levels), on the other hand, and despite strong efforts toward standardization and interoperation [2,7,8,15], it has become increasingly complex to choose between semantically equivalent services with different metrics of cost, performance, reliability, security and elasticity. Complexity arises (i) from the need to match the services, and the service quality level descriptions in different languages, with different protocols used to set up, monitor and steer them, as well as (ii) from the need to plan service utilization in order to optimize a user-specific trade-off of the aforementioned metrics, gathering and exploiting information about a multitude of service providers.

Cloud brokering is nowadays the rising approach to address those issues [17], with both open source (DeltaCloud [4,6]) and commercial solutions (CloudSwitch, Enstratus, Rightscale, Kaavo) already being available. Cloud brokers target interoperation between one user application and one provider, only considering multiple cloud interconnection for the restricted case of cloudbursting from the user's private cloud to a public one. Besides, removing API-related lock-in barriers with respect to providers can result in tying up the user to the broker management interface.

The Contrail approach to federations mainly focuses on provider-level integration of infrastructural services (IaaS) including a heterogeneous population of providers of computation, network, and storage services. As opposed to a broker-based approach, the end-user can exploit advanced inter-provider deployment and coordination, even for a single application, and benefit from state-of-the art federated security as well as federation-ubiquitous SLA mechanisms. These features constitute the basis for higher-level services (PaaS, SaaS) and allow to provide standardized guarantees to the platform user. While cloud brokering relies on and fosters a more competitive cloud resource market, Contrail federations also promote *cooperation* among several providers as a way to open new market opportunities, and fully addressing issues (i) and (ii) outlined before.

To this end, the Contrail federation [5] meets several design constraints. *Security* and *decentralization* are key ones: the many Federation Access Points

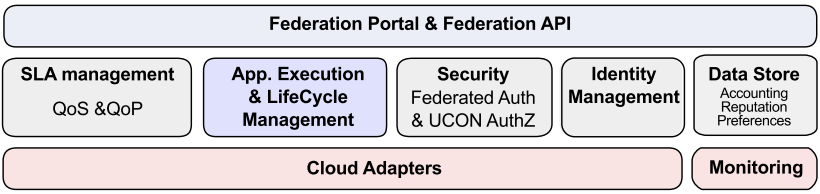

Fig. 2. A Federation Access Point. Each FAP implements the topmost layer of the Contrail abstract architecture, including the Federation Core, its GUI and API.

(FAPs, Fig. 2) implementing the top layer of the Contrail architecture are available to any federation user and can interact with any provider within the federation. Beside leveraging state-of-the-art federated authentication solutions [11] and authorization mechanisms [13], the *Data Store* module of the federation has built-in mechanisms for synchronizing critical data among FAPs. The Data Store provides permanent memory of many information flows (user accounting, reputation monitoring, violation monitoring) each one with its own synchronization policy. Each FAP can be co-located with a cloud provider, making the access to some resources cheaper, but the federation is free to choose resources for an application deployment request from any other provider in the federation, thus completely separating the interests of the cloud provider hosting a FAP and those of the federation/users.

Another key design aim was support for *efficient mapping* of applications under *complex SLA constraints*. The actual mapping of the application over one or more providers is done in the *Application Execution and LifeCycle Management* module (which includes the services of the *Application Lifecycle Manager* shown in Fig. 1), on the ground of several desiderata: the user needs, expressed by the application description and its SLA proposal; information gathered about provider reputation and available resources; the negotiation carried on by the SLA management mechanism. The *SLA management* module inside the FAP architecture is in charge of carrying the topmost level of the hierarchical SLA negotiation, as described in Section 4, trying to achieve SLA contracts with one or more providers that can overall satisfy the user-proposed SLA.

Application mapping exploits a set of heuristics in order to optimize the user-defined trade-off among application metrics, e.g. balancing economic cost and performance levels. To simplify this task, we employ a software toolkit which translates different parts of the application description (namely OVF files, SLA, deployment information) to and from several standard formats and into an in-memory graph representation. Graph-based optimization algorithms can be applied, and the whole structure can be modified, decomposed and translated in many ways to allow for composite deployment over possibly different cloud providers and SLA contracts. The system is designed to also monitor and control the application execution, to possibly perform resource migration or elasticity management.

Finally, achieving strong *Interoperability* and code flexibility was obviously a main issue in the federation architecture design. Toward the user, a classical REST approach has been followed, with tools that allow accessing the federation services via browser and command line. A great deal of features are made easy since the VEP, presented in Section 5, shields the Contrail federation from the specificities of providers about local deployment, monitoring, and SLA management. The FAP, implementing the operational functionalities of the Federation core (shown in Fig. 1), thus has the task to coordinate (via an extendable set of *cloud adapters*) different federation entities (one or more VEP instances, VIN and GAFS resources) for the sake of a specific application.

4 Service Level Agreements

Different cloud providers may have different interfaces to specify requirements, and not all of them provide automatic quotation for a required user configuration. Interacting with different providers by hand to find the best and cheapest one for a given application is a complex and time consuming task. The Contrail federation SLA Management layer automates this provider comparison and selection task and hides to the final user the complexity of interacting with multiple cloud providers.

The Contrail SLA Management layer allows to express user requirements about application QoS in a uniform way. The same SLA syntax is used by all cloud providers in the federation, enabling it to negotiate with and to compare multiple providers. To enable negotiation interoperability of different cloud providers with the Contrail federation, a SLA Management layer is added to each provider. This layer is able to understand the SLA syntax used by the federation and to automatically create SLA offers which will be proposed to the federation on behalf of the provider.

The model of interaction proposed by Contrail is based on multi-level negotiation of SLAs: a user negotiates a SLA with the Contrail federation, which in turn negotiates with several providers to select the best one that can satisfy all user's needs (see Fig. 3).

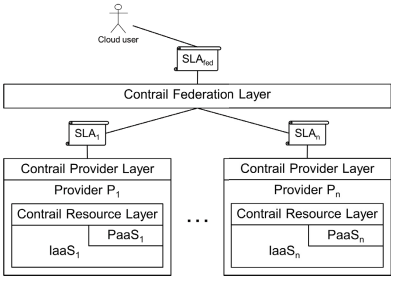

Fig. 3. Multilevel SLA negotiation between user, Contrail federation and multiple cloud providers

The Contrail provider SLA management layer works directly with that specific cloud provider's resources, while SLA management at federation level mediates between the SLA to be agreed with the end user and SLAs to be agreed with different cloud providers. In the negotiation phase the cloud user negotiates with the Contrail federation a pre-configured SLA template, over an arbitrary number of rounds, until they both agree on all the terms. On each round the federation SLA Manager provides to the user the best SLA offer for the given application selected after negotiating with multiple providers.

Each SLA offer is composed by multiple terms expressing various QoS guarantees about resources described by an associated OVF descriptor. The price of each resource is also expressed as a SLA term in the offer. The federation SLA Manager compares the SLA offers returned by each provider and then selects the best one with respect to criteria such as price and performance. To compare the offers returned by each provider, each SLA can be represented as a point in a multi-dimensional space: each term will be a dimension in this space. Coordinates of each SLA-point in this space will be defined to be proportional to the value of each term in that SLA. For terms to be maximised, such as the amount of memory of a VM, direct proportionality is used; while for terms to be minimised, such as price, the proportionality relation is inverse. This model allows the definition of a sort of distance between SLA offers and thus supports comparison between them.

The actual comparison between different SLA offers is done according to a prioritised list of criteria defined by the user, such as price, plus the fact that the specific application will be cpu-bound, memory-bound or I/O-bound. Further criteria for the comparison may include the "reputation" of each cloud provider (calculated by the Contrail federation as a function of the number of SLA violation observed) and user's preferences, either positive or negative, as a filter over providers. For example, a user may have registered a list of preferred cloud providers, or another may have specified a list of providers to be avoided.

SLA terms that will be supported by Contrail include specific characteristics defining the configuration of each VM, such as the amount of memory or the number of virtual cores, but also terms that may affect application performance, such as network bandwidth or the possibility to co-locate in the same host different VMs that must exchange large amounts of data, and even terms important at a legal or privacy level, such as the geographical location.

5 Virtual Execution Platform

The Virtual Execution Platform (VEP) [9,10,12] system offers a uniform way of representing and managing the resources of a cloud provider facilitating the tasks of data center administrators and of the Contrail federation, which interacts with heterogeneous cloud providers. Indeed, these providers might have different means to manage VMs or networks, different image formats that can be deployed on a physical host, different interfaces, or different contextualization methods for the physical resources. As such VEP enables the participation of the cloud provider in the federation seamlessly and it does proper VM contextualization and application lifecycle management. Additionally it publishes application events and metrics to the federation and SLA layers for application's monitoring and SLA enforcement.

Due to its capabilities of hiding the complexity of heterogeneous cloud providers, VEP enables interoperability at federation level through its RESTful interface based on the DMTF's Cloud Infrastructure Management Interface (CIMI) [8] standard. The CIMI model defines a framework for the application life cycle management on a cloud provider infrastructure, with applications generated from an Open

Virtualization Format (OVF) document [7]. VEP extends the CIMI API to support both the federation and the deployment of applications under SLA terms.

The Contrail VEP software is part of the business layer of a cloud provider, as depicted in Fig. 1 and it is installed on the provider data center. Nevertheless it could be also used as an external service, in which case it interacts remotely with the IaaS services through its external API.

Very few propositions currently exist to manage the whole application lifecycle at the IaaS level. The advantage of VEP over existing solutions for managing a cloud infrastructure is the integration of the support for SLAs, eg. with respect to VMware vCloud Suite [18], or the use of the OVF standard format for application description, eg. with respect to CompatibleOne [4]. The rest of this section describes the Contrail VEP features that enable the deployment of elastic and distributed applications, described with the OVF standard, on heterogeneous cloud providers and how these applications could be deployed within the constraints expressed in negotiated SLAs.

Standard OVF Support. Contrail and VEP [9] support standard OVF [7] descriptions without any need for extensions, compared to other propositions such as Claudia [3] which require extensions to manage application elasticity. The absence of extensions improves portability as existing OVF applications can be deployed on IaaS cloud providers without any modification.

IaaS cloud providers should guarantee rapid elasticity to dynamically adapt the allocated resources to the application's requirements. Elasticity support and the deployment over multiple cloud providers are facilitated by the VEP support for implicit as well as controlled deployment modes of OVF applications: in implicit mode, all OVF virtual resources are deployed and started; in controlled mode a user can explicitly specify which and how many OVF virtual resources listed in a *deployment document* are to be deployed. The controlled mode is indeed exploited by the Contrail federation for multi-provider deployments: each provider VEP receives a deployment document defining the OVF resources to deploy. This mode is also exploited to manage elasticity: submitting a new deployment document to an existing application adds the resources listed in this document to the application. To improve the dependability of long running applications, the mapping between the OVF description and the deployed virtual resources can be exploited by VEP to take an application snapshot and to package this snapshot in a new OVF document which can be exploited to re-deploy the application, possibly on another provider.

SLA Terms Support. The Contrail federation and the user negotiate the SLA associated to an application, which is then deployed by VEP respecting those negotiated constraints [12]. The SLA support during the deployment is achieved with the definition of the Contrail Constrained Execution Environments (CEEs), which can be derived from a negotiated SLA or made available as templates ready to be instantiated by users. The CEE on which a new application is to be deployed can be specified by the user, the federation acting on behalf of the user, or from default rules. It is possible to deploy multiple

applications on the same CEE, for instance applications sharing the same virtual network or storage.

A CEE defines a virtual infrastructure made of resource handlers and constraints where user applications are deployed and managed. Fig. 4 shows the mapping between the resources described in the application OVF document and the CEE resource handlers specifying the constraints which should be respected for the deployment of each resource. Each resource handler specifies the physical resources to be allocated for each virtual resource (virtual machine, storage, or network) instantiated in the infrastructure. Different types of constraints are supported in VEP concerning performance, security, placement or the number of virtual resources which can be allocated. For instance, constraints can specify relations between resources, such as affinity to allocate resources close to each other in order to improve interactions, or anti-affinity to increase dependability, for instance to place virtual machines on different data centers. The CEE also defines the monitoring configurations, which are then used by the provider and the federation to evaluate whether a SLA is enforced. New deployment requests for additional resources of an application can then be automatically generated by the SLA enforcement services in reaction to performance indicator deviations, or directly requested by the user. Adding new resources to the application does not necessitate any SLA re-negotiation as long as the CEE constraints are respected.

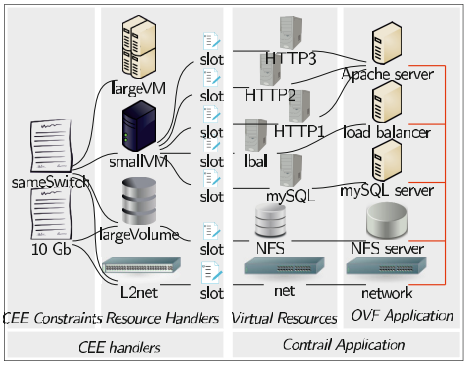

Fig. 4. Constrained Execution Environments and OVF mapping

VEP Features. In the previous paragraphs, we have presented the VEP support for SLAs through CEEs and DMTF's OVF standard [7] for application description (without any extension) as well as how VEP deals with elastic applications. Other features of the Contrail VEP not discussed in this section are: partial application deployment to allow multi-provider application deployment from the federation layer; advance reservation to guarantee resource provisioning in the future; application snapshots to improve dependability.

6 Conclusion

The Contrail project provides an open-source integrated approach to virtualization at the IaaS and PaaS levels [1]. This chapter discussed the challenges and the approach in Contrail to address interoperability issues and SLA-aware deployment of distributed applications across a federation of heterogeneous cloud providers. We described the major components of the Contrail architecture and

we outlined the key features and the design of the Contrail federation, SLA management, and VEP. These components allow dynamic transparent leasing of resources from multiple sources and ease the user access to cloud services. In a nutshell, Contrail ameliorates the effectiveness of the pay-as-you go approach and increases the end-user freedom in the provider selection. Moreover, strong SLA guarantees support both provider competition (on prices) and collaboration (provider aggregation) to access new market segments.

Contrail is still under intense development. More advanced policies and mechanisms are planned to support distributed deployment, sophisticated SLA management policies, and tighter integration with other Cloud providers (e.g. OpenStack, Amazon).

Acknowledgments. This work is partially funded by the FP7 Programme of the European Commission in the context of the Contrail project under Grant Agreement FP7-ICT-257438. The authors would like to thank the Contrail Consortium members.

References

1. Contrail project, `http://contrail-project.eu/` code available on, `http://contrail.projects.ow2.org/xwiki/bin/view/Main/WebHome`
2. Cloud Data Management Interface (CDMI) Version 1.0.2 (June 2012), `http://www.snia.org/cdmi`
3. Morfeo Claudia, `http://claudia.morfeo-project.org/`
4. CompaptibleOne, `http://www.compatibleone.org/`
5. Coppola, M., Dazzi, P., Lazouski, A., Martinelli, F., Mori, P., Jensen, J., Johnson, I., Kershaw, P.: The Contrail approach to cloud federations. In: International Symposium on Grids and Clouds (ISGC) (2012)
6. DeltaCloud, `http://deltacloud.apache.org/`
7. DMTF: Open Virtualization Format Specification (2010), `http://www.dmtf.org/standards/ovf`
8. DMTF: Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol, An Interface for Managing Cloud Infrastructure, v1.0.1 DSP0263 (2012), `http://www.dmtf.org/standards/cloud`
9. Harsh, P., Dudouet, F., Cascella, R., Jegou, Y., Morin, C.: Using open standards for interoperability issues, solutions, and challenges facing cloud computing. In: 8th International Conference on Network and Service Management (CNSM) (2012)
10. Harsh, P., Jegou, Y., Cascella, R.G., Morin, C.: Contrail virtual execution platform: Challenges in being part of a cloud federation. In: Abramowicz, W., Llorente, I.M., Surridge, M., Zisman, A., Vayssière, J. (eds.) ServiceWave 2011. LNCS, vol. 6994, pp. 50–61. Springer, Heidelberg (2011)
11. IETF OASIS WG: RFC6749 – The OAuth 2.0 Authorization Framework (2012)

12. Jegou, Y., Harsh, P., Cascella, R., Dudouet, F., Morin, C.: Managing OVF applications under SLA constraints on contrail virtual execution platform. In: 8th International Conference on Network and Service Management (CNSM) (2012)
13. Lazouski, A., Martinelli, F., Mori, P.: A prototype for enforcing usage control policies based on XACML. In: Fischer-Hübner, S., Katsikas, S., Quirchmayr, G. (eds.) TrustBus 2012. LNCS, vol. 7449, pp. 79–92. Springer, Heidelberg (2012)
14. mOSAIC Web Site, http://www.mosaic-cloud.eu
15. OGF Consortium OCCI WG (2013), http://occi-wg.org/
16. Pierre, G., Stratan, C.: ConPaaS: a platform for hosting elastic cloud applications. IEEE Internet Computing 16(5), 88–92 (2012)
17. Sundareswaran, S., Squicciarini, A., Lin, D.: A brokerage-based approach for cloud service selection. In: IEEE Fifth International Conference on Cloud Computing (CLOUD) (2012)
18. VMware: VMware vCloud Suite for Cloud Computing & Cloud Management (January 2013), http://www.vmware.com/products/datacenter-virtualization/vcloud-suite/overview.html
19. Wieder, A., Bhatotia, P., Post, A., Rodrigues, R.: Orchestrating the deployment of computations in the cloud with conductor. In: 9th USENIX conference on Networked Systems Design and Implementation (NSDI) (2012)