Cloud and Network facilities federation in BonFIRE

David García-Pérez[1], Juan Ángel Lorenzo del Castillo[2], Yahya Al-Hazmi[3], Josep Martrat[1], Konstantinos Kavoussanakis[4], Alastair C. Hume[4], Celia Velayos López[5], Giada Landi[6], Tim Wauters[7], Michael Gienger[8], and David Margery[9]

[1] Atos Research and Innovation Group, Barcelona, Spain
[2] Hewlett-Packard Laboratories, Bristol, United Kingdom
[3] Chair of Next Generation Networks, Technical University Berlin, Berlin, Germany
[4] EPCC, University of Edinburgh, King's Buildings, Edinburgh, United Kingdom
[5] Distributed Applications and Networks Area, i2CAT Foundation, Barcelona, Spain
[6] Nextworks s.r.l., Pisa, Italy
[7] Department of Information Technology, Ghent University, iMinds, Ghent, Belgium
[8] High Performance Computing Center Stuttgart (HLRS), Stuttgart, Germany
[9] INRIA Rennes - Bretagne Atlantique research center, Rennes, France

Abstract. In recent years we have seen how Cloud Computing is changing the way of doing businesses and how services are delivered over the Internet. This disruption is a major challenge for Service Providers and Independent Software Vendors when creating new services and software applications for the Cloud. BonFIRE[10] offers a federated, multi-site cloud testbed to support large-scale testing of applications, services and systems. This is achieved by federating geographically distributed, heterogeneous clouds testbeds where each exposes unique configuration and/or features while giving to the experimenters (users) an homogeneous way to interact with the facility. All those testbeds are controlled by a central set of services commonly denominated "Broker". Additionally, BonFIRE is federated with different network facilities like the Virtual Wall, FEDERICA and AutoBAHN to provide high-level interfaces to network control functionality, in order to simulate diverse network QoS scenarios, enabling vertical federation.

Keywords: Multi-cloud, Federation, Future Internet, Services, Testbed, Bandwidth on Demand, Network QoS

1 Introduction

Cloud computing is steadily gaining significance not only in the field of IT infrastructure and service hosting and operation, but also in telecommunication services. This encourages cloud providers to develop new technologies to enhance their services in order to increase customer satisfaction on the one hand; and

[10] http://www.bonfire-project.eu

to promote strategies to build new business models on the other. In this market, cloud service providers, having different kinds of resources offering different levels of service quality, succeed not only by competition but also by collaboration between each other. There are several advantages of such collaborations for both cloud service suppliers and cloud service customers. Among the advantages for cloud service providers are complementarity of the offered resources to improve resource utilization; and combination of services in order to offer efficient end-to-end solutions. On the other hand, customers have the ability to combine resources and services from different cloud computing providers to create their own environments [1].

Cloud service providers might offer different types of services, with different APIs, various delivery conditions and prices, etc. Therefore, collaboration across cloud infrastructures that interoperate requires mechanisms for resource federation. Federated resources have to be managed at the federation level so as to be offered to users in as uniform a fashion as possible. Several resource management mechanisms are to be specified and standardized at the federation level: resource description, discovery, reservation, provisioning, orchestration, monitoring, and release. Furthermore, mechanisms are required for authentication and authorization, resource control and information exchange across cloud infrastructures.

To this end, cloud federation [2–4] architectures should be designed to include the necessarily mechanisms and fulfill all collaboration requirements. The different Cloud federation architectures proposed in the literature [5] can be classified into three different models as follows:

- **Cloud aggregation**: [6] different partners cooperate to integrate their cloud infrastructures into a unique virtual facility. This architecture usually requires a higher degree of coupling.
- **Cloud bursting or Hybrid Clouds**: [7, 8] combines the existing on premise infrastructure (private cloud) with resources of one or more public clouds.
- **Brokering**: [6] employs a central "broker" component that intermediates between user requests and several Cloud infrastructures.

There are many challenges related to cloud federation that affect interoperability (e.g. different APIs) and portability (e.g. different VM formats) [9].

BonFIRE [10] offers a Future Internet, Federated testbed, that supports large-scale testing of applications, services and systems over multiple, geographically distributed, heterogeneous Cloud and network testbeds. The design of BonFIRE is based in four pillars: Observability, Control, Advanced Features, and Easy of Use. The BonFIRE infrastructure gives the experimenters the ability to control and monitor the execution of their experiments to a degree that is not found in traditional Cloud facilities. The available testbeds allow the evaluation of cross-cutting effects of converged service and network infrastructures. In order to do so, the testbeds expose a set of capabilities beyond those normally provided by production cloud services in a commercial environment, enabling what we define as "Research by Experimentation". In terms of classification, BonFIRE employs a central broker to expose a homogeneous API and interact

with the testbeds. Additionally, BonFIRE includes and federates with private and public clouds and can thus be used to model cloud-bursting scenarios.

The remainder of the paper is structured as follows; Section 2 introduces the different facilities federated and the reason why Brokering was selected as federation solution. Section 3 presents the BonFIRE federation architecture. Section 4 introduces the different tools developed to homogenize access to all Cloud sites. Section 5 introduces the different set of experiments used to validate the platform. Finally, Section 6, presents the conclusions and future work.

2 BonFIRE Federated Clouds and Networks

BonFIRE allows users to execute experiments that use compute, network, storage, site-link and router resources. The experiments are controlled by a set of central services, collectively called the Broker, that interact with the testbeds. In this way, the BonFIRE users can test their applications, services, or systems in different Cloud configurations (e.g. deploy across multiple multi-clouds, or experiment with different Cloud managers available through BonFIRE) using a single, homogeneous interface.

At the core of BonFIRE are seven Cloud testbeds that are located at EPCC (UK), INRIA (France), PSNC (Poland), HLRS (Germany), iMinds (Belgium), Wellness Telecom (Spain) and HP (UK) (fig. 1). All testbeds present different features regarding structure, networking, and resources. Users can also create resources on Amazon EC2 using the BonFIRE interface.

Integration of the BonFIRE Cloud facilities and their interconnection with external facilities is key in BonFIRE. Horizontal (i.e. of similar functionality) integration is achieved through the integration of the different facilities as well as Amazon. Additionally, BonFIRE offers vertical integration of Cloud and network facilities. This enables cross-layer testing to propagate service-level requirements down to the connectivity levels. The iMinds Virtual Wall facility allows experimentation using controlled, emulated networks. BonFIRE is also interconnected with two external facilities to provide vertical integration at the network level. These are the GÉANT AutoBAHN Bandwidth-on-Demand (BoD) system; and the FEDERICA computing network architectures e-infrastructure.

3 BonFIRE Federation Architecture

BonFIRE provides resource level operations through an implementation of the Open Cloud Computing Interface[11] (OCCI) standard. The OCCI was designed for remote management interactions with cloud computing infrastructures. It is an open protocol that is resource-vendor independent, cloud-manager neutral, and can be extended.OCCI acts as the user API for the service lifecycle management of BonFIRE-supported resources. Originally limited to compute, storage,

[11] http://www.occi-wg.org

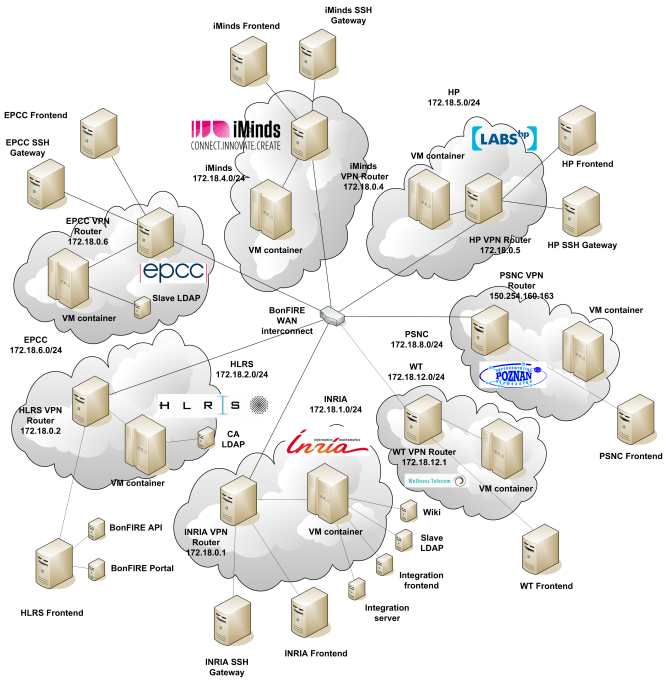

Fig. 1. BonFIRE Cloud Federation Infrastructure.

network resources, BonFIRE has extended OCCI to allow operations to site-links and routers, as required by the AutoBAHN and FEDERICA testbeds.

The BonFIRE implementation of OCCI enables defining different kinds of resources with various possible configurations. For instance, a user can create a compute resource and configure it on demand with one of multiple instance types (lite, small, medium, large, or custom), with one or multiple disks (disk image and external storages), with one or more network interfaces, etc. The BonFIRE OCCI supports several management operations and various states of resources. For instance, the possible management operations of a compute resource include: create, stop, resume, shutdown, and delete.

The Resource Manager (RM) (fig. 2) is the lowest layer of the BonFIRE architecture that may be accessed by the end-users, providing the entry point for the users to interact with BonFIRE at resource level. The RM exposes the BonFIRE OCCI interface and maintains the current set of experiments and all the resources associated to them. User operations on non-experiment resources are passed to the Enactor and then on to the appropriate interconnected testbed. User operations on experiment resources are executed at the RM level.

Due to the heterogeneity of BonFIRE Cloud testbeds, the mapping between BonFIRE OCCI and the testbed native interfaces was implemented in different layers. The role of the Enactor is to hide from the RM and from the OCCI the technical details of how to communicate with each specific testbed or external

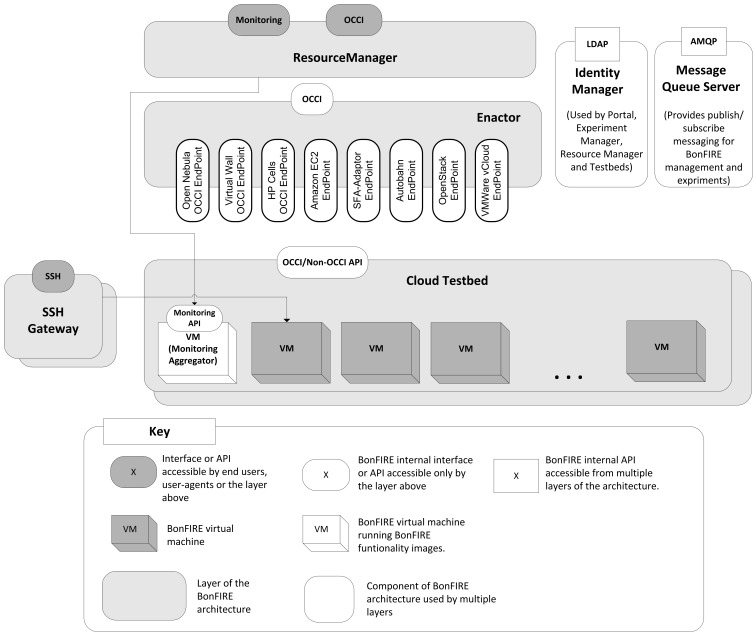

Fig. 2. Lowest level BonFIRE architecture.

facility. The Enactor receives OCCI requests from the RM and adapts them to a suitable testbed format. When the Enactor gets the information back from the testbed, it transforms the reply again into BonFIRE OCCI.

Canonical BonFIRE images are generated centrally to provide the Debian squeeze Linux distribution in different sizes for the Xen, KVM and ESX hypervisors. The BonFIRE OCCI supports contextualisation at the time of the creation of the VMs and all images also support this; for instance, software can be installed automatically after booting, and also parameters may be passed directly into the virtual machine on instantiation. As a federated infrastructure, every BonFIRE site receives the same contextualisation information in the VM creation request, but they are free to use it according to each site's own particularities.

BonFIRE offers a private address space for each infrastructure provider, e.g. the whole network subnet 172.18.0.0/16. Due to that, all the created VMs are assigned to a dedicated private network offered by a local provider in its BonFIRE address range. In order to connect the several VM instances at different locations, a BonFIRE-wide VPN was established, which allows communication between separated virtual machines and internal BonFIRE services. For the VPN network installation, the tinc software was used. The tinc service allows the creation of a decentralized network, so there is no single point of failure if one instance loses its connection. Figure 1 gives a brief overview of the whole BonFIRE infrastructure and shows the different IP ranges of the providers.

3.1 User Management

Identity management and authorization is done centrally in BonFIRE. On the one hand, a central LDAP database stores user credentials (username, password and ssh public key) and group membership. On the other, an authorization service stores site access rights as wells as usage limits and peak resource usage for each group. Indeed, groups are a key BonFIRE functionality, that enables efficient collaboration. Group membership is managed by the Principal Investigator of an experiment.

Interactions with BonFIRE services are authenticated against the LDAP database using HTTP basic authentication over ssl. The first service called must authenticate the user, and will carry user information in a dedicated HTTP Header when calling other services. This information is trusted by the service called if and only if the requester can themselves be authenticated using client-side authentication permitted by ssl. In particular, all testbeds in the federation must be configured to authenticate central servers and trust user and group information sent to them using HTTP headers.

The Resource Manager is in charge of doing all authorization by calling the accounting service to check, based on group membership, whether a given user can create or access resources. It will also create temporary credentials and add these, as well as the public keys of all users allowed to access resources, into the context information carried by the OCCI requests. Temporary credentials are used by code running inside the resources that needs to call the BonFIRE API.

Access to resources is secured through SSH. SSH gateways are operated by BonFIRE sites to allow access to the BonFIRE VPN. A classical master-slave configuration is used for LDAP, with slaves on all gateways. This approach avoids a single point of failure, relieves the load on the central database and propagates changes very quickly. VMs are configured with the public keys of users allowed to access them. This allows users to connect to any gateway and, from there, to any VM in the BonFIRE VPN.

3.2 Cloud Testbeds

Depending on Cloud Manager technology, BonFIRE currently offers the following five different types of Cloud testbed:

- **OpenNebula**: The currently operated OpenNebula version 3.6 includes an implementation of an OCCI server based on the OCCI draft 0.8. In order to provide valuable cloud functionality, additional fields of use were added by the BonFIRE developers in order to improve and extend the whole OCCI software stack of OpenNebula.
- **HP Cells**: The OCCI at HP Cells is completely stateless, so there is nothing that can get out of sync with the BonFIRE cental services or with the Cells state. BonFIRE-specific information such as groups, users, etc. are not stored, so the information retrieved on each request from the Enactor is filtered according to the permissions of the requesting user. This OCCI server was implemented specifically to support the BonFIRE project.

- **Virtual Wall**: The Virtual Wall emulation testbed is not a typical cloud environment, as it lacks the ability to dynamically add computes to an already running experiment. However, its functionality offers a first step to bridge the gap between network and cloud experimentation. The Virtual Wall offers the same OCCI resources as the other testbeds in BonFIRE, but their implementation is very different due to its underlying framework, Emulab. For instance, the Virtual Wall maps Compute resources to physical nodes, which prevents virtualisation, but allows the experimenter to take full control of the hardware. In response to the need of experimenters to share larger amounts of storage between different Compute resources, the Virtual Wall implements a notion of shared storage based on the Network File System (NFS).
- **VMWare vCloud**: vCloud does not offer by default an OCCI API. Similar to the case of HP Cells, an OCCI server was developed inside the BonFIRE project that interacts with the VMWare vCloud Director API to support VMWare Cloud facilities. The OCCI server is stateless, all the requests coming from the Enactor are translated and mapped to the proprietary API.
- **Amazon EC2**: The Amazon EC2 endpoint at the Enactor makes use of the API that Amazon provides to connect remotely to their Cloud services. The endpoint only allows to manage two kind of resources: storages and computes that are mapped to their Amazon equivalents, volumes or images and instances. In order to deal with the large volume of information returned, BonFIRE caches some OCCI queries in the Enactor, like listings of EC2's numerous storage resources.

3.3 Network testbeds

BonFIRE supports experimentation and testing of new scenarios from the services research community, focused on the convergence of services and networks. In order to support network experimentation, BonFIRE is federated with the iMinds Virtual Wall testbed; and is interconnected with two network facilities: FEDERICA and AutoBAHN.

The most distinctive features of the iMinds Virtual Wall are related to its networking capabilities. Whereas the other BonFIRE testbeds only provide a best-effort variant of the Network resource, the Virtual Wall implements three different types of Network resources: Default Networks that provide basic connectivity between two or more Computes; Managed Networks that provide controllable QoS (parameters that can be adjusted are bandwidth, packet loss rate and delay) over the network links; and Active Networks, that, on top of the functionality of Managed Networks, also provide the possibility to control the background traffic (UDP and TCP connections with dynamically adjustable packet size and throughput) on a network link. These networks provided by the Virtual Wall are emulated, using the Emulab software.

FEDERICA is an infrastructure composed of computers, switches and routers connected by Gigabit Ethernet circuits. Through the Slide-based Federation Architecture (SFA) paradigm, FEDERICA offers to BonFIRE experimenters iso-

lated network slices by means of virtualizing routers. This interconnection is aimed to help experimenters to investigate application performance through better control of the underlying network. The following changes were carried out in BonFIRE to incorporate these new network resources: the router resource was added to the BonFIRE OCCI and the network resource was enhanced with two new attributes: network link and vlan. Finally, since FEDERICA offers an SFA interface as federation API, it was necessary to implement an SFA endpoint at Enactor level. The FEDERICA SFA interface expects a unique XML request, where all the slice resources and their configuration are specified. This differs from the BonFIRE architecture, where each resource is requested in a single OCCI call. The main function of the BonFIRE SFA endpoint is to transform BonFIRE's OCCI information model to the SFA information model.

The federation between BonFIRE and the AutoBAHN beta-functionality offered by the GÈANT facility allows the experimenters to request QoS guaranteed network connectivity services between VMs deployed on EPCC and PSNC testbeds. Overcoming the Best Effort limitation of the public Internet, dedicated network services can be established on demand for each experiment, with guarantees in terms of bandwidth, reduced jitter and service reliability. This option is fundamental to offer a controlled connectivity between VMs, so that the experimenters can evaluate the performance of their applications in environments able to emulate a variety of network conditions.

In BonFIRE, a BoD service is represented by a new type of OCCI resource: the site_link. Once the resource is created, it can be used to connect two networks created in the BonFIRE sites at the edge of the site_link: the traffic between the VMs attached to these networks is routed through the dedicated service. The processing of the OCCI requests for site_link resources is managed at the enactor through a dedicated AutoBAHN end-point that is in charge of translating the OCCI specification into the AutoBAHN BoD service format. The Enactor endpoint acts as an AutoBAHN client.

4 Practical Experimentation Tools

In response to user-demand, BonFIRE exposes various tools and interfaces to its users. The main API is based on OCCI, an interface familiar to the target users, i.e. Cloud application developers. However, interacting with OCCI may be very cumbersome as the size of the deployments grows, and the format is not easily human-readable.

In order to increase usability, BonFIRE offers other client tools, building on the BonFIRE API. These include: an intuitive web Portal to interact graphically with all components; a BonFIRE Experiment Descriptor written in JSON or OVF; a general-purpose RESTful client that uses the Restfully library to allow interaction via Ruby scripting; and a set of Command Line Interface Tools that can be used interactive or programmatically. Extensive, tutorial-based user documentation accompanies the BonFIRE tools.

The feedback of the experimenters to the diversity of available tools was very positive. They found it very easy to start creating experiments with the web Portal and progress later to more complex scenarios in which they created their own programs based on the available tools to create and access their experiments. The BonFIRE Portal is key to this progression, as it includes intuitive builders for OCCI and for the experiment descriptor.

5 Validation of the platform

Validation of the federated cloud and network platform was designed into several phases. In the early stages of the project, three of the core partners of BonFIRE performed Cloud experimentation in several aspects: from application benchmarking in Cloud Computing to study of malicious paterns in applications [11, 12]. In a second phase, we performed two Open Calls, in the first one four independent companies or research groups performed experiments using BonFIRE with topics ranging from service composition, virtual clusters, security, or multimedia [13–15], in the second one, five new experiments ranging from Cloud elasticity, QoS in federated Clouds, testing of commercial home security and management solutions, multimedia interfaces, or orchestration between users and cloud resources[12]. Finally, BonFIRE has just started its Open Access initiative for new users to come and use for free the platform.

6 Conclusions and Future Work

BonFIRE enables Research by Experimentation founded on federation of heterogeneous, cloud and network testbeds and facilities. The BonFIRE Cloud federation model is based on a central broker component that interacts between the user requests, the experimenters, and the different infrastructures. This broker offers to the users an OCCI-based interface that exposes all the Cloud and network features as resources.

The BonFIRE interface exposes to the users an homogeneous view of the heterogeneous testbeds it encompasses. Still, the interface exposes the different characteristics that make each federated site unique. Additionally, user management is centralized; all sites are in one unique WAN network that eases communication between VMs located at different Clouds; and the VMs in all federated Clouds, independently of the employed Cloud Manager, can be contextualized in the same way.

The BonFIRE project is still ongoing. In terms of federation, BonFIRE is working towards a cache facility, which will accelerate resource listings from sites. Unlike other BonFIRE functionalities, this is implemented with tight coupling between sites and central services; the sites are enhance to push events about

[12] At the time of writting, those experiments were still being executed in the BonFIRE platform, more information about them it is is available in the project webpage: http://www.bonfire-project.eu.

the state of the resources in the BonFIRE Message Queue. BonFIRE is also investigating OpenStack as a potential target Cloud Manager.

BonFIRE is open for free use by organizations and individuals not within the consortium. Over 2013 and 2014 BonFIRE expects to engage an increasing number of experimenters accessing the cloud federated facility.

Acknowledgments. BonFIRE is funded by the EU 7th Framework Programme (FP7/2007-2013) under grant agreement number 257386 and 287938.

References

1. Al-Hazmi, Y. et al.: A Monitoring System for Federated Clouds. Proceedings of the 1st IEEE International Conference on Cloud Networking, 68–74, 2012.
2. Keahey, K. et al.: Sky Computing. IEEE Internet Computing, vol 13, no 5, 43–51, 2012.
3. Rochwerger, B. et al.: Reservoir-When One Cloud is Not Enough. Computer, 44–51, March 2011.
4. Vandenberghe, W. et al.: Architecture for the Heterogeneous Federation of Future Internet Experimentation Facilities. Future Network and Mobile Summit, July 2013.
5. Ferrer, A.J. et al.: OPTIMIS: A Holistic Approach to Cloud Service Provisioning. Future-Generation Computer Systems, vol. 28, 66–77, 2012.
6. Tordsson, J. et al.: Cloud Brokering Mechanisms for Optimized Placement of Virtual Machines Across Multiple Providers. Future Generation Compute Systems, vol 28, no. 2, 358–367, 2012.
7. Sotomayor, B. et al.: Virtual Infrastructure Management in Private and Hybrid Clouds. IEEE Internet Computing, vol. 13, no. 5, 14–22, 2009.
8. Moreno-Vozmediano, R. et al.: Multicloud Deployment of Computing Clusters for Loosely Coupled MTC Applications. IEEE Transactions in Parallel and Distributed Systems, vol. 22, no. 6, 924–930, 2011.
9. Petcu, D.: Portability and Interoperability Between Clouds: Challenges and Case Study. Lecture Notes in Computer Science, 66–74, 2011.
10. Hume, A. et al.: BonFIRE: A Multi-cloud Test Facility for Internet of Services Experimentation Proceedings of the 8th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, 1–16, 2012.
11. Ragusa, C. et al.: A Framework for Modeling and Excuting of Infrastructure Contention Experiments. MERMAT. 2nd Internation Workshop on Measurement-based Experimental Research, Methodology and Tools. FIA 2013.
12. Engen, V. et al.: Predicting Application Performance for Multi-Vendor Clouds using Dwarf Benchmarks. 13th International Conference on Web Information System Engineering WISE 2012.
13. Gomez, A. et al.: Experimenting Virtual Clusters on Distributed Clound environemtns using BonFIRE. FIRE Engineering Workshop, 2012.
14. Wajid, U. et al.: Testing Optimization in Service EcoSystems (TEOS). IEEE International Conference on Web Services, 2012.
15. Naqvi, S. et al.: Analysing Impact of Scalability and Heteroneneity on the Performance of Federated Cloud Security. 4th IEEE International Workshop on Security in e-Science and e-Research, in conjuction with The 2012 IEEE International Conference on Trust, Security, and Privacy in Computing and Communications, 2012.