BonFIRE: A Multi-cloud Test Facility for Internet of Services Experimentation

Alastair C. Hume[1], Yahya Al-Hazmi[2], Bartosz Belter[3], Konrad Campowsky[4], Luis M. Carril[5] Gino Carrozzo[6], Vegard Engen[7], David García-Pérez[8], Jordi Jofre Ponsatí[9], Roland Kűbert[10], Yongzheng Liang[11], Cyril Rohr[12], and Gregory Van Seghbroeck[13]

[1] EPCC, University of Edinburgh, King's Buildings, Mayfield Road, Edinburgh, UK
`A.Hume@epcc.ed.ac.uk`,
[2] Chair of Next Generation Networks, Technical University Berlin, Berlin, Germany
[3] Poznan Supercomputing and Networking Center (PSNC), Poznan, Poland
[4] Next Generation Network Infrastructures, Fraunhofer FOKUS, Berlin, Germany
[5] Centro de Supercomputación de Galicia (CESGA), Santiago de Compostela, Spain
[6] Nextworks s.r.l., Pisa, Italy
[7] IT Innovation Centre, Southampton, UK
[8] Atos Research and Innovation Group, Barcelona, Spain
[9] Distributed Applications and Networks Area, i2CAT Foundation, Barcelona, Spain
[10] High Performance Computing Center Stuttgart (HLRS), Stuttgart, Germany
[11] Computing Center, University of Stuttgart, Stuttgart, Germany
[12] INRIA Rennes - Bretagne Atlantique research center, Rennes, France
[13] Department of Information Technology, Ghent University, IBBT, Ghent, Belgium

Abstract. BonFIRE offers a Future Internet, multi-site, cloud testbed, targeted at the Internet of Services community, that supports large scale testing of applications, services and systems over multiple, geographically distributed, heterogeneous cloud testbeds. The aim of BonFIRE is to provide an infrastructure that gives experimenters the ability to control and monitor the execution of their experiments to a degree that is not found in traditional cloud facilities.

The BonFIRE architecture has been designed to support key functionalities such as: resource management; monitoring of virtual and physical infrastructure metrics; elasticity; single document experiment descriptions; and scheduling.

As for January 2012 BonFIRE release 2 is operational, supporting seven pilot experiments. Future releases will enhance the offering, including the interconnecting with networking facilities to provide access to routers, switches and bandwidth-on-demand systems. BonFIRE will be open for general use late 2012.

Key words: Multi-cloud, Future Internet, Internet of Services, Testbed, Bandwidth on Demand

1 Introduction

BonFIRE offers a multi-site cloud testbed that supports large scale testing of applications, services and systems over multiple, geographically distributed, heterogeneous cloud testbeds. BonFIRE targets the Internet of Services community and offers a test infrastructure that is ideal for performing experiments relating to distributed applications and services. BonFIRE's aim is not to provide a production environment for cloud applications, but instead to provide an infrastructure that gives experimenters the ability to control and monitor the execution of their experiments to a degree that is not found in traditional cloud facilities.

By targeting the Internet of Services community, BonFIRE aims to extend the reach of Future Internet experiment testbeds into the area of distributed applications and services. To further appeal to that community, BonFIRE adopts a cloud-based model that is familiar to many Internet of Services experimenters.

One of the key features of BonFIRE is to give experimenters the ability to control some of the many variables that affect the performance of distributed applications. For example, BonFIRE allows users to control network quality-of-service parameters such as latency, delay and packet loss using IBBT's Virtual Wall [1] network emulation facility. Further control of network performance between geographically distributed infrastructures is anticipated through future interconnection with FEDERICA [2] and GÉANT AutoBAHN [3, 4].

Apart from network characteristics there are many other variables that can affect performance of an application running on a cloud infrastructure; examples include the load of the VMs running on the physical machine as well as disk and memory contention. Where BonFIRE cannot control variables, it provides low-level infrastructure metrics that can be used to monitor the infrastructure at a level that is typically hidden from users of cloud systems. Examples of infrastructure metrics include the load on a physical CPU, network throughput and disk I/O metrics. Experimenters will be able to use this low-level infrastructure monitoring data to understand, and account for, the impact of these variables on their experiment results.

By adopting a cloud-model to access the multi-cloud facility, BonFIRE provides a convenient resource-level interface for managing compute, network and storage resources on any of the BonFIRE testbeds. Thus BonFIRE supports geographically distributed experiments over multiple heterogeneous cloud testbeds including the ability to elastically adjust the resources used during an experiment. BonFIRE also supports a higher experiment-level interface that provides users with the ability to describe the whole experiment in a document. Additionally, BonFIRE provides built-in support for commonly used experiment functionality such as monitoring collectors and publishers.

2 The Testbeds

At the core of BonFIRE are five geographically distributed cloud testbeds that together offer 350 cores, with 700GB of RAM and 30TB of storage. An additional 2,300 multi-core nodes can be added to BonFIRE on specific request. Additionally, BonFIRE is working to interconnect with several external facilities to enhance the offering.

2.1 Core Testbeds

The BonFIRE core testbeds can be divided into two groups: OpenNebula based testbeds located at EPCC (UK), INRIA (France) and HLRS (Germany) and non-OpenNebula based testbeds located at IBBT (Belgium) and HP (UK). All testbeds are different from one another regarding structure, networking and resources. This heterogeneity is a key feature of the BonFIRE system.

The common point of the testbeds of EPCC, INRIA and HLRS is that they use OpenNebula as a cloud management solution and are currently running the Xen hypervisor. Apart from that, the sites are quite different: EPCC provides 96 cores on two nodes, INRIA 36 cores on 9 nodes and HLRS a total of 44 cores: 20 cores on five nodes at 2GB RAM and 24 cores on six nodes at 8 GB RAM.

IBBT's Virtual Wall is based on the Emulab network emulation software. Tight integration of Emulab's capabilities and the BonFIRE functionality makes it very much suited for network related experiments. The IBBT Virtual Wall consists of 100 nodes, of which 8 are available permanently to BonFIRE users throughout the project's lifetime. Each node has 4 cores at 4GB RAM and 4 to 6 Gigabit Ethernet interfaces, all of them connected to a non-blocking switch.

The HP infrastructure provides a virtual machine execution service. The virtual machines will be running in a Xen environment, augmented with network and storage virtualization.HP's infrastructure is set up with the current stable release of HP Cells [5] as a Virtual Infrastructure Manager. The HP facility provides 32 4-core nodes permanently dedicated to BonFIRE with access to up to 96.

2.2 On-Request Resources

On-request resources are resources that are not part of the BonFIRE facility on a permanent basis, but that are provided to users for large- scale experiments. Typically, a substantially larger collection of resources is available on-request than is offered by the pemanent infrastructure. The operation and investments of these facilities are too high to be exclusively provided for experiments but are operated as production level facilities. In order to allow large scale testing these resources can be made available for a specific experiment for a limited time.

As with the permanent resources, the kind and amount of on-request resources differs from site to site. In total, there will be around 2,200 cores available through on-request resources across all sites.

2.3 External Facilities

Integration of the internal multi-site cloud facilities and their interconnection with external (FIRE) facilities is key in BonFIRE. Horizontal integration is achieved through the integration of the different testbeds facilities as described in Section 2.1. Conversely, the vertical integration of the facilities (Cloud-and-Network) enables cross-layer testing to propagate service-level requirements down to the connectivity levels. BonFIRE will initially interconnect with two external facilities to provide vertical integration to the network level. These are GÉANT's Bandwidth-on-Demand (BoD) system and FEDERICA's slice-based infrastructure that gives access to routes, switches and servers.

A major challenge in the vertical federation of cloud and network is the need to provision on-demand network connectivity services among multiple cloud resources and sites by interfacing to multi-domain heterogeneous production networks. In this framework, BonFIRE integrates with Bandwidth- on-Demand (BoD) provisioning systems that have been developed by network operators to provide their end-users with dynamic and efficient systems for their on-demand network connectivity services. Due to the research-driven mission of the BonFIRE project and the positioning of the overall FIRE initiative, national Research and Education networks (NREN) and GÉANT are the preferential connectivity providers for BonFIRE. It is planned to inter-connect some of the available sites (PSNC, EPCC, HLRS) through their NRENs - PIONER, JANET and DFN respectively - and through the GÉANT network. This will enable to perform and demonstrate a cloud-to-network proof-of- concept prototype utilizing GÉANT BoD service implemented with AutoBAHN for establishing and monitoring dedicated high capacity connections between sites.

FEDERICA [2] was a European Project of the 7th Framework Program. Its main goal was to deploy an e-Infrastructure for researchers on Future Internet. Resources of this e-Infrastructure can be allocated to independent slices and assigned to different experimenters allowing them to have the complete control of the resources in their slice. This infrastructure is physically distributed around the whole of Europe and it is composed by routers, switches and servers. By interconnecting with a subset of FEDERICA's infrastructure, BonFIRE users will be able to perform new kinds of experiments over a controlled network. This includes configuring network topologies and routing protocols and setting quality of service (QoS) parameters.

3 Architectural Principles

The following architecture principles have been used to drive the design of the BonFIRE architecture. By explicitly specifying these key principles we aim to ensure a consistent architectural vision throughout the BonFIRE system by applying these key principles when making any architectural decisions. The principles are as follows:

1. It must always be possible for the BonFIRE multi-cloud facility to include testbeds over which BonFIRE has no control.
2. Always provide APIs to BonFIRE functionality, in addition to any BonFIRE graphical user interfaces (GUIs). These APIs will allow third parties to better integrate BonFIRE into their systems.

3. Allow experimenters full access to the specific functionality of particular testbeds. The heterogeneous testbeds within the BonFIRE multi-cloud differ for good reasons.
4. Allow higher-level functionality to exclude specific functionality of a testbed if this makes common tasks easier to achieve. Thus BonFIRE will give experiments the opportunity to choose the best approach for their specific task.
5. Support incremental adoption of the BonFIRE system by experimenters. Some experimenters will come to BonFIRE fresh and will be happy to invest effort learning the whole system in order to utilize its full power. Other users may wish to incrementally integrate BonFIRE functionality into existing experiments.
6. Support declarative specification of experiments as far as possible. Experimenters need only focus on what they want to deploy as an experiment and on the relevant conditions.

4 Architectural Functionality

This section presents an overview of the BonFIRE architectural functionality. These are the key functionalities of the BonFIRE architecture and are expressed independently of the architectural components needed to implement them.

4.1 Resources

BonFIRE allows experimenters to execute experiments that create and use various resources on the BonFIRE testbeds. BonFIRE supports three types of resources: compute, network and storage.

Compute resources are virtual machines (VMs) that are created and run for some part of the duration of an experiment. All BonFIRE virtual machines are created as part of an experiment and have a lifetime that does not exceed the lifetime of the experiment. Compute resources can be configured with application-specific contextualization information that can provide important configuration information to the virtual machine; this information is available to software applications after the machine is started.

Network resources can be used to connect the VMs to networks to allow the VMs to communicate with each other and possibly with the wider Internet. BonFIRE provides one network that any VM on any BonFIRE site can connect to. This network is also used by the BonFIRE services to manage and monitor the VMs in the experiment. Experiment-specific networks can be configured on the testbeds, according to the experiment's needs.

Storage resources are disks that can store data associated with an experiment. It is often important to decouple the lifetime of the storage resources from the lifetime of the experiment. This provides a persistence of data that allows experiment data to be written to a storage resource and be retrieved at a convenient time after the experiment has completed. BonFIRE supports two types of storage resource: operating system (OS) storage resources, and data block storage resources.

All BonFIRE compute, network and storage resources reside on the testbeds on which they are created and BonFIRE makes no attempt to hide this location information from the end user. End users are expected to plan their experiments with a detailed understanding of how they wish to distribute their resources among the various testbeds.

BonFIRE supports dynamically creating, updating, reading and deleting resources throughout the lifetime of an experiment. In order to aid the end user in managing the resources within an experiment, BonFIRE provides the concept of experiment resources that can be used as containers for the resources whose lifetime is limited to the duration of the experiment.

4.2 Monitoring

BonFIRE includes an extensible monitoring framework that allows users to log and monitor various metrics associated with an experiment's progress. There are three types of metrics that may be monitored according to the experiment's needs: VM metrics, application metrics and infrastructure metrics.

VM metrics provide system information regarding the status of the VM from the perspective of CPU, memory, disk space.

Application metrics can be defined by the experimenter according to the specific software applications being used by the experiment, for example the number of open database connections, or the value of an application counter.

Infrastructure metrics provide detailed information about the underlying hypervisor's system performance. For example, some experiments may need to monitor the hypervisor's CPU, how many VMs run on the same CPU, or what the load is on a specific CPU. This level of monitoring is typically not provided by production cloud systems but it is an essential feature of the BonFIRE experimental facility.

Users, and user-agents such as software systems, can access an experiment's monitoring information from a single source. BonFIRE provides an API to access monitoring information and also a graphical tool to display metric values and metric value graphs in real time while an experiment is executing.

4.3 Elasticity

Elasticity is the ability to create and delete resources, typically virtual machines with associated storage and networking, while an experiment is running. Usually elasticity actions are performed when specific conditions occur. For example, a new VM may be created when the overall system load is high. The BonFIRE monitoring functionality can be used to detect when these experiment-specific conditions occur. BonFIRE aims to provide three levels of elasticity support:

- Manual: where experimenters manually observe the monitoring system and create or delete resources as desired.

- Programmed: where the experimenter writes a program that uses the monitoring API to detect when specific conditions occur and programmatically creates or deletes the resources as desired. According to the experimenter's preferences this program may be executed either on a BonFIRE virtual machine or on a machine outside of the BonFIRE infrastructure.
- Managed: where the user specifies the elasticity policies in a high- level experiment descriptor and BonFIRE creates or deletes the resources according to the specified rules.

The BonFIRE resource API supports the control of resources on all testbeds in the BonFIRE infrastructure. It is therefore possible for an control program running on a VM on one testbed to create resources on another testbed. This support for cross-site elasticity is a key feature of BonFIRE.

4.4 Experiment Descriptors

Experiments can interact with the BonFIRE system at two different granularities: resource level and experiment level granularity.

When using resource level granularity, experimenters explicitly send a request to BonFIRE for each resource create, update, read or delete operation. With the experiment level granularity the experimenter describes the experiment in a single document and sends one request to BonFIRE to submit the experiment. Once submitted, the experiment is managed by BonFIRE and the individual resource level operations are performed by BonFIRE.

BonFIRE provides resource level operations through an implementation of the Open Cloud Computing Interface (OCCI) [6]. OCCI provides create, read, update and delete operations to compute, storage and network resources.

With experiment level granularity the experimenter describes their experiment in a single document that we call the Experiment Descriptor (ED). There are five parts to an experiment descriptor:

1. Specification of the initial deployment of compute, storage and network resources.
2. Specification of the monitoring metrics.
3. Specification of elasticity rules containing both trigger conditions (specified with respect to the monitoring metrics) and the actions to perform when the conditions are met.
4. Specification of general rules that specify trigger conditions and actions to perform that are not necessarily related to elasticity.
5. Specification of a shutdown sequence for stopping the experiment.

BonFIRE uses Open Virtualization Format (OVF) [7] as its primary experiment descriptor. OVF is a standard for the specification of the deployment of virtual resources and being developed within the Distributed Management Task Force (DMTF).

OVF will support the first part of the experiment descriptor described above. Parts 2, 3, 4 and 5 will require extensions to standard OVF in several directions.

Other projects have already extended OVF to support specific cloud requirements such as elasticity [10] and BonFIRE will aim to adopt the same extensions where possible.

Release 2 of BonFIRE supports experiment descriptors that specify the initial deployment of compute, storage and network resources, but does not support the specification of monitoring metrics or elasticity rules in one document. This does not mean that experiments requiring monitoring and elasticity are nor supported by BonFIRE. Such experiments simply have to configure the monitoring metrics and elasticity rules separately within the virtual machine images used by the experiment, rather than specifying them within an experiment descriptor.

As an alternative to the OVF format, BonFIRE will also support a very simple experiment descriptor format that provides the core functionality of specifying compute, network and storage resources in a single document.

4.5 Scheduling

Many commercial cloud facilities present a model of almost-infinite resource availability: the user gets as many resources as they apply for. Most BonFIRE testbeds, however, offer a small number of compute nodes as part of the permanent infrastructure. BonFIRE therefore operates at a different end of the resource spectrum from commercial cloud facilities and hence must operate in a different way regarding the scheduling of resources.

The problem is made interesting by the fact that BonFIRE combines cross-site deployment with the notion of the experiment. It is possible that the user may have a preference to start specific compute resources on specific sites; if one site cannot fulfill its requirements, while the others can, experiments may fail to run as expected. Also, BonFIRE will allow the user to leave the location unspecified, so the system needs a mechanism to resolve this in a way that the resources can be deployed.

This leads to the idea of *Atomic Experiment Scheduling*, whereby BonFIRE considers an experiment in its entirety before attempting deployment. The intention is that BonFIRE does not attempt deployment of any experiment resources until all involved BonFIRE sites indicate that they can allocate the required resources.

Future releases of BonFIRE will provide atomic scheduling over all the OpenNebula-based facilities. This will require BonFIRE-specific extensions to OpenNebula.

5 Architectural Components

The architectural components of BonFIRE can be seen in the architecture overview diagram show in Fig 1. The BonFIRE architecture consists of five layers each building on the functionality of the layer below it. Each layer exposes its functionality via a set of APIs. The APIs of the lowest two layers, the

Testbeds and the Enactor, can only be used by the layer above them. The APIs of the next two layers, the Broker and the Experiment Manager, can be used by end users, agents acting of behalf of end users, and the layer above them.

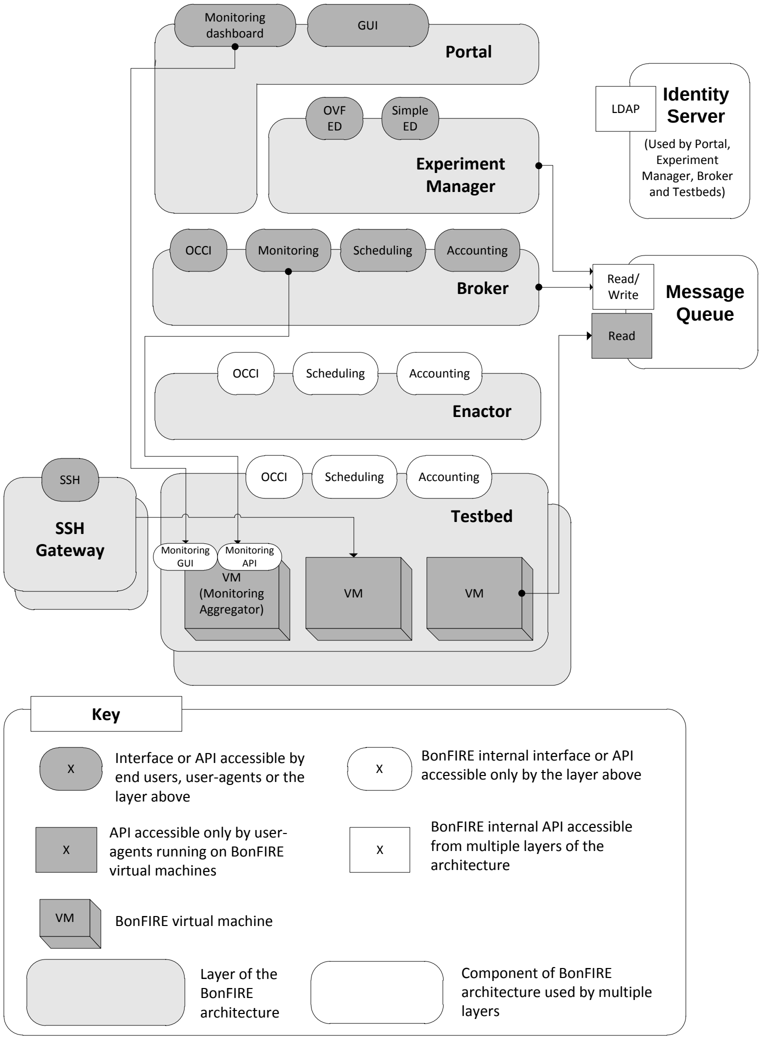
Monitoring dashboard
GUI
Portal
OVF ED
Simple ED
Experiment Manager
LDAP
Identity Server
(Used by Portal, Experiment Manager, Broker and Testbeds)
OCCI
Monitoring
Scheduling
Accounting
Broker
Read/ Write
Read
Message Queue
OCCI
Scheduling
Accounting
Enactor
SSH
SSH Gateway
OCCI
Scheduling
Accounting
Testbed
Monitoring GUI
Monitoring API
VM (Monitoring Aggregator)
VM
VM
Key
X
Interface or API accessible by end users, user-agents or the layer above
X
BonFIRE internal interface or API accessible only by the layer above
X
API accessible only by user-agents running on BonFIRE virtual machines
X
BonFIRE internal API accessible from multiple layers of the architecture
VM
BonFIRE virtual machine
Layer of the BonFIRE architecture
Component of BonFIRE architecture used by multiple layers

Fig. 1. BonFIRE architecture. The accounting and scheduling APIs shown here are not supported by the current version of BonFIRE.

5.1 Testbeds

The lowest layer of the BonFIRE architecture are the testbeds. The testbeds are the cloud infrastructures that provide compute, network and storage resources that can be used by BonFIRE experiments. Currently BonFIRE has three different types of testbeds: the Cells cloud infrastructure provided by HP; the Virtual Wall network emulation infrastructure provided by IBBT; and three cloud infrastructures implemented using the OpenNebula toolkit for cloud computing and offered by EPCC, HLRS and INRIA.

Theoretically testbeds may expose their functionality via any API they choose but to simplify the initial implementation all testbeds currently expose their basic resource manipulation functionality using a version of the Open Cloud Computing Interface (OCCI) [6].

5.2 Enactor

The role of the Enactor is to shield the technical details of how to communicate with each specific testbed from the higher level Broker. The Enactor receives OCCI requests from the Broker and transforms them onto suitable requests for the appropriate testbeds. The transformed request is then sent to the testbeds and the response received. The response is then transformed into a common format and passed to the Broker.

5.3 Broker

The Broker is the lowest layer of the BonFIRE architecture that may be accessed by end users and provides the entry point for experimenters to interact with BonFIRE at the resource level granularity.

The Broker maintains the current set of experiment resources and all the compute, storage and network resources currently used by the experiment. User operations on the compute, storage and network resources are passed to the Enactor and then onto the appropriate testbed. User operations, such as create and read, on the experiment resource are typically executed at the Broker because the testbeds know nothing about the concept of an experiment. Some operations such as deleting an experiment require the Broker to interact with the testbeds via the Enactor.

The Broker currently exposes two APIs: OCCI for managing resources; and a Monitoring API. The OCCI interface is used to create, read, update and delete compute, network and storage resources. The OCCI is extended from the proposed standard to include BonFIRE specific requirements the experiment resource, explict indication of the location at which a result must be created, and also additional network elements to specify the bandwidth, loss rate and latency for emulated networks on the Virtual Wall.

5.4 Experiment Manager

The role of the Experiment Manager is to take a description of an experiment as described in an experiment descriptor and manage its execution through multiple calls to the Broker. Experimenters can create new managed experiment resources and specify the corresponding experiment descriptor.

BonFIRE supports two different formats of experiment descriptor, OVF and the BonFIRE simple experiment descriptor. For each format the Experiment Manager contains a parser that maps the experiment descriptor onto a common data model that describes the experiment. This allows the rest of the Experiment Manager architecture to be decoupled from the choice of experiment descriptor format.

The data model is then validated to check that it is consistent and that the labels used to identify locations, templates, etc. are valid. When an experiment descriptor request is sent to BonFIRE, the validator is the last component to execute before the response is returned. This response indicates whether or not a valid experiment descriptor was provided, but cannot indicate whether it could be successfully deployed as deployment is carried out later in a background task. It is therefore important that the validator detects as many errors in the experiment descriptor as it possibly can.

The managed experiments are then placed in a queue, and deployed in turn in a background task. In a future version, the testbeds will be polled to check resource availability, before either deploying each experiment or holding in the queue until the necessary resources become available. At present, the Experiment Manager attempts to deploy every experiment in the queue. If the location of any resources is left unspecified in the experiment descriptor, the scheduler will choose a suitable location.

Next, the planner plans the order in which the various resources should be created. Network and storage resources will need to be created before the compute resources that use them. Additionally, property constraints between virtual machines may also influence the order in which compute resources must be created. For example, if a client virtual machine must be configured at boot time with the IP address of the server virtual machine, then the server virtual machine must be created first in order to determine its IP address.

Once the initial deployment of resources has been planned, the resources are created in the appropriate order via calls to the OCCI API of the Broker. Property values are also passed to VMs on creation, for example to ensure the server's IP address is written to the appropriate contextualisation section when the client virtual image is created.

5.5 Portal

The Portal offers the experimenter a graphical interface to the BonFIRE capabilities. It has a view of the experimenter's data, the running experiments, and the available platform capabilities (for example, the list of sites). The Portal is the aggregated view of the whole platform, and is the simplified entry point for the experimenter. It is also a way to present to the experimenter information from the Broker or the cloud sites, for example, to monitor experiment progress.

5.6 Message Queue

The role of the Message Queue is to support the sending of messages from the Broker and Experiment Manager to running virtual machines. The Broker and Experiment Manager write notifications of changes to an experiment's lifecycle state onto the Message Queue. Applications running on the virtual machines may subscribe to receive notification of the state changes associated with an experiment. One application of such a notification system would be for virtual machines to package up the experiment result data when the experiment state changes to terminated.

5.7 SSH Gateways and the BonFIRE WAN

The availability of publicly addressable IP addresses varies across the various BonFIRE testbeds. Some testbeds have very few, or even zero, such IP addresses to allocate. This obviously creates a challenge when one VM in a experiment must communicate with another that does not have a publicly addressable IP. To overcome this problem BonFIRE implements a WAN to which all VMs can be connected. This WAN can be used for both BonFIRE's control data and experimenters' application data. The WAN is implemented using VPN technology. The BonFIRE WAN can be a major bottleneck and should not be used for large scale inter-testbed communication that is crucial to an experiment. Instead in such cases publicly addressable IPs should be used wherever possible.

The SSH Gateways provide a means for experimenters to SSH into virtual machines that are not publicly addressable. Each BonFIRE site provides an SSH gateway that experimenters can connect to using SSH. From these gateways experimenters can then SSH to their VMs over the BonFIRE WAN.

5.8 Identity Server

The Identity Server provides BonFIRE's identity management functionality. User credentials are stored in a central LDAP database. To avoid a single point of failure the database is cloned on a master-slave approach on each site. The slaves on the sites are also used for user authentication on the sites. The web interface of the Identity Management allows users to manage their credentials. There is also an administrative web- interface which gives the BonFIRE administrators full control of all existing user accounts. The passwords are never seen in clear text by the administrators, as they are stored in hash format. For each user the Identity Server stores the username, password, groups and also optionally the user's SSH public key. If available, the testbeds use this public key to preconfigure all virtual machines such that only that user can SSH to them.

6 Experiments

BonFIRE is currently being used by seven experiments. This section gives an overview of two of these experiments and explains why BonFIRE is particularly suited to the task.

6.1 QoS-Oriented Service Engineering for Federated Clouds

This experiment is one of the three driving experiments in the BonFIRE project, which is carried out by the IT Innovation Centre at the University of Southampton. The experiment investigates techniques to better understand and predict the Quality of Service (QoS) achieved by service based applications when running on cloud infrastructures. The complexities of determining QoS requirements for service based applications in the cloud has given rise to a new class of service engineering tools within the Platform-as-a-Service (PaaS) layer for modelling, analysing and planning. While it is in the interest to a PaaS provider not to over- provision, it is important that the application performance is stable and the user experience is acceptable. This is not trivial since service based applications are often highly interactive and the usage patterns may fluctuate significantly depending on user demand.

QoS terms are typically set out in a Service Level Agreement (SLA) based on low level infrastructure terms such as CPU speed, disk space, etc. However, the customers (typically the application users) are often more interested in application-level parameters, e.g., the number of dropped frames for a multimedia streaming application. The gap between the terms infrastructure providers offer and what the users are interested in can be large. This results in a complex relationship between application performance and resource parameters, which is greater still for applications deployed across federated clouds.

Several service engineering tools have been developed in the EU IST IRMOS project to address the challenges above [8]. This experiment focuses on prediction of application performance based on a generic description of infrastructure resources (based on application benchmarking scores), to enable transferability across different providers, particularly with federation in mind. The hypothesis put to test in the experiment is that such a description of resources not only allows for prediction of application performance, but may enable more accurate predictions compared with using typical infrastructure parameters. Moreover, we hypothesize that using QoS terms in SLAs will improve overall efficiencies for all cloud stakeholders due to increased accuracy of requirements achieved by a simplification of service planning and adaptation models, and increased market adoption and flexibility due to the simplification of the federation between Platform and Infrastructure stakeholders.

The heterogeneous resources and federated testbeds in BonFIRE is essential to this experiment to be able to address the above hypotheses. The resource control offered in BonFIRE ensures repeatable experimentation, and the built-in monitoring facilities allow a more in-depth analysis than what is possible elsewhere; particularly infrastructure monitoring as that allows for correlation of results obtained on the virtual resources with what actually happened on the physical infrastructures. This is important for addressing resource reliability in the cloud, which is a factor one needs to consider when predicting application performance to estimate QoS. For more details about the experiment and initial results, please refer to [9].

6.2 Virtual Clusters on Federated Sites

The *Virtual Clusters on Federated Sites* experiment is funded through the first BonFIRE open call and is being carried out by Centre of Supercomputing of Galicia (CESGA). The aim of this experiment is to research the feasibility of using multiple cloud environments for the provision of services which need the allocation of a large pool of CPUs or virtual machines to a single user (as High Throughput Computing (HTC) or High Performance Computing (HPC)). The use case covers dose computation for radiotherapy treatments based on Monte Carlo methods, developed in the eIMRT project [11].

The experiment studies questions related to the usage of virtual clusters in a distributed cloud environment. A first set of questions are related to the time that the deployment and enlargement of such cluster need to be operational and the influence other simultaneous operations have in the process. Previously in other experiments in grids, local clusters and even commercial cloud providers, interferences with other users and processes have been observed which affect the final quality of service. The objective is to understand better how to manage these virtual clusters to guarantee a low time to solution or latency (this means, the time since the cluster has been requested to the end of the service). Also, the elasticity functionality will be analyzed, using the application performance monitoring as a trigger for the change in the size of the cluster.

A second set of experiments will investigate the usage of the distributed capability of cloud providers (federated or multi- site) in order to protect the service against failures. A virtual cluster will be deployed divided in two sets, and the characteristics of the network (latency, bandwidth and packet loss rate) will be changed to study the effects in the performance of the cluster. The radical situation of losing part of the cluster will be simulated changing the bandwidth to zero, when the other part should recover from it to guarantee that the customer receives the solution on time.

The main metric to measure in the experiments is the time to execute one operation as function of the factors which can affect this time, as the number of machines that compose the cluster, the size of each virtual machine, etc.

Because clouds are shared environments which are not completely under control of the experimenter, several measurements must be done. To automate the process of executing the experiments, an Experiment Agent should be developed. This agent will communicate with the BonFIRE infrastructure and will control the execution as well as record the data locally to make analysis later.

The results and data acquired during the experiment should permit CESGA to develop the policies and business rules to include in the applications under development at the institution which use the Software as a Service model. This new model will be developed after the end of the experiment.

An experiment like this needs the BonFIRE infrastructure because the usage of commercial providers does not guarantee enough control of the factors which can affect the results. Installing a local infrastructure for executing it is too complex and time consuming. Additionally, BonFIRE includes a site where the network can be controlled. This is a facility that CESGA can not deploy currently. BonFIRE, therefore, offers a unique cloud infrastructure where new cloud concepts and applications can be experimented with.

7 Current Status and Future Work

As of January 2012 release 2 of BonFIRE is operational and is being used by the three embedded experiments and the four experiments funded by BonFIRE's first open call. These open call experiments are feeding back comments and requirements that will be addressed in subsequent releases of BonFIRE. A new set of experiments funded by a second open call are planned to begin using the BonFIRE facility in September 2012. Subsequently, it is planned to open BonFIRE up to a wider set of users later in 2012.

The resource-level API provided by the Broker is working very well and will continue to provide the core functionality of BonFIRE. The higher-level functionality offered by the Experiment Manager will be extended to support all five levels of the experiment descriptor as described in Section 4.4.

Future BonFIRE releases will give experimenters greater control over the underlying infrastructure. This will include the ability to specify the physical host a virtual machine can be placed on. Research will also be carried out to investigate how well various points of resource contention with other cloud users can be simulated and hence controlled by experimenters. Such resource contention points are CPU, memory, disk and network usage. Where such contention cannot be controlled BonFIRE allows it to be monitored via the infrastructure monitoring functionality. Further research will be carried out to discover how well these infrastructure metrics can be used to predict and, therefore explain, the performance of an application running on a cloud.

BonFIRE will expand vertically to support the dynamic interoperation among some BonFIRE sites through the GÉANT network. This will automate the network control for real multi-site experiments and offer network as primary resource for the experiments with its key configuration parameters (bandwidth, delay, etc.). In release 3, the BonFIRE system will be interfaced to the GÉANT AutoBAHN system (Automated Bandwidth Allocation across Heterogeneous Networks). The user/cloud (BonFIRE) to network (GÉANT) interface will be based on the AutoBAHN User Access Point interface(UAP): the AutoBAHN client will be integrated into the BonFIRE Enactor to issue BoD request and implement consequent actions on the respective LAN sides (end-point). Also other BonFIRE system components will be adapted to cope with the new WAN network resources: in particular, the Portal will require new tabs/pages to describe the new resources; the OCCI will be extended to include specific fields for BoD network resources; the Experiment Manager and Broker will require data model adaptations. These near-future BonFIRE architecture and components modifications will make available on-demand network connectivity services between selected sites/resources bound to the experiments: user-controlled network QoS parameters and possibility to reserve actual (i.e. not emulated) network resources both in immediate-permanent and advance reservation mode will definitely enhance the Future Internet (FI) experiment coverage area.

Additional vertical expansion and the ability to federate with other FI testbeds will be demonstrated by connecting BonFIRE with a subset of the FEDERICA facility. This will require BonFIRE to connect to the Slice-based Facility Architecture (SFA) used by many of the networking-focused FI testbeds.

Acknowledgements. BonFIRE is funded by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 257386. The authors wish to acknowledge the contributions of: Frédéric Gittler (HP Labs), Kostas Kavoussanakis (EPCC), David Margery (INRIA), Josep Martrat (Atos), Eilidh Troup (EPCC), Constantino Vázquez Blanco (Universidad Politécnica de Madrid), Celia Velayos (i2CAT) and Tim Wauters (IBBT).

References

1. Virtual Wall, http://www.ibbt.be/en/develop-test/ilab-t/virtual-wall
2. FEDERICA Project, http://www.fp7-federica.eu/
3. GÉANT2 Bandwidth on Demand (BoD) User and Application Survey (DJ.3.2.1), http://www.geant2.net/upload/pdf/GN2-05-086v11.pdf
4. Definition of Bandwidth on Demand Framework and General Architecture (DJ3.3.1), http://www.geant2.net/upload/pdf/GN2-05-208v7_DJ3-3-1_GEANT2_Initial_Bandwidth_on_Demand_Framework_and_Architecture.pdf
5. HP Labs cloud-computing test bed projects - Cells as a Service, http://www.hpl.hp.com/open_innovation/cloud_collaboration/projects.html
6. Open Cloud Computing Interface, http://occi-wg.org/
7. Open Virtualization Format (OVF), http://www.dmtf.org/standards/ovf
8. Metzger, A., Boniface, M., Engen, V., Phillips, S., and Zlatev, Z.: Towards Critical Event Monitoring, Detection and Prediction for Self-adaptive Future Internet Applications. Proc. of the 1st Int. Workshop on Adaptive Services for the Future Internet (2011)
9. Phillips, S., Engen, V. and Papay, J.: Snow White Clouds and the Seven Dwarfs. In Proc. of the IEEE Int. Conf. and Workshops on Cloud Computing Technology and Science (2011)
10. Rodero-Merino, L., Vaqueroa, L. M., Gil, V., Galán, F., Fontán, J.,Montero, R. S., Llorente, Ignacio M.: From Infrastructure Delivery to Service Management in Clouds. Future Generation Computer Systems 26, 1226–1240 (2010)
11. González-Castaño, D. M., Pena, J., Gómez, F., Gago-Arias, A., González-Castaño, F. J., Rodríguez-Silva, D. A., Gómez, A., Mouriño, C., Pombar, M., Sánchez, M.: eIMRT: A Web Platform for the Verification and Optimization of Radiation Treatment Plans. J. Appl. Clin. Med. Phys. 10(3), 2998 (2009)