G-Lab-ToMaTo-Teagle Federation

Konrad Campowsky, Fabian Eichhorn
Next Generation Network Infrastructures
Fraunhofer FOKUS
Berlin, Germany
{konrad.campowsky,fabian.eichhorn}@fokus.fraunhofer.de

Christian Henke
Architektur der Vermittlungsknoten
Technische Universität Berlin
Berlin, Germany
christian.henke@tu-berlin.de

I. Introduction

. The broad field of Future Internet research requires diverse infrastructures that serve different needs across domains and layers. Federation between different experimental facilities is foreseen to address those requirements as single testbeds cannot offer the full range of technologies that are required by the Future Internet community. This abstract outlines an approach for integrating the G-Lab experimental facility with the Teagle federation framework by using the Topology Management tool (ToMaTo).

A. The Teagle Federation Framework

Teagle is a resource federation framework that has been developed in the context of the Panlab II project [1]. An important conceptual feature of the Teagle approach is that it aims at being a truly generic design that is able to deploy, manage and subsequently offer arbitrary resources and services [2].

The fundamental technical management authority of a Teagle domain is a Panlab Testbed Manager (PTM) which has been described in detail in [3]. A PTM's main functionality is the execution of generic provisioning operations on resources under its control. These resources are distinctly typed, uniquely named, and can expose further specific operations in addition to PTM's generic operations. Resources follow a dynamic lifecycle; resource types can be instantiated, these instances can be worked with, (re-)configured, and finally de-provisioned from the system. A high level overview of a PTM domain and its architecture is given in Fig. 1Figure 1.

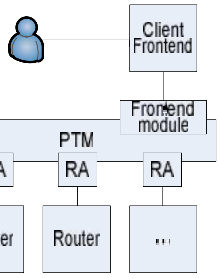

Figure 1: Structural overview of a PTM domain

Given the high heterogeneity of resource types a PTM has to support, it obviously cannot be directly aware of all resource specific semantics. An abstraction layer is made up of so-called *resource adapters* (RA) to which the PTM delegates the task of addressing resource specific types of communication. These resource adapters can be viewed as device drivers in the sense that they possess detailed knowledge about semantics of the resources instances they are responsible for. At its core level, a PTM itself is completely unaware of this nature and acknowledges resource instances merely as fundamental entities.

B. The G-Lab Testbed Facility

The G-Lab testbed facility [4] is based on the *PlanetLab* infrastructure [5] and provides a clustered environment distributed all over Germany to different customers from the academic world. This environment allows researchers to book slices of infrastructure in order to experiment their concepts. The infrastructure which can be booked consists of virtual server with specific processing power, storage and interconnection capabilities in terms of throughput and bandwidth. According to specific requirements regarding processing power, storage and connectivity slices are allocated to a specific project. PlanetLab is one of the platforms working under the umbrella of the GENI [6]. It is a global research network in the form of a distributed computing platform, designed to support the development of new network services. The actual control framework software used by G-Lab is called *PlanetLab Central* (PLC). Its primary focus lies on multiplexing the computing resources of *nodes* (servers) through distributed virtualization. Nodes run a minimal version of a Unix operating system and are divided into virtual containers. The resources (reserved CPU percentage, memory and bandwidth allocations, etc.) bundled by such a virtual container are called a *sliver*. PLC then groups these slivers into *slices* which are owned and subsequently administered by the party requesting the instantiation of a slice. Note that a sliver always belongs to exactly one distinct slice.

C. Topology Management Tool

The Topology Management Tool (ToMaTo) allows researchers to define complex network topologies and deploy them on the G-Lab facility. These topologies can include components like virtual machines (both KVM [8] and OpenVZ [9] based), switches, hubs and routers. It further more enables the user to specify network link parameters like bandwidth, delay and jitter.

Even though ToMaTo offers its users a convenient graphical user interface, its full functionality is also exposed through an XML-RPC API that accepts a description of a topology in a defined XML format.

The Topology management tool and its capabilities are described in more detail in [10].

II. Resource Management and Federation

A. Design Considerations

Our architectural design for integrating the G-Lab facility with the Teagle framework takes into consideration that ideally resources acquired from G-Lab should be seamlessly usable with other Teagle resources and furthermore support the same capabilities. This means that the PTM must also be enabled to make use of slivers offered by G-Lab in the same way it would utilize resources under its direct control.

Although G-Lab does not directly support the PTM's notion of a resource hierarchy, this requires little efforts. After acquiring resources from G-Lab, the PTM must merely setup appropriate RAs for further configuration. For example, for virtual machines this implies programmatically accessing the machine through SSH and deploying a RA that allows installing further software packages on the machine.

While ToMaTo offers virtual machines based on either the KVM or OpenVZ virtualization technologies, only the KVM variety is fully suitable for seamless usage in a Teagle testbed. In contrast to KVM - which is a hypervisor based virtualization solution - OpenVZ is a container based solution and is unable to support all features that other VMs in Teagle offer. Consequently, our implementation will only create KVM virtual machines.

B. Prototype Implementation

The obvious choice for integrating any type of resource into Teagle is by writing an appropriate RA for a PTM.

Consequently, this is what we did. The ToMaTo Adapter translates the generic CRUD operations it receives from the PTM into appropriate XML-RPC requests that are dispatched towards the ToMaTo API. The adapter itself is a wrapper around the ToMaTo interfaces. It stores no information about created resources or their configuration itself. It rather relies on the ToMaTo API which it queries each time information about resources offered by ToMaTo is required, thus avoiding a split-brain problem.

The following list shows the resource types exposed by the RA towards Teagle and their mapping on ToMaTo resources:

- Topology: A virtual resource that serves as a grouping of resources. It represents a testbed created in ToMaTo.
- VNode: A virtual machine. This type maps to a Node in a ToMaTo topology.
- Switch: Representation of a ToMaTo switch.
- Router: Representation of a ToMaTo router.
- Link: Represents the connection of a VM and a switch.
- Network: Representation of an IP based network

Any other ToMaTo related aspects are handled transparently to the user. The Teagle user can design topologies from these types using the Teagle frontend tools. The user can create a number of Virtual machines and form a network by interconnecting the machines with switches. Optionally, the user can choose to deploy a Link resource to define this interconnection. This allows the user to specify advanced configuration parameters like bandwidth and loss rate. Fig. 2 shows a ToMaTo topology created in Teagle.

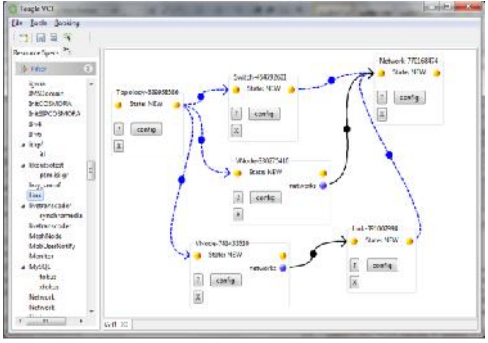

Figure 2:A ToMaTo topology created in Teagle' VCT Tool

After creating the specified topology on G-Lab, the ToMaTo Adapter will deploy additional RAs to facilitate provisioning on the created G-Lab nodes. Thus, the user can use the G-Lab nodes like any other VMs created in Teagle.

III. Conclusion and Outlook

In this article we have shown how the G-Lab facility can be enabled for federation with other experimental facilities by integrating it with the Teagle federation framework.

However, many things are left to be done at the PTM side. The process of selecting resources is not yet completely transparent for a TEAGLE user. In the future, the placement of virtual nodes will have to be decided by the PTM itself. Also, currently the user still needs to create a topology resource to contain other resources from G-Lab. This is a limitation that does not exist with other VMs in Teagle and which we plan to remove in the future.

[1] Website of Panlab and PII European projects: http://www.panlab.net

[2] Konrad Campowsky, Thomas Magedanz, and Sebastian Wahle. Interoperability in Heterogeneous Resource Federations. In International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM 2010). ICST/Springer, 2010.

[3] Campowsky, K., A. Gavras, B. Harjoc, T. Magedanz, and S. Wahle. "Pan-European Testbed and Experimental Facility Federation – Architecture Refinement and Implementation." Inderscience International Journal of Communication Networks and Distributed Systems(IJCNDS), Special Issue: Recent Advances in Test-bed Driven Networking Research, 2010

[4] Website of the G-Lab experimental facility: http://www.german-lab.de/

[5] Website of the PlanetLab experimental facility: http://www.planet-lab.org/

[6] National Science Foundation, GENI website: http://www.geni.net

[7] Website of the Topology Management Tool: http://tomato.german-lab.de/

[8] Website of the Kernel Based Virtual Machine (KVM) project: http://www.linux-kvm.org/page/Main_Page

[9] Website of the OpenVZ project: http://wiki.openvz.org/Main_Page

[10] Dennis Schwerdel. ToMaTo - a network experimentation tool. In International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM 2011). ICST/Springer, 2011