A Virtual Machine Server for the Future Internet

Rafael dos S. Alves, Luís Henrique M. K. Costa, Miguel Elias M. Campista,
Leonardo G. Valverde, Pedro da S. Pisa, Carlo Fragni, Tiago N. Ferreira,
Igor M. Moraes and Otto Carlos M. B. Duarte
{santos,luish,miguel,valverde,pisa,carlo,tiago,igor,otto}@gta.ufrj.br

Universidade Federal do Rio de Janeiro - GTA/COPPE/UFRJ

Abstract— **In this work we develop a virtual machine server (VMS) to instantiate and control virtual routers in a pluralist network substrate based on Xen [1]. The server provides services to the piloting plane of the future Internet architecture. Our implementation relies on Web services and on a virtualized systems management library. In order to demonstrate the server operation, we have also implemented a graphical user interface where the user plays the role of the piloting system.**

I. Introduction

Different architectures for the Future Internet were recently proposed. These architectures can be divided in two, purist and pluralist. Purist architectures propose a protocol stack flexible enough to meet current and future application requirements. On the other hand, pluralist architectures accommodate parallel protocol stacks to cope with multiple service requirements. Virtualization has emerged as a vital tool for both types of architectures, especially the pluralist ones [2], [3]. For pluralist architectures, the network element can be divided into many virtual elements running different operating systems and protocol stacks. In the purist approach, the computer virtualization can be used to test new protocols. Therefore, independently of the architecture, virtual networks creation requires multiple virtual network elements distributed on physical network elements. In order to automate the process, a virtual machine server (VMS) is developed in this work.

II. Architecture of the VMS

In the pluralist architecture, the virtual machine server provides services to the network piloting system. Our VMS lies between the Piloting System, which takes decisions on behalf of network users, and the pluralist network substrate, implemented by Xen virtualization platform. The piloting plane is an intelligent system which senses the network, makes decisions, and acts enforcing changes in virtual routers and links. For example, if the piloting plane detects a performance bottleneck on some physical node, a decision to migrate some virtual machine to another physical element can be taken. In this case, the piloting plane sends a message to the virtual machine server which performs the migration.

The virtual machine server was implemented as a Web Server, i.e., the messages sent from the piloting plane for the virtual machine server are SOAP messages. The programming language used was Java, and the Web services library was Axis2. The Libvirt library, which provides control functions for different virtualization platforms including Xen, was used to gather information and take actions on the virtual routers.

Currently, our virtual machine server already implements a *topology discovery service* in which all the physical and virtual nodes from the network are discovered and its connections are mapped, a *status service* that returns the main parameters of the physical machine and virtual routers running on it, such as the IP addresses and the number of virtual CPUs allocated to each virtual machine, a *migration service* that can be used to live migrate a virtual machine between physical machines, a *measurement service* which gathers a number of parameters, such as memory utilization and CPU load. This last one can be used online by the piloting plane to make decisions and take actions using other VMS services.

The *graphical user interface* (GUI) allows the user to act as the piloting system. Implemented in Python with the OpenGL library, the GUI gives the user a graphic view of the physical and logical network topologies. It also allows the user to obtain information of each node and each link of the networks. Besides, our GUI implements graph operations such as shortest path discovery and redrawing algorithms to help the users to better understand the network controlled.

III. Conclusions and Future Work

The proposed virtual machines server can create new virtual elements on desired physical hosts. The server can also take administrative tasks and serve as a controller either for pluralists and purists solutions for the Future Internet. In our future work, the virtual machine server will store preconfigured OS images, which are migrated and initiated in physical hosts.

Acknowledgments

This work was supported by FINEP, FUNTTEL, CNPq, CAPES, and FAPERJ.

References

[1] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, “Xen and the art of virtualization,” in *Proceedings of the nineteenth ACM Symposium on Operating Systems Principles - SOSP03*, Oct. 2003.

[2] T. Anderson, L. Peterson, S. Shenker, and J. Turner, “Overcoming the internet impasse through virtualization,” *ACM Computer*, vol. 38, no. 4, pp. 34 – 41, Apr. 2005.

[3] J. He, R. Zhang-Shen, Y. Li, C. yen Lee, J. Rexford, and M. Chiang, “Davinci: Dynamically adaptive virtual networks for a customized internet,” in *ACM CoNEXT*, Dec. 2008, pp. 1–12.