

Extending the Planetlab Usage Model for Distributed Heterogeneous Research Testbeds *

Roberto Canonico, Giorgio Ventre
Dipartimento di Informatica e Sistemistica
Università di Napoli Federico II
via Claudio 21, 80125 Napoli, ITALY
{roberto.canonico, giorgio}@unina.it

Salvatore D'Antonio
Consorzio Interuniversitario Nazionale
per l'Informatica - CINI
Laboratorio ITeM-CINI
via Cinthia 4, 80126 Napoli, ITALY
saldanto@unina.it

Abstract

Distributed research testbeds play a fundamental role in the evaluation of disruptive innovations for Next Generation networks. In recent years, the main research funding agencies have promoted several initiatives aimed at building large-scale infrastructures that could be used to support research in the area of networking and distributed applications. To maximize the benefit/cost ratio, multiple concurrent experiments should be able to use the infrastructure resources at the same time. However, concurrency of resource usage might lead to interference and produce meaningless and not replicable results. This problem has already been addressed in several testbeds, usually by the adoption of proper mechanisms to guarantee the isolation of experiments. The OneLab research project aims at building a European-scale research infrastructure federated with PlanetLab. One of the peculiarities of OneLab is the heterogeneity of access networking technologies (e.g. UMTS, WiFi, WiMax, WMNs, ...). The PlanetLab usage model, based on the concept of slice and implemented by means of the VServer virtualization system in the testbed nodes, is not completely suitable to face the intrinsic heterogeneity of resources in OneLab. In this paper we propose a few extension to the PlanetLab usage model that we intend to introduce in OneLab in order to better support heterogeneous resource sharing. The main purpose of this paper is to present work-in-progress and to stimulate discussion on this specific issue among participants of the ROADS workshop.

1 Introduction

The availability of large scale real-world distributed research infrastructures has been widely recognized as a fundamental necessity to drive the evolution of the future Internet [1]. The largest research testbed currently in operation is PlanetLab, made of almost 800 network nodes (Linux-based end systems) located at about 400 institutions spread all over

*This work has been funded by the FP6 IST-2005-034819 project "OneLab: an open networking laboratory supporting communication network research across heterogeneous environments".

the globe. OneLab is a EU-funded research project aimed at building a European-scale research infrastructure federated with PlanetLab. One of the peculiarities of OneLab is the heterogeneity of access networking technologies (e.g. UMTS, WiFi, WiMax, WMNs, ...) made available to experimenters. Since one of the objectives for OneLab was to be federated with PlanetLab, OneLab has borrowed from PlanetLab some basic assumptions and design choices. In particular, to maximize the benefit/cost ratio, the infrastructure is being designed with the objective of allowing multiple concurrent experiments at the same time. When several experiments use the same node, both computational and communication resources are shared. How concurrency is managed strongly influences the testbed usage model. In this paper we present a pragmatic approach we have decided to pursue in order to define and implement a proper usage model for heterogeneous resource sharing in OneLab.

2 Resource sharing models in a multi-experiment research testbed

An important goal for a distributed research testbed is to share a set of physical resources among multiple experiments that might either run concurrently or sequentially or in combination. In order to ensure that experiments do not interfere with each other, proper resource usage mechanisms have to be defined and implemented.

The simplest resource sharing mechanism consists in relying on the native CPU and packet schedulers implemented in the nodes if the testbed to let several processes, possibly belonging to different experiments, share both computational and communication resources. This approach does not provide any isolation among concurrent experiments. Hence, it is viable only if shared resources never get exhausted by any single experiment, thanks to overprovisioning.

A more reliable approach, usually referred to as *slicing*, consists in allocating a subset of physical resources to each experiment. Slicing is usually implemented by means of virtualization. Virtualization is a widely used technique in which a software layer multiplexes lower-level resources among higher-level software programs and systems.

3 Usage models of existing research infrastructures

In this section we provide a survey of the approaches to resource management and allocation which have been adopted

by some research projects in the field of distributed experimental testbeds.

PlanetLab is a geographically distributed platform for deploying, evaluating, and accessing planetary-scale network services. In order to run short-term experiments or deploy continuously running services PlanetLab users acquire a slice, which is a collection of virtual machines (VMs) spread around the world. Slices run concurrently on PlanetLab, acting as network-wide containers that isolate services from each other. An instantiation of a slice in a particular node is called a sliver. In the current version of PlanetLab, VMs are implemented on physical machines by means of the Linux VServer technology. VServer implements a soft partitioning concept based on Security Contexts which permits the creation of many independent Virtual Private Servers (VPS) simultaneously running on a single physical node. By means of so-called contexts, VServer hides all processes outside of a given scope, and prohibits any unwanted interaction between a process inside a context and a process belonging to another context.

In PlanetLab, slice creation and resource allocation are decoupled. When a slice is first created, a best effort service is associated with it and resources are acquired and released by the slice during its entire lifetime. Therefore, slices are not bound to sets of guaranteed resources. Such an approach has been deliberately chosen in the original PlanetLab design. PlanetLab, in fact, has not been designed for controlled experiments, but to test services in real world conditions. Replicability of experiments, hence, has not been considered an important goal [3].

Emulab [4] is a software that runs shared testbeds for development, debugging, and evaluation of networked systems and applications. The Emulab's resource usage approach relies on the solution of the combinatorial optimization/constraint satisfaction problem of giving users resources that match their request based on the knowledge of the available resources. Emulab allows two kinds of experiments: immediate experiments and queued experiments. When Emulab experiments are submitted, they are given a duration, i.e. a time bound after which the resources used for that experiment will be revoked. Users may change or remove this duration, but they must justify any change. Thus, experimenters are encouraged not to hold resources for longer than necessary. This procedure enables fair sharing among users.

4 Defining the OneLab usage model

Our analysis suggests that the virtualization techniques currently used in PlanetLab (namely, VServers and VNET) don't seem to adequately fit the requirements of a heterogeneous testbed like OneLab. In particular, since OneLab aims at introducing novel wireless networking technologies into the experimental environment, a few challenges seem not to find in PlanetLab a proper solution. Just as an example, the fact that many experiments involving wireless connections are very much topology-dependent, leads us to the conclusion that a precise knowledge of each node characteristics is to be maintained in the central repository of nodes (PLC, in PlanetLab terminology). Moreover, in the case of UMTS, WiMax and WMNs, the current flat resource sharing policy adopted in PlanetLab does not provide sufficient guarantees in terms of isolation.

The Wireless Virtualization Working Group of the GENI Project has recently issued a preliminary report [2] in which it is stated that overprovisioning cannot be applied to the

wireless spectrum. The scarceness of wireless spectrum, calls for a wider range or resource partitioning models in order to support a reasonable range of concurrent experiments in a wireless testbed.

Similar considerations hold for other particular resources (an UMTS network interface, as well as an IP video camera or a network processor card) that may be made available only in a few nodes of a heterogeneous testbed like OneLab. In the absence of sophisticated fine-grained resource sharing mechanisms, a time-based Space Division Multiple Access (SDMA) model may be adopted. Such a model assigns the PLC the responsibility to manage a time based advance resource reservation system that may grant exclusive use of particular resources only to a given slice. A similar system, named Sirius, has been partially implemented in PlanetLab version 4, but its use is optional and, in any case, not suitable to manage more specific nodes configurations. Hence, we are currently investigating viable extensions of the Sirius system and of the PLC database that may be used to manage peculiar resources in OneLab.

5 Conclusions

In this paper we have reported some preliminary considerations that have emerged so far in the definition of a proper usage model to be adopted for the OneLab research testbed. The peculiarity of resources made available only in a few of the testbed nodes calls for a usage model that allows users to control precisely and reliably the usage of resources in the nodes belonging to a given slice. A reliable time-based reservation mechanism, able to control and manage the use of specific resources at each node is one of the components to be added to the infrastructure. In our opinion, however, such a mechanism should not only operate at a centralized level (e.g. in the PLC) but should also cooperate with the virtualization mechanisms used to implement slivers in the nodes. An analysis of the capabilities of VServer and an investigation of the opportunities offered by different virtualization techniques (like Xen or OpenVZ) is currently under way.

References

- [1] Tom Anderson, Larry Peterson, Scott Shenker, and Jonathan Turner (Eds.). Report of NSF Workshop on Overcoming Barriers to Disruptive Innovation in Networking. Technical Report GENI Design Document 05-02, GENI, January 2005.
- [2] Sanjoy Paul and Sridhar Seshan. Virtualization and Slicing of Wireless Networks. Technical Report GENI Design Document 06-17, GENI Wireless Working Group, September 2006.
- [3] Larry Peterson, Vivek Pai, Neil Spring, and Andy Bavier. Using PlanetLab for Network Research: Myths, Realities, and Best Practices. Technical Report PDN-05-028, PlanetLab Consortium, June 2005.
- [4] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar. An integrated experimental environment for distributed systems and networks. In *Proc. of the Fifth Symposium on Operating Systems Design and Implementation*, pages 255-270, Boston, MA, December 2002. USENIX Association.