GENI Racks: InstaGENI Design and Deployment and Changes to the PlanetLab Networking Model

Andy Bavier
Princeton University
acb@cs.princeton.edu

Joe Mambretti
Northwestern University
j-mambretti@northwestern.edu

Rick McGeer
HP Labs
rick.mcgeer@hp.com

Rob Ricci
University of Utah
ricci@cs.utah.edu

I. Introduction

The National Science Foundations Global Environment for Network Innovations (GENI) is an effort to build an environment for large-scale networking experimental research in novel network services and architectures. Deploying an environment to support these research activities requires a platform of sliceable, reservable compute and network resources that are deeply programmable, widely deployed, highly instrumented and directly accessible to end-users at many layers.

The GENI Mesoscale deployment envisions a network of distributed clusters ("GENI racks") running the OpenFlow framework and supporting the GENI AM API, an emerging standard for federation and interoperability. InstaGENI, a collaborative partnership effort of HP, Northwestern, Princeton, the University of Utah, and the Open Networking Institute, is one of the two funded GENI rack deployment efforts. InstaGENI features a lightweight, expandable cluster design featuring integration with the FlowVisor OpenFlow Aggregate Manager (FOAM), ProtoGENI and PlanetLab Aggregate Managers, and L2 connectivity to national research networks. InstaGENI racks will federate with existing Slice Authorities such as GPO GENI, ProtoGENI, and PlanetLab Central, enabling researchers in these communities to allocate resources across the InstaGENI deployment.

II. Background

PlanetLab [1], [2] was designed as a deployment platform for distributed systems as overlay networks, from network measurement systems to wide-area stores to content distribution networks to distributed hash tables. Each project receives a *slice* of the available PlanetLab resources, implemented using lightweight OS containers on a shared Linux kernel. PlanetLab can be regarded as a distributed Cloud offering the Linux environment as Platform as a Service.

Emulab [3] is a Cloud system that offers clusters of computational resources connected by virtual networks with guaranteed bandwidth and settable loss and latency. Unlike PlanetLab, Emulabs computational resources are generally physical nodes with customizable operating system images installed on the bare metal (i.e., Hardware as a Service). Emulab is designed for short-lived network experiments under controlled conditions, whereas PlanetLab is designed for long-lived services and observations on the open Internet. The two systems are complementary pillars in the GENI network experimentation ecosystem. In the context of GENI, Emulab has evolved into the ProtoGENI project.

The OpenFlow protocol [4] allows deep programming of network switches, with experimental and production traffic running over the same network infrastructure. An OpenFlow switch is treated as a simple flow table, and an OpenFlow controller uses a standardized protocol to modify flow entries. FlowVisor is an proxy enabling an OpenFlow network to be sliced among multiple controllers, similarly to the way that PlanetLab and Emulab slice compute resources. OpenFlow has been adopted by a number of switch vendors, including HP on its E-series (nee Procurve) switches.

III. InstaGENI Design

The original Internet grew exponentially in the 1990s because its core software and protocols could be deployed on almost any computer and its messages could be carried by a wide variety of existing communications systems. In this spirit, InstaGENI is committed to immediately building a live, highly distributed experimental facility, capable of running any existing GENI research experiment on an intercontinental scale, and using existing software modules and COTS hardware components. The base InstaGENI design is engineered for affordability: it is easy to expand a simple, cheap base installation. Our design goal in each component was to choose a minimal, expandable configuration.

A. Rack Hardware

The base InstaGENI rack consists of five experiment nodes, one control node, an OpenFlow switch for internal routing and data plane connectivity to the Mesoscale infrastructure, and a small control plane switch/router.

The InstaGENI rack has been designed for expandability, while providing standalone functionality capable of running most ProtoGENI experiments or as an exceptionally capable PlanetLab site. The base computation node is the HP ProLiant DL360 G8. Each experiment node has 12 cores, 48GB RAM (4GB/core), and 1TB disk. The experiment nodes are used for transient storage only; the control node provides permanent image and user storage with a 4 TB RAID array. The control node is a quad-core machine with 12GB RAM. Remote monitoring and management is enabled via the HP iLO3, a separately-powered card with separate network connection in the server chassis.

Each experiment node features four 1 Gb/s ports; three are connected to the OpenFlow switch and one to the control switch. The OpenFlow switch is an HP ProCurve (now E-Series) 5406 switch with v2 linecards. The control connection for the wide area goes through the HP ProCurve 2610 (24 port) switch. The 2610 switch also carries the iLO connections.

B. Core Software and Networking

Each InstaGENI rack runs the GENI AM API [5], which enables each rack to advertise and allocate resources as part of the federated GENI ecosystem. Two Aggregate Managers run on the control node: the ProtoGENI AM and the FlowVisor OpenFlow AM (FOAM). The ProtoGENI AM is responsible for managing and configuring the compute nodes and setting up VLANs for experimenters. FOAM partitions the network flow space and enables experimenters to plug in OpenFlow controllers to manage the experiment data planes.

In addition to Hardware as a Service, InstaGENI provides two images for slicing up compute resources on a single node: an OpenVZ-based image and a next-generation PlanetLab image. Both provide experimenters with the ability to create lightweights VMs using OS containers, and to connect the VMs to public Internet or to the Mesoscale network infrastructure via VLANs and OpenFlow. The PlanetLab network stack is discussed in more detail below.

The experimental data plane of each InstaGENI rack is connected via L2 to the GENInet backbone, a set of dedicated VLANs in the the Internet2 and NLR networks in the U.S. The InstaGENI experimental environment provides options for creating WAN networks based on dynamic or static VLANs and tunnels provisioned over the core foundation channels.

IV. PlanetLab on InstaGENI

InstaGENI racks can run the PlanetLab node image to provide lightweight and robust slicing of compute resources in environment familiar to experimenters; we refer to these nodes as IG-PL nodes. This PlanetLab image uses Linux Containers (LXC) as the virtualization technology. All PlanetLab nodes on all InstaGENI racks are controlled from a single MyPLC [6] instance (IG-PLC). IG-PLC exports the GENI AM API and federates with PlanetLab Central and other GENI Slice Authorities. All IG-PL nodes appear as a single GENI Aggregate.

The InstaGENI PlanetLab image uses Linux Containers (LXC) as the virtualization technology instead of the traditional Linux VServers [7]. VServers is a mature OS container technology but it requires patching the kernel, which introduces overhead on the PlanetLab developers. LXC has been adopted by the Linux developers for inclusion in the mainline kernel; IG-PL nodes run a stock Fedora kernel.

From the standpoint of users, the change from VServers to LXC is mostly transparent: experiments that used to work with VServers still works with LXC. The main difference is in network virtualization. With VServers, all PlanetLab slices share the external NIC, and a custom module called VNET provides isolation so that each slice sees only its own traffic. LXC virtualizes the kernel networking stack on a per-slice basis, so that each slice can now customize its forwarding table, bridge interfaces, run a firewall, adjust TCP parameters, etc. The per-slice capabilities provided by LXC are essentially the same as those of the VINI/Trellis system [8], [9].

A. New Networking Model

A key goal of GENI is to allow slices to program the network at L2. The IG-PL image supports this goal by connecting slices to the rack's OpenFlow data plane at L2. PlanetLab's MyPLC software stack is also used in a number of other testbeds (PlanetLab, MeasurementLab, VINI, etc.) that have different networking requirements. The networking modes supported by the IG-PL image satisfies the requirements of all these testbeds, and are very similar to the methods that other virtualization platforms use to connect VMs to the network:

- *PlanetLab*: Each node has a single public IP address. Every slice has a virtual NIC with its own private IP address, and connects to the public network through L3 NAT. A custom kernel module allows a process inside a slice to bind directly to external ports.
- *Measurement Lab*: Each slice has a virtual NIC with its own unique public IP address, and connects via a L2 software bridge to the physical NIC.
- *VINI/Trellis*: Each slice has its own virtual network topology implemented by L3-over-L2 tunnels. Slices contiain multiple virtual interfaces that are bridged at L2 to other virtual NICS representing tunnel endpoints.
- *InstaGENI*: Each slice contains one or more virtual NICS that are bridged at L2 to virtual NICs representing VLANs. Each virtual NIC has its own MAC address.

The IG-PL image uses Open vSwitch (OvS) [10], which supports the OpenFlow protocol, to implememt L2 software bridging. We plan to let experimenters plug an OpenFlow controller into a slice's data plane in the IG-PL kernel, but we leave the details to future work.

V. InstaGENI Deployment

By the end of 2012, InstaGENI racks will be running at 8 sites. The InstaGENI project plans to deploy 32 racks at campuses across the U.S. over three years. InstaGENI will shortly provide one piece of a comprehensive distributed environment for building the next generation of network services.

References

[1] A. Bavier et al. Operating System Support for Planetary-Scale Network Services. (NSDI '04), May 2004.
[2] L. Peterson and A. Bavier and M. Fiuczynski and S. Muir. Experiences Implementing PlanetLab. (OSDI '06), November 2006.
[3] B. White et al. An Integrated Experimental Environment for Distributed Systems and Network. (OSDI '02), December 2002.
[4] OpenFlow Network Foundation. http://www.openflow.org
[5] GENI AM API. http://groups.geni.net/geni/wiki/GAPI_AM_API
[6] PlanetLab MyPLC. http://svn.planet-lab.org/wiki/MyPLCUserGuide
[7] Linux-VServer project. http://linux-vserver.org
[8] S. Bhatia et al. Trellis: A Platform for Building Flexible, Fast Virtual Networks on Commodity Hardware. (ROADS '08), December 2008.
[9] A. Bavier and N. Feamster and M. Huang and L. Peterson and J. Rexford. In VINI Veritas: Realistic and Controlled Network Experimentation. (SIGCOMM '06), September 2006.
[10] Open vSwitch. http://openvswitch.org