

German-Lab Experimental Facility

Dennis Schwerdel, Bernd Reuther, Paul Müller
University of Kaiserslautern, Germany
{schwerdel, reuther, pmueller}@informatik.uni-kl.de

Robert Henjes, David Hock
University of Würzburg, Germany
{henjes, david.hock}@informatik.uni-wuerzburg.de

I. INTRODUCTION

The G-Lab project aims to investigate concepts and technologies for future networks in a practical manner. Thus G-Lab consists of two major fields of activities: research studies of future network components and the design and setup of experimental facilities. Both is controlled by the same community to ensure, that the experimental facility fits to the demand of researchers. Researchers gain access to virtualized resources or may gain exclusive access to resources if necessary. The goal of the G-Lab project is that theoretical research and the experimental facility will converge into a Future Internet. Thus it is important that the experimental facility is flexible enough to adapt to the needs of the experiments and ultimately become a research field itself.

The G-Lab project has started in 2008 as a distributed joint research and experimentation project for Future Internet studies and development. Initially this BMBF¹ funded project was distributed across six universities in Germany: Würzburg, Kaiserslautern, Berlin, Munich, Karlsruhe, and Darmstadt.

In the working group 7 (Experimental Facility), a distributed experimental facility consisting of wired and wireless hardware with over 170 nodes, which are fully controllable by the G-Lab partners, is built up and managed. This platform provides a facility to G-Lab working groups (e.g., 1-6) to test their proposed approaches and ideas for the Future Internet architecture. The whole network of the platform is distributed into individual clusters at the six different locations within Germany. Kaiserslautern serves as the head of administration with an increased number of nodes.

The first version of platform was available at March 2009 and first experiments took place at the commencement of April. Since then the second phase of the G-Lab project has started. This phase includes 9 projects with both academical and private partners contributing specialized hardware like sensor networks and energy-efficient nodes to the experimental facility.

This work introduces the current setup of the experimental facility, describing the available hardware, management of the platform, the utilization of the Planet-Lab software and the user management.

II. EXPERIMENTAL FACILITY DESIGN

For the design of the experimental facility it has been an important point to use existing solutions, adapt them if needed

¹„Bundesministerium für Bildung und Forschung“, German Federal Ministry of Education and Research

and integrate them. Thus it was possible to build up a running testbed very quickly. The usage of free and mostly open source software solutions allowed to use the full budget for hardware equipment and also makes it easy to adapt the used software.

A. Hardware Setup

Each of the six sites in phase 1 has set up a local cluster consisting of 25 cluster nodes (Kaiserslautern: 59, Berlin: 15) interconnected by a Cisco switch. All nodes feature 16 GB of system memory, two quad-core processors, fast hard-drives and four gigabit network interfaces. Each site has a head node with an even more powerful processor and 12 hard-drives and two network nodes with four additional network interfaces.

The nodes are connected to the switch with at least two network interfaces, leaving the option to wire the other interfaces directly when needed by an experiment. Additionally all nodes can be managed by a service processor that has a separate network interface. These management interfaces are logically separated from the normal interfaces with only the head node being connected to both network parts.

The public network part is routed to the Internet connecting the clusters of the different sites.

B. Headnode Structure

The physical head node machine hosts several virtual machines offering services to the experimental facility. One virtual machine on this host at each site, the so called local head node, is dedicated to the local cluster management. As virtualization solution VMWare ESXi 4 is being used but other solutions like Proxmox are also being examined. Currently the physical head node in Kaiserslautern (main site) hosts virtual machines for monitoring (section II-E), Planet-Lab Central (section II-D), a file server, the head node software and various machines for testing purposes.

The head node software manages and controls all local nodes at a site. It provides the following services:

- Administration of the local network segment using DHCP
- Provision of boot images for the associated nodes using PXE netboot (see section II-C).
- Administration of access to the management interfaces of the local nodes via VPN².
- Proxy for monitoring that allows the central monitoring server to monitor the management interfaces (see section II-E).

²Virtual Private Network

This system is provided as a bootable ISO image that stores local changes to a disk. So all sites have the same base system with local modifications which allows for easier development.

C. Flexible Software Deployment

The local head node software of the local site provides customized boot images for the other nodes via PXE³ boot. In the context of German-Lab we define three categories of boot images:

- 1) Planet-Lab boot image (described in section II-D).
- 2) Virtualization boot image: This kind of boot image provides virtualization with access for all German-Lab users. Users can use nodes booted with this image to run customized images. As virtualizers we have developed a boot image using VirtualBox and currently develop a boot image using Xen and KVM.
- 3) Custom boot images: This kind of boot image contains a system designed by a user and only allows access to a limited user group specified by the system itself.

There is a clear trade-off between access for more users and more privileges for users. The German-Lab experimental facility allows both, access for all users to almost all nodes (Planet-Lab software is the default) and full access to a few nodes if needed.

A central management platform for distributing boot images and assigning them to the nodes has been developed and is currently being tested in Kaiserslautern and then will be deployed to the other head nodes. This front-end allows all users to schedule their resources and to set up nodes with their boot images.

D. Planet-Lab

Planet-Lab is a software, that allows to virtualize nodes using the VServer technology and which provides a central managing and control platform. There is also a testbed called Planet-Lab with which we do not share resources at the moment. The Planet-Lab software consists of a central server called Planet-Lab Central (PLC) and a boot image for all nodes. On the PLC all sites, users and nodes can be configured and a custom boot image for each node can be generated.

In German-Lab the PLC runs in a virtual machine on the head node in Kaiserslautern. The PLC controls and manages all running Planet-Lab nodes except for the boot process, which is controlled by the local head node.

E. Central Monitoring

A dedicated virtual server in Kaiserslautern is used for the monitoring infrastructure. Currently the resource usage, hardware health and availability of all nodes and service processors are monitored. The software Nagios is being used to collect monitoring data of individual hosts and services and notify administrators by e-mail when problems occur. Different tools are used to configure the monitoring and visualize the results.

The Planet-Lab part of the facility is monitored using Planet-Lab specific software (CoMon and MyOps) that has

been adapted in cooperation with Planet-Lab developers in Princeton.

F. Identity Management

The user management is an important part in an experimental facility supporting project. Especially the organization of the identity of an user and his access rights is a critical issue for designing an experimental facility. In case of the G-Lab project the user management is necessary in two different areas, the infrastructure services, and the testbed platform itself.

For services like the private Planet-Lab installation an account synchronization has been realized, so that a central LDAP database serves as master environment. This can easily be extended to future services, if required. The management of the central database follows a role based access control (RBAC) oriented model. This supports also the verification of the integrity of the stored user data.

III. OFFERED SERVICES

Besides the experimental facility the work package 7 offers other technical services to foster the collaboration of the German-Lab projects.

Mailing lists and the central website to represent the projects and allow access to news, publications and press information and a wiki to share information are hosted in Würzburg. Access to a common file server, version control storage, project management and bug tracking is hosted in Kaiserslautern.

IV. CURRENT & FUTURE WORK

On a technical level the platform can currently be used to run various software either in the Planet-Lab system, in a virtualized system or in a custom system on dedicated hardware. This provides a maximum of flexibility for experimenters and thus increases the usability of the platform.

To ensure the sustainability and continuous development of the platform a G-Lab-Association will be founded. The association will be joined by the partners of the industry and first and second phase of the project though others could also take part in.

Collaborations and federations with various testbeds all over the world like Geni, Planet-Lab, Onelab and CoreLab have been or will be negotiated. This will help to further enhance the German-Lab experimental facility and will also allow certain experiments to share the resources of these federated partners.

ACKNOWLEDGEMENTS

The authors want to thank the following persons:

Supporters: Andy Bavier, Steven Soltesz, Vivek Pai

Student workers: Andreas Teuchert, Waldemar Huber, Nicholas Gray, Johannes Wacker

³Preboot Execution Environment