German-Lab Experimental Facility

Dennis Schwerdel[1], Daniel Günther[1], Robert Henjes[2], Bernd Reuther[1], and Paul Müller[1]

[1] *University of Kaiserslautern, Integrated Communication Systems Lab*
{schwerdel,guenther,reuther,pmueller}@cs.uni-kl.de
[2] *University of Würzburg, Institute of Computer Science*
henjes@informatik.uni-wuerzburg.de

Abstract. 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 resource if necessary. We present 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.

1 Introduction

Today's Internet has a large economic influence but is based on legacy mechanisms and algorithms from the 70ies and 80ies. The rapid evolution of applications and transport technologies demands for changes even of core technologies of the Internet. Thus several research efforts worldwide currently investigate concepts and technologies for future networks. The goal of the G-Lab project is to foster experimentally driven research to exploit future network technologies.

The G-Lab project [1] has started in 2008 as a distributed joint research and experimentation project for Future Internet studies and development. Initially this BMBF[3] funded project was distributed across six universities in Germany: Würzburg, Kaiserslautern, Berlin, München, Karlsruhe, and Darmstadt. G-Lab can be divided in two major work areas, the Future Internet research and the experimental platform.

Multiple research groups focus on theoretical and practical studies from architectural questions to routing, mobility and security. The goal of the G-Lab project is not limited to explore theoretical possibilities and novel ideas but also to use experimental approaches to verify the derived results while using the experimental facility. To investigate the functional aspects of novel Internet architecture approaches (e.g. routing,

[3] German Federal Ministry of Education and Research, „Bundesministerium für Bildung und Forschung"

addressing, control, monitoring & management aspects) and their interaction with each other is such an intricate task which could not be validated only by analytical research and methods.

The project is composed of 8 working groups that are dedicated to different aspects of future Internet research: project coordination, architecture, routing, wireless and mobility, monitoring, QoS and security, service composition and the experimental facility. In the working group 7, 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 with Kaiserslautern as the main site. The first version of platform was available at March 2009 and first experiments took place at the commencement of April.

The goal if the G-Lab project is that theoretical research and the experimental facility will converge into a Future Internet as depicted in Figure 1. 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. With this G-Lab avoids the situation that the platform providers offer their services but nobody is going to use it.

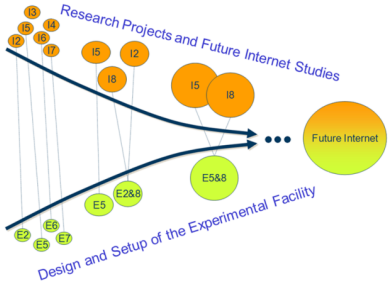


Fig. 1: German-Lab philosophy

2 Hardware Equipment

The hardware equipment consists of three types of nodes and one switch per site. The nodes can be classified in the following categories types:

Normal Node: This is the standard type of node, which can be used to run networking tests and computations.

Network Node: The second node type is designated for special networking tests requiring more network interfaces.

Head Node: The last type is acting as a head node of the local site. It has the task of managing the local site.

After vigorous scrutiny, Sun Microsystems and Cisco have been chosen as hardware provider for the facility. The technical equipment can be seen in table 1.

Table 1: Node hardware

Node Type	Chassis	CPU	RAM	Disk	Network
Head node	Sun Fire X4250	2x Xeon Quad E5450 3.0 GHz	16 GB	16x 146 GB SAS	4x 1 GBit
Network node	Sun Fire X4150	2x Xeon Quad L5420 2.5 GHz	16 GB	4x 146 GB SAS	8x 1 GBit
Normal node	Sun Fire X4150	2x Xeon Quad L5420 2.5 GHz	16 GB	4x 146 GB SAS	4x 1 GBit

All the nodes include a dedicated service processor, i.e. a small computer that allows to control and monitor the hardware remotely with a special management network interface. Each site has one head node, two network nodes and a variable amount of normal nodes as shown in table 2.

Table 2: Node counts

Site	Head Nodes	Network Nodes	Normal Nodes
University of Kaiserslautern	1	2	56
University of Würzburg	1	2	22
Karlsruhe Institute of Technology	1	2	22
University of Munich	1	2	22
University of Darmstadt	1	2	22
Universitiy of Berlin	1	2	12

The networking equipment consists of a layer-3 switch from Cisco Systems (Catalyst 4500 E Series).

3 Experimental Facility Design

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

3.1 Network Setup

All nodes of a site are located in one network segment interconnected by the switch, which has been split into two virtual switches using VLANs. The public part contains all interfaces of the normal and network nodes and all except one interface of the head node. The private part contains all management interfaces of the service processors and

one normal interface of the head node. Both networks are completely separated and only the public network has an uplink to the Internet. With this separation the access to the service processors can be controlled by the head node.

Public IP addresses are needed for all interfaces of each node (except management interface). The addresses are distributed by the head node using DHCP. Global DNS records are managed by the main site (Kaiserslautern), a site-specific zone is delegated to each site to allow decentralized DNS management.

Some sites have policies denying externally controlled nodes with IP addresses in the address range of that site, because some access rules are based on IP ranges. In this situation special firewall rules have been set up that blocks all communication between the nodes and the rest of the site except a few defined proxy hosts.

3.2 Headnode Structure

In the initial design of the experimental facility the head node has an operating system running directly on the hardware, which has early been recognized as being inflexible. Now the head node has been virtualized and separated in a couple of virtual machines. This has some major advantages:

- Different functionality can be separated into separate virtual machines. This even allows for different operating systems (e.g. Fedora Linux and Debian Linux) running on these machines.
- Virtual machines allow easy backups with snapshots of running machines.
- Virtual machines can be cloned and the clone can then be used for development and testing purposes, it can even be sent to other sites.
- The virtualization host provides a remote control (e.g. console login) over virtual machines which is an additional way of access in case the virtual machine is not working properly.

As a virtualization solution VMWare's ESXi 4 is being used but other solutions like Xen and VirtualBox are also being examined. Currently the head node in Kaiserslautern (main site) has virtual machines for monitoring (section 3.5), Planet-Lab Central (section 3.4), a file server, the head node software and various machines for testing purposes.

Headnode Software The headnode 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 3.3).
- Administration of access to the management interfaces of the local nodes via VPN[4].
- Proxy for monitoring that allows the central monitoring server to monitor the management interfaces (see section 3.5).

This system is provided as an 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.

[4] Virtual Private Network

3.3 Flexible Software Deployment

The headnode software of the local site provides boot images for the nodes via PXE[5] Netboot. Thus any boot image can be booted on any node. In the context of German-Lab we define three categories of boot images:

1. Planet-Lab boot image (described in section 3.4): This allows a node to boot the Planet-Lab software which is the default. This boot image contains a part that is specific to each node.
2. Virtualization boot image: This kind of boot image provides virtualization with access for all German-Lab users. Thus users can use nodes booted with this image to run custom software images by means of the used virtualization technology. As virtualizers we have developed a boot image using VirtualBox[2] and currently develop a boot image using Xen[3] and KVM[4].
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. Planet-Lab provides a very good virtualization when measured in the number of concurrent users that it allows, but it is very limited in the hardware access it provides (e.g. only TCP and UDP sockets, no raw sockets). Custom boot images can provide full hardware access and also allow for kernel modifications but restrict the number of users that can access the node.

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 is being developed.

3.4 Planet-Lab Usage

Planet-Lab[5, 6, 7] 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 (for which the software has been designed) with which we do not currently share resources.

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. In the Planet-Lab testbed the boot image is booted from a CD or a USB device but in German-Lab that has been modified to be used as a PXE boot image that is provided by the head node software at each site.

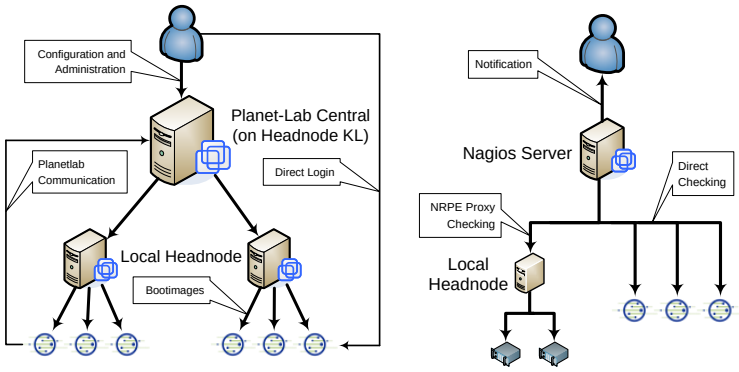
Figure 2a shows how the Planet-Lab software is used in German-Lab. The user configures its node on the PLC, which then provides a custom boot image. This boot image is used on the local headnode to boot the node via PXE. Once the node is booted, the node only communicates with the PLC and the user.

[5] Preboot Execution Environment

3.5 Central Monitoring

The monitoring of the entire infrastructure is also part of the goal. A dedicated virtual server in Kaiserslautern is used for the monitoring infrastructure. The software Nagios[8] is being used to collect monitoring data of individual hosts and services and notify administrators by e-mail when problems occur. Information that is currently monitored is:

- Resource usage (CPU, memory, disk, etc.) on all virtual machines
- Hardware health of all nodes (using the service processors)
- Availability of all nodes and service processors



(a) Planet-Lab usage (b) Monitoring

Fig. 2: German-Lab Structure

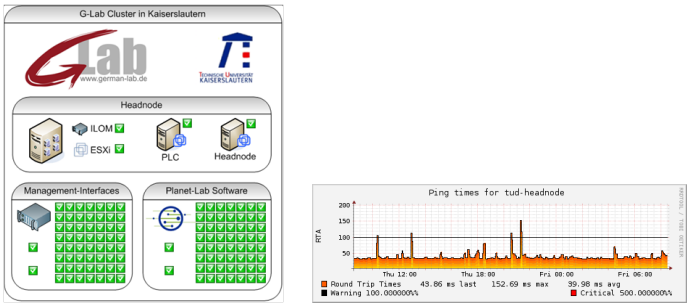
Some of this information is not visible for the monitoring server like resource usage on distant hosts and host and service information of hosts that are not visible from the server like the service processors. To allow the monitoring of these hosts and services the Nagios Remote Plugin Executor (NRPE)[9] software is being used as a proxy. NRPE is a server that allows specified hosts (i.e. the G-Lab monitoring server) to execute preconfigured commands. With this proxy both internal data and hidden hosts can be checked.

To configure the data for the Nagios software (e.g. hosts, services, check commands, users), Nagios Administrator[10] is used. The monitoring information can be visualized in two ways (see Figure 3):

1. A structure diagram gives the current state of each host or host group with green, yellow or red lights. The NagVis[11] software is used for this purpose.
2. Using PNP4Nagios[12] the history of monitored values can be visualized in a timeline graph for each host and each service.

The web-frontends of Nagios, the Nagios Administrator and both visualization tools have been combined in a central website[13]. Of course all monitoring information is also being stored in log files so that future visualization or analysis can work on the history too.

The G-Lab monitoring architecture has been valuable since it was deployed and helps to detect and solve problems quickly. Problems that can be fixed without hardware change have frequently been solved within a few hours.



(a) NagVis (b) PNP4Nagios

Fig. 3: Nagios frontends

3.6 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 in public available experimental facility design. In case of the G-Lab project the user management is necessary in two different areas, the infrastructure services, and the testbed platform itself.

The infrastructure services consist of the internal and external project documentation area, mailing lists, help desk, and software management. Whereas the testbed itself can be divided into management and experimenter view. The experimenter requires access to the nodes and testbed resources on several layers. As standard software in G-Lab, the Planet-Lab environment is used, also for the management of access rights. For deploying and operating specialized images a central account management is provided.

The administration of the users and system resources is done by a distributed administration team organized as a sub project of the overall G-Lab project. Each site might have some equipment, but at least users for the facility equipment. The approach distributes the responsibilities for the users assigned to a specific site to a representative of this site. This procedure requires additional role and access rights assignments for an extended group of identities. For example the headnodes, the node management and monitoring, and the private Planet-Lab node administration are typical tasks, which are

delegated to site representatives. Also a site representative has to organize the experiments and the resource usage of that site.

Figure 4 shows an architectural overview of the technical structure of the G-Lab identity and role dependency management. In general a central LDAP server stores the users identities in a separate subtree, which is suborganized in subtrees containing the users of a specific site. A basic rule is, that a identity is not associated with any access rights. This is organized in a separate tree, the so called group tree. Each service is represented by a unique group, which grants its members access to this server. A third separate subtree organizes virtual identities on machine level, so that each site has its own system level access user. This enables a fine grained and easy manageable environment on site level, even in case of changes. For services like the private Planet-Lab installation an account synchronization will be realized, so that the central LDAP database serves as master environment. This can easily be extended to future services, if required. The management of the central database is done by a set of scripts, which respect a set of defined default roles for specific tasks. Also these scripts verify the integrity of the stored user data.

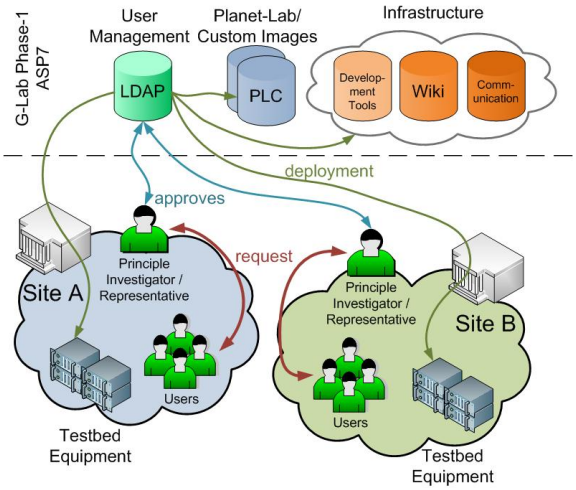


Fig. 4: Identity Management

4 Conclusion & 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 directly on the hardware. This provides maximal flexibility for experimenters and thus increases the usage of the platform. In the future the platform will be extended by a frontend

that allows all users to schedule their experiments and to set up nodes with their boot images. Also the components of the platform will be integrated even more. Monitoring experiments directly will become possible with the CoMon[7] software.

The German-Lab platform has been developed for nearly a year and is running now approximately nine months. At this point an evaluation of some decisions from the very beginning can be done. The first lesson was that virtualization is very important. It provides so many useful features even to a single system that it should be used on servers regardless whether multiple systems are needed or not. The migration from a system that is running on the hardware to a virtualized system can be complicated. An important observation is that monitoring can be very helpful when developing a testbed. So monitoring is not an additional component, it must be integrated into the architecture and should be developed as early as possible. It has also been discovered that VMWare ESXi does not provide all functionality that is needed to develop and run the experimental facility. Especially cloning or snapshotting of running virtual machines or live migration are not supported by the free version. An extended analysis of existing virtualization solutions can be worth the time because migrations from one virtualizer to another can be complicated.

To ensure the sustainability and continuous development of the platform one 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 it. The participation in the association was given special attention in the announcement of the second phase. In the past months there were several conversations, especially with industrial partners, in order to clarify whether such a platform could be used under the commercial terms and conditions. It has been experienced that manufacturers are interested and forced by quality control services to test and verify their products in a "real" environment before bringing it into the market. Which gives a developed platform extra importance in commercial market besides many infrastructure providers also shown the interest to test their product in "post-IP" environment.

4.1 Emulations of network properties

In the current G-Lab environment the network link characteristics between the clusters are excellent. In the real Internet is this not the case. We can observe different link characteristics, such as packet loss, packet delay and jitter. To provide the experiments of the G-Lab project with realistic environments these network characteristics must be emulated.

As future work we plan to create different emulation scenarios. The first scenario is the simple link emulation between two hosts, the second is the emulation of multihoming and the third is the emulation of a complex network. We also plan to develop a measurement experiment to measure the three developed scenarios. In the emulation scenarios we want to emulate different datalines like DSL, WLAN or Satellite. The value of the network parameters should be configurable at run time, which might be necessary for some experiments.

Various tools are in use today to design models equivalent or similar to actual network environments. We plan to analyze different tools like the Network Simulator, Traffic Control and Dummynet.

Bibliography

[1] German-Lab Project: German-Lab Website <http://www.german-lab.de>

[2] Sun Microsystems, Inc.: VirtualBox Website <http://www.virtualbox.org>

[3] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T.L., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xen and the art of virtualization. In Scott, M.L., Peterson, L.L., eds.: SOSP, ACM (2003) 164–177

[4] Unknown Author: Kernel Based Virtual Machine (KVM) Website <http://www.linux-kvm.org>

[5] Peterson, L.L., Roscoe, T.: The design principles of planetlab. Operating Systems Review **40**(1) (2006) 11–16

[6] Peterson, L.L., Bavier, A.C., Fiuczynski, M.E., Muir, S.: Experiences building planetlab. In: OSDI, USENIX Association (2006) 351–366

[7] Park, K., Pai, V.S.: Comon: a mostly-scalable monitoring system for planetlab. Operating Systems Review **40**(1) (2006) 65–74

[8] Nagios Enterprises LLC: Nagios Website <http://www.nagios.org>

[9] Galstad, E.: Nagios NRPE Documentation. Sourceforge.net (May 2007)

[10] secure-net-concepts GbR: Nagios Administator Website <http://www.nagiosadmin.de>

[11] NagVis Project: NagVis Website <http://www.nagvis.org>

[12] Linge, J.: PNP4Nagios Website <http://www.pnp4nagios.org>

[13] German-Lab Project: German-Lab Monitoring <http://nagios.german-lab.de>