

DSL-Lab: a Low-power Lightweight Platform to Experiment on Domestic Broadband Internet

Gilles Fedak, Jean-Patrick Gelas, Thomas Hérault, Victor Iniesta, Derrick Kondo, Laurent Lefèvre, Paul Malecot, Lucas Nussbaum, Ala Rezmerita,
Olivier Richard

► **To cite this version:**

Gilles Fedak, Jean-Patrick Gelas, Thomas Hérault, Victor Iniesta, Derrick Kondo, et al.. DSL-Lab: a Low-power Lightweight Platform to Experiment on Domestic Broadband Internet. 9th International Symposium on Parallel and Distributed Computing (ISPDC'2010), Jul 2010, Istanbul, Turkey. 2010. <inria-00502888>

HAL Id: inria-00502888

<https://hal.inria.fr/inria-00502888>

Submitted on 16 Jul 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DSL-Lab: a Low-power Lightweight Platform to Experiment on Domestic Broadband Internet

Gilles Fedak, Jean-Patrick Gelas, Thomas Herault, Victor Iniesta, Derrick Kondo, Laurent Lefèvre, Paul Malécot, Lucas Nussbaum, Ala Rezmerita and Olivier Richard
INRIA, Université de Lyon, Université Paris XI, Université Joseph Fourier Grenoble, LORIA/Nancy-Université

Abstract—This article presents the design and building of DSL-Lab, a platform to experiment on distributed computing over broadband domestic Internet. Experimental platforms such as PlanetLab and Grid’5000 are promising methodological approaches to study distributed systems. However, both platforms focus on high-end service and network deployments only available on a restricted part of the Internet, leaving aside the possibility for researchers to experiment in conditions close to what is usually available with domestic connection to the Internet. DSL-Lab is a complementary approach to PlanetLab and Grid’5000 to experiment with distributed computing in an environment closer to how Internet appears, when applications are run on end-user PCs. DSL-Lab is a set of 40 low-power and low-noise nodes, which are hosted by participants, using the participants’ xDSL or cable access to the Internet. The objective is to provide a validation and experimentation platform for new protocols, services, simulators and emulators for these systems. In this paper, we report on the software design (security, resources allocation, power management) as well as on the first experiments achieved.

Keywords-xDSL; experimentation platform

I. INTRODUCTION

The last decade has seen the development of distributed applications allowing the sharing of computing resources (CPU, storage or communication) at large scale (GRID or P2P). According to the deployment context (institutional networks or Internet) or to the application context (e-science, file sharing), these systems have renewed several research topics around large scale distributed systems.

In parallel with this evolution, high-speed Internet access has become common in home families; ADSL (Asymmetric Digital Subscriber Line) lines are wide-spread and fiber optic communication is now reaching the market. The progress realized by these technologies allows Internet provider to offer their customer an Internet connection comparable, in term of bandwidth, to 10 years old local network (up to 100Mb/sec). However, the architecture of a network of home PCs interconnected by ADSL presents special characteristics:

- the physical characteristics of the network differ substantially from the LAN characteristics, already well studied, because of the asymmetric communication performance (download/upload) and the internal ISP topologies ;
- within each family home, users share their Internet connection between several machines, using wired and/or

WiFi local network as well as NAT and Firewalls to protect their network ;

- new classes of network appliance, beside the regular PC joins this network: wifi phones, media center and IPTV, Network Attached Storage, Networked gaming console, and etc. Furthermore, the network resource might be shared between several network demanding applications (VOIP, P2P, gaming).

From the methodological point of view, evaluating the impact of broadband Internet characteristics over distributed applications present several challenges. Building an experimental testbed is a methodological approach which has already been proved to give scientists a useful tool as witnesses successful projects like PlanetLab [1], Grid’5000 [2] or DAS3 [3]. PlanetLab is a distributed platform to evaluate new classes of network services such as content delivery network, overlay network, network measurement services and many more. PlanetLab interconnects dedicated servers, spread over Internet, usually made available by research laboratories. Grid’5000 is another experimental platform to study distributed systems composed of 5000 CPU distributed over 8 sites in France. Grid’5000 is designed as a cluster of clusters interconnected by 10 Gb/s fiber links. The platform forms a VPN to launch experiment where the entire software stack is totally reconfigurable and customizable. For security issues, Grid’5000 is totally isolated from the Internet, which contrasts with PlanetLab.

While Grid’5000 and PlanetLab are valuable platforms for the study of distributed systems, the environments they provide for experiments (respectively High Performance clusters and Internet access through academic networks) are very different from the environment where most distributed applications are executed nowadays: Internet access at users’ home, using technologies such as DSL or cable. The DSL-Lab platform intends to fill this gap, by providing nodes called *DSLnodes*, located at users’ home and using their Internet connection, available permanently for scientists to run experiments on distributed systems such as desktop computing or peer-to-peer systems. At the present stage, the platform is composed of 40 nodes, covering the major French Internet Service Providers. It features a large variety of network connection (NAT, firewall) with several *DSL*

boxes¹ technologies.

Because of the constraints due to the deployment of DSLnodes at volunteers' home, the DSL-Lab middleware which manages the platform presents several innovative features compared to existing experimental platforms. We designed and developed a full software stack to make the platform lightweight for volunteers hosting DSLNodes in terms of installation, configuration and administrative task, and secure by ensuring that only authenticated users can access DSLnodes. DSL-Lab is *green* in many aspects : beside selecting low power hardware and avoiding virtualization to keep the hardware configuration and consumption minimal, the software guaranties that DSLnodes are powered off when not used. For its users, DSL-Lab is deeply reconfigurable, allowing them to deploy their own custom Linux OS. Despite being distributed on the Internet, the testbed appears to its users as a private virtual cluster, managed by a specialized version of OAR [4], a job scheduler Grid'5000 users are familiar with, which allows collaborators to reserve nodes and launch experiments.

In this article, we describe the architecture of DSL-Lab, the design decisions and implementation concerning hardware, operating system, security, network connectivity, resources management, power management and experiment deployment. We also report on several experiments performed on this testbed, either to measure its network characteristics, power consumption and estimate its hosting cost or to conduct quantitative performance evaluation of distributed computing software. We also show the flexibility of the platform by connecting it to PlanetLab and Grid'5000 during the IDHAL experiment.

The rest of the document is organized as follows: in Section 2 we describe the architecture of DSL-Lab, in Section 3, we present the early results and on-going experiments, in Section 4 we present Related Works, and we conclude in Section 5.

II. DSL-LAB ARCHITECTURE

The major part of the platform is hosted at individuals' homes, implying several unusual requirements when designing the platform. We call *volunteers* our users who agreed to host DSLnodes and *experimenters* the researchers who are using the platform. Users are asked to sign an agreement which stipulates, among others, that experimenters will be respectful of the privacy and security of the volunteers, and that volunteers are not malicious and do not try to interfere voluntarily with the experiments running on DSL-Lab. The following requirements were collected under the assumption that the agreement rules are observed.

¹DSL boxes such as FreeBox, NeufBox, LiveBox or DartyBox are the equipment given by Internet providers. These terminals usually feature more than a simple DSL modem and offer a large set of applications such as local network router, firewall, NAT, port forwarding, wifi radius, community chili-spots, NAS, telephony such as VoIP, SIP, media center, TV over IP, Video on Demand etc...

Table I
DISTRIBUTION OF DSLNODES PER DSL PROVIDER

Corporation	Number	Market share	DSL provider
Free	21	25,2 %	Free, Alice
SFR	4	31,1 %	SFR, Neuf, Cegetel, Tele 2
Orange	2	49,6 %	
Other DSL	5	2,1 %	
Noos	1		Noos, Numericable (french cable network)
Comcast	2		US cable network
	2		Assignment pending
Labs	3		LAN (DSLNodes used for development)

- DSLnodes have to be as unobtrusive as possible in terms of appearance, size, noise, light emission and power consumption;
- the platform should not compromise the security of volunteers by making their network reachable from the outside world;
- node management should be done entirely remotely and should not require any on-site intervention from the volunteers;
- for experimenters, the platform should appear as a traditional cluster in terms of providing all the tools they are familiar with.

A. Hardware and Network

To meet the previously mentioned requirements, we had to select specialized hardware so that it would be powerful enough for conducting all our experiments, but low profile enough so that it won't disturb volunteers. We selected the Neo CI852A-4RN10 barebone, which belongs to the Mini-ITX class of PC, characterized by a small size form factor, an absolute silence, thanks to the absence of fan or moving part, and low power processor. It is powered by an Intel Celeron M 1 GHz processor, 512 MB of RAM and storage is ensured by 2 Gb of Compact Flash storage. The node also has 4 Ethernet ports so that it can be used eventually as a router to conduct networking experiments. An interesting feature is that this hardware does not allow volunteers to interact with the nodes, which prevents the node from being used for personal purpose other than the scientific experiments. The total cost for the 40 DSLnodes and the central server is approximately 26K€.

The DSL-Lab central server is connected to the French academic backbone (Renater) through the University Paris-Sud network. This backbone is large and peers almost directly with all French ISP, so network quality depends mostly on the ISP and the volunteer's last mile.

As of January 2009 (see Table I), the DSLnodes were distributed as follows: 32 were hosted on the French major DSL providers (Orange, Free, Neuf, Tele2), 1 on the French

cable network (Numericable), 2 were hosted on the US cable network, 3 were distributed to Orsay, Lyon and Grenoble laboratories for development purposes and the remaining 2 were waiting for new volunteers. Some of our nodes may switch to fiber networks in 2010.

The DSL “network” is very heterogeneous and offers a wide range of different environments. There are 4 technologies allowed in France: ADSL, ADSL2, ADSL2+ and ReADSL. In addition, some volunteers have the ability to fine tune their ADSL link. For instance, some are using the “fastpath” which reduces the ping delay but is more prone to transmission errors. Also the phone line length, ranging from 50 m to several kilometers for the set of DSLnodes, and the domestic electrical environment impacts the quality of the DSL link and available bandwidth and latency.

In this project, we had to develop our own software stack, which provides the following features:

B. Remote OS Deployment

The DSL-Lab system is able to deploy remotely a new OS on every DSLnode without asking for volunteer intervention. One of the major concerns when designing the DSL-Lab platform was to avoid as much as possible volunteer intervention on the nodes. So, we needed to be able to re-install (in case an experimenter breaks the installed system by mistake) and upgrade (for security reasons, or to install additional software) the whole software stack, including the operating system installed on DSLnodes.

It was chosen not to use an approach based on virtualization, i.e. using a virtual machine containing the experimentation system running on the node. Firstly, VM-based virtualization solutions, such as Xen, require dedicated RAM for the virtual machine. For cost and power consumption reasons, the nodes that were chosen only provide 512 MB of RAM, which is too little to run an host system and a virtual machine and the guest OS. Secondly, virtual machines often cause a performance overhead, or limit what is possible to do with the system. For example, on PlanetLab, a special API has to be used to manipulate raw sockets. On DSL-Lab, direct access to the bare machine is provided to experiments, without any limitations.

To be able to deploy or upgrade the experimentation system on DSLnodes, two disk partitions are used for two different systems:

- a small partition (5 MB) with a read-only, minimal system based on a customized TTYLinux distribution is used for remote OS deployment and as a fallback in case of problems. As such, the experimenter is not allowed to modify it in any circumstance. This minimal system has been thoroughly tested to ensure maximum reliability;
- a larger partition, with the experimentation system based on a full fledged Debian GNU/Linux distribution, that experimenters are allowed to modify.

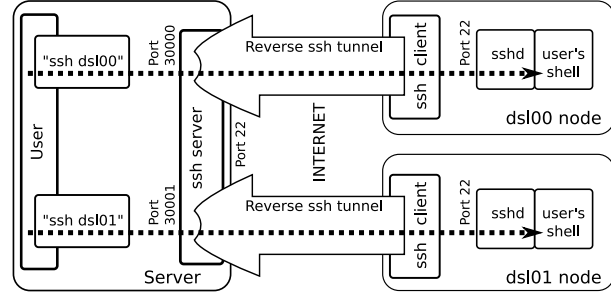


Figure 1. Establishment of connectivity layer within DSL-Lab using tunnel and reverse ssh. The node automatically maintains an SSH connection to the DSL-Lab server, providing a tunnel that users can use to connect to the node.

The bootloader is configured to alternate booting on the first and the second partition.

The first partition will never be modified. It contains the bootstrap code that connects to the central DSL-Lab server, checks for an updated version of the experimentation operating system or a special request to re-install the DSLnode, and proceeds accordingly. After checking if re-installation is necessary, the DSLnode reboots the experimentation system, and awaits the experimenter’s instructions.

Using that system, it is possible to safely and completely upgrade all DSLnodes remotely, allowing experimenters to deploy their own customized Linux distribution.

However, our current implementation is vulnerable to local attacks. For example, a malicious experimenter could alter the experimentation system and make it unbootable, requiring manual intervention to get the system back up and running. A solution could be to use a hardware Watchdog to solve this case, but the one provided on DSLnodes failed to work. Another problem is that the minimal system is not write-protected. Hence, a malicious experimenter could also modify that system. PlanetLab, for example, uses a read-only medium (CDROM, write-protected USB key) to avoid this problem.

C. Connectivity and Security

The platform is managed in such a way that only identified experimenters have access to it. Experimenters first log into the server, which acts as a gateway and provides remote access to each DSLnode within a VPN through SSH.

To minimize the number of listening ports visible from the Internet, and avoid entering in conflict with the volunteer’s security policy, there is no publicly accessible SSH daemon running on DSLnodes. Instead, the DSLnode establishes an SSH connection, including an SSH tunnel to its own SSH server port, to the DSL-Lab central server. That SSH TCP tunnel can later be used by authenticated experiments to connect to the DSLnode using SSH (see Figure 1). This creates a VPN that connects the nodes together without modifying volunteers’ security settings. To prevent the drop

of the reverse connection caused by a TCP timeout in the case of no activity, we configured SSH to use a small keep-alive. In addition, the connection aliveness is checked every 10 minutes, and the ssh client from the DSLnode to the DSL-Lab server is restarted if necessary.

As the nodes are distributed to volunteer’s home, we do not trust the software and private keys that they contain. One malicious volunteer, or the person who compromised their network may retrieve the private key and try to connect to the SSH server with it. For this reason, the key only permits to create the SSH tunnel and cannot execute anything on the DSL-Lab central server.

On DSLnodes, only 4 ports, 2 TCP and 2 UDP, are inevitably used for experiments. Even if the experimental software behind those ports are not proved to be secure, they are running for short periods of time and with unprivileged accounts. To open direct connections between hosts, we are integrating a new solution called PVC (Private Virtual Cluster), which is described in the next section.

D. Resources and Power Management

Most of the DSL-Lab experimenters are familiar with the Grid’5000 platform. To leverage their knowledge acquired on Grid’5000, we have adapted the Grid’5000 batch scheduler, called OAR, to the DSL-Lab platform so that: 1/ experimenters have a similar work environment and 2/ it would eventually facilitate the connection of both platforms. Thanks to OAR, several experimenters may reserve some nodes in advance and deploy their experiments simultaneously.

Because DSLnodes are hosted on a volunteer basis, a request of the volunteers is that the DSLnode does not waste power. Besides selecting thrifty hardware, the system ensures that the DSLnodes are powered-off when not used, thus reducing electricity consumption. To do so, DSLnodes periodically boot and check with the server if they have been reserved using OAR. The node then stays up (if the next reservation starts soon enough), or re-schedules its next wake-up time accordingly. Wake-ups are done using ACPI alarms, and have proved to be very reliable. Even if not reserved, the nodes wake up on a regular basis to check if new reservations have been created. Their wake-up times are distributed within the work day, so that experiments are guaranteed to be able to access a node in less than 30 minutes, even without reserving a node previously.

III. FIRSTS EXPERIMENTS AND RESULTS

In this section, we report on the first experiments that were ran on DSL-Lab. The power consumption of DSLnodes is first evaluated, to verify that hosting a DSLnodes do not cost an unreasonable price. We also measure the bandwidth and latency of the network connection available from the DSLnodes. Two different kinds of distributed systems were also studied. First, PVC, that allows the creation of a private

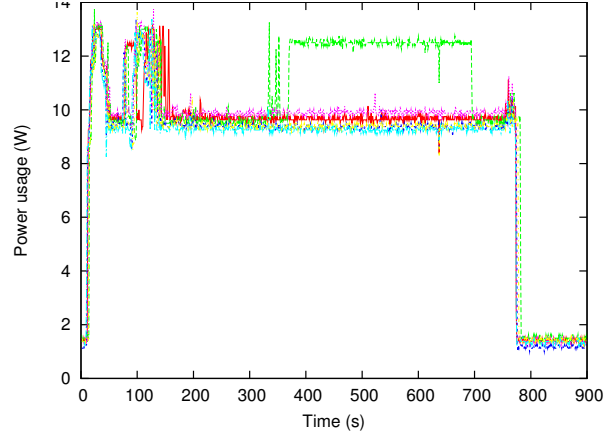


Figure 2. Power consumption of 6 DSLnodes during the boot process, low or full usage (during CPU intensive usage), and shutdown.

Table II
LATENCY OF 11 DSLNODES - MARCH 2009

City	ISP	DSL latency	Tot. latency
Chaville	Free (v.5)	22.18 ms	25.14 ms
Lyon	SFR	37.75	46.47
Versailles	Free (v.5)	4.07	7.14
Lyon	Free	33.5	42.38
Grenoble	Free (v.5)	44.61	54.92
Le Touvet	Free (v.4)	46.85	57.76
Paris	Free (v.5)	33.63	37.23
Saint Gratien	Free	37.86	41.85
Bezons	Free	14.27	17.05
Lyon	SFR	27.29	36.47

virtual cluster. Then, several Desktop Grid scenarios are executed on DSL-Lab, providing a real-world experimentation environment. Finally, the possibility to connect DSL-Lab with PlanetLab and Grid’5000, is demonstrated.

A. Power Consumption of DSLNodes

First, we evaluate the power consumption of DSLnodes, in order to price DSLnode hosting as our volunteers are not refunded for the electricity consumed. Linux’s frequency scaling is enabled using the on-demand governor and nodes are shutdown when unused by experimenters. Figure 2 presents real power usage as measured with a power sensor [5]. A node consumes around 1.5W when turned off (due to wake-on-LAN and software power switch) ; 9-10W when the CPU is idle and 13-14W at 100% CPU load. According to the French regulated electricity price, the cost per hour is 0.00182€ if the DSLnode is reserved and 0.000195€ otherwise. If we assume usage of the platform to be 8 hours a day, 5 days a week, 11 months a year, it costs 3.57€/year for a volunteer to host a DSLnode.

B. Latency and bandwidth

Table II presents the latency observed from DSLNodes on two french DSL providers Free and SFR. ADSL latency is

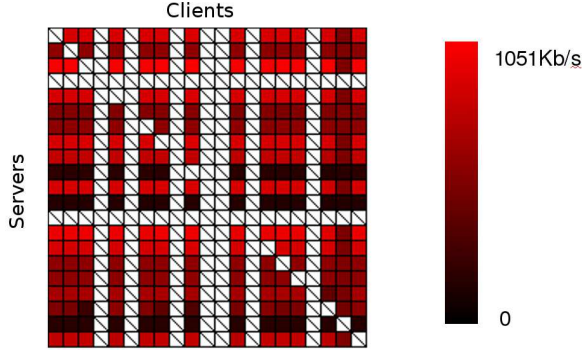


Figure 3. Communication matrix between half of the DSLnodes. The color shows the TCP/IP bandwidth averaged over 8 measures.

between DSLNode and their corresponding DSLAM (i.e. provider’s modem). Total latency is the latency between the DSLNode and the DSL-Lab server. Both latencies are measured using the ICMP Ping protocol. This shows that most latency is introduced during ADSL line traversal. This is mainly due to the interleaving methods that delays data over time to resist interferences. There are different interleaving settings reflected by different ADSL latency.

Then, we characterize our network by building a communication matrix between DSLNodes. Figure 3 shows the DSLnode to DSLnode communication bandwidth where direct connection between DSLNodes is possible. Blanks shows that for some nodes incoming communications are restricted by NAT or firewalls. Because ADSL is an asymmetric protocol that have a significantly slower up-link than down-link, communications speed are limited by sending node uplink bandwidth. Up-link uses a small number of DSL channels and bandwidth is thus dramatically limited. The up-link capacity is also limited by physical line conditions which also explain the great differences between DSLnodes.

Determining realistic simulation or emulation parameters is often challenging, especially when dealing with networks like DSL. DSL-Lab allowed the capture of typical latency, bandwidth and packet loss rates for DSL. Those parameters were later used during the validation and the comparison of several network emulators (NISTNet, Dummynet, TC/Netem) [6].

C. Direct Communications between DSLnodes: Private Virtual Cluster (PVC)

The objective of Private Virtual Cluster (PVC) [7] is to provide, in a transparent way, an execution environment for existing cluster applications over nodes distributed on the Internet. Thanks to its capability to dynamically connect firewall-protected machines, without any intervention of domain administrators, without modification of the existing security policy, and with minimal performance overhead, PVC can serve as a middleware in order to test the network performance, with standard tools.

Table III
CONNECTIVITY TECHNIQUES USED ON DSL-LAB PLATFORM

ISP	Nodes	Connection establishment method
Free v4	5	UPnP or Traversing-TCP
Free v5	16	UPnP or Proxy
SFR	4	UPnP or Traversing-TCP
Orange	2	UPnP or Traversing-TCP
Comcast	1	Traversing-TCP
Noos	1	UPnP or Traversing-TCP
Lab.	2	Proxy

The validation of the PVC design was performed using the DSL-Lab testbed. Initially DSL-Lab made it possible to survey the various characteristics of the firewalls included in the DSL boxes.

According to the different kinds of existing firewalls, four different connectivity techniques were integrated in PVC: UPnP, TCP Hole punching, Traversing TCP and Proxy. The TableIII summarizes the connectivity techniques which allow direct connections between DSLNodes depending on the DSL provider and the DSL boxes technologies. The development, testing and evaluation of the connectivity technique has been done on DSL-Lab.

A large part of the DSLnodes is connected to the Free provider, who supplies two types of box ADSL (FreeBox v4 and v5). The version FreeBox v4 implements a classical dynamic NAT with port translation. The version v5 of the FreeBox improves the quality of the router by implementing a stateful firewall, but forbids the Traversing-TCP method to open direct communication. After 2007, this type of router became more and more frequent on broadband Internet, which was impacting dramatically the effectiveness of the PVC framework when running on DSL-Lab. Thanks to this real world deployment, the issue was identified, and it led to the implementation of the proxy communication method within PVC framework.

Once the connectivity technique for each type of firewall was identified and implemented in PVC, two additional types of experiments were performed on the DSL-Lab testbed. The first one evaluated the overhead of the system for the connection establishment using a specific test suite. The second one demonstrated PVC’s capabilities to execute unmodified MPI applications (NAS benchmarks, MPIPOV, DOT) deployed over a set of machines behind firewalls blocking incoming connections, connected to the Internet by ADSL connections.

D. Evaluation of Desktop Grid Platforms

DSL-Lab is the platform of choice to perform experiments on Desktop Grid middleware. This approach complements Grid’5000 where the experimental conditions are controlled and reproducible but far from real life Internet condition. BitDew [8] is a middleware dedicated to data management on Desktop Grid. In addition with traditional micro-

benchmarks performed on Grid'5000, several real-life scenarios have been run on DSL-Lab. These scenarios include multi-source file distribution, fault-tolerance, collective file distribution with replication and wide-area file transfers featuring P2P protocols (BitTorrent) as well as Amazon S3 and IBP protocols. The experiments on the DSL-Lab platform have shown that the BitDew middleware was able to cope with complex scenarios in situations close to real-world deployment. We report on two scenarios that were previously published in [8] to illustrate the specificity of DSL-Lab with respect to quantitative performance evaluation.

The first scenario aims at illustrating data resiliency feature of BitDew. The scenario consists of the following : 5 data replica are created and scheduled to 5 DSLnodes. Every 20 seconds, a crash is simulated by killing the BitDew process on the DSLnode owning the data, and simultaneously a new host arrival is simulated by starting BitDew on another DSLnode. The scenario verifies that data replica are automatically rescheduled and downloaded by new hosts.

During the execution on the scenario on DSL-Lab, dates of the various stages are recorded : node arrival, data schedule, beginning and end of data download and crashes. Figure 4 shows the Gantt chart of the experiment where the horizontal axis is the time and the vertical axis represents the hosts. The Gantt chart presents the main events of the fault tolerance scenario: red box is the time between the arrival of a host and the start of the file transfer waiting time, the blue box shows the download duration and red star indicates a node crash. The right most vertical axis gives the bandwidth obtained during the file download whose size is 5MB. One can observe a waiting time of 3 seconds before the download starts, which is due to the failure detector.

We can also observe a great variation in the communication performance between the hosts. This can be explained by the difference of service quality between the various Internet Service Providers and by the fact that users' applications consuming bandwidth might be running at the time of the experiment. ISPs advertise that the ADSL download bandwidth is up to 20Mbit/s and 1Mbit/s for upload. But these bandwidths are not always observed in practice because of phone line length and attenuation. For instance, DSL06 is dramatically slow compares to other DSLnodes. After running these scenarios on DSL-Lab, the BitDew middleware has been enhanced with a monitoring mechanism which allows to detect faults during file transfer to improve the failure detector responsiveness.

Real world parallel applications often require collective communication to exchange data between computation steps. The second scenario experiment with a communication pattern similar to the *All-to-all* collective associated with data replication. All-to-all file transfer starts with a set of nodes, each one owning one different file. At the end of the collective, all the nodes should own all the files.

On the DSL-Lab platform, experimental conditions can

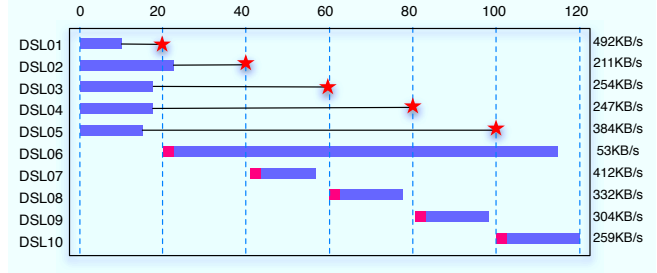


Figure 4. Evaluation of Bitdew in presence of host failures. The rightmost part of the graph presents the bandwidth obtained during the file transfer.

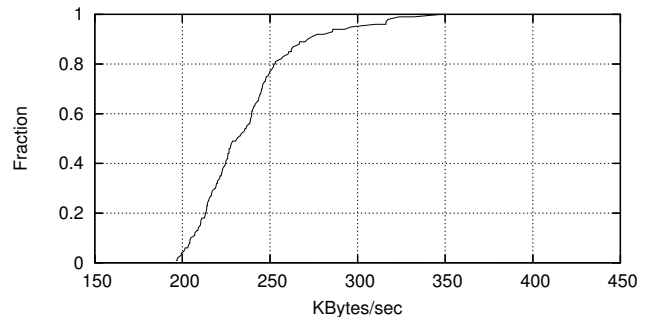


Figure 5. Cumulative Distribution Function (CDF) plot for collective all-to-all as measured on the 16 DSL-Lab nodes.

vary. One run was executed every hour during approximately 12 hours. During the collective, file transfers are performed concurrently and the data size is 5MB. We measure, on each DSL-Lab node, the duration of the all-to-all collective and the Figure 5 shows the cumulative distribution function (CDF) plot for the bandwidth obtained during the all-to-all 5MB file transfer. We can observe great variations in the measurement between consecutive runs. Under this circumstances of low reproducible experiments, it is harder to conclude about the performances compared to experiences run within the Grid'5000 platform.

In partnership with C. Cerin from Université Paris XIII, we are now evaluating a new Desktop Grid protocol called BonjourGrid [9], which is dedicated to service discovery.

E. Integrating DSL-Lab with other platforms

In the IDHAL experiment, the possibility of connecting various experimental platforms together was investigated. Several sets of nodes, from different platforms, were used: DSLnodes, Grid'5000 nodes, nodes from PlanetLab, desktop computing nodes located at a university in Grenoble, clusters in Luxembourg and Porto Alegre (Brazil), and virtual machines. Two challenges arose: being able to connect those machines together despite the differences in the way the platforms are connected together, and using all the platforms together ; the aim was to be able to use them together to do a distributed computation.

Two approaches were combined to connect all the platforms together. On most of the nodes, a virtual private network (VPN) tunnel was set up to a well-connected single node using *OpenVPN*. (The possibility to use several nodes as VPN servers did not prove necessary.) On the PlanetLab nodes, where setting up a tunnel was not possible, TakTuk [10] was used to set up an overlay communication network.

After connecting all the nodes together, we focused on using them all for a single computation, using the KAAPI library [11] to compute instances of the N-Queens problem. Since the nodes had very different resources (both compute power and network), it did not make sense to simply split the computation equally on all nodes. Instead, the work-stealing capabilities of KAAPI allowed it to efficiently share the computation on all nodes. While the goal of this experiment was mainly to prove the feasibility of using so many different resources together, it was still possible to solve several instances of the N-Queens problem, using different combinations of nodes.

During these experiments, several nodes encountered problems and suddenly stopped functioning, outlining the need for improvements in KAAPI in the area of fault tolerance.

IV. RELATED WORKS

Prior to DSL-Lab, some platforms were already dedicated to scientific research on large-scale Internet distributed systems. In this last section, we describe some of them and how DSL-Lab differs by providing a complementary experimental setup.

Grid'5000 [2] is a French academic cluster of clusters spread over 9 French locations. It gathers about 4800 cores of 4300 processors that are part of 1600 nodes. Due to its main purpose for experimentation and in contrast to traditional grids, it has been designed to be easily re-configured and managed. All experiments are isolated from each other and from the Internet. Grid'5000 features a wide variety of high-speed networks and high performance computers. Other similar platforms like DAS-3 [3] from the Netherlands achieve similar goals.

DSL-Lab tries to retain all the features that have made Grid'5000 a success by adapting them to residential Internet nodes. We use the same OAR scheduler [4] but, as detailed previously, added power saving features and improved disconnected host management. Another example is the deployment software. For this, Grid'5000 includes KaDeploy [12] but this optimizes communications only for clusters and requires server management cards (IPMI). We had to rewrite a similar tool able to deal with Internet communications. The second difference is that our software is not able to take deployed OS control back due to lack of hardware support on our nodes. The third difference is

the lack of reproducible experimental conditions, which, similarly to PlanetLab is due to the Internet network.

Apart from the isolated grids world, there are platforms that include some real Internet links between their nodes. Experiments would be feasible with Volunteer Desktop Grids, but even if they offer a great number of nodes, the experimenters have very limited control of the volunteer's resources. In particular, there is no guarantee that volunteered desktop would provide full CPU power and availability 7 days a week [13]. In comparison, dedicated nodes provide a more flexible experimental setup.

PlanetLab [1] is a worldwide platform gathering about 900 nodes from 472 laboratories locations. It was originally a centralized network managed by the Princeton University, but it now federates various administrative domains and projects. Nodes are shared among experimenters using virtualization (Linux VServer). In contrast, DSL-Lab allows, optionally, exclusive full access to nodes in order to prevent interactions with others experiments (required on low bandwidth links).

The fact that there is no virtualization also allows DSL-Lab nodes to require less power and to consume less hardware resources than PlanetLab nodes. For example, the absence of virtualization allows us to use up to 2 GB of storage and 512 MB of RAM, which is the full system capacity. The lack of virtualization also suppresses the additional network latency during network time measurements, thus providing more precise measurements.

One of the commonly noticed limitations of the original PlanetLab project is that nodes offer only a small heterogeneity as they are mostly located in laboratories. Several projects have been added to the PlanetLab federation to deal with that issue.

Firstly, the EU project OneLab, the European branch of PlanetLab, have made some studies that cover radio links such as 802.16, UTMS and WiFi. They plan to work on the other types such as Bluetooth. They also provides network monitoring services.

Secondly, SatelliteLab [14] is an extension of PlanetLab which includes nodes on the edges of Internet. The testbed consists of *planets* nodes formed by PlanetLab nodes and *satellites* nodes which are desktops, laptops and PDA connected to the Internet through DSL, Wifi, Bluetooth and cellular links. The requirements of PlanetLab hosts are very high. Server class machines running the PlanetLab OS must be configured with a static and public IP address, and this requirement excludes most of the nodes on the edges of Internet. With SatelliteLab, the application code of an experiment runs on the planets, whereas the satellites only forward network traffic. In contrast, the DSL-Lab software stack has been designed to run on a machine with very low requirements (less than 2 GB of storage) hosted on a domestic DSL, which allows experimenters to run experiments directly on each node of the platform.

Besides PlanetLab, Emulab, originating from Utah University, is a software and a platform of 450 nodes. Experiments are driven using a similar interface to the Network Simulator. It also offers access to some wireless links, sensor networks, access to PlanetLab nodes and can use emulation. Experiments, even if they get exclusive access to nodes, can be swapped out and may share local networks with other nodes. DSL-Lab has only one node per broadband line and thus does not suffer from this limitation. EmuLab is still being improved with projects such as FlexLab that try to restart experiments automatically as long as unrealistic conditions are detected by its network observations.

V. CONCLUSION

Grid'5000 and PlanetLab have paved the way to establish new category of research tools for science of distributed computing. DSL-Lab is a complementary platform which allows scientists to experiment on the broadband DSL Internet. DSL-Lab consists of *DSLnodes*, located at the edge of Internet available permanently for scientists to run experiments on distributed systems such as desktop computing or peer-to-peer systems.

In this article, we described the design and the architecture of the DSL-Lab platform, as well as experiments using the platform that show its usefulness. To summarize our main achievements:

- We have built a new experimental platform deeply reconfigurable by its users. After a careful selection of the hardware, we have designed and developed a complete software stack which features OS deployment, resource management, power efficiency, security and connectivity. The platform is currently deployed over 40 nodes since April 2007. Thanks to its lightweight and low-cost design, the platform can be easily extended with additional resources.
- DSL-Lab has lead to several advances in evaluating Desktop Grids (for example, BitDew, PVC, Bonjour-Grid). To the best of our knowledge, none of the existing Desktop Grid middleware have been designed and developed using an experimental testbed in the context of the DSL broadband Internet. We are now opening the platform to other areas of distributed computing research (self-stabilization, mobile computing etc...).
- We have successfully connected DSL-Lab with two other main experimental platforms, namely PlanetLab and Grid'5000. While these early results are promising, we plan to focus our next work on bridging Grid'5000 and DSL-Lab in order to make DSL-Lab available for the entire Grid'5000 community.

ACKNOWLEDGMENTS

This research has been supported by the French ANR "Young Researcher" program (see <https://www.dsllab.org>) under grant JC05_55975.

REFERENCES

- [1] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, "Planetlab: an overlay testbed for broad-coverage services," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 3, pp. 3–12, 2003.
- [2] F. Cappello and all., "Grid'5000: A large scale and highly reconfigurable grid experimental testbed," in *GRID '05: Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing*, Washington, DC, USA, 2005.
- [3] "DAS-3: The distributed ascii supercomputer 3," <http://www.cs.vu.nl/das3/>.
- [4] N. Capit, G. D. Costa, Y. Georgiou, G. Huard, C. Martin, G. Mounié, P. Neyron, and O. Richard, "A batch scheduler with high level components," in *Cluster computing and Grid 2005 (CCGrid05)*, 2005.
- [5] A.-C. Orgerie, L. Lefèvre, and J.-P. Gelas, "Chasing gaps between bursts : Towards energy efficient large scale experimental grids," in *PDCAT 2008 : The Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies*, Dunedin, New Zealand, Dec. 2008.
- [6] L. Nussbaum and O. Richard, "A comparative study of network link emulators," in *Communications and Networking Simulation Symposium (CNS'09)*, San Diego, USA, 2009.
- [7] A. Rezmerita, T. Morlier, V. Néri, and F. Cappello, "Private virtual cluster: Infrastructure and protocol for instant grids," in *12th International Euro-Par Conference*, Germany, 2006.
- [8] G. Fedak, H. He, and F. Cappello, "BitDew: A Programmable Environment for Large-Scale Data Management and Distribution," in *Proceedings of the ACM/IEEE SuperComputing Conference (SC'08)*, Austin, USA, November 2008.
- [9] H. Abbes, C. Cérin, J.-C. Dubacq, and M. Jemni, "Bonjourgrid: A community decentralized desktop grid." xW'08 : XtremWeb users group workshop.
- [10] "TakTuk: Adaptive large scale remote executions deployment," <http://taktuk.gforge.inria.fr/>.
- [11] T. Gautier, X. Besson, and L. Pigeon, "Kaapi: A thread scheduling runtime system for data flow computations on cluster of multi-processors," in *PASCO '07: Proceedings of the 2007 international workshop on Parallel symbolic computation*, 2007.
- [12] Y. Georgiou, J. Leduc, B. Videau, J. Peyrard, and O. Richard, "A tool for environment deployment in clusters and light grids," in *Second Workshop on System Management Tools for Large-Scale Parallel Systems (SMTPS'06)*, Greece, 2006.
- [13] P. Malecot, D. Kondo, and G. Fedak, "Xtremlab: A system for characterizing internet desktop grids (abstract)," in *Proceedings of the 6th IEEE Symposium on High-Performance Distributed Computing*, 2006.
- [14] M. Dischinger, A. Haeberlen, I. Beschastnikh, K. P. Gummedi, and S. Saroiu, "Satellitlab: adding heterogeneity to planetary-scale network testbeds," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 315–326, 2008.