SEVENTH FRAMEWORK PROGRAMME
Theme 3
Information and Communication Technologies

Deliverable D2.3
Testbed usage report

Grant Agreement number:	**287581**
Project acronym:	**OpenLab**
Project title:	**OpenLab: Extending FIRE testbeds and tools**
Funding Scheme:	**Large scale integrating project**
Project website address:	**www.ict-openlab.eu**
Date of preparation:	**June 19th, 2013**

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Document properties

People	
Document Editor:	Jordan Augé (UPMC)
Authors:	Yahya Al-Hazmi (TUB) Jordan Augé (UPMC) Scott Kirkpatrick (HUJI) Sándor Laki (ELTE) Jorge E. Lpez de Vergara Mndez (UAM) Ciro Scognamiglio (UPMC) Michael Sioutis (UPMC) József Stéger (ELTE) Alexander Willner (TUB)

Abstract

This document presents the progress with respect to usage and activity monitoring and reporting in the OpenLab project, and more specifically to the task 2.3 of Work package 2 as per the Project's "Description of Work".

Contents

List of Figures

List of Tables

Acronyms

Introduction

This document presents the progress with respect to usage and activity monitoring and reporting in the OpenLab project, and more specifically to the task 2.3 of Work package 2 as per the Project's "Description of Work" [8].

This section presents an introduction to the notions of usage and activity. It explains why monitoring them is important for the testbeds and the federation, presents a set of metrics of interest, and suggest how they could be measured.

The testbeds involved in OpenLab are of different nature and maturity, and do not all present the same level of functionalities. It was thus important to make a census of the different testbeds, to raise awareness about the importance of such monitoring, to learn about what was currently being done, and to be able to share both tools and best practices. This work This work would be an input to improv

Section 2 reports about the different tools that are used and/or developed within OpenLab, that could be used for collecting and reporting useful information in a more general context, both at the testbed and at the federation level.

Section 3 presents the results that have been collected within the project, and analyze them. This task guides the decision taken within the project with respect to usage and activity monitoring, and serves as an input to the next section.

Finally, Section 4 presents the tool that has been put in place to mutualize the resources about usage and activity monitoring, named FreeTRex. This tool is used both to collect information about the different testbeds, present available tools and share best practises.

A last section (Section 5) is attached to this deliverable while it is not directly related to usage and activity monitoring. It reports on the outcome of the first OpenLab plugfest, during which the ability of cross-testbed tools to function on multiple testbeds have been evaluated. While the scope of this plugfest was larger than this of this document, work described in subsubsection 5.3 is of particular interest to the present context.

1 Usage and activity monitoring and reporting

Responsible: UPMC, HUJI + contributions welcome

1.1 Motivations for collecting usage and activity data

Usage and activity monitoring and reporting is often considered a nice-to-have functionality. It is in fact essential for running and operating platforms such as those in OpenLab. We describe in the following the three main aspects.

Daily engineering tasks and planning Obviously, getting monitoring data and alerts assessing the health of the platform helps daily debugging and operations. Better knowing users and their requirements, as well as digging through historical data is fundamental to identify bottlenecks and plan future platform evolutions [10].

For testbeds that are running on the public internet such as PlanetLab, the support has to handle security incidents linked to the activity of the slice. Proper knowledge of the experiments being carried on can help providing a fast and accurate answer.

Reporting to sponsors and to the public Of utmost importance also is the reporting to sponsors and to the general public. Huge investments of public money are done in the projects, and such metric represent simple proofs that the platform is used and has an interest for researchers, for the industry, etc. Sponsors need to asses that they funding is well invested, and understand who it benefits. This is also the opportunity to advertise to the general public the use of these infrastructures, both at events and through the public project website. Some use case could be highlighted showcasing innovative features, and would help understanding the impact of the testbeds.

Guiding policies and informing about platform value Finally, usage and activity monitoring can support the definition of federation policies, as well as assess the value of platforms contributing heterogeneous resources, with respect to the user demand [6].

1.2 How to monitor usage and activity

Usage and activity monitoring and reporting consists in detecting, measuring, understanding and representing the usage of testbed resources, and the related user activities.

Control plane We will first look at metrics that can be measured automatically, by monitoring the control plane of the testbed. While each testbed is made of different software bricks, usually homemade, we can still understand its basic mechanisms by looking at the component we are using to federate testbeds, the Slice-Based Federation Architecture (SFA).

SFA defines four objects corresponding to the main entities exposed by the testbeds:

Authorities: These represent testbeds, parts of testbeds to which trust or rights may be delegated, and/or communities of users.

Resources: These consist of nodes, links, or any other experimental resource exposed to the users.

Users: These are experimenters wanting access to resources.

Slices: A slice is the basic unit of interaction between users and resources. One can think of a slice as corresponding to an experiment, and encompassing all users and resources associated with that experiment.

These objects and their relationships are convenient for defining several basic metrics:

- the number and type of resources provided by the testbed (eventually grouped by authority);
- the number of (active) users of the testbed (eventually grouped by authority);
- the number of (active/finished) experiments, or slices, and their characteristics in terms of associated users, resources and their duration.

We see that slices that are at the heart of the architecture are also the most interesting aspect of usage and activity monitoring. They consist in what experimenters are using the testbed for. As such, we recommend collecting additional information about these experiments when users are requesting access to those slices. In PlanetLab Europe for example, users are requested for a description of their experiments and a URL presenting their project. As explained later, this information comes very important for the testbed support to diagnose the cause of potential security incidents, etc. This minimal set of information will be enriched on the portal we will offer to experimenters, but we will take care of not asking too much to experimenters no to discourage them to use the testbed, or we will make sure that they can get some benefits in return.

Experimental plane The metrics we have just presented can provide information about the usage of the platform. Through, it might not be sufficient to reveal user activity on the different platforms. Suppose that we plan to display user activity in real time on a map, most of the user actions on the control plane will be punctual and quite unfrequent: users will book a slice and add resources, then they will perform one of several experiments by directly (or indirectly) accessing the resources, and never contacting the control plane again.

Without additional monitoring of the experimental plane, we cannot distinguish whether a slice is idle or not. One possibility could be to refer to some tools that will instrument the interaction of the user with the experimental plane, such as OMF. Another one could be harvesting the different logs of the machine (ssh connections for example), etc. Again this would add information, but we would miss activity from non-interactive experiments.

We have in fact evidence of such activity, though the impact of the experiment on the resources the testbeds provide. It can be measured in terms of CPU activity, memory consumption and bandwidth usage. This is exactly what is provided by the different measurement and monitoring solutions in the context of OpenLab. As such, the different tools we are developing and integrating for monitoring are also of interest to this deliverable. And the same will hold for federation tools allowing the aggregation of this data. The challenge will be to define the appropriate activity metrics.

These solutions could easily be extended to aggregate usage and activity information itself, since the different testbeds and tools will likely construct databases with the evolution of this information with time.

2 Monitoring tools

This section briefly overviews the monitoring tools that are available in different OpenLab testbeds and can provide data for usage and activity reporting at testbed level. At federation level, information from heterogeneous data sources must be gathered and presented to the users. To this end, we also introduce the Manifold framework that gives an easily extensible way to testbed operators to make their testbeds manageable through MySlice [3] which is expected to be the primary resource browsing interface of OpenLab. As an alternative to Manifold, we also show that the SPQR framework originally developed to harmonize network measurement data can also be used as a common interface for usage and activity data stored in heterogeneous databases for different testbeds and tools. SPQR can integrate and harmonize heterogeneous repositories storing testbed specific usage statistics, and provides a semantic query interface to query the usage metrics in the same way for all the integrated testbeds, hiding the underlying differences including the different unit metrics that may be applied.

Responsible: ELTE/UAM + contributions for each testbeds/tool developer

2.1 Tools available for testbed monitoring

2.1.1 PlanetLab

PlanetLab monitoring tools can be divided into three categories:

Infrastructure monitoring: Monitoring of the infrastructure of the testbed is done with Nagios[1] and NRPE[2]. Nagios is installed on a dedicated server and uses NRPE to remotely execute Nagios plugins on other servers. The NRPE service is deployed on the servers hosting all PlanetLab Europe services. Automatic deployment on the servers of the NRPE service is done using a custom configuration of Puppet[3], thus giving the advantage of easily adding and removing service monitoring components from a central point and rendering the infrastructure dynamic.

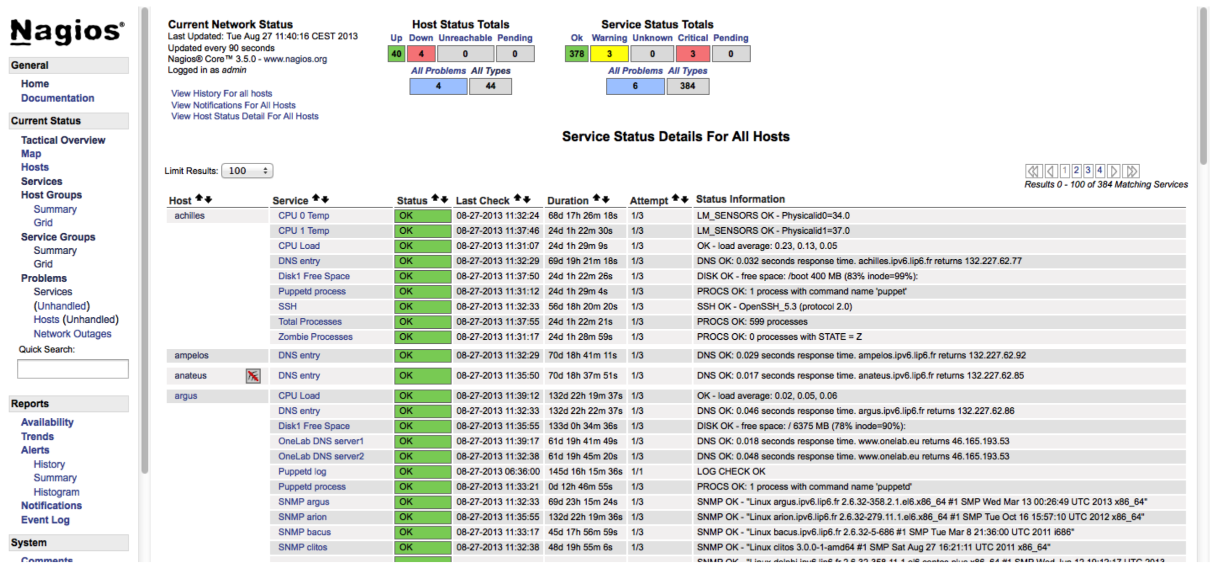
Services monitored include machines health status and specific PlanetLab services or components, in particular we take into account service availability, software malfunctioning (by monitoring logged information) and security checks.

Node operation monitoring: Nodes and their operational state are monitored mainly through the use of MyOps[4]. MyOps allows node and site monitoring and also implements escalation policies. Detailed information on node operation are reported:

1. status: if online or in a filesafe state due f.i. to hardware malfunctioning.
2. ping response time, traceroute and DNS status.
3. service and firewall status: SSH and PlanetLab node manager service in particular.

[1]http://www.nagios.org
[2]http://nagios.sourceforge.net/docs/nrpe/NRPE.pdf
[3]http://puppetlabs.com
[4]http://www.cs.princeton.edu/ soltesz/dl-myops/MyOps.pdf

Figure 1: Nagios monitoring of the services and infrastructure of PlanetLab Europe



Current Network Status
Last Updated: Tue Aug 27 11:40:16 CEST 2013
Updated every 90 seconds
Nagios® Core™ 3.5.0 - www.nagios.org
Logged in as *admin*

Host Status Totals

Up	Down	Unreachable	Pending
40	4	0	0

All Problems	*All Types*
4	44

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
378	3	0	3	0

All Problems	*All Types*
6	384

Service Status Details For All Hosts

Results 0 - 100 of 384 Matching Services

Host	Service	Status	Last Check	Duration	Attempt	Status Information
achilles	CPU 0 Temp	OK	08-27-2013 11:32:24	68d 17h 26m 18s	1/3	LM_SENSORS OK - Physicalid0=34.0
	CPU 1 Temp	OK	08-27-2013 11:37:46	24d 1h 22m 30s	1/3	LM_SENSORS OK - Physicalid1=37.0
	CPU Load	OK	08-27-2013 11:31:07	24d 1h 29m 9s	1/3	OK - load average: 0.23, 0.13, 0.05
	DNS entry	OK	08-27-2013 11:32:29	69d 19h 21m 18s	1/3	DNS OK: 0.032 seconds response time. achilles.ipv6.lip6.fr returns 132.227.62.77
	Disk1 Free Space	OK	08-27-2013 11:37:50	24d 1h 22m 26s	1/3	DISK OK - free space: /boot 400 MB (83% inode=99%):
	Puppetd process	OK	08-27-2013 11:31:12	24d 1h 29m 4s	1/3	PROCS OK: 1 process with command name 'puppet'
	SSH	OK	08-27-2013 11:32:33	56d 18h 20m 20s	1/3	SSH OK - OpenSSH_5.3 (protocol 2.0)
	Total Processes	OK	08-27-2013 11:37:55	24d 1h 22m 21s	1/3	PROCS OK: 599 processes
	Zombie Processes	OK	08-27-2013 11:31:17	24d 1h 28m 59s	1/3	PROCS OK: 0 processes with STATE = Z
ampelos	DNS entry	OK	08-27-2013 11:32:29	70d 18h 41m 11s	1/3	DNS OK: 0.029 seconds response time. ampelos.ipv6.lip6.fr returns 132.227.62.92
anateus	DNS entry	OK	08-27-2013 11:35:50	70d 18h 37m 51s	1/3	DNS OK: 0.017 seconds response time. anateus.ipv6.lip6.fr returns 132.227.62.85
argus	CPU Load	OK	08-27-2013 11:39:12	132d 22h 19m 37s	1/3	OK - load average: 0.02, 0.05, 0.06
	DNS entry	OK	08-27-2013 11:32:33	132d 22h 22m 37s	1/3	DNS OK: 0.046 seconds response time. argus.ipv6.lip6.fr returns 132.227.62.86
	Disk1 Free Space	OK	08-27-2013 11:35:55	133d 0h 34m 36s	1/3	DISK OK - free space: / 6375 MB (78% inode=90%):
	OneLab DNS server1	OK	08-27-2013 11:39:17	61d 19h 41m 49s	1/3	DNS OK: 0.018 seconds response time. www.onelab.eu returns 46.165.193.53
	OneLab DNS server2	OK	08-27-2013 11:32:38	61d 19h 45m 20s	1/3	DNS OK: 0.048 seconds response time. www.onelab.eu returns 46.165.193.53
	Puppetd log	OK	08-27-2013 06:36:00	145d 16h 15m 36s	1/1	LOG CHECK OK
	Puppetd process	OK	08-27-2013 11:33:21	0d 12h 46m 55s	1/3	PROCS OK: 1 process with command name 'puppetd'
	SNMP argus	OK	08-27-2013 11:32:33	69d 23h 15m 24s	1/3	SNMP OK - "Linux argus.ipv6.lip6.fr 2.6.32-358.2.1.el6.x86_64 #1 SMP Wed Mar 13 00:26:49 UTC 2013 x86_64"
	SNMP arion	OK	08-27-2013 11:35:55	132d 22h 19m 36s	1/3	SNMP OK - "Linux arion.ipv6.lip6.fr 2.6.32-279.11.1.el6.x86_64 #1 SMP Tue Oct 16 15:57:10 UTC 2012 x86_64"
	SNMP bacus	OK	08-27-2013 11:33:17	45d 17h 56m 59s	1/3	SNMP OK - "Linux bacus.ipv6.lip6.fr 2.6.32-5-686 #1 SMP Tue Mar 8 21:36:00 UTC 2011 i686"
	SNMP clitos	OK	08-27-2013 11:32:38	48d 19h 55m 6s	1/3	SNMP OK - "Linux clitos 3.0.0-1-amd64 #1 SMP Sat Aug 27 16:21:11 UTC 2011 x86_64"

4. PCU status: a Power Control Unit permits operators to remotely control and check node hardware health status.

Site monitoring reports for each PlanetLab Europe registered site the number of active nodes, number of used slices, if enabled and eventually if a penalty has been applied to the site. As for PlanetLab Europe policies a site has to have at least two operating nodes, failing to do so MyOps automatically applies a penalty and eventually disables the site if no resources are active for a longer period of time. MyOps also mantains an history of node and site status information.

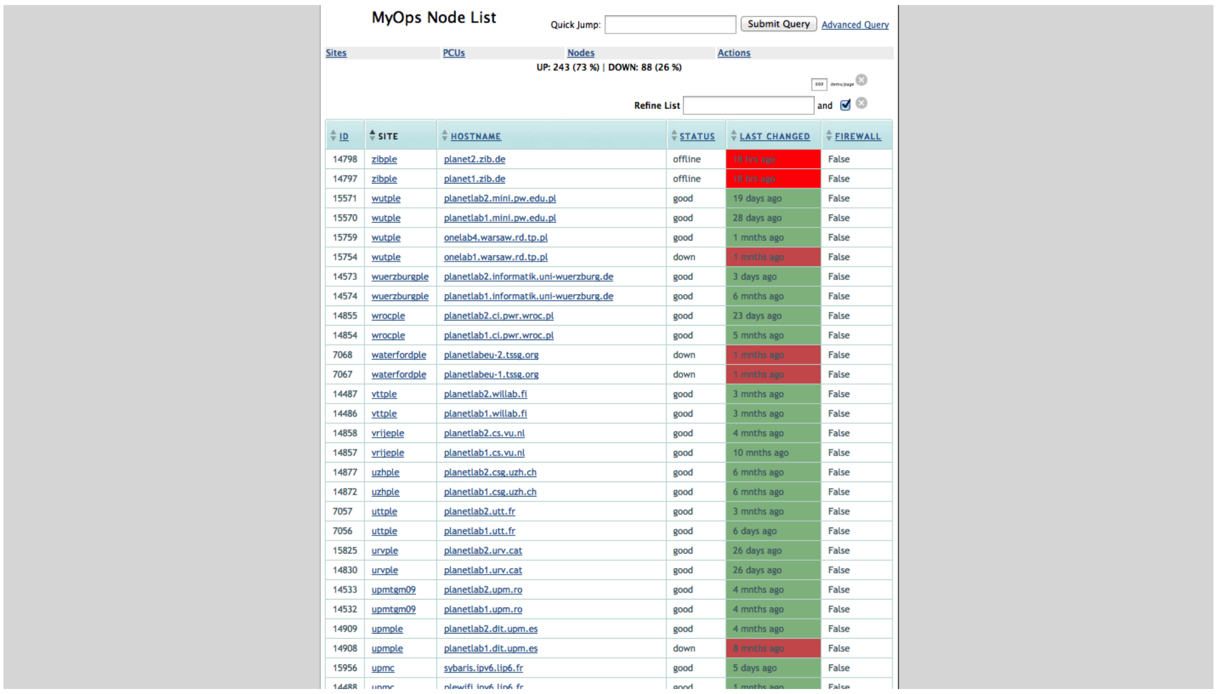
Users can access MyOps through the web interface[5] or programmatically by using a very simple RESTful[6] API interface.

Experiment monitoring: Slicestat is a service used for experiment monitoring that runs on each node and that exposes information on the experiments running on the node. Such information can be retreived and analysed by simply accessing the service on the node on port 3100 with the http protocol. For example `http://ple2.ipv6.lip6.fr:3100/slicestat` will report informations on slivers running on the UPMC hosted node. Slicestat reported information is retreived and stored on a central server every 15 minutes, the resulting data can be analysed and used for displaying experiment resource usage statistics as shown in figure 3.

As part of the CoMon[7] project for monitoring infrastructure of PlanetLab, Slicestat only supports collecting resource usage statistics of Linux-VServer based containers. PlanetLab Europe is currently migrating towards the use of LXC based containers for managing slivers on

[5]http://monitor.planet-lab.eu/monitor/
[6]http://en.wikipedia.org/wiki/Restful
[7]http://comon.cs.princeton.edu

Figure 2: MyOps monitoring node status on the testbed

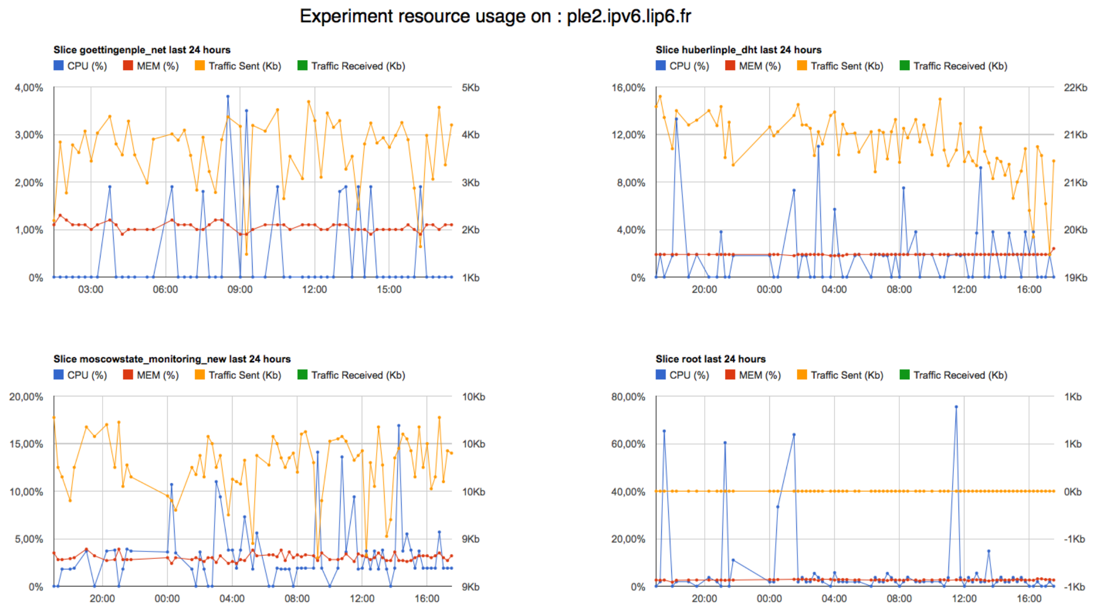


MyOps Node List

UP: 243 (73 %) | DOWN: 88 (26 %)

ID	SITE	HOSTNAME	STATUS	LAST CHANGED	FIREWALL
14798	zibple	planet2.zib.de	offline	18 hrs ago	False
14797	zibple	planet1.zib.de	offline	18 hrs ago	False
15571	wutple	planetlab2.mini.pw.edu.pl	good	19 days ago	False
15570	wutple	planetlab1.mini.pw.edu.pl	good	28 days ago	False
15759	wutple	onelab4.warsaw.rd.tp.pl	good	1 mnths ago	False
15754	wutple	onelab1.warsaw.rd.tp.pl	down	1 mnths ago	False
14573	wuerzburgple	planetlab2.informatik.uni-wuerzburg.de	good	3 days ago	False
14574	wuerzburgple	planetlab1.informatik.uni-wuerzburg.de	good	6 mnths ago	False
14855	wrocple	planetlab2.ci.pwr.wroc.pl	good	23 days ago	False
14854	wrocple	planetlab1.ci.pwr.wroc.pl	good	5 mnths ago	False
7068	waterfordple	planetlabeu-2.tssg.org	down	1 mnths ago	False
7067	waterfordple	planetlabeu-1.tssg.org	down	1 mnths ago	False
14487	vttple	planetlab2.willab.fi	good	3 mnths ago	False
14486	vttple	planetlab1.willab.fi	good	3 mnths ago	False
14858	vrijeple	planetlab2.cs.vu.nl	good	4 mnths ago	False
14857	vrijeple	planetlab1.cs.vu.nl	good	10 mnths ago	False
14877	uzhple	planetlab2.csg.uzh.ch	good	6 mnths ago	False
14872	uzhple	planetlab1.csg.uzh.ch	good	6 mnths ago	False
7057	uttple	planetlab2.utt.fr	good	3 mnths ago	False
7056	uttple	planetlab1.utt.fr	good	6 days ago	False
15825	urvple	planetlab2.urv.cat	good	26 days ago	False
14830	urvple	planetlab1.urv.cat	good	26 days ago	False
14533	upmtgm09	planetlab2.upm.ro	good	4 mnths ago	False
14532	upmtgm09	planetlab1.upm.ro	good	4 mnths ago	False
14909	upmple	planetlab2.dit.upm.es	good	4 mnths ago	False
14908	upmple	planetlab1.dit.upm.es	down	8 mnths ago	False
15956	upmc	sybaris.ipv6.lip6.fr	good	5 days ago	False
14488	upmc	plewifi.ipv6.lip6.fr	good	1 mnths ago	False

Figure 3: Slicestat monitoring statistics for slivers on a PLE node



nodes, and as Slicestat is incompatible with this new infrastructure, the resource monitoring service will be replaced by Collectd[8], an actively developed and supported open source project that can report the same information as slicestat and also can be easily extended through the use of plugins.

PlanetLab usage monitoring and statistics Other tools exist for the reporting of general information on the testbed usage and monitoring. In particular statistic on various aspects of the testbed are collected and displayed on `http://stats.planet-lab.eu`. This service reports testbed user and site member statistics, information on node availability and some basic detail on the type of resources available on the testbed.

Within the user interface of the MySlice[9] portal a plugin has been developed to monitor activity of the testbed. The plugin has been developed in the context of a project named PRES VIEW by four students during their master thesis. It can display information on site and user activity geolocalized on a map and can be used to retreive such activity by specifying a date range and other filters. The information will be displayed on a map built using the google map API as displayed in figure 5. Information displayed includes user activity through the MySlice portal and the PLCAPI (logins, slice creation etc.) and resource usage (slice activity on selected nodes).

[8]http://collectd.org
[9]http://www.myslice.info

Figure 4: PlanetLab Europe general statistics on testbed usage and status

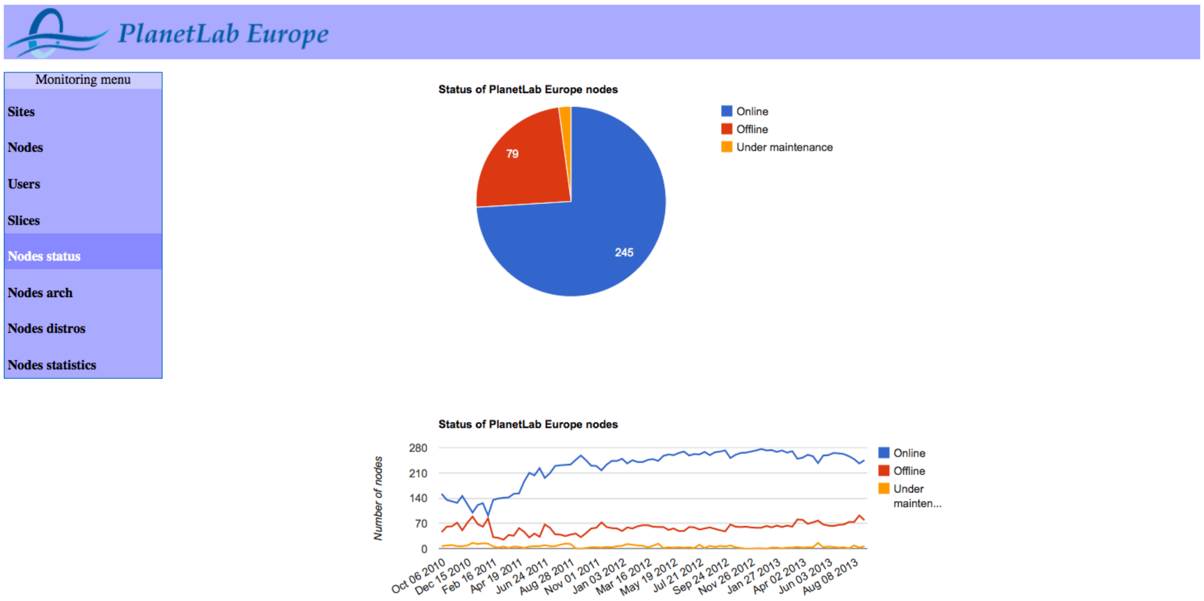
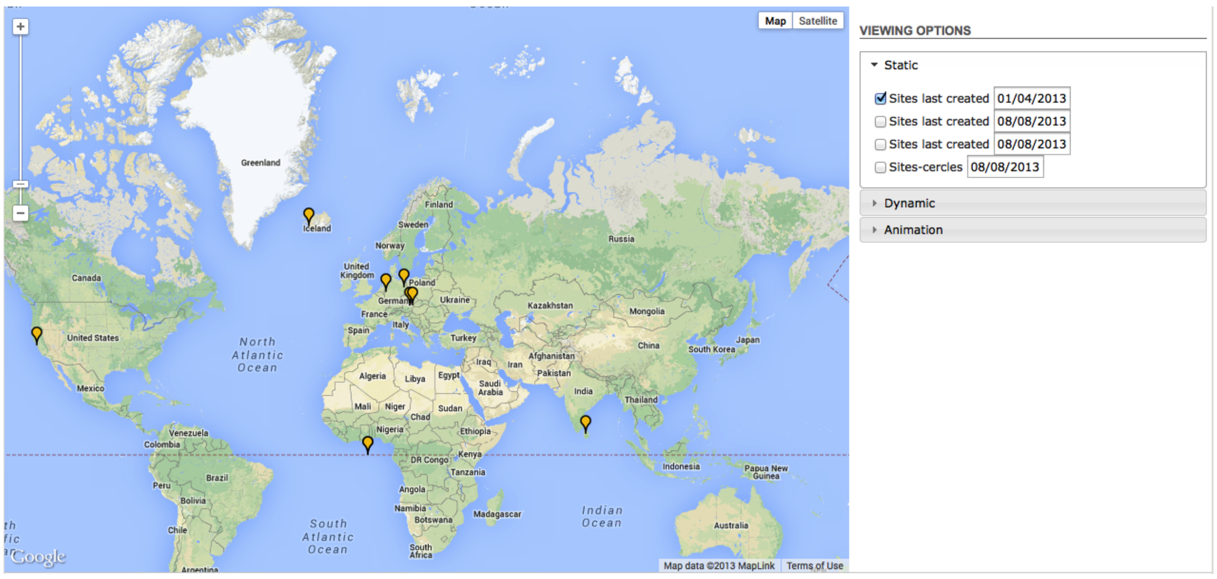


Figure 5: The PRES VIEW service displays site and user activity in time



VIEWING OPTIONS

Static

- ☑ Sites last created 01/04/2013
- ☐ Sites last created 08/08/2013
- ☐ Sites last created 08/08/2013
- ☐ Sites-cercles 08/08/2013

Dynamic

Animation

PlanetLab Europe use cases another additional instrument used to monitor user activity on the PlanetLab Europe testbed has been the selection and report of use cases[10]. In PlanetLab Europe users are required to describe their experiment when first creating their slice, this information is useful to identify the type of experiment and the type of usage expected (e.g., network load, traffic type or resource usage load). Together with monitoring history on the slices and nodes we can identify the most active and interesting experiments and by the means of a survey collect more detailed information.

PlanetLab Europe Web site monitoring and statistics Useful information on how the testbed is accessed and which features are more requested is gathered by monitoring the PLE Web site and Portal. The web site access statistics are collected and updated every day using AWStats[11], an opensource Perl based log analysis tool. While AWStats provides us with generic access statistics, we also use Google Analytics to monitor access to specific sections of the web site, documentation and the portal, thus enabling us to have a more complete overview of the user activity.

2.1.2 ETOMIC at UPNA

ETOMIC testbed at UPNA provides an application to show its usage statistics. The application is available at http://casper.tlm.unavarra.es/graficas/index.php. Figure below shows the application:

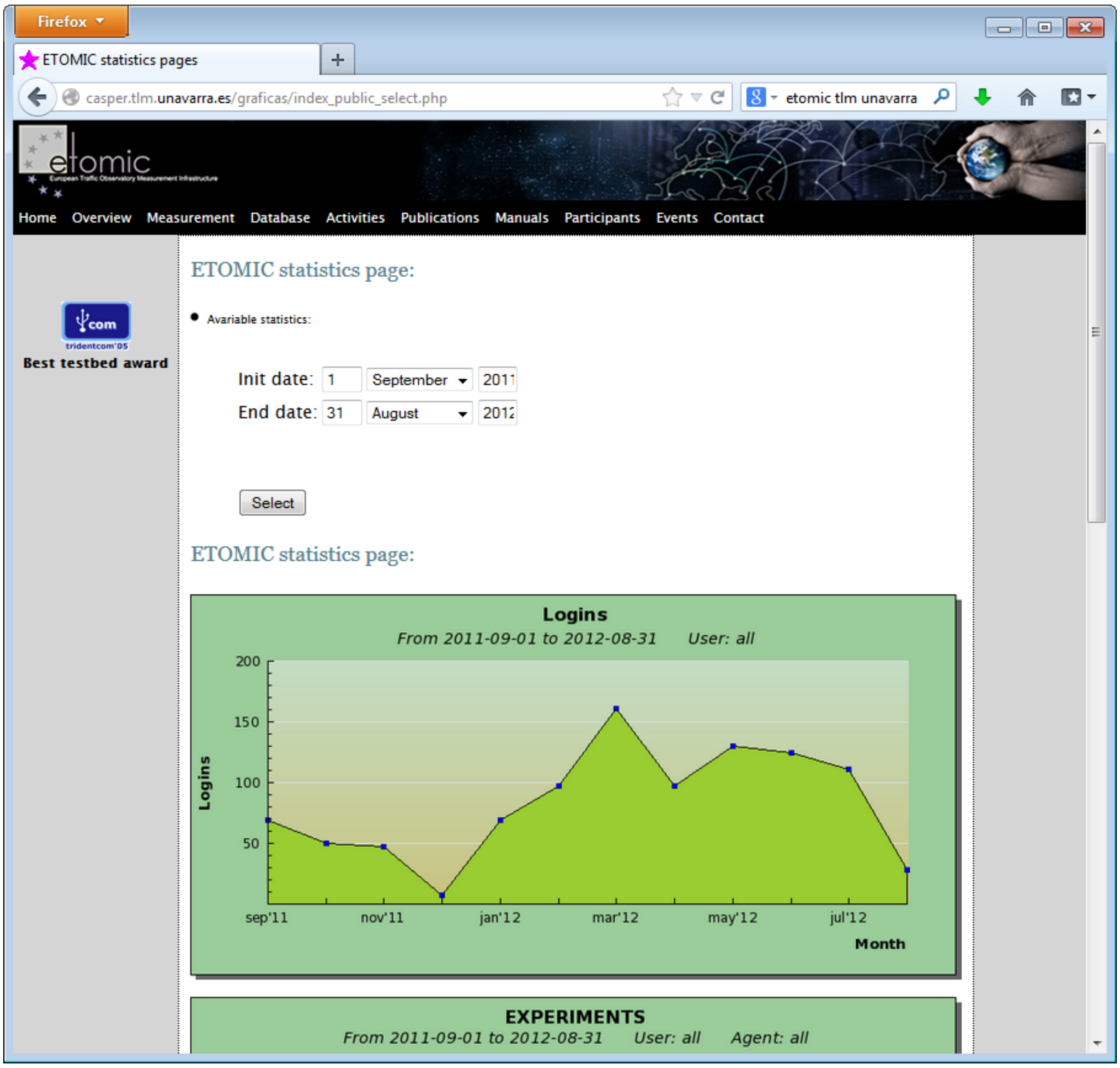
The application is very simple to use: the user just selects statistics period to show (based on an initial date and a final date), and then press Select button to obtain the usage statistics. When the user sends the dates, the application queries its database, and generates several plots with the number of logins, experiments performed, Mbytes downloaded, and the running experiments at each agent. All these statistics are represented grouped in months, as shown below.

2.1.3 SONoMA-v2 periodic measurements

First of all, SONoMA [11] is not a testbed, but a network measurement tool that can work in various environments. As described in D2.2 [5] the SONoMA network measurement framework was originally developed for the network measurement community, providing them with an easy-to-use tool to carry out large-scale network measurements from heterogeneous networking elements (PlanetLab node, BlackFin-based APE boxes [9], etc.). In the past half year, the framework has been redesigned, especially focusing on questions like how it could be more flexible and extensible, supporting the needs of infrastructure monitoring. To this end, we extended the framework with many new features:

1. Flexible tool extension. In the former SONoMA version, the SONoMA measurement agents had to implement the predefined measurement types, and provide a SOAP-based web service interface through which the central management system were able to contact them. In the new SONoMA framework, agents do not deal with measurement implementations, since the new model assumes that there are great many underlying tools for different

[10] http://www.onelab.eu/index.php/services/testbed-access/use-cases.html
[11] http://awstats.sourceforge.net



ETOMIC statistics page:

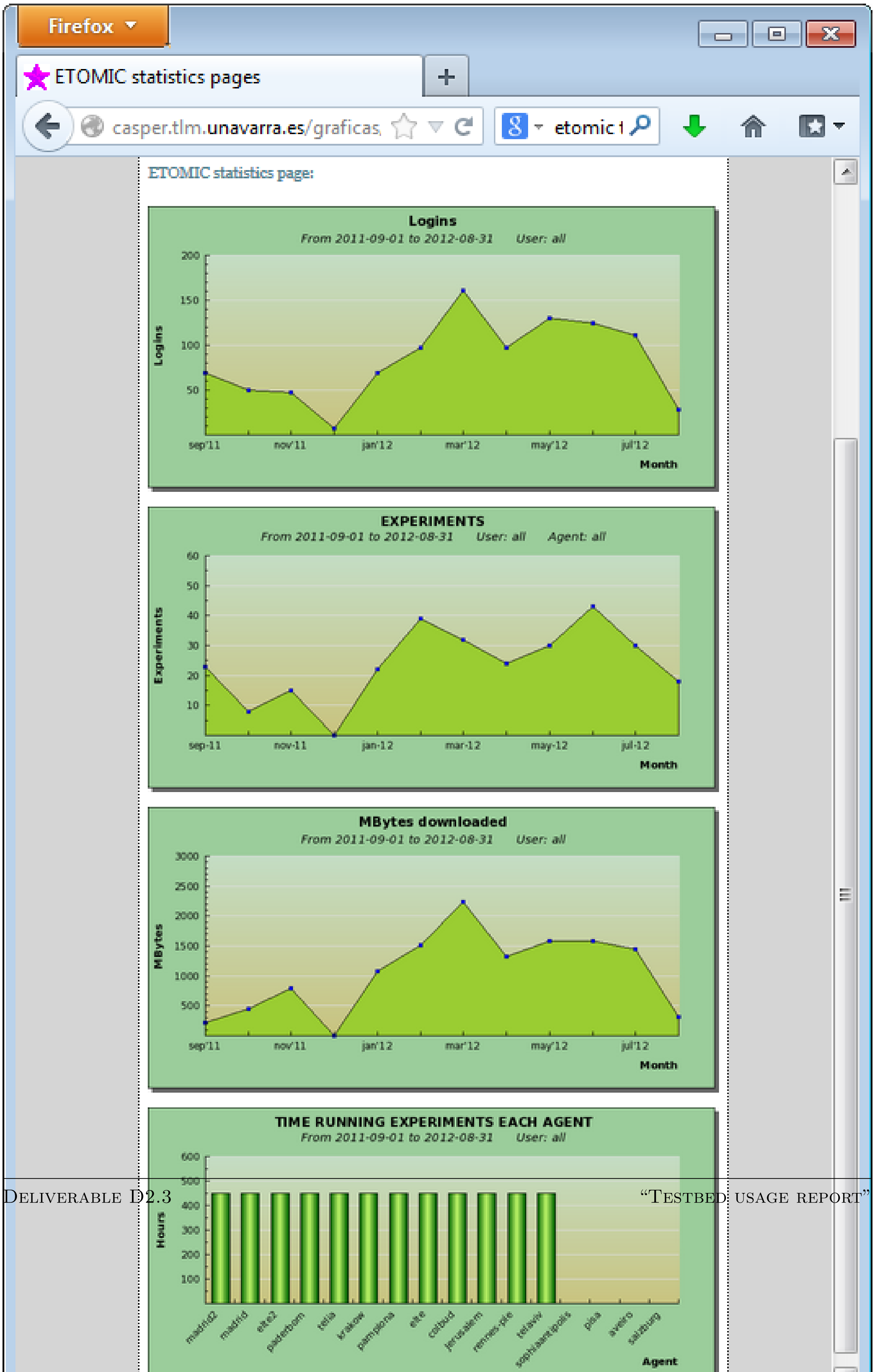
- Avariable statistics:

Init date: 1 September 2011
End date: 31 August 2012

ETOMIC statistics page:

Logins
From 2011-09-01 to 2012-08-31 User: all

EXPERIMENTS
From 2011-09-01 to 2012-08-31 User: all Agent: all


Firefox
ETOMIC statistics pages
casper.tlm.unavarra.es/graficas
etomic t
ETOMIC statistics page:
Logins
From 2011-09-01 to 2012-08-31 User: all
Logins
200
150
100
50
sep'11
nov'11
jan'12
mar'12
may'12
jul'12
Month
EXPERIMENTS
From 2011-09-01 to 2012-08-31 User: all Agent: all
Experiments
60
50
40
30
20
10
sep-11
nov-11
jan-12
mar-12
may-12
jul-12
Month
MBytes downloaded
From 2011-09-01 to 2012-08-31 User: all
MBytes
3000
2500
2000
1500
1000
500
sep'11
nov'11
jan'12
mar'12
may'12
jul'12
Month
TIME RUNNING EXPERIMENTS EACH AGENT
From 2011-09-01 to 2012-08-31 User: all
Hours
600
500
400
300
200
100
madrid2
madrid
elte2
paderborn
tella
krakow
pamplona
elte
colbud
jerusalem
rennes-ple
telaviv
sophiaantipolis
pisa
aveiro
salzburg
Agent

passive and active network and other measurements. This model also enables to extend the framework with new tools (e.g. new bandwidth estimation command line tools, etc.). Currently supported metrics include available bandwidth (iperf-based), round trip delays (using ping or fping), cpu load and memory usage (from /proc).

2. Harmonizing network tools and network measurements. Various testbeds could provide different tools and capabilities to measure the same metric or different versions of the same tool may result different outputs that need to be handled. To harmonize different outputs, the new framework requires the testbed operator to define the expected output for each tools, with the proper metric type and unit.

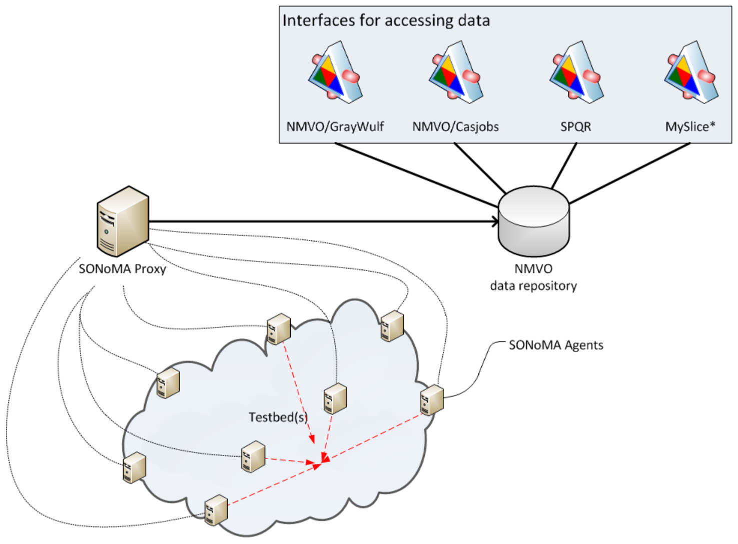
3. Periodic measurements. In contrast to the former SONoMA, where measurements were only carried out on demand, the new variant allows to define periodic and continuous measurements as well. In this way, the platform can be instrumented to perform active or passive measurements periodically, e.g. in every 10 minutes, providing testbed users, operators or other testbed components with accurate and up-to-date information on the available resources and their states. Since numerous command line tools are available that can be used from this framework, with the support of periodic measurements the new SONoMA tool could provide the capability of continuous monitoring.

4. Permanent storage of data. One of the key advantage of the original SONoMA system is that all the measurement data is automatically stored in a permanent repository called Network Measurement Virtual Observatory (nmVO), enabling researchers to query and analyze their historical data via a SQL-based querying interface. The new SONoMA variant also supports this feature, but this is done more efficiently in a two-stage fashion. After the measurement results are sent back to the central management system, the collected data is first stored in a temporal SQLite repository, serving as a fast first stage database. The records from this database are then transferred to the permanent database of nmVO. Since the new SONoMA framework supports various measurement tools, and it is easily extensible, the back-end schema must also be flexible, supporting the extension with new metrics and tools.

Architecture The architecture overview of the SONoMA framework for periodic measurements is depicted in Figure 6. The SONoMA proxy plays the role of a central management system that controls external and periodic measurements, collect and store measurement data. All the data are permanently stored in our nmVO repository and made available for OpenLab and other network measurement communities through various query interfaces. We note that nmVO gateway for MySlice which is the primary platform for users to browse testbed resources is still under development and the first release is expected to be published in 2013 September.

In contrast to the former SONoMA version, the new system follows a pull-based approach to distribute network measurement tasks to the measurement agents, thus each agent sends requests to the proxy periodically to obtain a list of measurement tasks to be carried out. As illustrated in Figure 7, agents first place the incoming tasks into a task queue from where they are processed and executed one after another. Each monitoring task uses an underlying tool that has previously been installed on the host. Various tools can provide different interfaces from

Figure 6: Architecture overview of the SONoMA framework (* The nmVO gateway for MySlice is still under development.)



command line to web services as well. Through its driver system the framework can interoperate with various network measurement tools. However, different tools returns their results in various formats using heterogeneous data units which must be handled, converted and stored in a local cache. Finally, the framework periodically moves the data from the temporal storage to the permanent database of nmVO.

Deployment Currently, the new SONoMA framework is deployed in PlanetLab system, but after testing we are planning to deploy it in other testbeds as well. The new agents are running in a dedicated slice, called elteple_sonomaperiodic, and carry out ping and available bandwidth measurements in every 10 minutes in a full-mesh fashion between all the PlanetLab nodes.

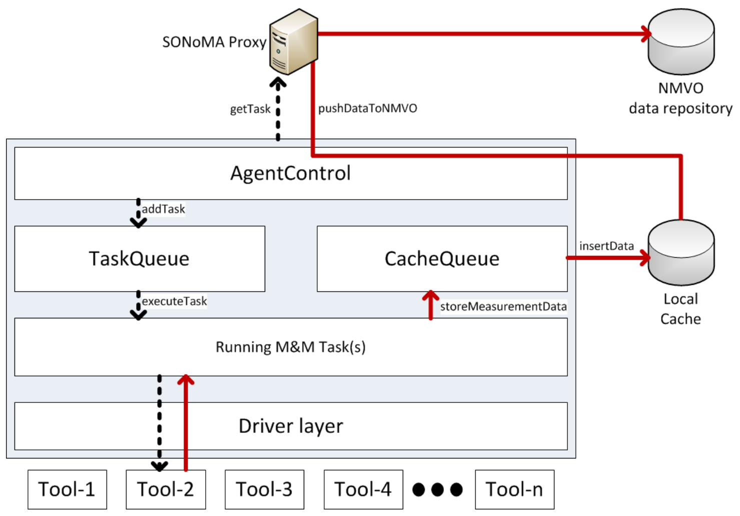
We have to note that though the agent are running in a virtual slice now, the framework could also work in the substrate, in the physical machines as well.

Data access for periodic measurements SONoMA provides a mechanism which automatically stores the measurement results in the nmVO repository permanently. All the data is available for OpenLab testbed users through the web interfaces of nmVO (both nmVO/CasJobs[12] and nmVO/GrayWulf[13]).

Manifold gateway for MySlice integration MySlice is the main platform of OpenLab where users could browse the available resources and can receive a complete view of their status,

[12]http://nm.vo.elte.hu
[13]http://graywulf.vo.elte.hu/ui

Figure 7: Internal structure of a SONoMA agent. Dashed arrows represent the calls between different components, while the red solid lines illustrate the measurement data flow from the tools to the permanent storage.



including monitoring and usage statistics. To make the periodic measurement data collected by SONoMA available through MySlice, a Manifold gateway for nmVO and SONoMA is under implementation in close collaboration with UPMC and ELTE.

2.2 Tools available at the federation level

At federation level, it is crucial to harmonize heterogeneous data collected in different testbeds by various tools. In this section we describe how OpenLab handles this issue by providing users with an uniform access to all the collected data. *(UPMC)*

2.2.1 MANIFOLD / TopHat

MANIFOLD [2] is a data aggregation component designed to interconnect heterogeneous data providers. It is used to build several services on top of it, such as TopHat [7], which provides a measurement and monitoring aggregation service for the federation of testbeds, and MySlice [3], a portal for the federation of testbeds. It is currently used for aggregating information about testbed resources, and measurement and monitoring data, and present it to the user through a user-friendly GUI. It provides intuitive interfaces for navigating and browsing the set of available data, and propose appropriate visualization components. As such, it is also considered for usage monitoring to collect data available in different services or databases, expose it in the context of testbeds, resources, users or experiments.

Data representation MANIFOLD has been designed for interconnecting heterogeneous data providers. It assumes a uniform representation of data in a tabular format similar to what is found in relational databases, and whose flexibility has been much assessed in the literature. This representation is a fundamental aspect of the framework which makes it feasible and scalable. In particular, it assumes that all data can be identified through an appropriate naming (and thus that an underlying semantic has been adopted). The software does not need to know about the naming schemes to transport and process the data; it is thus evolutive and independent to evolutions of the data.

Platform adaptation As existing platforms do not necessarily stick to this representation scheme, it allows for gateways to be developed, playing to role of an adaptation layer abstracting the heterogeneity in technologies (and eventually semantics). The framework natively supports multiple users and is able to provide support for managing users' accounts on the different platforms.

Query language As part of its core library, MANIFOLD proposes a simple query language, largely inspired from SQL, allowing to request information aggregated from several platforms, independently of the format of the data,or the technology behind.

Platforms just need to advertise the data they can provide, and the framework is able to make the necessary glue to provide connectivity to all of them. Just like SQL does not depend on the underlying database schema, our query language makes it possible to address all advertised information with a fixed set of API calls.

Data composition Beyond the gateway, the component does not require any knowledge of this underlying semantic to transport and process the data. We nevertheless encourange platforms to agree and share at least parts of the semantics – through the use of ontologies for instance. This will enable the component to transparently infer the relationships in the data, and to compose the different data sources. This architecture thus makes it possible to enhance the value of existing platforms well beyond the value of each platform considered individually, answering queries overlapping two or more sources of information.

Architecture MANIFOLD provides two interfaces for experimenters: the web-based GUI and the programmatic XML-RPC API. In addition, it provides a library in the form of a Python API for people who wish to build tools on top of MANIFOLD. These three interfaces are supported by three separate components, illustrated on Figure 8: the core functionalities are provided through a Python library, which in turn is exposed through an XML-RPC API for remote use, and by the web GUI.

Building on this framework The component advertises available aggregated data as if it were a platform itself. This allows the web interface to build on top of it to allow users to browse, navigate and make sense of the vast amount of available information. This is performed through a set of visualization plugins or widgets guiding users through some process. Our experience operating PlanetLab Europe shows that such an interface is fundamental for users to exploit the full richness of the tool.

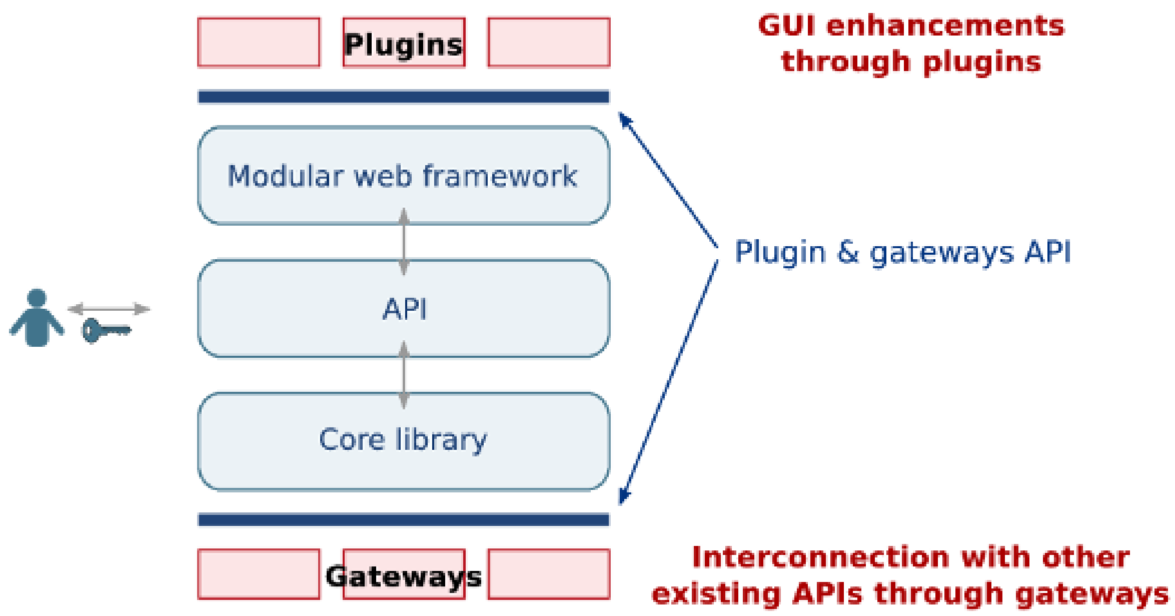

Plugins
GUI enhancements through plugins
Modular web framework
API
Core library
Plugin & gateways API
Gateways
Interconnection with other existing APIs through gateways

Figure 8: MANIFOLD architecture

Currently, gateways allow access to measurement and monitoring datasets, allowing indirectly to track usage of data. We plan to extend available datasets to cover usage and activity monitoring data as well.

The framework is designed to be easily extensible via its plugin mechanism, allowing both new parts of the interface to be added, for navigation or visualization purposes, or even more complete applications or workflows. The portal functionality described later is such an example. Plugins benefit from several modules making development easier such as full support for asynchronous queries. The abstraction framework we have just described allow them to be isolated from the heterogeneity of the different interconnected platforms: they just need to issue queries with the right semantic to be put in relation with the right platforms and receive requested data. As a general principle, the layering of functionalities makes it possible to use them separately.

Proper visualization components will allow to better illustrate and understand user activity. As an example, the visualization component initially developed for PlanetLab Europe, and introduced in Section 2.1.1 is being integrated into Manifold. I will be possible to integrate both usage data and measurements from various sources, to translate them into activity events that can me displayed on a map, conveying visually what is taking place on the testbeds.

Tracking usage of the tool itself As Manifold is being used internally as the central building block MySlice and TopHat, and thus plays a central role in the portal offered to users, it is of utmost importance to provide a consistent tracking of usage of the tool itself. This will allow to better know whether people are using the tool and how, understanding the queries of interest for them, as well as collecting information from a central place that are transversal to different tools and platforms. We expect that it could reveal new patterns into the usage of testbeds and tools.

In addition to tracking the number of users, we remark that the query paradigm that is at the core of the tool is a convenient way to log user activity. Manifold proposes to log both incoming requests submitted to the tool, as well as subqueries issued to the platforms that are involved in answering the request. Of course, such data is also available through the same query mechanism. Let's illustrate the potential use of such information through three use cases:

Basic platform usage statistics: By inspecting the objects requested into the queries, we can get an overview of information of interest by the different platforms. We might for example learn that a lot of users are interested in traceroute or bandwidth measurements in some testbed.

Accountability: Manifold acts as a mediator between the user and the source of data, which itself needs to collect usage information. Because of the potential use of caching, or because the authentication method might not always support proper delegation, we might be unable to inform all the platforms about the use of the data they produced. This is why it is crucial to collect such information, and made it available transparently to the platforms.

Tracking of user behaviour: In our last example, we will dig even further into understanding user behaviour. When a user is browsing a list of resources to make up its slice in MySlice, it will issue a successing of queries, filtering data according to the metric of interest for him (eg. node with low CPU usage), add columns (e.g. city or node architecture), then select a node, and continue this process until it has selected a set of resources for its experiment. Looking at the succession of queries might help us understand what information is of importance for the user, what characteristics of resources he is interested in, and what is his process to choose its resources. Hence, we expect that much more can be harvested from the events we are logging than simply usage statistics.

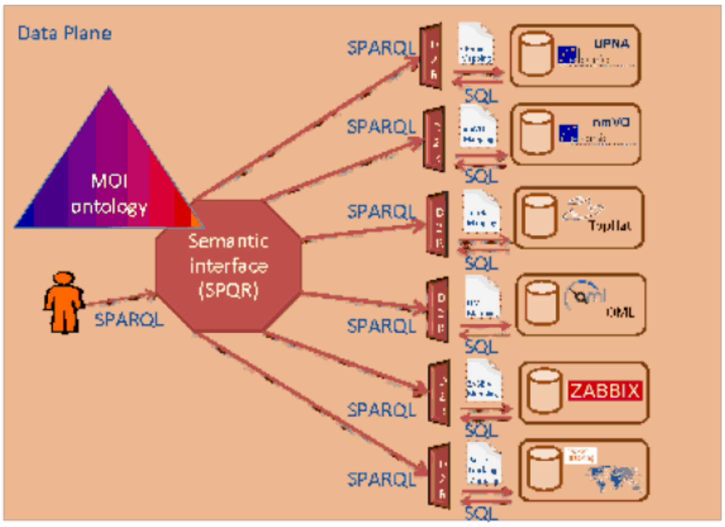
The reader is invited to refer to the project website for further technical insights into the component. MySlice will be launched as a beta version of the OneLab portal in the coming months, so we don't have any usage statistics to report yet, but we expect this mechanism will help us track how the tool is used progressively by the different users.

2.2.2 SPQR: A semantic alternative

Though the semantic processor and query rewriter (SPQR) is not so mature technology as Manifold, it provides an ontology-driven alternative to integrate different data sources storing testbed specific usage and activity reporting data in various format. SPQR is a tool developed at UAM to integrate different network measurement repositories based on a semantic approach. This way, all measurement repositories can be federated. Figure 9 shows a general view on how SPQR can be used to integrate different network measurement tools available at OpenLab.

The idea behind SPQR is to have a common ontology for all network measurements (the MOI ontology developed at ETSI) and several translators from SQL databases to the ontology (D2R), using mappings rules between the ontology and the database schema. Then, when a user queries for network measurements using the ontology, SPQR analyzes the query and send just that part that can be solved by any of the databases. Then, when SPQR has all answers, it merges them and sends a combined response to the user.

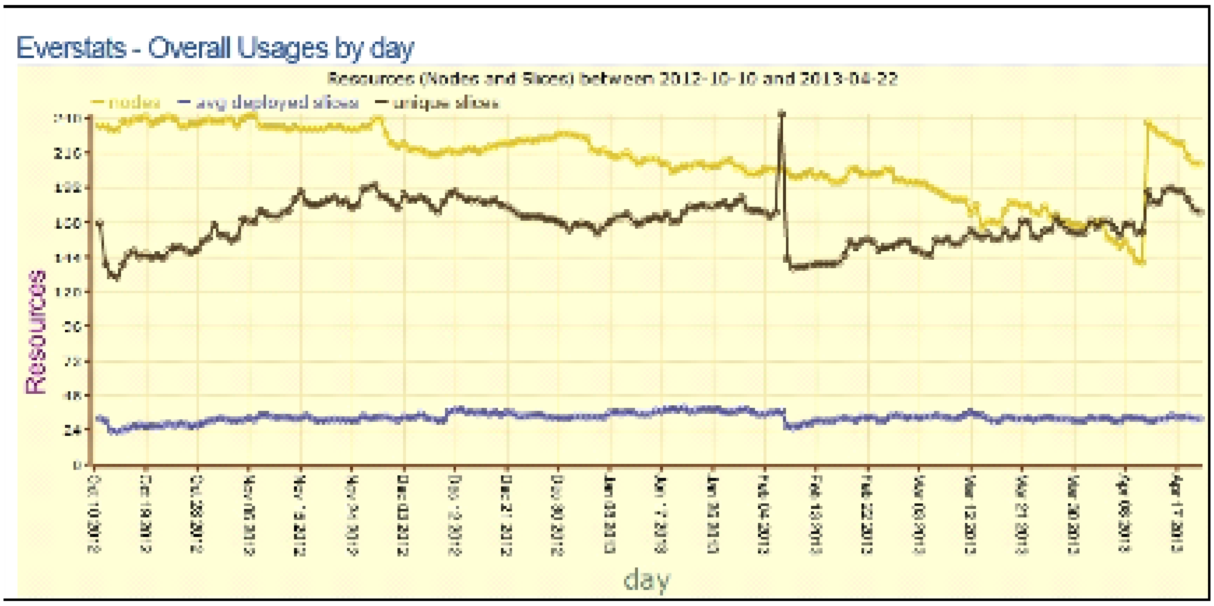
Figure 9: Schematic overview on how SPQR integrates various data sources



During the project, we have defined mapping files between the SQL database schema and the MOI ontology of the following network measurement tools:

- ETOMIC at UPNA
- ETOMIC nmVO at ELTE
- TopHat
- OML tools (OML-Iperf)
- Zabbix
- Packet Tracking

Figure 10: Everstats monitoring Slices activity



3 Collected statistics and analysis

This section is associated to subtask T2.3-1 in the DOW [8].

Responsible: HUJI

3.1 Everstats PlanetLab Europe monitor

The Everstats PlanetLab Europe monitor, running at HUJI provides a long-term monitor of activity in the PlanetLab universe, getting its activity statistics from the CoMon service, and offering plots over selected intervals of the slices and nodes getting the highest utilization. In figure 10 we show the daily levels of nodes active (the upper, yellow line ranging about 200 or so, of slices defined and kept renewed (the middle, brown line which stays above 150 for the period shown, which covers 4Q2012 and the first half of 2013), and slices seen in action during the period (the lowest, dark blue line, ranging from 20-40).

This informaion is publically available, at http://everlab.cs.huji.ac.il:3000/planetstats . and can be interactively configured to cover the desired period. The figures are backed up with tables which can be sorted into ascending or descending order in any chosen data column to highlight cpu consumers, network users, etc.

We see in figure 11 that the total cpu hours has varied about a typical figure of 3000-4000 cpu hours/day, summed over all the PLE nodes during the period shown.

The bandwidth monitored is separated into sending and receiving components. In late 2012, several slices were capturing data far in excess of the data they sent out, leading to unbalanced communications workload. Since early 2013, however, this seems to have come back into balance, although individual slices with very unbalanced communications are still seen (figure 12).

Figure 11: CPU hours utilisation on PLE

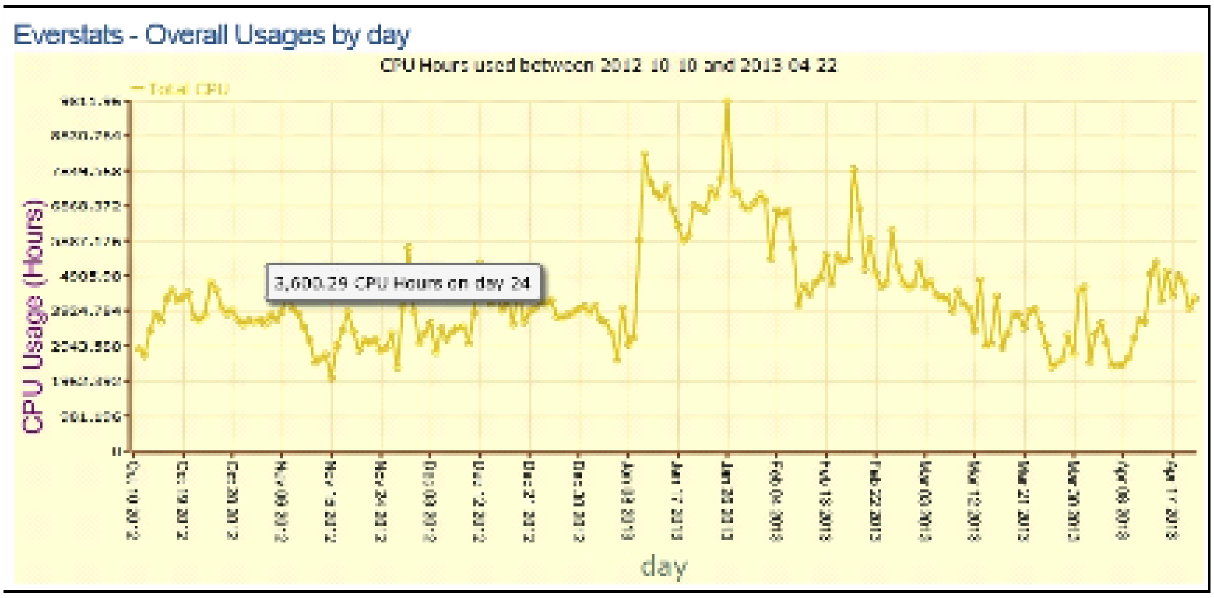


Figure 12: Network bandwidth utilisation

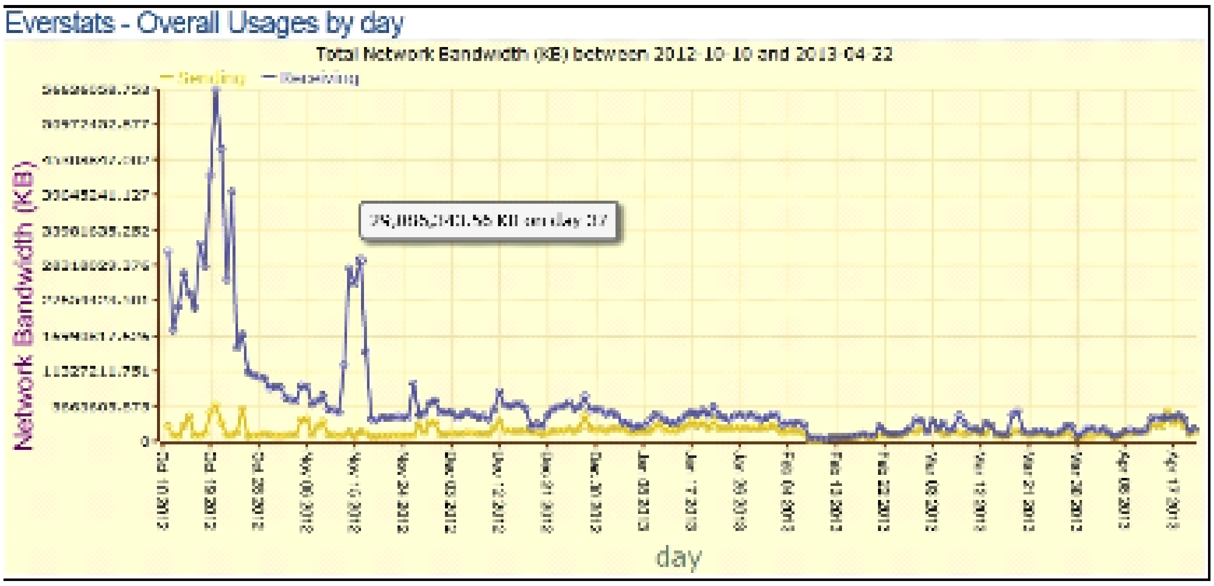
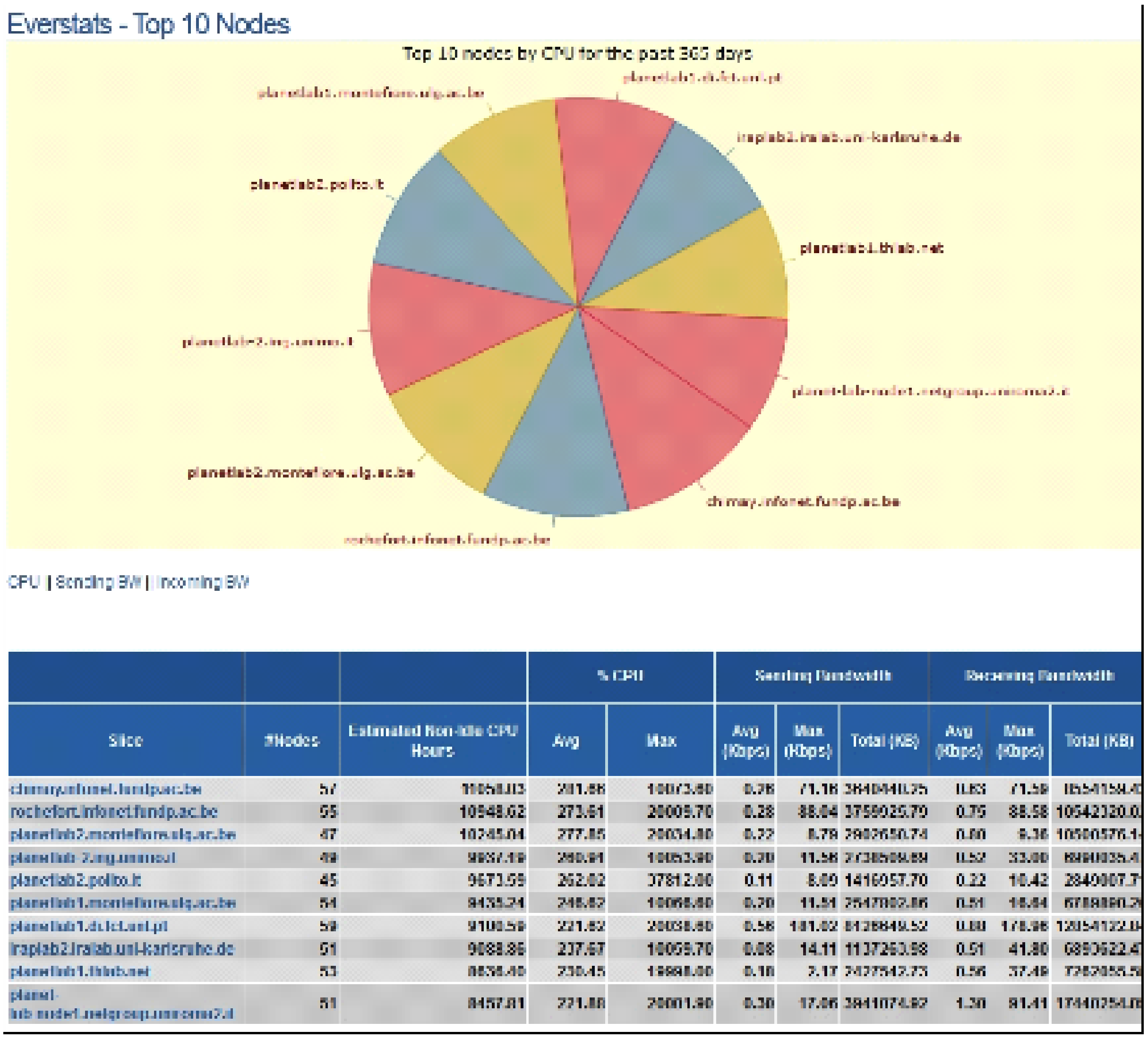


Figure 13: Top 10 nodes sorted by CPU usage


Everstats - Top 10 Nodes
Top 10 nodes by CPU for the past 365 days
planetlab1.montefiore.ulg.ac.be
planetlab1.di.fct.unl.pt
iraplab2.iralab.uni-karlsruhe.de
planetlab2.polito.it
planetlab1.thlab.net
planetlab-2.ing.unimo.it
planet-lab-node1.netgroup.uniroma2.it
planetlab2.montefiore.ulg.ac.be
chimay.infonet.fundp.ac.be
rochefort.infonet.fundp.ac.be
CPU | Sending BW | Incoming BW

			% CPU		Sending Bandwidth			Receiving Bandwidth		
Slice	#Nodes	Estimated Non-idle CPU Hours	Avg	Max	Avg (Kbps)	Max (Kbps)	Total (KB)	Avg (Kbps)	Max (Kbps)	Total (KB)
chimay.infonet.fundp.ac.be	57	11058.03	281.86	10073.80	0.26	71.16	3840448.25	0.63	71.59	10554158.4
rochefort.infonet.fundp.ac.be	55	10948.62	273.61	20009.70	0.28	88.04	3759025.79	0.75	88.58	10542320.0
planetlab2.montefiore.ulg.ac.be	47	10745.04	277.85	20034.80	0.22	8.79	2902650.74	0.60	9.36	10500576.1
planetlab-2.ing.unimo.it	49	9837.19	280.81	10053.90	0.20	11.56	2738509.89	0.52	33.00	6990035.4
planetlab2.polito.it	45	9673.59	262.02	37812.00	0.11	8.09	1416057.70	0.22	10.42	2840007.7
planetlab1.montefiore.ulg.ac.be	54	9435.24	246.62	10066.80	0.20	11.51	2547002.86	0.51	16.64	6788860.2
planetlab1.di.fct.unl.pt	59	9100.59	221.62	20038.80	0.56	181.02	8128849.52	0.88	178.96	12054122.8
iraplab2.iralab.uni-karlsruhe.de	51	9088.86	237.67	10059.70	0.08	14.11	1137263.58	0.51	41.80	6893622.4
planetlab1.thlab.net	53	8836.40	230.45	19998.00	0.16	2.17	2427542.73	0.56	37.49	7262055.5
planet-lab-node1.netgroup.uniroma2.it	51	8457.81	221.88	20001.90	0.30	17.06	3841074.82	1.30	81.41	17440254.8

Everstats also presents data on individual node or slice characteristics by the day throughout the measurement period. In the following graphs we chart a year of activity by the node and by the slice. The natural output format for these is either a pie chart (for the biggest contributor) or a table with interactive sort capability. Thus the year's distribution of total cpu consumption by nodes is shown in figure 13.

with the top 10 nodes (by cpu usage) usage data displayed in a table below its pie chart. If we select the top 10 nodes by bandwidth outgoing, the distribution (at the top) is less uniform (figure 14).

Figure 14: Top 10 nodes sorted by outbound network traffic utilisation

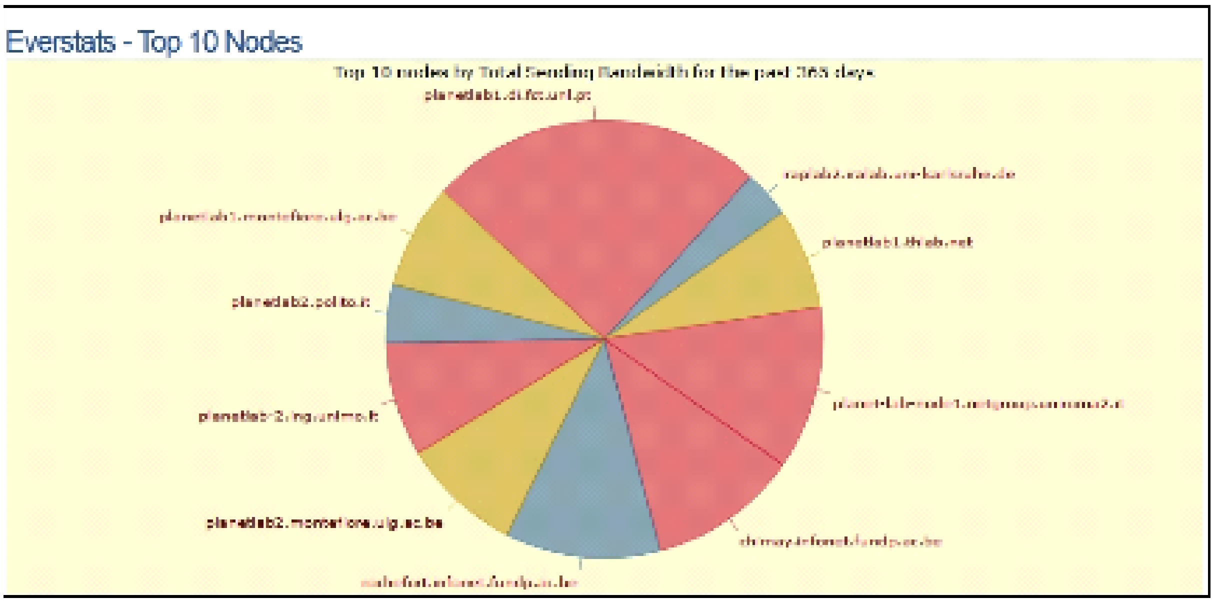
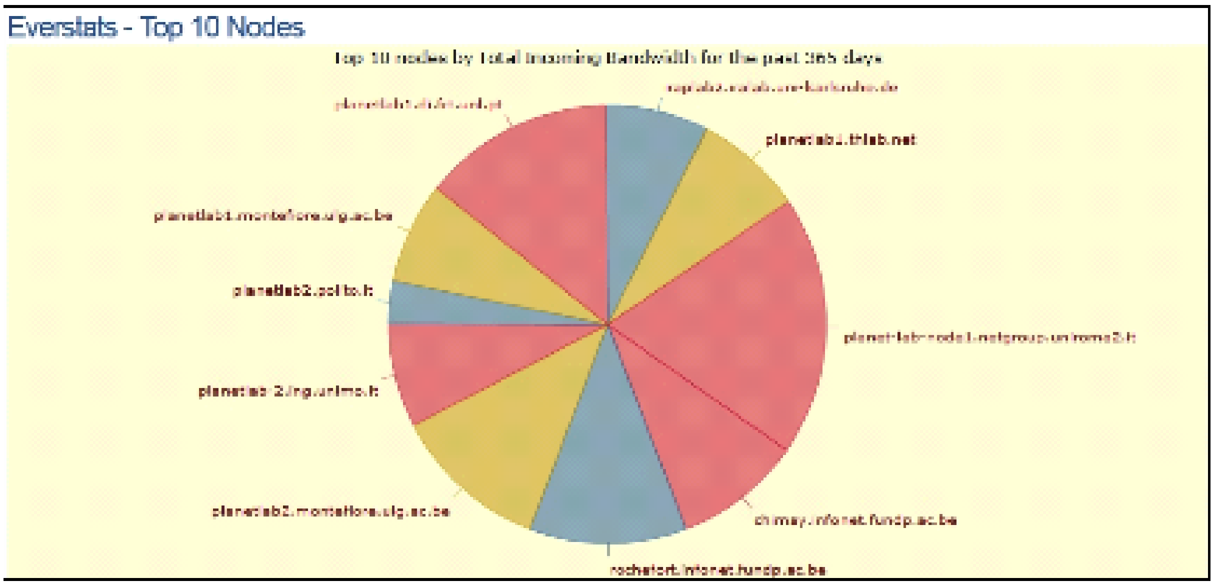
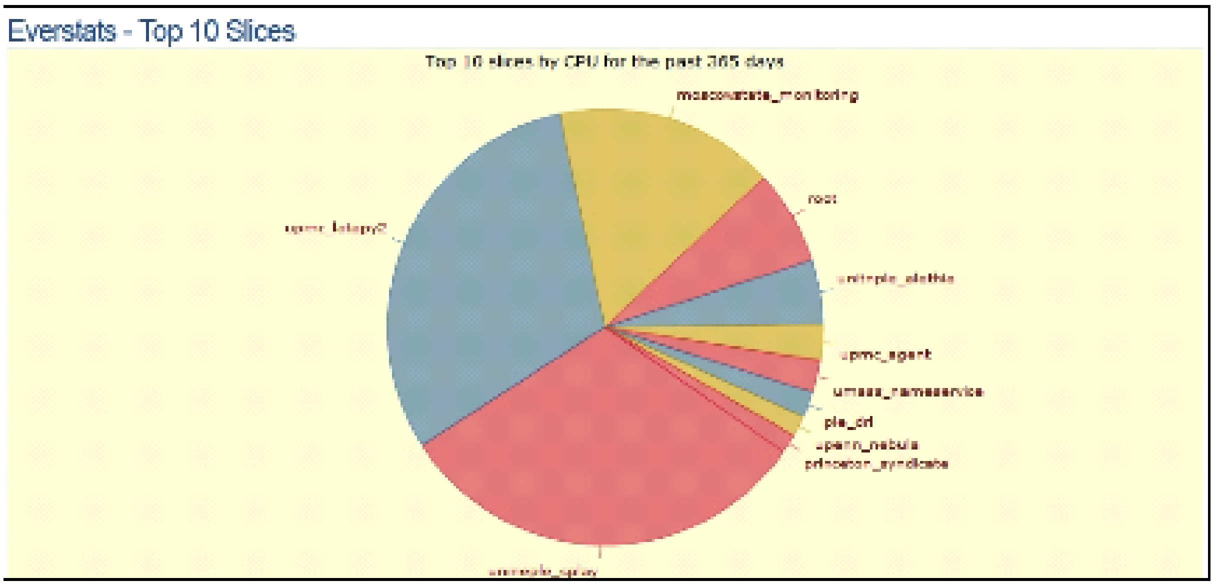


Figure 15: Top 10 nodes sorted by inbound network traffic utilisation



Note these are not the same nodes as the leading cpu consumers. In figure 15 we show the distribution of bandwidth-incoming nodes for the past year.

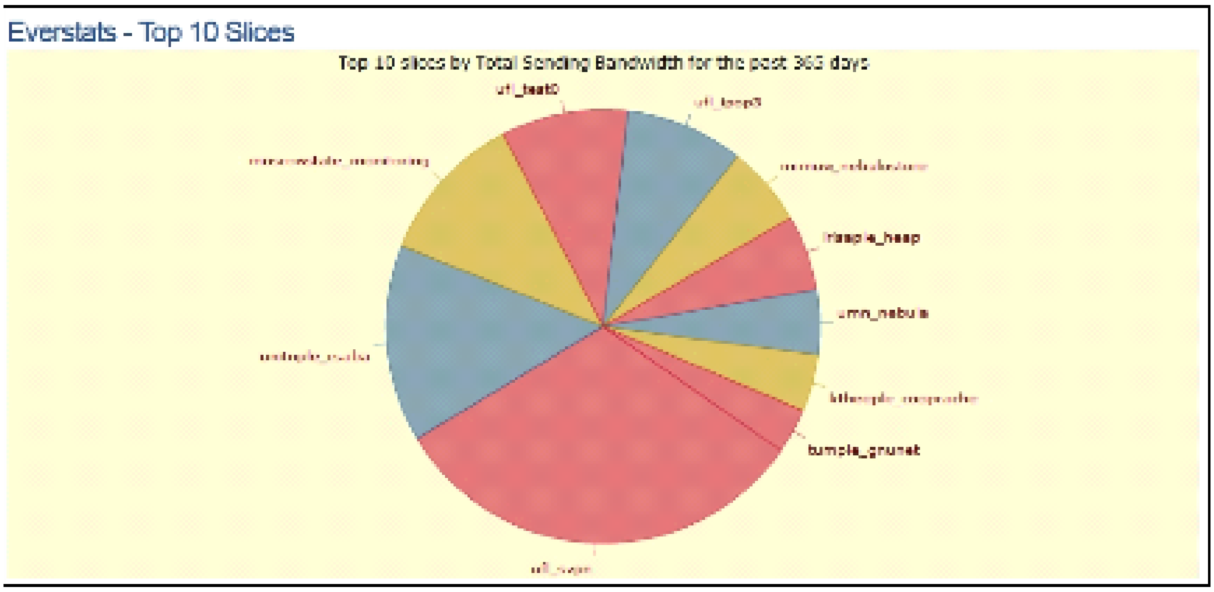
Figure 16: Top 10 slices sorted by CPU utilisation



We can also extract relative slice activity statistics for the past year. First by cpu consumption as shown in figure (16), then ordered by outgoing bandwidth consumption (figure 17), and finally the top 10 slices, sorted by receiving bandwidth over the past year (figure 18).

We notice that both slices and nodes that appear to make leading use of PlanetLab Europe are well distriuted across the European countries and that a wide variation of distinct types of usage is evidenced. The database contains more than 1000 defined slices (although it seems that not more than 500 of them were ever simultaneously in effect). 100 of the slices seen at some point in the past year used more than 100 nodes in their definition. Another 200 slices exploited 10 or more nodes for their processes. But a total of 900 slices with 1 or more nodes defines showed some cpu and communications activity in the past year. We don't think this background of lighly used slices is a problem or sign of users failing to succeed with their PlanetLab applications. More likely this is the educational use, in courseware, of PlanetLab for hands-on experience in distributed computing – an application which we hope to increase in the future.

Figure 17: Top 10 slices sorted by outbound network traffic utilisation



4 OpenLab Usage reporting architecture

This section is associated to subtasks T2.3-2 and T2.3-3 in the DOW [8].

Responsible: UPM-C/HUJI + contributions from each testbeds + TUB for freeTRex

This section will describe how the architecture and best practices that have been put in place in OpenLab, from the testbed to the federation level: how the tools have been deployed, operated and integrated. What manual operations are being done, how is data collected, stored, displayed, etc. It will be important to dedicate a subsection to the FreeTRex portal.

4.1 The Testbeds in OpenLab

A number of different and diverse testbeds are available in the OpenLab project. Each of them has been build with a certain type of applications in mind and usually uses its own contol framework. It is an objective of OpenLab to enable transparent access to combinations of resources from different testbeds for advance and large scale future Internet experiments

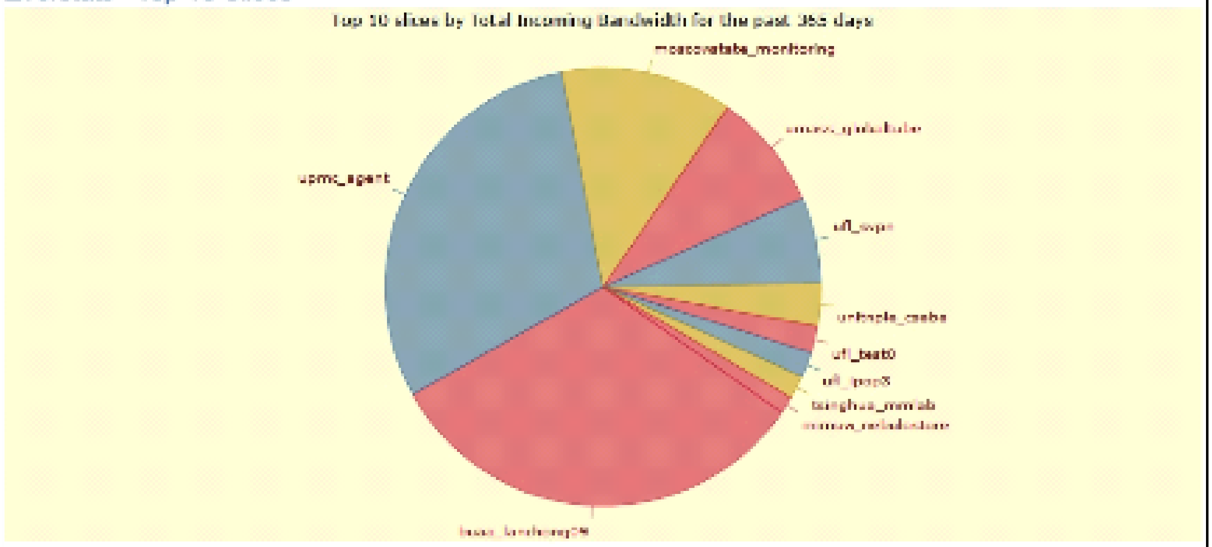
4.1.1 NITOS (Network Implementation Testbed using Open Source code)

NITOS [14] is an OMF-based wireless testbed located in a campus building at UTH in Volos, Greece. It consists of 45 nodes equipped with a mixture of Wi-Fi and GNU-radios, as well as cameras and temperature and humidity sensors. Two programmable robots provide mobility. This publicly available testbed supports experiments across all networking layers. In addition to OMF, the testbed employs locally developed tools: the NITOS scheduler, a resource reservation

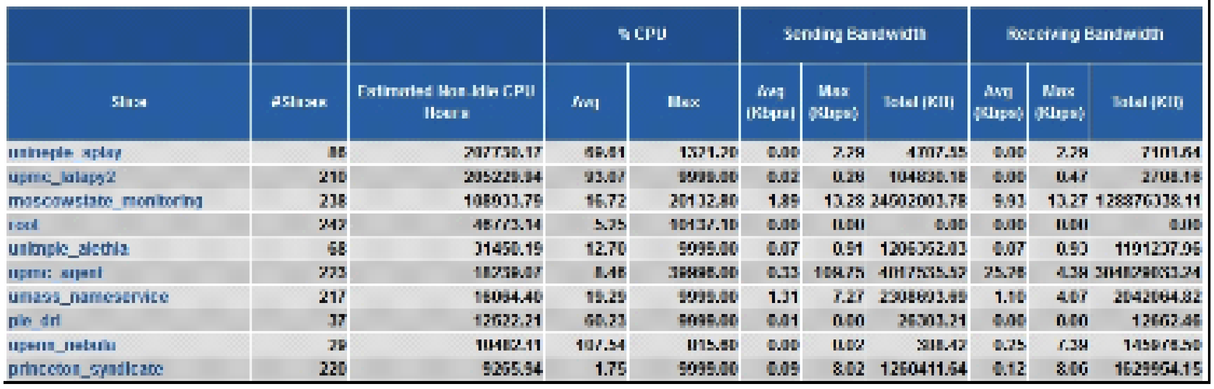
[14] nitlab.inf.uth.gr/NITlab/index.php/testbed

Figure 18: Top 10 slices sorted by inbound network traffic utilisation

Everstats - Top 10 Slices



CPU | Sending BW | Incoming BW



			% CPU		Sending Bandwidth			Receiving Bandwidth		
Slice	#Slices	Estimated Non-Idle CPU Hours	Avg	Max	Avg (Kbps)	Max (Kbps)	Total (KB)	Avg (Kbps)	Max (Kbps)	Total (KB)
unineple_aplay	86	207730.17	69.61	1321.20	0.00	2.28	4707.35	0.00	2.28	7101.64
upmc_lalapy2	210	205220.84	93.07	9999.00	0.02	0.26	104830.18	0.00	0.47	2708.16
moscowstate_monitoring	238	108033.79	16.72	20132.80	1.89	13.28	24502003.78	9.93	13.27	128876338.11
root	242	48773.14	5.75	10137.10	0.00	0.00	0.00	0.00	0.00	0.00
unitmple_alethia	68	31450.19	12.70	9999.00	0.07	0.91	1206352.03	0.07	0.93	1191237.06
upmc_agent	273	18259.07	8.48	39998.00	0.33	109.75	4017535.57	25.78	4.38	304829033.24
umass_nameservice	217	16064.40	19.29	9999.00	1.31	7.27	2308693.69	1.10	4.07	2042064.82
ple_drl	37	12622.21	60.23	9999.00	0.01	0.00	26303.21	0.00	0.00	12662.46
upenn_nebula	29	10482.11	107.54	815.80	0.00	0.02	308.42	0.25	7.38	145978.50
princeton_syndicate	220	9265.94	1.75	9999.00	0.09	8.02	1260411.64	0.12	8.06	1629954.15

application, and TLQAP, a topology and connectivity monitoring tool. In OpenLab, under the guidance of COSMOTE, a major ISP provider in Greece, and with the help of Alcatel-GR, NITOS will be extended to meso-scale (WiMAX/3G/LTE), with a base-station and mobile end-user handsets.

4.1.2 w-iLab.t

The w-iLab.t testbed[15] is a wireless mesh and sensor network infrastructure deployed across three floors of the IBBT office building in Ghent, Belgium. It contains 200 locations, each equipped to receive multiple wireless sensor nodes and two IEEE 802.11a/b/g WLAN interfaces. Wi-Fi and sensor networks operate simultaneously, allowing complex and realistic experiments with heterogeneous nodes and multiple wireless technologies. In addition, shielded boxes accommodate nodes that can be connected over coax cables to RF splitters, RF combiners and computer controlled variable attenuators, thus allowing fully reproducible wireless experiments with emulated dynamically changing propagation scenarios. With an in-house designed hardware control device, unique features of the testbed include the triggering of repeatable digital or analogue I/O events at the sensor nodes, real-time monitoring of the power consumption, and battery capacity emulation.

4.1.3 DOSTEL

The DOTSEL testbed at ETH Zrich is focused on delay-tolerant opportunistic protocols and applications. It is composed of 15 Wi-Fi equipped Android Nexus One devices that are carried by staff members, and five Wi-Fi a/b/g ad-hoc gateways. With OpenLab, DOTSEL will be enlarged to 25 nodes and add 3G capability.

4.1.4 PLE (PlanetLab Europe)

PLE[16] is the European arm of the global PlanetLab system, the worlds largest research networking testbed, which gives users access to Internet-connected Linux virtual machines on over 1000 networked servers located in the United States, Europe, Asia, and elsewhere. Nearly 1000 scientific articles mention the PlanetLab system each year(1), including papers in such prestigious networking and distributed systems conferences as ACM SIGCOMM, ACM CoNEXT, IEEE INFOCOM, ACM HotNets, USENIX/ACM NSDI, ACM SIGMETRICS, and ACM SIGCOMM IMC. Researchers use PLE for experiments on overlays, distributed systems, peer-to-peer systems, content distribution networks, network security, and network measurements, among many other topics. (2) Established in 2006 and developed by the OneLab initiative, PLE is today overseen by four OpenLab partners: UPMC, INRIA, HUJI, and UNIPI. UPMC handles testbed operations and INRIA co-leads, along with Princeton University, development of MyPLC, the free, open-source software that powers PlanetLab. The PlanetLab Europe Consortium has 150 signed member institutions: mostly universities and industrial research laboratories, each of which hosts two servers that it makes available to the global system. These institutions are

[15]http://www.ibbt.be/en/ilabs/ilab-t/wireless-lab
[16]http://www.planet-lab.eu

home to 937 users. On a typical recent day, 244 were connected to on-going experiments. OpenLab extends both the PlanetLab software and the PlanetLab Europe Consortium.

4.1.5 HEN (Heterogeneous Experimental Network)

HEN[17], built between 2005 and 2010 by UCL, provides 100 server-class machines with between 6 and 14 NICs each, interconnected by a Force10 E1200 switch with 550 Gigabit ports and 24 10-Gigabit ports. This infrastructure allows emulation of rich topologies in a controlled fashion over switched VLANs that connect multiple virtual machines running on each host. The precise control of topology and choice of end-host operating system possible on HEN are particularly valuable facilities to networking and distributed systems researchers. Many dozens of researchers actively use HEN: at Stanford University, the University of Lancaster, NYU, the Nokia Research Centre, and NEC Labs Europe, to name a few. UK- and EU-funded projects, including the EPSRC-funded Virtual Routers project, EPSRC-funded ESLEA project, EU FP7-funded Trilogy project, and EU FP7-funded CHANGE project, have all generated the bulk of their experimental results on HEN. Results have been published in prestigious networking and distributed system venues including ACM SIGCOMM, ACM HotNets, USENIX/ACM NSDI, USENIX Security, ACM CCR, ACM CoNEXT, Presto, FDNA, PMECT, ICDCSW, and LSAD. OpenLab extends HEN to support multi-homed operation and interconnection with other FIRE-supported testbeds, yielding a powerful platform for experimental research on multi-path transport and application protocolsa hot area of interest to todays networking community.

4.1.6 The WIT IMS testbed

The TSSG/WIT NGN IMS testbed[18] is an Irish nationally-funded initiative serving telecom firms seeking to develop or test NGN services. It provides them with advanced multimedia services, such as conference calling and handling of presence information. The testbed is a carrier grade NGN platform based on the Ericsson IMS Communications System (ICS). The SIP based horizontal network architecture includes an Ericsson IMS core and the components for managing sessions, addressing, subscriptions and IMS inter-working components with the relevant gateways for connectivity to other networks. The testbed has recently been upgraded with pico/femto cells to allow secure remote access to the test facility. The network also includes support systems for handling provisioning, charging, device configuration and operation and maintenance. Clients include IP centrex companies, a location based service provider, and developers of pico/femto cell technology. International customers have conducted testing in the area of IMS security and testbed interconnection using the GSMA Pathfinder service operated by Neustar. In OpenLab, the WIT testbed will be enhanced by the development and integration of a P2P/NGN QoS reservation mechanism that will allow OpenLab experimenters to test application level P2P traffic routing algorithms.

[17]mediatools.cs.ucl.ac.uk/nets/hen
[18]http://ngntestcentre.com

4.1.7 OSIMS - An Open Source IMS experimentation platform

The University of Patras IMS testbed supports PSTN testing scenarios: calls between a PSTN network and any PSTN number (including international and mobile numbers); and calls between IP phones (either soft phone or hard phone) and any PSTN number (including international and mobile numbers). The testbed has been used in numerous interoperability experiments with the carrier grade network of Telecom Austria, and the NGN testbeds of Siemens AG in Munich and Telefnica TID in Madrid. It is currently hosts experiments from the FP7 VITAL++ project. Integration of the testbed into the Teagle framework was carried out under the PII project. In OpenLab, the testbed will be enhanced to incubate P2P/NGN QoS reservation algorithms and establish experimentation paths taking advantage of the OpenFlow protocol.

4.1.8 ETOMIC (European Traffic Observatory Measurement InfrastruCture)

ETOMIC[19] is a high precision (10s of nanoseconds) network measurement testbed featuring dozens of Internet-connected nodes globally synchronized with GPS clocks. More than 100 users from 34 institutions run over 100 experiments per month. Created in 2004-05 within the FP6 EVERGROW Integrated Project, it was awarded the Best Testbed Award at TridentCom 2005. In OneLab2, ETOMIC opened interfaces to MySlice (described above) to transparently provide measurements to PlanetLab users. OpenLab will continue to promote ETOMICs experimental plane interoperability.

4.2 Statistic collection in PlanetLab Europe

4.2.1 Slicestat deployment on the PLE nodes

Slicestat is a small daemon running in a slice environment that provides information on resource usage of the slices currently running on the same node. A dedicated slice has been deployed in PLE, slicestat is installed through an init script that is executed once when the slice is created on the node. The init script executes some basic operations and downloads and installs the rpm package and relative dependencies. The installation script included in the package automatically set up and starts the service. Once finished, accessing the node on port 3100 will return the current resource usage in the form of a tab delimited text file. Slicestat uses a feature of PlanetLab, enabled through the use of tags[20], that permits the service to directly access information on the node on the local running virtual machines. This type of access is usually restricted to the administrator user and also not accessible from a virtual environment. Slicestat is then able to retrieve such information in a controlled way to avoid the machine to be compromised.

4.2.2 Collecting and storing monitoring data

Monitored data is collected and stored on a central server. The software running on the stats server is responsible for collecting the data by accessing the slicestat server via the http protocol

[19]etomic.org
[20]https://www.planet-lab.eu/planetlab/doc/PLCAPI.html

and then store the result on the local database. Access to the slicestat sevice on each node of the PLE testbed is done concurrently to minimise time delays and avoid bottle necks. The node hostname, slice name and timestamp are added to the collected data for each reading and stored on database. We used PostgreSQL[21] and set up a partitioned database to optimise the storage facility and render it more scalable. Moving avarages based on time slots are built with the use of stored procedures. We calculate avarages for each 15 minutes, 1 hour and 1 day, while we generate avarages for longer period of time when needed when analyising or reading the data.

4.2.3 Data analysis and result display

Statistics for each slice, avarages per node or site are displayed using the Google Chart API[22]. To build the graphs using Google Chart data must be in the JSON[23] format. The stats server provides also a local RESTful service that can be used to query directly the stored data and provide results in this format. This service has been deployed using Python and the Flask[24] framework.

4.3 Free-T-Rex platform

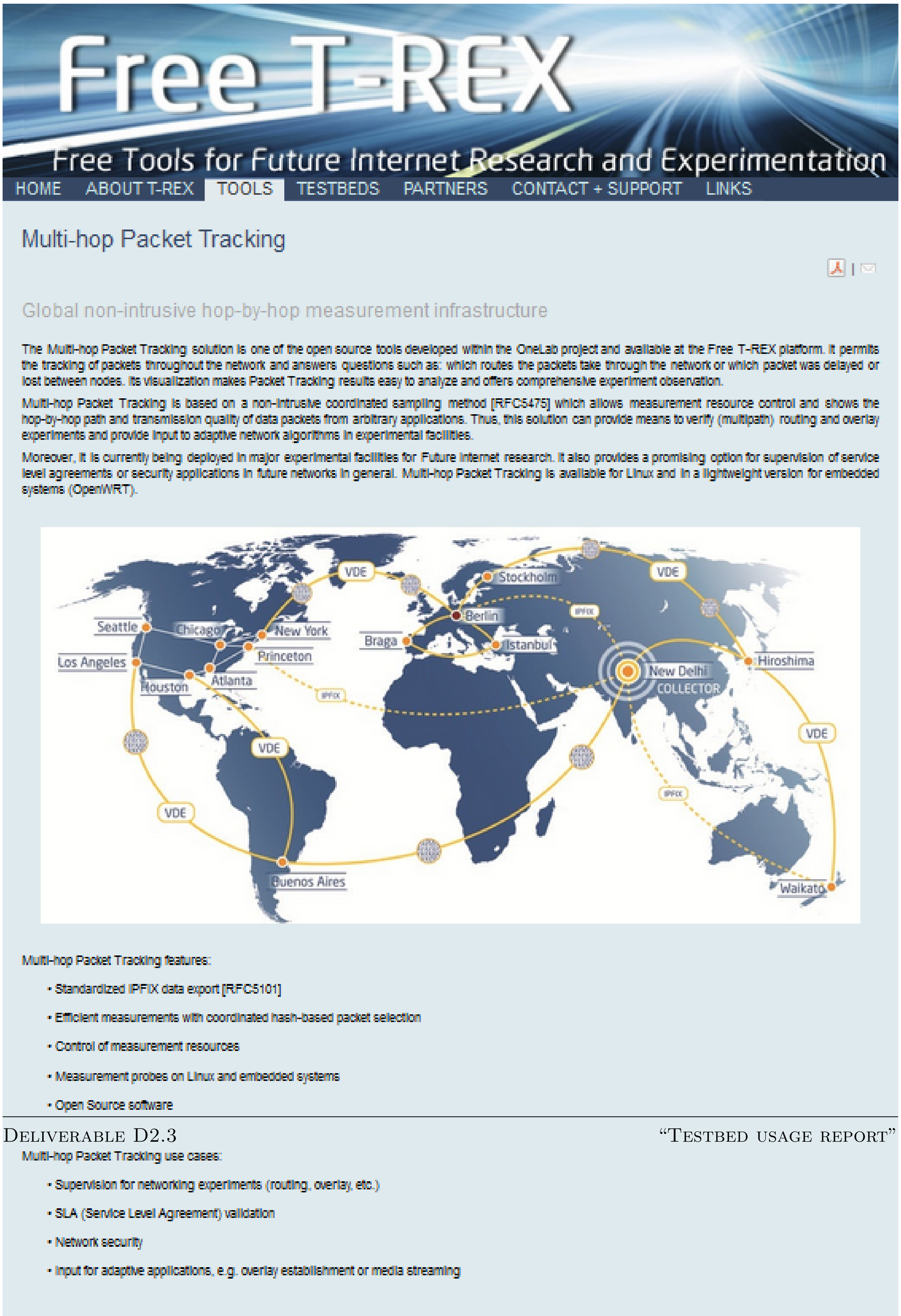
Free T-REX [1] is a platform for testbed users (researchers), testbed providers and tool developers in the area of Future Internet research and experimentation. It provides information about several useful tools and as well access to tools that are deployed in global Future Internet experimental facilities and can be used by researchers in their experiments. It also supports testbed providers in selecting suitable tools for instrumenting their infrastructures.

In addition to this, Free-T-Rex is used by OpenLab in order for providing reports about testbed usage and testbed statistics, and as well for sharing knowledge gained through users experience exchanges.

Free-T-Rex includes a list of tools that are used by both testbed providers and users. Figure 1 shows a screenshot of Free-T-Rex Platform that is opened to one of the experimentation tools it supports, namely Multi-hop Packet Tracking. You will see detailed information about Packet Tracking: general tool description, its features, its applicable use cases, and access to the software and documentation. Furthermore, if you click on the TESTBED tab, you will see a list of testbeds with high level information about them and reports, or links to reports about testbed usage and testbed statistics.

[21]http://www.postgresql.org
[22]https://developers.google.com/chart/
[23]http://www.json.org/
[24]http://flask.pocoo.org/

Figure 19: Screenshot of Free-T-Rex Platform showing detailed information about one of the experimentation tools "Packet Tracking"



Multi-hop Packet Tracking

Global non-intrusive hop-by-hop measurement infrastructure

The Multi-hop Packet Tracking solution is one of the open source tools developed within the OneLab project and available at the Free T-REX platform. It permits the tracking of packets throughout the network and answers questions such as: which routes the packets take through the network or which packet was delayed or lost between nodes. Its visualization makes Packet Tracking results easy to analyze and offers comprehensive experiment observation.

Multi-hop Packet Tracking is based on a non-intrusive coordinated sampling method [RFC5475] which allows measurement resource control and shows the hop-by-hop path and transmission quality of data packets from arbitrary applications. Thus, this solution can provide means to verify (multipath) routing and overlay experiments and provide input to adaptive network algorithms in experimental facilities.

Moreover, it is currently being deployed in major experimental facilities for Future Internet research. It also provides a promising option for supervision of service level agreements or security applications in future networks in general. Multi-hop Packet Tracking is available for Linux and in a lightweight version for embedded systems (OpenWRT).

Multi-hop Packet Tracking features:

- Standardized IPFIX data export [RFC5101]
- Efficient measurements with coordinated hash-based packet selection
- Control of measurement resources
- Measurement probes on Linux and embedded systems
- Open Source software

Multi-hop Packet Tracking use cases:

- Supervision for networking experiments (routing, overlay, etc.)
- SLA (Service Level Agreement) validation
- Network security
- Input for adaptive applications, e.g. overlay establishment or media streaming

5 Outcome of the first interoperability trials

5.1 Introduction

This document summarizes the outcomes of the first OpenLab plugfest [4], organized in the UPMC premises in Paris, between the 23rd and 25th of January 2013. It builds on [12], and was open to participants from across the FIRE community, as well the GENI community and others.

The interoperability trials (also known as bake-offs or plugfests), organized at Months 18 and 27 of the OpenLab project are the occasions at which progress is assessed. They serve a dual objective. Within the project, they are the opportunity to obtain a status of the advancement of the integration and inform the various scientific workpackages. Beyond this, they represent an opportunity to disseminate achievements and work in progress in the different related projects, as well as foster collaboration, and adoption and reuse of the solutions developed and proposed in each of them.

A major objective of such events is to enable a set of experiment control tools that are currently used in just one or a limited set of testbed environments to operate across multiple testbeds. This will make it possible to define an experiment once and rerun it in different environments: either multiple real-world environments, such as multiple wireless testbeds, each with their own particularities; or a range of real-world, emulated, and simulated environments, allowing different mixes of uncontrolled and controlled environmental parameters. These enhancements will allow for greater repeatability and comparability of experiments.

The degree of interoperability of experiment controllers – their ability to function across multiple testbeds – has been testbed and the outcome is described in this report. Although experiment controllers are the main focus, some aspects of the control plane as well as testbed-specific technologies are also covered, since it is equally important for further interoperability and repeatability.

The event was organized during 2 and a half days, and welcomed 38 participants from 13 institutions and 8 countries, spanning the following ongoing projects: OpenLab, FED4FIRE, FIT Equipex, F-Lab and NOVI (see Section A.1). It was divided in two parts. Since it was the first event of the kind, the first day was dedicated to acquiring some knowledge about the work realized by the different partners that were invited to give tutorials (for components of interest for the federation), or shorter presentation or demos about their tools, progress or integration results (see Section A.1). It was then decided to organize the remaining two days around a set discussion sessions (morning of day 2), and more or less formal working groups (four thematic sessions were organized about those four topics that gathered a critical mass of people – SFA, OMF/OMF, MySlice, and NEPI –, to benefit from the presence of their main contributors.

The minutes of the discussions sessions – *Interoperability of experiment controllers* and *Integration of measurement systems* – are presented respectively in Sections 5.2 and 5.3. They are important in that they structure and give directions for the different working groups.

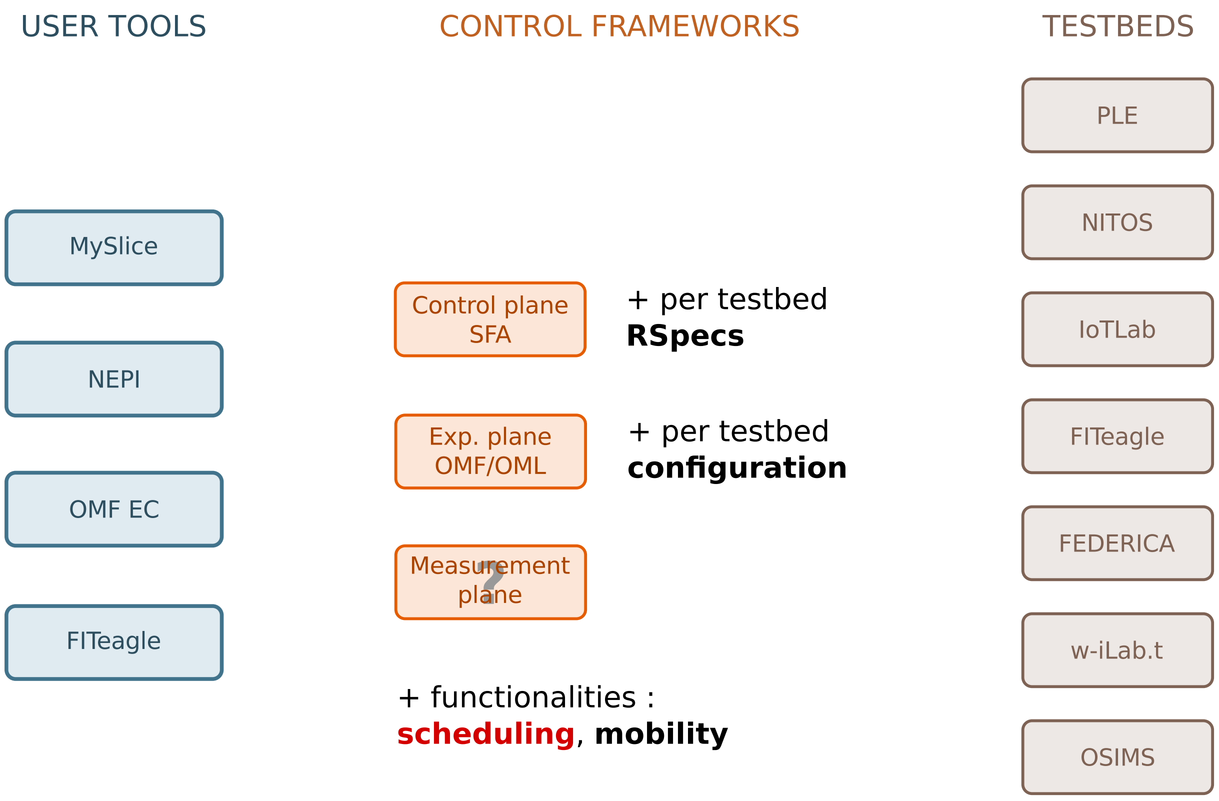


Figure 20: Interoperability of experiment controllers: an overview

5.2 Interoperability of experiment controllers

5.2.1 Overview

Figure 20 gives an overview of the different entities that we are dealing with in this subsection. On the right side we have the different testbeds providing resources to the federation. On the left side, we find the various user-facing tools that provide control over the federation. The vision here is the one of interoperabilityi, with multiple (though limited) entry points to the federation. The idea is to support a small set of tools, covering the needs and required functionalities of the different testbed user communities.

In order to allow an interoperable communication between the tools and the testbeds, avoiding the N by N scalability issue, federated control framework have been proposed for both the control plane – SFA – and for the experimental plane – OMF[25]. They are represented in the middle of the picture. The third brick has to deal with measurements, and it is currently not specified. This aspect will be discussed in Section 5.3.

[25]Currently, there is an ongoing effort to standardize the protocols behind the OMF component, under the nme FRCP, *Federated Resources Control Protocol*

Testbed	Status	Software	Notes
PLE	✓	SFAWrap	scheduling OK
NITOS	✓	SFAWrap	scheduling OK
w-iLab.t	not yet	ProtoGENI	
FITeagle	✓	SFAWrap	
OSIMS	✓	SFAWrap	
FIT/IoTLAB	✓	SFAWrap	scheduling OK
FEDERICA	✓	SFAWrap	
VirtualWall	✓	ProtoGENI	

Table 1: Implementation status of SFA on the different testbeds

5.2.2 Control plane management through SFA

SFA adoption status

It has been decided in the project to adopt the SFA architecture to provide the federation with a means of authenticating and authorizing users, and allowing them to browse and reserve resources to form a slice, or an experiment. We report in Table 1 the status of this effort for different testbeds, both within (upper part) and outside the OpenLab project. The software used for bringing this functionality to the platform is also indicated.

SFA proposes a distributed and secure thin waist for allowing a global federation of testbeds. As such, it does not take into account testbed specifics, which are dealt with in an upper layer, in the form of resource specifications (RSpecs), which currently take the form of XML documents with a given structure. There are as many instances of such document formats as there are different testbeds, in order to account for the differences in properties.

This makes it necessary for the different user tools to deal with each one version in order to properly interpret and present the information to the user[26].

Still, for the top level of the document, or for common functionalities such as scheduling or mobility, it is possible to converge towards a common representation. This is what has been done so far by three testbeds which all have reservation functionalities: NITOS, IoTLAB – formerly SensLAB – and PLE. All three testbeds propose a common representation format in the RSpecs, despite the heterogeneity of their underlying schedulers (homemade for PLE and NITOS, a wrapper for OAR in the IoTLAB case). This simplifies the support of scheduling in user tools.

Testbed support status in MySlice (SFA)

The status of various testbed support in MySlice is presented in Table 2.

Testbed support status in NEPI (SFA)

While NEPI already provide a set of testbed adapters, the support of new testbeds through SFA is being done through MySlice (either through the integration of the core library and relevant gateways, or by calls to the API).

[26]In the Fed4FIRE project, a new proposal for RSpecs, based on a standard RDF container, and well-defined semantics will overcome such issues, which will only put as a requirement the knowledge of the semantic, and will otherwise allow for a default best-effort treatment.

Testbed	Status	Notes
PLE	✓	scheduling OK, demo
NITOS	✓	scheduling OK, demo
w-iLab.t	n/a	n/a
FITeagle	!	in progress, to test
OSIMS	!	tested soon
FIT/IoTLAB	✓	scheduling OK, demo
FEDERICA	!	to be tested
VirtualWall	!	to be testbed

Table 2: Testbed support status in MySlice for SFA

Testbed	Status	Notes
PLE	✓	can be make more stable
NITOS	✓	native
w-iLab.t	✓	native
FITeagle	planned	
OSIMS	no	
FIT/IoTLAB	✓	scheduling OK
FEDERICA	no	
VirtualWall	✓	native

Table 3: Implementation status of OMF on the different testbeds

Testbed support status in OMF EC (SFA)
To the best of our knowledge, OMF Experiment Controller does not support or plan to support SFA interfaces.

5.2.3 Experimental plane management through OMF

At the different of SFA, we are in a situation where the solution has to date not been agreed by all testbeds, which some other ones are native OMF. Current status is reported in Table 3[27].
OMF adoption status
Like in the previous case with SFA, OMF presents a standard interface, and testbed specificities represented with various configuration options. A smooth support by user tools (especially in GUIs) is then again dependent on the support of these options.
Testbed support status in MySlice (OMF)
MySlice currently has no support for experiment control, and instead relies on third party tools, such as NEPI, for experiment control. The handover between MySlice and NEPI has been

[27]At the time of writing this report, a new version – OMFv6 – is being released which might have some impact or influence on its adoption. Also, the underlying protocol is being proposed as a standard, under the name FRCP (Federated Resources Control Protocol).

the object of one working group since it involves agreeing on how to pass the data between tools, as well as avoiding multiple authentications from the user.

Testbed support status in NEPI (OMF)

OMF support in NEPI is in development and has been the object of a working group.

Testbed support status in OMF EC (OMF)

OMF is the native control framework for the OMF Experiment Controller. The interface with the different testbeds is realized through OEDL (ruby) scripts.

5.3 Integration of measurement systems

5.3.1 Overview

We are now focusing our attention on the *measurement plane* brick that was not considered previously in Figure 20. There is currently no federated control framework specialized for measurements, such as SFA and OMF for respectively the control and experimental planes. It is the purpose of this subsection to discuss about possible realization of such a component, that would accommodate measurement and monitoring needs of the different testbeds and user communities, as well as possibly integrate external types of measurements platforms.

5.3.2 Towards a measurement plane interface ?

Three types of measurements sources can be identified:

- **testbed measurements**: these are measurements and monitoring information produced by the testbed. They generally provide some data about the testbed resources or its substrate (eg. wireless signals from the NITOS testbed);
- **third-party platforms**: third party platform can give various types of measurements, which can also belong the testbed substrate. This is for example the case of the public internet on which PlanetLab is deployed (any source of Internet measurements), or simply because those platforms are running on top of a testbed (eg. CoMon for monitoring system information about PlanetLab);
- **user-defined measurements**: while not being made public, those are the measurements collected by the user for its own experiment purpose (eg. through OML, or traces in NEPI).

The challenge here comes from the diversity of sources and measurements to be accommodated, since the measurement integration layer has to reach a balance between uniformity (for tractability purposes) and heterogeneity (the value of measurements): multiple API and data formats; archived vs on-demand measurements; etc. The adopted solution has to propose a convenient query and transport protocol, and integrate well in user tools to support the various stages of an experiment. Finally, it might require the support of authentication and authorization solutions (as it is being done in the context of testbed federation).

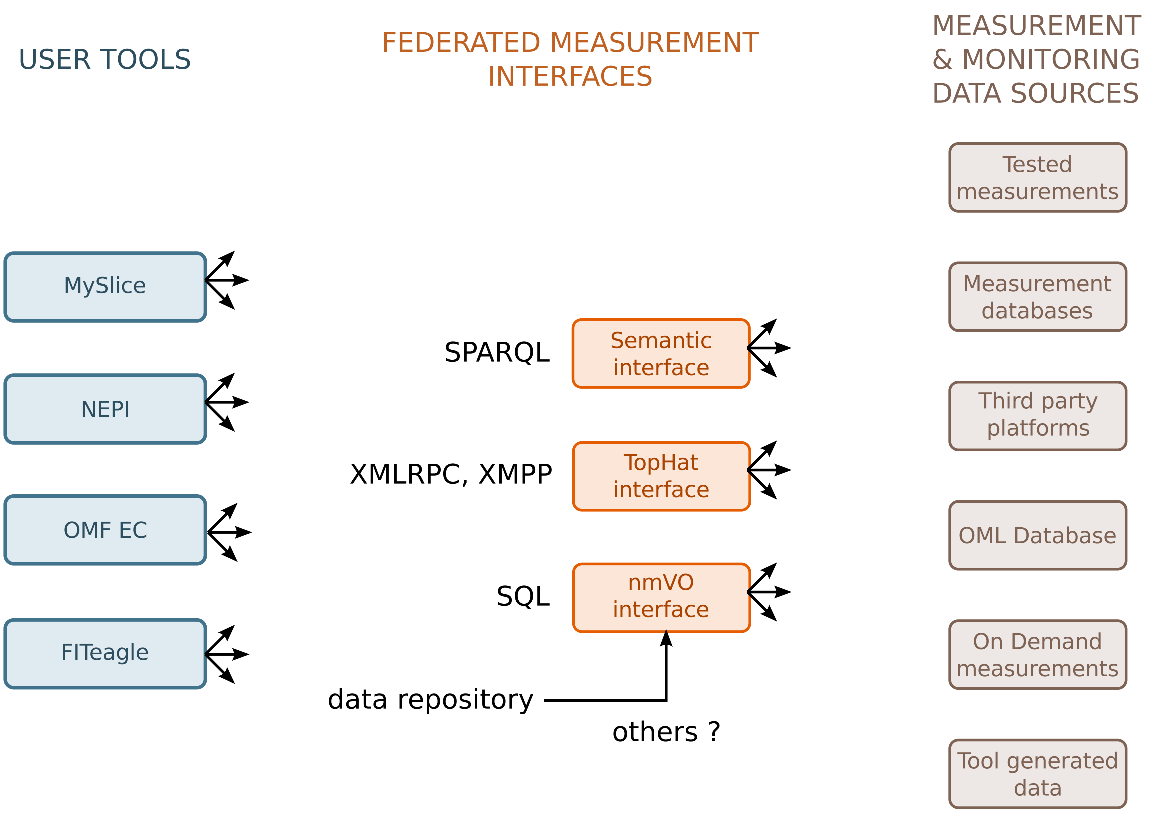


Figure 21: Integration of the different measurement systems

5.3.3 Candidate components and their status

Figure 21 represents various classes of data sources on the left, covering various data sources and services available and required by experiments on federated testbeds. The same set of users tools as considered before is represented on the right. Finally, the set of candidate solutions is displayed in the middle.

This report won't enter much into the details of those solutions and of the different data sources as it is the object of [5], which will be due soon after this report.

5.3.4 Integration status of measurements in user tools

Integration of measurements in user tools is currently limited. MySlice embeds some information aggregated by Topat during the slice creating phase, allowing user to select resources according to a wide range of criteria about testbeds resources and measurement about the substrate. It also allows users to trigger on-demand measurements trough its API during an experiment, or to retrieve past and archived measurements. NEPI is also currently considering the integration of Topat measurements, and this is the object on ongoing discussions.

5.4 Future directions

The objective of the plugfest was the opportunity to assess the advancement of the integration of the various components developed and deployed in many major projects involved in the federation of networking testbeds. The agenda made it possible that people get informed of the current context, and at the same time benefit from the colocation of many technical people that can best inform about the various tools. This allowed to solve several technical issues but also think on how future directions can be made more consistent with the project objectives, and leave the event having the basic knowledge, eventually the software deployed and knowing how to progress.

We had very positive feedback about this plugfest, and especially about the working group sessions, which gives many incentives to organize future such events. Suggestions include less time dedicated to presentations – possibly lightning talks to give updates about the progression –, a theme and an objective that should be reached at the end, and potentially some coding sessions.

6 Conclusion

This document has summarized the different activities performed in the context of the OpenLab project, related to usage and activity monitoring and reporting.

The most important outcome of this work has been to identify metrics of interest, and put in place a tool, the FreeTRex website, so that we can centralize and share information about the different results, software components and best practises that could be collected. By displaying such information publicly, we expect that newer platform can benefit from what has been done, and have incentives to open up their information also.

This work received its input from a census that was performed among the different partners, requesting information about those metrics, and the different tools that were exploited. Both the list of tool and the analysis are also joined to this document.

We expect our contribution to start tackling an important challenge in the testbed federation, the issue of usage and activity monitoring. While is does not provide a definite answer on how to proceed, we expect it to provide useful resources as well as a strong base on which to build further.

A Workshop agenda and list of participants

A.1 Technical agenda

The full version of the agenda including the different presentations is available online: `http://openlab-plugfest.npafi.org`.

DAY 1 - Wednesday, January 23rd: TUTORIALS, PRESENTATIONS, DEMOS

09H30 - 10H00	*Welcome coffee (25-26/105)*
10H00 - 10H10	Introduction (Jordan Aug, UPMC)

Components ***(approx. 30 min talks and 10 min discussion)***

10H10 - 10H50	**SFA & SFAWrap tutorial** (Mohamed Amine Larabi, INRIA) slides (PPT)
10H50 - 11H30	**MySlice tutorial** (Jordan Aug, Loc Baron, UPMC) slides (part 1) (PDF) video (part 2) (OGV) slides (part 3) (PDF)
11H30 - 12H10	**NEPI tutorial** (Alina Quereilhac, INRIA) slides (PDF)

Integration efforts ***(approx. 15 min talks/demos and 10 min discussion)***

12H10 - 12H30	**NEPI/OMF integration** (Julien Tribino, Alina Quereilhac, INRIA; Pieter Becue, iMinds) *No slides yet!*
12H30 - 14H00	*Lunch (@ L'Ardoise - Jussieu Campus)*
14H00 - 14H25	**NEPI/MySlice integration** (Lucia Guevgeozian, INRIA) slides (ODP)
14H25 - 14H50	**FITeagle/MySlice integration** (Stefan Harder, Mitja Nikolaus, TUB) slides (PPT)
14H50 - 15H15	**IoT-LABR, Integration of SFA, OMF/OML, MySlice** (Frdric Saint-Marcel, Sandrina Avakian, Anthony Garcia, INRIA) slides (PPT)

15H15 - 15H40	**How to run mobile experiments @ iMinds w-iLab.t** (Vincent Sercu and Pieter Becue, iMinds) slides (PPT)
15H40 - 16H05	**NOVI Services to Control, Manage and Monitor Virtual Testbeds in SFA-enabled Platforms** (Jzsef Stger, ELTE) *No slides yet!*
16H05 - 16H40	*Coffee Break (25-26/105)*
16H40 - 17H05	**Making NITOS SFA-compliant - NITOS Scheduler** (Harris Niavis, UTH) slides (PPT)
17H05 - 17H30	**Building an OMF 6 Resource Controller for OpenFlow support** (Kostas Choumas, UTH) slides (PPT)
17H30 - 18H00	Preparation of working groups

DAY 2 - Thursday, January 24th : WORKING GROUPS

09H00 - 09H30	*Welcome coffee*
09H30 - 12H30	**Working groups & discussion sessions**: 09H30 - 11H00 : interoperability of experiment controllers 11H00 - 12H30 : integration of measurement systems
12H30 - 14H00	*Lunch (@ L'Ardoise - Jussieu Campus)*
14H00 - 16H00	**Working groups (continued)** - Main topics: MySlice, NEPI, OMF/OML
16H00 - 16H30	*Coffee Break*
16H30 - 18H00	**Working groups (continued)** - Main topics: MySlice, NEPI, OMF/OML

DAY 3 - Friday, January 25th : WORKING GROUPS & WRAP-UP

09H00 - 09H30	*Welcome coffee*
09H30 - 11H30	**Working groups (continued)**
11H30 - 12H30	**Wrap up session**

List of participants

Name	Institution	Country
Jordan Aug	UPMC	FR
Sandrine Avakian	INRIA	FR
Susanna Avessta	UPMC	FR
Loc Baron	UPMC	FR
Pieter Becue	iMinds	BE
Kostas Choumas	UTH	GR
Christoph Dwertmann	NICTA	AU
Timur Friedman	UPMC	FR
Anthony Garcia	INRIA	FR
Lucia Guevgeozian	INRIA	FR
Stefan Harder	TUB	DE
Mohamed Larabi	INRIA	FR
Giuseppe Lettieri	UNIPI	IT
Jorge López de Vergara	UAM	ES
Harris Niavis	UTH	GR
Mitja Nikolaus	TUB	DE
Thierry Parmentelat	INRIA	FR
Mario Poyato Pino	UAM	ES
Alina Quereilhac	INRIA	FR
Salma Rebai	TSP	FR
Frdric Saint-Marcel	INRIA	FR
Guillaume Schreiner	UDS	FR
Ciro Scognamiglio	UPMC	FR
Vincent Sercu	iMinds	BE
Michael Sioutis	UPMC	FR
Jószef Stger	ELTE	HU
Christos Tranotis	UoP	GR
Julien Tribino	INRIA	FR
Erkan Valentin	UDS	FR
Wim Venderberghe	iMinds	BE
Pierre Vigreux	UPMC	FR
Mohammed Yasin	UPMC	FR
Roberto Quilez	INRIA	FR
Anne-Sophie Tonneau	INRIA	FR
Marouen Mechtri	TSP	FR
Thiago Abreu	UCBL/Lyon	FR
Nguyen Nghi	UCBL/Lyon	FR
Stefan Covaci	TUB	DE

References

[1] Free-t-rex website, 2013.

[2] Manifold website, 2013.

[3] Myslice website, 2013.

[4] Openlab plugfest website, 2013.

[5] Yahya Al-Hazmi, Alexander Willner, Tom Pfeiffer, Jordan Augé, Marc-Olivier Buob, Alina Quereilhac, Lucia Guevgeozian, Sándor Laki, János Szüle, and József Stéger. Interoperability of tools and data repositories. Technical report, 2013.

[6] Panayotis Antoniadis, Serge Fdida, Timur Friedman, and Vishal Misra. Federation of virtualized infrastructures: sharing the value of diversity. In *Proceedings of the 6th International COnference*, page 12. ACM, 2010.

[7] Thomas Bourgeau, Jordan Augé, and Timur Friedman. Tophat: supporting experiments through measurement infrastructure federation. In *Proceedings of TridentCom'2010*, Berlin, Germany, 18-20 May 2010.

[8] The OpenLab Consortium. Description of work, annex to contract, June 2011.

[9] István Csabai, Attila Fekete, Péter Hága, Béla Hullár, Gábor Kurucz, Sándor Laki, Péter Mátray, József Stéger, Gábor Vattay, Felix Espina, Javier Aracil, et al. Etomic advanced network monitoring system for future internet experimentation. In *Testbeds and Research Infrastructures. Development of Networks and Communities*, pages 243–254. Springer, 2011.

[10] Fabien Hermenier and Robert Ricci. How to build a better Testbed: Lessons from a decade of Network Experiments on Emulab. *Proceedings of TridentCom'2012 – Thessaloniki, Greece – June 22-23, 2012*, 2012.

[11] Béla Hullár, Sándor Laki, József Stéger, István Csabai, and Gábor Vattay. Sonoma: A service oriented network measurement architecture. In *Testbeds and Research Infrastructure. Development of Networks and Communities*, pages 27–42. Springer, 2012.

[12] Alina Quereilhac, Alexander Willner, Yahya Al-Hazmi, Christos Tranoris, Pieter Becue, Vincent Sercu, Jordan Augé, Thierry Rakotoarivelo, Andras Gulyas, Gergely Biczok, and Papagianni Chrysa. Experimental plane – experiment controllers. Technical report, 2012.