D6.2 First progress report on the Experimental Execution Environment

Due Date	Month 18
Delivery	Month 18+1week, PO agreed
Lead Partner	BADW-LRZ
Dissemination Level	PU
Status	Final
Approved	Internal review yes
Version	V1.0

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 671564.

DOCUMENT INFO

Date and version number	Author	Comments
01.03.2017 v0.1	Vytautas Jančauskas	First draft
23.03.2017 v0.2	Vytautas Jančauskas	Incorporating comments by Helmut Heller and Stephan Hachinger
01.04.2017 v0.3	Vytautas Jančauskas	Incorporating comments by Tomasz Piontek and Saad Alowayyed
04.04.2017 v0.4	Vytautas Jančauskas	Incorporating comments by Bartosz Bosak, Tomasz Piontek
06.04.2017 v1.0	Vytautas Jančauskas	Final edits

CONTRIBUTORS

Contributor	Role
Vytautas Jančauskas	WP6 leader and author of this deliverable
Tomasz Piontek	WP5 leader/ internal reviewer
Saad Alowayyed	WP2 contributor/ internal reviewer
Helmut Heller	WP6 contributor
Stephan Hachinger	WP6 contributor

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

1 Executive summary

The aim of Work Package 6 (WP6) is to provide an Experimental Execution Environment (EEE) for the project's scientists and external users of ComPat tools. EEE aims to be a collection of software, computing resources and custom developed tools that allows users to submit simulations to be performed at the projects' computing sites. To this end the sites have to provide a common software infrastructure and the EEE must have a single, unified job submission system. The submission system we use is QCG (https://www.qoscosgrid.org/), developed at the Poznan Supercomputing and Networking Center. The uniform software infrastructure is provided using environment modules (http://modules.sourceforge.net/). ComPat specific naming conventions are used for these modules to achieve consistency. It is important for the goals of the project that the user does not know or care in which particular site their software is run. The computing sites and computing nodes on those sites have to be automatically chosen in order to optimise energy consumption and in order to best suit a particular application.

Before the Month 18 of the project, a user's manual for the EEE was prepared. It is available on a publicly accessible wiki website. It is being maintained and updated constantly by WP6 members. The manual can be found at http://compat-eee-wiki.drg.lrz.de (and also in the appendix). The manual is prepared with the intention of allowing users to start executing their simulations on the EEE.

Along with the manual we have implemented and hosted on a virtual machine at LRZ a maintenance database. It is used to schedule and announce planned downtimes. Computing sites are asked to fill in a form to add new scheduled downtimes and the users can view them via a website. Alternatively, this information can be gathered from other sources, such as databases or public websites. For example, we use a Cron job to get this information from the GOCDB system. The maintenance database takes the form of a RESTful web service with the API available on the wiki. Users can query the list of upcoming, current and past scheduled downtimes by following this link: https://nagios-compat.drg.lrz.de:5000. They need to authenticate using their grid certificate and this information is only accessible to people who are part of the EEE.

2 EEE users' manual

We have prepared the first version of the EEE users' manual. The purpose of the manual is to provide project users with instructions on how to start using the EEE. This means: getting a Grid certificate, registering on the various supercomputing sites, and using the common ComPat software infrastructure (e.g. instructions on using QCG). We also provide EEE specific information to the project's developer community. DokuWiki was chosen as the platform to host the manual. We chose a Wiki format because this allows for easy collaborative editing of the document. For example, it allows for more frequent updates, with contributing members being able to edit the pages independent of one another. We chose DokuWiki because it is simple to install and uses plain text for storing wiki pages. This allows for automatic generation of wiki pages if such a need arises. For example, scheduled downtimes can be reported as an automatically updated wiki page (functionality planned). The wiki can be reached via the main project web site. It can also be found by following the link.

http://compat-eee-wiki.drg.lrz.de/dokuwiki/doku.php

In summary, it provides the following information:

- Instructions on how to obtain a Grid certificate. Grid certificates are used as a means of authentication in this project. There are currently two possibilities that the project's computing sites support: a "real" Grid certificate or a PLGrid SimpleCA certificate. The latter one can be used by partners who for some reason have trouble acquiring certificates from https://www.eugridpma.org/.
- How to register on project's supercomputing sites. The procedure currently is different on each participating supercomputing site. This is due to different policies in different organizations.
- A software request form. This is used to request software modules to be made available on project's sites.
- A list of available environment modules.
- Environment module naming conventions.
- Technical information like the APIs for querying the maintenance database and service uptime statistics.
- QCG user's manual: http://www.qoscosgrid.org/trac/qcg/wiki/user_information.

3 EEE hardware infrastructure

Currently the hardware for the EEE is provided by three supercomputing sites. We summarize the available machines below. One of the goals of the project is to find ways to assign computing jobs to

machines that are most suited for executing them – be it in terms of energy efficiency or compute time, or both. Therefore, a heterogeneous hardware stack is an advantage.

3.1 LRZ

Leibniz Supercomputing Centre of the Bavarian Academy of Sciences and Humanities (LRZ) makes SuperMUC (Figure 1: SuperMUC) available to the project. SuperMUC currently consists of two phases (physically separate subsystems) – Phase 1 and Phase 2.

Figure 1: SuperMUC

Summary of hardware available in each phase is given below. At the time of this writing, users of the ComPat middleware (QCG) only have access to Phase 1 of the machine. Integrating Phase 2 is planned for the near future. SuperMUC uses the LoadLeveler job scheduling system by IBM.

Table 1: LRZ Hardware

	Phase 1			**Phase 2**
Installation date	2011	2012	2013	2015
Processor	Westmere-EX Xeon E7-4870 10C	Sandy Bridge-EP Xeon E5-2680 8C	Ivy-Bridge (IvyB) and Xeon Phi 5110P	Haswell Xeon Processor E5-2697 v3
Number of nodes	205	9,216	32	3,072
Number of cores	8,200	147,456	3,840 (Phi)	86,016
Peak performance (PFlop/s)	0.078	3.2	0.064 (Phi)	3.58
Memory (TB)	52	288	2.56	194
Power consumption (MW)	<2.3			~1.1

3.2 PSNC

Poznan Supercomputing and Networking Center (PSNC) currently makes two machines available to the ComPat project – Inula and Eagle (Figure 2: Eagle). A summary of their capabilities is provided in the table. Eagle uses SLURM and Inula uses PBS for job scheduling. Inula also provides access to GPGPU cards (Nvidia Tesla M2050). It has 205 GPGPU nodes.

Figure 2: Eagle

Table 2: PSNC Hardware

	Eagle	**Inula**
Processor	Intel Xeon E5-2697	AMD (Interlagos), Intel Xeon E5-2697
Number of nodes	1,233	681
Number of cores	32,984	3,360
Peak performance (PFlop/s)	1.4	0.1384
Memory (TB)	120.6	6.56
Power consumption (MW)	n/a	n/a

3.3 STFC

At the time of this writing Science & Technology Facilities Council's Hartree Centre provides their Neale machine (Figure 3: Neale) to the ComPat project. Neale is an immersion cooling (environmentally friendly) machine.

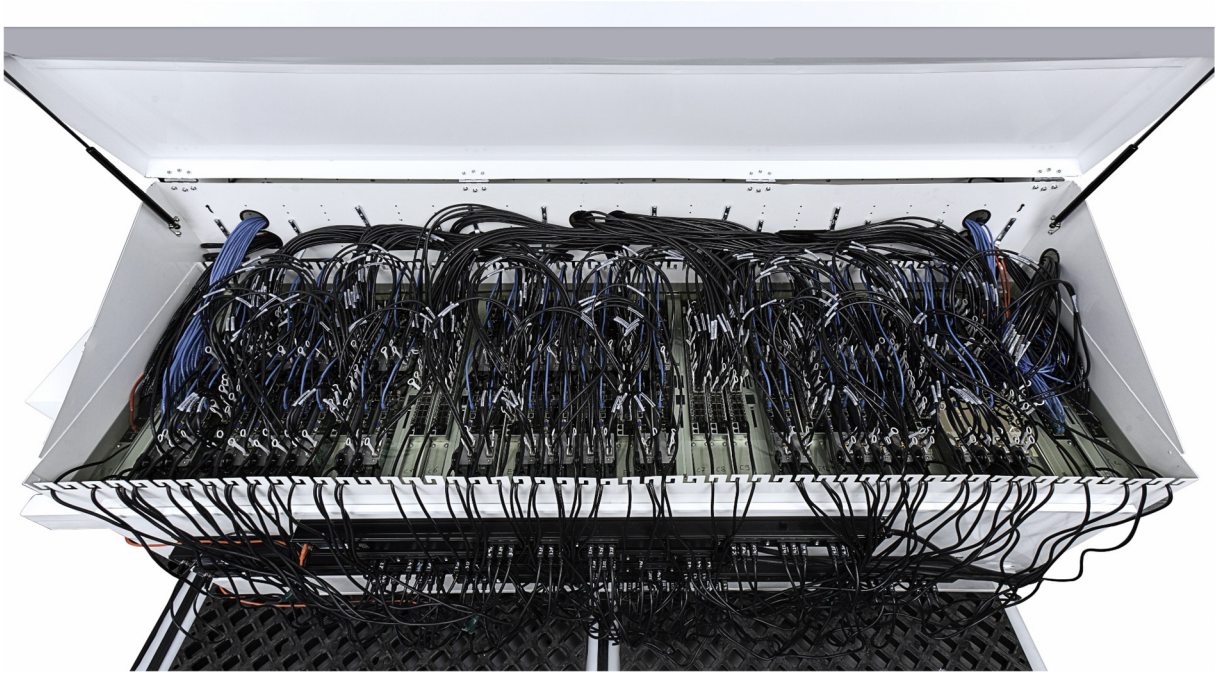

Figure 3: Neale

Table 3: STFC Hardware

	Neale
Processor	Intel Xeon (Ivy Bridge E5-2650v2. 2.6GHz)
Number of nodes	120
Number of cores	1,920
Peak performance (PFlop/s)	n/a
Memory (TB)	7.68
Power consumption (MW)	n/a

3.4 Energy monitoring

We work together with WP4 in order to make sure energy consumption data is available. This work is described in WP4's deliverable.

4 EEE software infrastructure

In the following sections we provide the current state of software organisation and software solutions we have developed. Primary focus is to provide a consistent user experience between computing sites and allow for easy deployment of software on those sites. We monitor all relevant software services running on our sites using Nagios (Figure 4: Nagios).

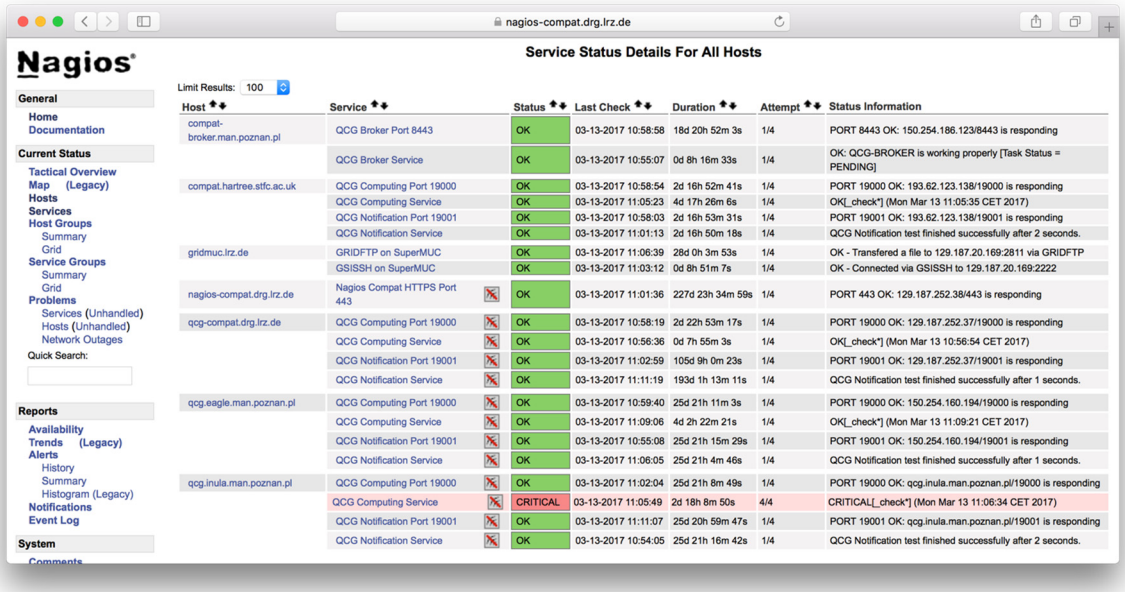

Figure 4: Nagios

4.1 VOMS

It was planned in the initial description of work for this work package that we will use Virtual Organisation Membership Service (VOMS) in order to simplify the process of getting access to the EEE and therefore ComPat related services. However, as noted in D6.1 and as was discussed with the leaders of other work packages (namely WP5), we only have users internal to the project. This simplifies the user registration procedure. Because of the aforementioned, setting up VOMS does not provide enough benefit compared to the effort required. Therefore, it was decided not to rely on it and keep the simpler system of granting users permissions to use resources individually.

4.2 EEE module system

The ComPat project's computing resources are by the project's very nature split between different computing sites located in different organizations and different countries. The users of these resources should not be expected to adapt to each site to run their software. One of the goals of the project is to automatically run jobs on hardware most suited for that particular job. To this end, a unified software infrastructure is needed across the sites. This means: commonly used libraries have to be built the same way and provided using the same naming conventions. This unified software infrastructure is provided via environment modules (http://modules.sourceforge.net/). On each computing site software is organized in to an agreed upon directory structure. The users are meant to follow the directory structure when building their own software. Commonly used libraries and tools that are specific to the project also have associated module files for easy access. This partly relies on sites providing a shared

directory that can be accessed by all the users that are part of the project. This directory can be accessed via the $COMPAT_SHARED variable. The directory structure is as follows:

$COMPAT_SHARED/
- /Astro
- /Bio
- /Common
- /Fusion
- /Modules

Astronomy related codes will be built under the Astro directory, biology, computational chemistry and other life science codes will be built under Bio and nuclear fusion simulations will be built under Fusion. The Common directory will contain libraries and other commonly used software, such as MUSCLE2. Finally, the Modules directory contains environment module files. This directory will have to be added to the $MODULEPATH variable in order for the module system to find it. The contents of the Modules folder are synchronized between the different computing sites using a Globus endpoint. The endpoint is called compat-module-system-2 and it's contents can be easily transferred to any participating site for use. In the table below we summarize the modules currently available to EEE users.

Table 4: EEE Modules

Module	Description
compat	Will load ComPat environment variables, for example $COMPAT_SHARED, which is used to access the ComPat shared directory.
compat/common/ruby	RUBY programming language needed by MUSCLE2.
compat/common/muscle2	MUSCLE 2 - Multiscale Coupling Library and Environment is a portable framework to do multiscale modeling and simulation on distributed computing resources.
compat/common/amuse	AMUSE - Astrophysical Multipurpose Software Environment.
compat/common/namd	NAMD - a parallel molecular dynamics code designed for high-performance simulation of large biomolecular systems.
compat/common/amber	AmberTools – a package of molecular simulation programs.

Here is the relevant section of the module avail command on the SuperMUC, for example:

```
------------ /gpfs/work/pr92ge/di25pul/Modules ------------
compat/1.0(default)                    compat/common/muscle2/compat-1.1
compat/app/fusion/1.0(default)         compat/common/namd/2.10
compat/common/amber/16                 compat/common/ruby/1.9.3(default)
compat/common/amuse/089e701            compat/dev/fusion/1.0(default)
compat/common/muscle2/612248f(default)
```

4.3 Maintenance database

The maintenance database is used to store scheduled downtimes. Scheduled downtimes differ from unscheduled ones in that they are known in advance. This is an important distinction when calculating service availability statistics. The periods of scheduled downtime don't count towards downtime for that particular service or machine. Currently there are several ways to add a scheduled downtime to the downtime database. One of them is the scheduled downtime web-form (Figure 5: Scheduled downtime entry form). It has to be filled in by hand by authorised users.

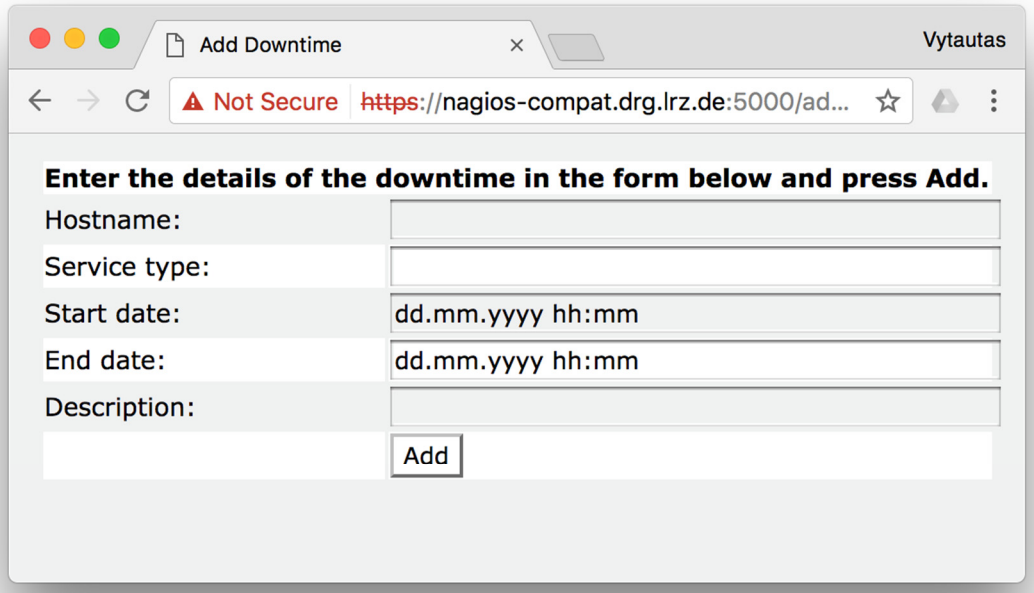

Figure 5: Scheduled downtime entry form

Another option is to automate the database update procedure. This is only possible if the site in question provides scheduled downtimes in a well defined format. At the moment this is only true of PSNC where downtimes can be queried using a REST interface. PSNC has agreed to mark downtimes that apply to ComPat related services with a special flag. This way we can get the downtimes in XML format by using the following query:

```
wget --no-check-certificate --certificate=/home/nagios/.globus/usercert.pem \\
    --private-key=/home/nagios/.globus/userkey.pem --private-key-type=PEM \\
    -O gocdb.xml \\
    https://goc.egi.eu/gocdbpi/public/?method=get_downtime&scope=&service_extensions=(ComPat=yes)
```

The resulting XML file is then parsed and the downtimes matching the query are added to the database.

The database can further be queried to get data about current and upcoming scheduled downtimes. Examples of queries and their results are given below.

https://nagios-compat.drg.lrz.de:5000/upcoming

[{"service_type": "QCG Compute", "hostname": "compat.hartree.stfc.ac.uk", "start_date": 1488351600.0, "end_date": 1488387600.0, "description": "Maintenance day - system at risk"}, {"service_type": "QCG Compute", "hostname": "compat.hartree.stfc.ac.uk", "start_date": 1489561200.0, "end_date": 1489597200.0, "description": "Maintenance day - system at risk"}]

https://nagios-compat.drg.lrz.de:5000/downtime

[]

In the second case there are no scheduled downtimes happening right now so the result is an empty list. As can be seen in the above examples, the API uses a simple JSON based data format, where downtimes are given in a list, each downtime consisting of the following information:

```
{
“hostname” : string,
“service_type”: string,
“start_date” : timestamp,
“end_date” : timestamp,
“description” : string
}
```

So far this service is used by QCG to gather information about upcoming and ongoing service downtimes.

Supplementing the maintenance database, we have an availability database. The availability database runs on the same virtual machine and periodically (once every hour) captures the state of services that are monitored by Nagios.

4.4 Service availability statistics

Key performance indicators (KPIs) for measuring service availability were proposed in D6.1. In order to calculate the value of these performance indicators we need to track service availability. This is

done via Nagios. We implemented a database that is updated by a cron job that captures Nagios state and writes it to the database. The job is executed every hour and writes a record that shows the state the service was in along with other metadata. This gives us an option to calculate service availability statistics in various ways. One such way measures the percentage of time that the service was functioning correctly over a user specified time period. That allows us to estimate the quality of service availability in the project and highlights problem areas. We provide a web service that is accessible to all members of the project:

https://nagios-compat.drg.lrz.de:5000/performance/dd.mm.yyyy/dd.mm.yyyy

To access the indicators, users indicate the start and end dates of the period they are interested in in the template URL. Service availability statistics are calculated for that period and plots are generated and downloaded to the browser. An example of that is given in (Figure 6: Service availability statistics). This shows service availability from 8 March 2017 to 1 April 2017. The start of this period corresponds to when we have finally finished integrating various sites in to the EEE. Namely, we have finished integrating the Neale machine at Hartree in March of this year. Please note that each Nagios monitoring probe has its own pie chart in the figure. In D6.1 the following table was proposed for evaluating service availability:

Excellent	**Very Good**	**Good**	**Mediocre**	**Poor**
Above 99%	99%-95%	95%-85%	85%-70%	Below 70%

As can be seen, of the services we monitor (currently), 15 would fall under Excellent, 3 under Very Good, 4 under Good and 1 under the Poor category. The monitoring of qcg.inula.man.poznan.pl QCG computing service has revealed problems which will be subject to further investigations and application of WP6 procedures to improve the stability and availability of the resource. The KPIs are expected to improve since we have finished integrating all the services in to the EEE only recently.

Figure 6: Service availability statistics

4.5 Software request form

ComPat users can request software and corresponding environment modules to be provided on all sites by using the ComPat EEE software request form. The form can be accessed from the ComPat wiki (http://compat-eee-wiki.drg.lrz.de/dokuwiki/doku.php?id=software_request). Users authenticates with their DN and specify the software package they need.

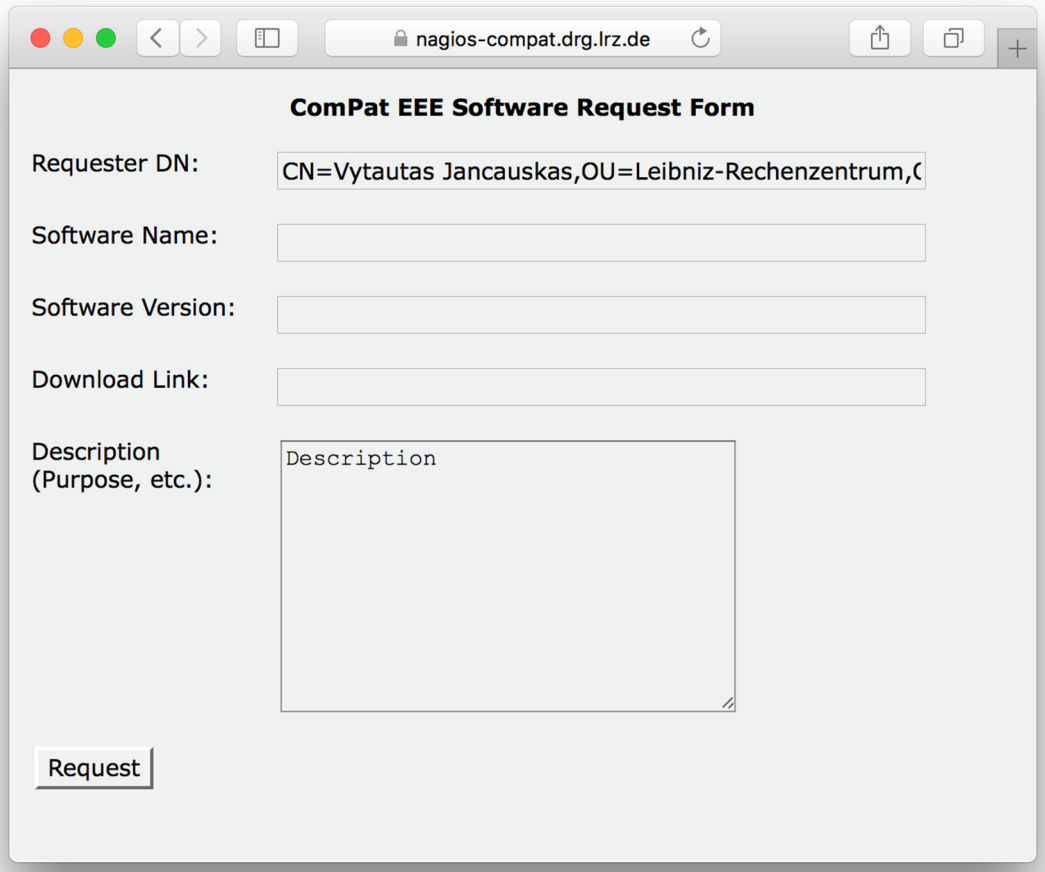

Figure 7: Software request form

5 Other issues

In this section we explain various solutions and issues that have come up as part of maintaining the EEE.

5.1 SimpleCA certificates

In order to allow researchers who do not have European grid certificates to participate in the project a solution was needed. Grid certificates are used to authenticate in order to use computing resources. Most sites participating in the project will hand out Grid certificates. Users not part of European

institutions, like our project partners from ITMO, will generally not have these certificates. In order to allow them to use the project resources, our computing sites have decided to allow users with certificates issued by PLGrid SimpleCA certificate authority. These certificates can be obtained by following the instructions of the user manual found by following the instructions for registering on the PSNC here: http://compat-eee-wiki.drg.lrz.de/dokuwiki/doku.php?id=psnc_instructions. The process used to obtain these certificates involves a project technical manager approving the users, therefore there is no risk of misuse by people not belonging to the project. This is just a catch-all way to help users who may not have Grid certificates to conveniently get access to the ComPat infrastructure.

5.2 Work-around for computing sites that do not allow outside access

Some computing sites do not allow outside access from compute nodes. Some others severely restrict outside access from within login nodes too. For example, the SuperMUC system does not allow Internet access from either. In practice, this causes problems when building modern software that often relies on ad-hoc package managers (i.e., Anaconda for Python.) It also means it is very hard to use systems that need a feedback of information in order to function. Examples of such systems are QCG and RADICAL Cybertools. Both of which we need to deploy. QCG because it is a vital part of the ComPat software infrastructure (being the middle-ware of choice for ComPat) and RADICAL Cybertools because some users want to use it. The procedure of integrating these tools at these sites is documented in detail in the D5.2 deliverable.

5.2.1 QCG

QCG is the ComPat middleware suite of choice. It is expected that EEE users will use QCG for their job submissions and most ComPat services are in turn based on QCG. At LRZ, the QCG server runs on a virtual machine within the LRZ network. This virtual machine is allowed to connect to SuperMUC, the feedback mechanism relies on an SSH tunnel.

5.2.2 RADICAL Cybertools

RADICAL Cybertools is a distributed computing middleware suite. Some of our users rely on it to execute their software. It depends for its operation on having access to an outside MongoDB (an open source, document-based) database. This is not possible to do on SuperMUC in general. SuperMUC worker nodes don't have access to the outside, only to the log-in nodes. Furthermore, SuperMUC login nodes are somewhat limited in that they can only be accessed from specific IPs and block ingoing and outgoing traffic from everywhere else. For RADICAL Cybertools we have negotiated privileged access to agreed-upon IP addresses so that outgoing SSH traffic is now possible.

6 Conclusions

Before the month 18 we have achieved a unified and stable execution environment, built on top of resources provided by our partners. We have also prepared and made available to project users the EEE users' manual. The manual is actively being updated by project members. Since the last deliverable QCG was deployed on all computing sites and it is ready for use by project members (jobs can be submitted via QCG to all project's sites). RADICAL Cybertools were deployed on SuperMUC. We have implemented a scheduled downtime database, software request form, and service availability tracking.

In the upcoming months we will continue contributing to the manual, creating software modules and deploying them on sites. We will also take care of getting new users using project's infrastructure and making sure users can deploy their software, especially when intervention from the sites administrators is required.

7 Annexes

The user's manual is presented as part of this deliverable as described in the DOW. It was exported at 2017-04-04 to PDF and attached to this document.

Experimental Execution Environment (EEE) for the ComPat project

This wiki contains information relevant to users and developers of the Experimental Execution Environment of the ComPat project .

Currently, there are the following resources (sites/machines) in the EEE:

1. SuperMUC at LRZ
2. Eagle, Inula at PSNC
3. STFC

Users' Manual

EEE Modules and Conventions

Performance Indicators

Software Request

Maintenance Database API

ComPat User DNs

QCG User's Guide

If you have any suggestions for the wiki please contact me:

Vytautas.Jancauskas@lrz.de

Users' Manual

Information here should help the users of the Experimental Execution Environment (EEE) of the ComPat project get their applications running.

Getting a grid certificate

In order to use the resources that the ComPat EEE makes available you have to have a grid certificate. Grid certificates bind a cryptographic key to a Distinguished Name (DN) and is used as a way to identify you as a valid computing resource user on all the different sites of the computing grid.

Now for the practical points:

1. Figure out who is your Grid CA (Certification Authority which will issue your grid certificate) - usually you need to choose the closest CA (see https://www.eugridpma.org/members/worldmap/)
2. Request a certificate from the CA
3. Install that certificate in your browser and in your ~/.globus folder (i.e. .globus folder inside your home directory - also for Windows)

LRZ provides some documentation on certificates (which partly apply to other computing centres as well) on:

https://www.lrz.de/services/compute/grid_en/certificate_en/person-certificate_en/

Further LRZ information on grid computing, tools (globus toolkit, etc.) and handling which you might find useful:

https://www.lrz.de/services/compute/grid_en/

https://www.lrz.de/services/compute/grid_en/grid-middleware_en/globus_guide_en

Site Specific Instructions

Instructions for getting on SuperMUC

Instructions for getting on PSNC

Instructions for getting on STFC

Information about ComPat resources

SuperMUC Resource Info PSNC Resource Info STFC Resource Info

SuperMUC Instructions

Send an email with the personal information template from below filled in to grid-support@lrz.de to request an account on SuperMUC.

If you're a Mac or Linux user, you can extract your DN from your certificate with the following command (the point of the Python script is to automagically convert the DN to the right format):

```
openssl x509 -in ~/.globus/usercert.pem -noout -subject | python -c "import
sys; print ','.join(reversed(sys.stdin.read().strip()[9:].split('/')[1:]))"
```

Template Email Request

```
Dear grid-support,

I would like to request an account on SuperMUC as part of
the ComPat project.

My personal information is as follows:

Academic Title (if any):
First Name:
Last Name:
Nationality:
Office Address:
Telephone no:
Email address:
Certificate DN (distinguished name, also called subject) :

Many thanks in advance.

Best rgds,
Your name
```

Getting on SuperMUC

To get on SuperMUC you will need gsissh which is part of the Globus Toolkit . Download and install the version for your operating system. On a Mac it will be put under `/Library/Globus/bin`. You will want to add it to your `.bash_profile` path. You will want to set-up your trust roots by running `myproxy-get-trustroots -b -s myproxy.lrz.de -p 80`. Also, you need to create a proxy (a short-lived copy) of your certificate by running `grid-proxy-init` and typing your certificate passphrase. After that you can log on by running `gsissh gridmuc.lrz.de -p 2222`.

You can put your intermediate results on $SCRATCH and use $WORK to store your data and software.

Optional: You can create directories under $WORK and make them readable/writable by the ComPat project group (pr92ge) to share software or data with your colleagues. Use the command `chmod g+rw $WORK`.

To get data in and out of SuperMUC you can use GridFTP command (`globus-url-copy -r ./filetoupload.txt gsiftp://gridmuc.lrz.de/~/` to copy from your machine to your home on SuperMUC; -r means recursive and also copies directories). Or use or the convenient Globus Online website.

To use Globus Online follow the instructions below:

1. Create a Globus Online account here .
2. Go to https://www.globus.org/app/transfer and choose `Globus ID` as your organisation, and click `No thanks, continue` when it's asking if you want to use an existing account.
3. When using the Globus Online web app you need to specify gridmuc as your endpoint.
    1. Upload a proxy certificate to a myproxy server. We recommend using `myproxy.lrz.de` and you can do this by running `myproxy-init -l` *`username`* `-s myproxy.lrz.de -p 80`. This *username* can be freely chosen and is not related to anything else. When asked for myproxy passphrase invent a new password specific to myproxy.
    2. In the Globus Online website select GridMUC as one of the end-points. Just type GridMUC in one of the search fields and it should pop-up.
    3. In the GridMUC endpoint log-in window specify the user name and password you used in the `myproxy-init` command. Screenshot
4. To set-up the other endpoint you need to install Globus Connect Personal by following the steps outlined on the website. **BEWARE**: there seems to be a bug in the Linux version of Globus Connect. After following the usage instructions you will receive a very non-descript error message. You can ignore it (just click OK), but you have to quit Globus Connect application and start it again before you can connect.
5. In Globus Online specify the name you chose above as the other endpoint.
6. If you did everything right, the transfer window should look something like this.

Extensive instructions on using SuperMUC can be found on the LRZ website:

Getting on SuperMUC

Using LoadLeveller on SuperMUC

SuperMUC currently uses LoadLeveller as it's batch system. You can find some common usage scenarios, queue descriptions and other information by following the link below.

https://www.lrz.de/services/compute/supermuc/loadleveler/

Accessing the Shared Directory

To access the shared directory that contains tools and data that are meant to be shared between the ComPat group, once logged in to SuperMUC, do:

```
cd $WORK
cd ../di25pul
```

This is the shared directory. The agreed upon directory tree structure is:

```
di25pul/
```

```
Astro/
Bio/
Common/
Fusion/
```

You should put commonly used tools (NAMD, MUSCLE2, AMUSE, ...) in the Common subdirectory.

Access to PSNC/PLGrid resources

To create an account and to get access to PLGrid infrastructure please follow the procedure below:

Create an account

go the the https://portal.plgrid.pl page. You will be redirected to the log-in page. at the bottom of the log-in page please click **“Nie masz konta? Zarejestruj sie”** (Don't have an account? Register)

Fill in the questionnaire:

1. **Imie** - First name.
2. **Nazwisko** - Last name.
3. **Login** - The proposed user name. It has to start with “plg” prefix and must have 6-15 letters. It cannot start with “plgrid” string.
4. **Haslo** - Password. Min 10 characters, at least 5 different, at least 1 digit, cannot contain first name, last name and login, it must have at least one capital letter and small letter.
5. **Potwierdz haslo** - Confirm the password.
6. **Adres e-mail** - Email (a real one, a confirmation message will be sent to it).
7. **Telefon** - Phone (you can cheat here).
8. **Czy jestes zarejestrowany w bazie Nauka Polska** - Are you registered in Polish Science Database - choose “Nie” - No.
9. **Warunki korzystania z Infrastruktury PLGrid** - PLGrid usage rules - select “Akceptuje warunki” - (I accept).
10. **Czy jestes Robotem?** Are you a robot? - answer according to the real state.

Press **“Wyslij formularz rejestracyjny”** - Submit a registration form.

Confirm registration

Go to the mailing box you provided during the registration and confirm the registration by clicking the first link in the mail you got

Add affiliation

1. Login to the https://portal.plgrid.pl using your credentials
2. Click **Afiliacje** (Affiliations) on the left site (confirmation of the registration is needed for that)
3. press **Dodaj** (Add) button
4. **Dodawanie Afiliacji** - Adding affiliation. **Aby dodac afiliację wybierz swój status** - to add affiliation select your status
5. Select - **Podopieczny pracownika jednostki naukowej** - “Charge” (person in someone else care)
6. **Twoj opiekun** - your supervisor
    1. **Imie** - first name - Tomasz
    2. **Nazwisko** - last name - Piontek
    3. e-mail: piontek@man.poznan.pl

7. **Organizacja naukowa** - scientific organization
    1. **Jednosta** - Unit - **Polska Akademia Nauk** (Polish Academy of Science). start typing Polska and you should get hint
    2. Press **“Dodaj podjednostke”** - Add sub-unit
    3. provide **Instytut Chemii Bioorganicznej** (Institute of Bioorganic Chemistry). Type “Instytut C” to get hint
    4. **Strona internetowa** (web page) - www.man.poznan.pl
    5. **Nazwa Afiliacji** (name of the affiliation) - PSNC
    6. **Dziedzina Nauki** - Science domain
    7. **Kategoria** - Category - select “Nauki inzynieryjne i techniczne”
    8. **Podkategoria** - subcategory - choose something
    9. **Dzial** - Department - choose something
8. **“Zaufany agent“** - trusted agent (leave **“Nie”** - No)
9. **Data Weryfikacji Afiliacji** - Affiliation Expiration Data - leave the date proposed by the system. You will be able to prolong it later
10. Press **“Wyslij Afiliacje do weryfikacji”** (Send affiliation for verification)

Please send a message to piontek@man.poznan.pl and wait until your affiliation will be confirmed by me. (you should be informed by the mail)

Applying for services

1. Login again to the https://portal.plgrid.pl on the left side click **“Uslugi”** (Services)
2. Press the green button - **Zarzadzaj Usługami** (Manage Services). You will be redirected to the Applications and Services Catalogue service.
3. Use the **“Szukaj”** (Search) Edit to search for **“Globalny Dostęp QosCosGrid”** (Global Access to QCG)
4. Apply for QCG by pressing the **“Aplikuj”** (Apply for) button
5. **IMPORTANT:** To have local access to PSNC clusters (optional) please apply also for **“Lokalny dostęp do klastrów PCSS”**.

Requesting for the certificate (needed to use QCG)

1. go back to the portal.plgrid.pl and click **“Certyfikaty”** (Certificates) link.
2. SimpleCA - press **“Wygeneruj certificate”** - Generate Certificate
3. **Podaj haslo do konta** - provide account's password and press **“Generuj”** (Generate)
4. **Zapisz certyfikat na dysku** - save on disk

Requesting for the testing grant

1. in the portal.plgrid.pl on the left side please click “Granty”. (You will be redirected to the Bazaar sytem).
2. in the Bazaar please select “Granty testowe” (on the left panel, just below “Granty”)
3. press green button “Włącz grant testowy” (apply for testing grant).
4. to make the testing grant the default one please select “Granty”. On the list of grants please click the one which name is “plg<USER><current year><a or b>. On the grant page (top/right corner) click “Ustaw grant jako domyślny” (set grant as default).

Make the compatpsnc grant the default one

1. in the portal.plgrid.pl on the left side please select Granty→Moje Granty (Grants→My grants)
2. find the compatpsnc grant on the list and select it
3. on the grant page in the top-right corner click “Ustaw grant jako domyslny” button (Set as default).

ANOTHER ISSUES

Re-creation of the certificate

Starting from 2017 web browsers will not accept SHA-1 certificates (old type of certificates). Because SimpleCA certificates were SHA-1 all the users (who created their certificates before 1 August 2016) are requested to re-create certificates following the procedure:

1. go the the PLG portal (portal.plgrid.pl)
2. on the left side select “certyfikaty” - certificates
3. in the “SimpeCA” section press “Usuń” (remove) button
4. in the same section press “Wygeneruj certyfikat” (Generate certificat) button.
5. provide password to your plg account (“Podaj haslo do konta) and press Generuj (generate).
6. store your certificate and upload to browser if you want.

IMPORTANT: The key of the certificate will be encoded with the same password as you have to the PLGrid portal. Previously it was possible to have various passwords for portal and certificate.

PROBLEMS

PLEASE REPORT ALL PROBLEMS TO qcg@plgrid.pl

Do not blame us for portal stability and functionality. It was written by Cyfronet team from Cracow.

STFC Instructions

In order access Hartree Centre systems you must first request access to our service administration system 'SAFE'; you can do this by clicking on this link:

https://um.hartree.stfc.ac.uk/hartree/

You then need to associate your account to the COMPAT project “HCEEC004-COMPAT” with PI Neil Morgan. To this purpose, please ask Neil Morgan for a password:

Neil.Morgan@stfc.ac.uk

Details of our machines and all associated user guides can be found here:

http://community.hartree.stfc.ac.uk/wiki/site/admin/home.html#guide

Any issues should be raised via a submission of a ticket to the Hartree Helpdesk via this email address:

hartree@stfc.ac.uk

EEE Modules and Conventions

The following instructions currently only apply to SuperMUC, however all sites should eventually stick to similar conventions.

Getting Started

Add the ComPat modules directory to the MODULEPATH environment variable by appending the following line to your `~/.bash_profile`:

```
export MODULEPATH=$MODULEPATH:/gpfs/work/pr92ge/di25pul/Modules
```

Then run `module load compat`.

Environment Variables

The command below will take you to the shared directory of the ComPat project.

```
cd $COMPAT_SHARED
```

The `$COMPAT_SHARED` directory is structured as follows:

```
$COMPAT_SHARED/
  /Astro
  /Bio
  /Common
  /Fusion
  /Modules
```

Available Modules

Use `module load name` to load the module by name and `module help name` to see information about that module. Use `module unload name` to unload the module and undo the changes to the environment.

To see all ComPat modules: `module avail compat`

Module	Description
compat	Will load the standard ComPat environment.
compat/common/ruby	Ruby
compat/common/muscle2	MUSCLE2
compat/common/amuse	AMUSE
compat/common/namd	NAMD

Performance Indicators

You can see an overview for the percentage of the time each service was functioning properly during a given time period by going to the link below. Modify the link to reflect the dates you want.

https://nagios-compat.drg.lrz.de:5000/performance/01.07.2016/01.10.2016

Software Requests

Currently, to request software to install on sites that are part of the EEE you have to fill in the form given below. You will need to authenticate using your Grid certificate.

Software Request Form

Maintenance Database API

The maintenance database query and update service resides on port 5000 of the nagios-compat.drg.lrz.de server. Authentication is done using Grid certificates. Your DN has to be manually added to the grid-mapfile on the server. To test authentication you can execute the following:

```
curl --cert ~/.globus/certificate:password
https://nagios-compat.drg.lrz.de:5000/test
```

It will return either {“authentication” : “succeeded”} or {“authentication” : “failed”} with HTTP codes 200 and 401 respectively, depending whether you provide a valid certificate.

API Documentation

URL	Method	Description
/downtime	GET	Will produce a complete list of current downtimes.
/downtime	POST	Will add a new scheduled downtime to the database. Return it's id.
/downtime/<id>	DELETE	Delete the record specified by <id>.
/upcoming	GET	Will return all upcoming downtimes.
/current_and_upcoming	GET	Will return current and upcoming downtimes.

Data format

```
[
    {
        "hostname" : string,
        "start_date" : timestamp,
        "end_date" : timestamp,
        "description" : string
    },
    ...
]
```

Examples

```
$ curl --cert certificate:password --data
"start_date=1234&end_date=4321&description=abcd"
https://nagios-compat.drg.lrz.de:5000/downtime/qcg.inula.man.poznan.pl
```

ComPat User DNs

```
C=PL/O=GRID/O=PSNC/CN=Tomasz Piontek
C=PL/O=GRID/O=PSNC/CN=Piotr Kopta
C=DE/O=GridGermany/OU=Leibniz-Rechenzentrum/OU=VER/CN=Stephan Hachinger
C=DE/O=GridGermany/OU=Max-Planck-Gesellschaft/OU=Rechenzentrum
Garching/CN=David Coster
C=DE/O=GridGermany/OU=Max-Planck-Gesellschaft/OU=Rechenzentrum
Garching/CN=Olivier Hoenen
C=UK/O=eScience/OU=UCL/L=EISD/CN=ulf schiller
C=DE/O=GridGermany/OU=Max-Planck-Gesellschaft/OU=Rechenzentrum
Garching/CN=Alberto Bottino
C=DE/O=GridGermany/OU=Max-Planck-Gesellschaft/OU=Rechenzentrum
Garching/CN=Bruce Scott
DC=org/DC=terena/DC=tcs/C=NL/O=Universiteit van Amsterdam/CN=S.A. Alowayyed
saloway1@uva.nl
C=UK/O=eScience/OU=Warwick/L=UOW/CN=keeran brabazon
C=UK/O=eScience/OU=Warwick/L=UOW/CN=dirk schubert
C=DE/O=GridGermany/OU=Max-Planck-Gesellschaft/OU=Rechenzentrum
Garching/CN=Onnie Luk
C=DE/O=GridGermany/OU=Leibniz-Rechenzentrum/CN=Vytautas Jancauskas
C=UK/O=eScience/OU=UCL/L=EISD/CN=james suter
C=UK/O=eScience/OU=UCL/L=EISD/CN=david wright
C=UK/O=eScience/OU=UCL/L=EISD/CN=robin richardson
C=UK/O=eScience/OU=Warwick/L=UOW/CN=oliver perks
C=US/O=National Center for Supercomputing Applications/CN=Andre Merzky
```