



CONTRACT NUMBER 508830

DEISA
**DISTRIBUTED EUROPEAN INFRASTRUCTURE FOR
SUPERCOMPUTING APPLICATIONS**

European Community Sixth Framework Programme
RESEARCH INFRASTRUCTURES
Integrated Infrastructure Initiative

FIRST SA4 ANNUAL REPORT

Deliverable ID: DEISA-DSA4-3
Due date : April, 30, 2005
Actual delivery date: May, 15, 2005
Lead contractor for this deliverable: IDRIS-CNRS, France

Project start date : May 1st, 2004
Duration: 5 years

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
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	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Table of Content

Table of Content.....	1
1. Executive Summary.....	2
2. Introduction	3
3. Main principles of organisation of User Support Service.....	4
4. Common Production Environment.....	5
4.1 CPE software stack.....	5
4.2 Interface to access the software.....	6
5. Software monitoring for the Common Production Environment.....	7
6. Preliminary User Documentation.....	10
7. Support of scientific Joint Research Activities.....	11
8. Miscellaneous	12
9. Future activities	13
10. References and Applicable Documents	14
11. List of Acronyms and Abbreviations	14

1. Executive Summary

The main objective of this service activity is to deploy all the actions required to allow scientific users to adopt to and use the DEISA supercomputing infrastructure. This is mainly done by providing and maintaining a *Common Production Environment* on all the platforms of the infrastructure, and providing also documentation on its usage, training sessions to help the porting and optimisation of the applications and a decentralised *Help Desk* service.

This document gives a detailed overview of all the tasks achieved or currently undertaken in the *User Support and Applications Service Activity (SA4)*, and also a brief overview of the activities planned for the next year. This constitutes the report of the first annual activities.

This document is publicly available.

2. Introduction

During this first year, the *Applications and User Support Service* Activity mainly concentrated its efforts on the following tasks:

- ? the study and definition of the organisation of the *User Support Service*, according to the choices of the *DEISA Operational Model*,
- ? the definition and implementation of a *Common Production Environment*,
- ? the selection, experiment, adaptation and deployment of an accompanying framework to monitor this *Common Production Environment*,
- ? the writing of the preliminary DEISA user documentation,
- ? the help provided to the users of the scientific *Joint Research Activities*.

3. Main principles of organisation of User Support Service

After discussions between the *Service Activities* during the first months of the project, the DEISA *Operational Model* has been defined and agreed upon during the summer of 2004. An important point which explains some choices made to organise the *User Support Service* is that each user has a unique entry point to the global infrastructure, called their *Reference Site* (or *Home Site*), which is the one on which they usually work, independently of the DEISA project.

A direct consequence is that each user will have an interactive access only to this Home site, not to the others, and that they will address help requests to their local user support service only, even if their jobs were executed on other sites. It will be the responsibility of these services to work together in a global and strongly coordinated way, to handle the questions and solve the problems which concern other sites. This is why the user support service of each site will be in direct contact with the services of all other sites, and will have full access to all the computers of the global infrastructure.

According to this general *Operational Model*, the questions relative to the access and usage of the various software products needed by the users to run their applications, and the ways for them to get information and help, were also discussed and agreed last summer. These choices were described in the D-SA4-1 deliverable (*SA4 Service Definition and Operation*) [1].

We summarise here the main points:

- ? The *Common Production Environment* is the common set of software available on each platform. It must have a high level of coherence between homogeneous platforms, and a lower level between heterogeneous platforms. Such a CPE is obviously necessary to give a logical coherence of the infrastructure from the software point of view, and to allow transparent job migration inside subgroups of homogeneous computers.
- ? A common user interface has been defined and set up to access the software of the CPE, based on the *Modules* tool.
- ? A framework to test the availability and the coherence of the CPE has been defined and set up, based on the *INCA* tool.
- ? A common set of documentation must be available, with some general documents and some other specialised manuals. Preliminary general documentation (*Primer* and *FAQ*) has already been written and has been available for a few months.
- ? The organisation of the DEISA *User Support Service* has been decided, according to the *Operational Model*, as previously stated. In order to collect information and problems reported by the users, a centralised *Trouble Ticket System* will be used.

4. Common Production Environment

The definition of a unified *Common Production Environment* (CPE) for the software is a major requirement in a distributed infrastructure like DEISA. It must have a very high level of coherence inside each subgroup of homogeneous computers and a lower level across the other subgroups. If some software, such as public domain ones or commercial but portable ones (for instance third party numerical libraries and parallel debuggers) can be the same on heterogeneous computers, other software such as compilers are usually specific to one computer or at least one manufacturer.

The necessity for such a CPE, provided and synchronously maintained by all the partners, is also the direct consequence of our choice not to use a centralised general repository for all the software, even for each subgroup of homogeneous computers. It would have been technically possible, using the global file systems available in each super-cluster, but this option was discarded because it would have been a major single point of failure for the all components of a super-cluster.

Some general rules were also defined between the partners to manage the evolutions of the CPE of a super-cluster, because it is obviously required to maintain and guarantee its coherence over time, with the introduction of new versions of existing software, as well as of new software.

The CPE can be characterised by three main components:

- ? a coherent set of software packages,
- ? a uniform interface to access the software,
- ? a framework to monitor the software (this will be described in section 5).

4.1 CPE software stack

The CPE has been divided into six categories:

- ? the *environment*, which does not include any software, but defines the environment needed by a DEISA user (especially the so-called DEISA environment, which defines how to access the CPE itself, the different parts of the global file systems, etc.),
- ? the *shells* (it has been decided to support only Bash and Tcsh),
- ? the *compilers* (it has been decided to support up to now the Fortran, C, C++ and Java ones),
- ? the *libraries*, especially the communication, numerical and graphical ones, which can be relative to one kind of platform only (as for the numerical library offered generally by each manufacturer), or public domain ones available on most platforms, or third party commercial ones, but nevertheless compatible with all major systems,
- ? the *tools*, which can also belong to one of the three preceding species (specific to a manufacturer, public domain or commercial ones), to help the users with various tasks (editing of files, development of scripts, parallel debugging, etc.),

- ? the *applications*, which are executable codes or whole packages available to run simulations in some specialised fields (in material sciences, bio-informatics, computational chemistry, etc.), and which can be also either commercial or in the public domain.

The preliminary set of software components, which constitutes the initial CPE for the core sites, has been agreed between the partners, based on the requirements of the applications of the users involved in the scientific JRAs.

4.2 Interface to access the software

A common interface must be offered to the users to access the software of the CPE. This is useful because, from the user point of view, it simplifies both the access and use of the software, by providing a uniform method. However, it is also a critical requirement of job migration between homogeneous computers. This is because, due to local system administration policies, it is highly likely that software will be installed in different places on the different systems. This is especially true for software that is not part of the standard system installation.

A survey was made during the first months of the project, comparing the two main public domain tools available, *Modules* [7] and *SoftEnv* [9]. The first one was chosen, mostly because, in addition to fulfilling our needs (and in any case having capabilities rather close to those of *SoftEnv*), it was already used by most partners, and therefore already familiar to a large fraction of our users.

Each component of the CPE is accessible by using a dedicated interface based on *Modules* and called a *modulefile*. As previously stated, for each piece of software available to be run on a platform, a corresponding *modulefile* must be present, which will be internally different to hide the specific characteristics of the installation of this software on each platform.

During the development of these *modulefiles*, special care has been taken to make the *Modules* commands at least analogous, if not identical, between heterogeneous computers. For this reason, the preliminary subsets of these *modulefiles* have been developed in parallel on IBM AIX and NEC Super-UX systems, and later tested on a Linux environment.

The usage of this framework has been defined to be as simple and straightforward as possible for the users, but with the acceptance of some additional complexity in its development. The robustness of the system has also been emphasized, in order to prevent users from defining incompatible choices that would later create unexpected problems that may be difficult to diagnose.

IDRIS is responsible for developing these *modulefiles* and for delivering them to all other partners. For those having the same kind of computer, the adaptations are rather minor and mostly consist in changing various pathnames, depending on the local conventions used to install the software. For those having other kinds of computers, there is more work to do, as some new *modulefiles* must be added for the specific software (compilers, libraries, tools) available only on these systems.

5. Software monitoring for the Common Production Environment

In a distributed hardware, but also software, infrastructure like DEISA, it is an absolutely necessity that the administrators, the user support services, and even in some aspects the users themselves, have an updated view and a detailed status of the software environment.

The monitoring of such a distributed software environment has become a requirement in all important grid projects, even if the necessities are rather high and sometimes difficult to handle, especially in a heterogeneous context. The development of such a general framework is a huge task by itself.

The main features required in such tools are the ability:

- ? To verify the accessibility of the compilers, libraries, tools and applications (including version numbers) installed on the different computers and also, when possible, their current behaviour. Under this aspect, it is important that some basic tests, repeated at specified intervals of time, guarantee not only that a specific piece of software is installed and currently accessible, but also that its main functionalities are working correctly.
- ? To verify the status of the installations after software upgrades. Often, large pieces of software (mainly libraries and applications) include their own system of tests, but some changes in a software product may badly impact other installed software, so it is interesting to integrate more specific and cross functional tests.
- ? To offer both to the centre's staff and, with fewer details, to the users, an updated view of what is installed and available on the various computers of the distributed infrastructure. Both the administrators and the persons in charge of the operation must be alerted when some DEISA software component is not available or not working correctly. Also, user support teams will obviously find this information very helpful to diagnose the various problems reported, and even more helpful if it concerns platform other than their own.

A survey of the available public domain monitoring tools and an evaluation of their capabilities has been done during the summer of 2004. In fact, very few tools currently exist. Most of them have very serious drawbacks that prevent us using them: very limited capabilities; not a highly modular architecture with not enough flexibility to allow us to adapt it easily; or a deep integration inside Globus [5], etc. The Nagios tool [8] has a general, powerful and flexible architecture that we were looking for, but is mainly oriented to alarm users about problems on some sensible software or hardware components, rather than to give a detailed overview of a software stack infrastructure, which is what we need.

This is why we have chosen the INCA tool (*Test Harness and Reporting Framework*) [6], developed by the San Diego Supercomputer Center and the Argonne National Laboratory for the TeraGrid project [10]. Even if it was developed for the needs of this project, it was developed from the beginning with the requirement in mind of not limiting it to the specific requirements of TeraGrid, but allowing it to be easily adapted to other grid infrastructures.

INCA is specifically designed to periodically run a collection of validation scripts, called *reporters*, with the purpose of collecting two different kinds of information:

- ? the version of the software installed (*version reporters*),
- ? the availability and the correct operation of this software (*unit reporters*).

The collected information is cached on the INCA server, and can be archived to produce a historical representation of the status of the resources of a grid.

The architecture of INCA is composed of three levels:

1. a centralized *Server*, which collects, manages and publishes all the data produced by the reporters, and which itself includes two major components:
 - ? a centralized controller, named *collector*, which collects data from distributed controllers and forwards them to the depot,
 - ? a *depot* for data caching and archiving, implemented as a Web service.
2. *clients*, installed on each resource to be validated, responsible for gathering data from the resource on which they are installed, and which includes itself two components:
 - ? a *reporter suite*, which is a collection of tests to be performed, that is to say a collection of reporters along with their scheduling policies (e.g. the frequency of their execution),
 - ? a *distributed controller*, which executes reporters according to their scheduling policies, and forwards data to the collector.
3. *data consumers* (such as the Web-based one), which are generic tools able to get data from the depot Web service and to display them in a user-friendly style.

CINECA, in charge of this task, has installed the INCA tool; adapted it in a test case environment to the DEISA framework, specifically for the components of our CPE, for both the software itself and for the interface provided by the *Modules* tool (as INCA originally used the *SoftEnv* tool); tested and evaluated it. The results were very positive. Both our defined needs and the level of flexibility required to allow us to adapt the tool without too much effort were satisfied. After this successful testbed period, a technical workshop has been organised in spring 2005 to share the knowledge and experiences of the leading partner of this task, and to allow others to easily install, configure and deploy this tool in their sites.

As for the *modulefiles* for the Common Production Environment described in the previous section, one partner, here CINECA, is responsible for the development of all the configuration files required to set up INCA; both to define the software components that we want to monitor and to describe the way in which we want the tests to be performed and reported. The other partners are in charge of adapting these various files to their particular environment.

This framework to monitor all the main components of the distributed software infrastructure is obviously the place to supply the DEISA *Information System*, managed and set-up among the SA3 *Resource Management* activities, with information concerning the software infrastructure. This integration method is currently being discussed and will be implemented soon.

A last point to emphasise on this topic is that this was the opportunity to start a fruitful technical collaboration with the INCA development team, and in this way with a TeraGrid group. The version that we used this winter for our preliminary experiments had some portability problems, some pitfalls difficult to handle at first glance, and also suffered from some deficiencies in its installation documentation. After problem reports and various exchanges with the developers, these problems were solved, and we saw with satisfaction during the spring that in the next official version, great improvements have been made and that the installation is now almost straightforward, even in our context. This is the benefit of a good technical collaboration.

6. Preliminary User Documentation

The preliminary DEISA documentation was prepared at the beginning of the year, as expected for the D-SA4-2 deliverable [2], and was made available on the Web server at this time. It is up to now composed of the first version of the *Primer* (in both paper and Web versions) [4] and of a preliminary version of the general DEISA *FAQ* (in Web version only) [3].

The *Primer* is divided in several chapters which try to cover all the important fields where users need information to be able to access and conveniently use the DEISA infrastructure:

- ? the general overview of the project, and of the sites, resources and services available,
- ? the way to access the infrastructure (projects, accounts, certificates, access procedures, passwords),
- ? the file system organisation, usage and services,
- ? the *Common Production Environment* definition, components and usage,
- ? the job submission procedures, using either the batch subsystems, the UNICORE interface [11] or (later) portals,
- ? the access to documentation and user support.

As previously mentioned, a preliminary version of the general DEISA *FAQ* is also available, which is mainly another way to find the answers to some very common questions faster than looking inside the full version of the *Primer*.

First versions of these documents were prepared last January to be in time for the required date of the D-SA4-2 deliverable, but it is obvious that they will continuously evolve, along with the forthcoming more specialised documentation, to reflect the evolution and improvement of the environment. A general update of both the *Primer* and the *FAQ* is planned in early summer 2005, taking into account some (minor) changes, already done or planned to take place soon, in the software environment already installed and configured.

7. Support of scientific Joint Research Activities

During this first year, the complete infrastructure has not necessarily been accessible to users to run their applications. However, all the scientific *Joint Research Activities* (JRA 1-6) have nevertheless been given access to all the core sites to prepare their work for the moment when the infrastructure would be available for them to run their applications in production mode.

For some JRAs, the users themselves have done this work, with the help of their local user support service, and for other JRAs some engineers of the computing centres have done this work for the users.

All the projects included in these JRAs have had a direct and strong connection with one of the core partners (RZG for JRA 1 and 3, IDRIS for JRA 4 and 6, CINECA for JRA 5), except for JRA 2 which has this connection with EPCC, but in fact has also carried out work and experiments on the systems of two of the four core sites.

For each of these JRAs, the main task was to port and test the codes (sometimes there was only one large code and other times whole packages with various codes and associated utilities) on all the computers of the other core sites, or at least on some of them. In some cases, the application related *modulefile* has been developed. Additionally, for third party software, a licence server has been deployed when necessary (this was needed by JRA 5).

The *User Support Service* of all these sites gave, of course, the expected support to these users, but also had exchanges and co-ordination with the services of the other sites as required. Some sites chose to ask to these users to work on their DEISA test platforms, and some others on their production platforms, according to various local issues.

The assessment of all these experiments can be briefly summarised, emphasising that only three kinds of real problems were diagnosed:

- ? One of the main problems that the users had with some sites was to fulfil the local administrative procedures required to open an account on these sites. For some partners, these procedures were heavy, asking for instance for the signatures of several persons and consequently leading to some delays longer than expected. These sites later tried to simplify their procedures for the special case of the DEISA JRA users, but in any case these problems were specific to the intermediate phase, when it was not possible to use the DEISA administrative procedures to open accounts.
- ? The second important problem that some users had was related to certain security issues and to some special connectivity needs (to run daemons, to ask for licence servers, to use specific network ports, etc.), which were not provided by default. All these problems were relatively easy to solve, after agreements between the *User Support Services* and the *Network and Security Services* of the partners.
- ? Some users also had special requirements about certain software, or specific software versions, and in some cases extra installations of software components were required.

8. Miscellaneous

A few other activities should briefly be mentioned here:

- ? One of our first tasks was, as stated in our initial work plan, to make an audit at the beginning of summer 2004 about the actual organisation of the *User Support Services* of the partners, and about their assets and domains of competency, to give us a general idea of the current situation and of the possible ways to benefit from the existing various strengths. A summary was given in the D-SA4-1 deliverable [1].
- ? RZG has made some performance tests on real applications (using a code from JRA 3) to check both the real usage of the network connecting the core sites and the performance of the multi-site GPFS between IBM AIX machines. These tests were initially delayed, due to some problems in the early versions of MS-GPFS, some of which were rather long to solve. However, after this was completed, satisfactory results for input/output measurements were achieved, with up to 400 Mbit/s for write operations and 842 Mbit/s for read operations (the limitation comes from the 1Gbit/s network, since notably higher throughputs were measured locally).
- ? As previously mentioned, a centralised *Trouble Ticket System* is a real need in a decentralised infrastructure such as DEISA. Most of the partners already use locally such a system for their own needs, but a common and centralised one is required for DEISA. An analysis of the tools actually used by the partners was done by FZJ during the spring of 2005, and the set-up of the DEISA configuration of the tool selected is planned for the early summer. The main goals for this system are to:
 - ? give to all sites a global view of the situation of the virtual platform by looking at all the problems opened (it will help each *User Support Service* to solve the problems reported to them that concern in fact the computers of other sites, having access to the information on problems reported to other sites),
 - ? allow to keep the whole history of the situations and problems reported,
 - ? allow to generate reports and statistics, to have a general view of the *User Support* activity and of its historical evolution.

9. Future activities

As it has been clearly described in previous sections, the main efforts of this *Service Activity* this year were devoted to the preparation and building of the software layers required and in the preliminary tasks (definition of the *Service Operation*, first documentation, etc.) to be able to offer later a high level of support to the users of the infrastructure.

However, the direction of our efforts will change during the next year. Obviously, all the main tasks described before (*Common Production Environment*, monitoring of this CPE, basic documentation) will remain major priorities and will continue to be maintained, developed and improved. However, most of the work in these areas is already complete, and our main efforts will now be concentrated in others tasks:

- ? The special support for the selected projects of the *DEISA Extreme Computing Initiative* (DECI), which will be constituted by some of the most important projects of the European scientific community, and so of major importance for the impact of the whole DEISA project on computational sciences.
- ? The organisation of training sessions and workshops on *High Performance Computing* topics (especially on parallelisation, optimisation, grid computing, application coupling).
- ? The special care to give to applications with a high level of parallelism (using a huge number of processors).
- ? The offering of a specialised local support, for a limited period of time, to help some teams to port and optimise their applications to the DEISA infrastructure.
- ? The integration of computational monitoring and steering services, potentially developed between partners, in some applications.
- ? The integration of real-time visualisation techniques in some applications, if such needs are expressed by some projects.
- ? The building, test, documentation and release of the *DEISA Cluster Resource Management Package* (DCRMP), in collaboration with the SA3 *Service Activity* which leads this task. The DCRMP will include all the public domain middleware packages used inside DEISA, the aim being to facilitate their installation and usage both inside DEISA and potentially elsewhere.

10. References and Applicable Documents

- [1] DEISA D-SA4-1 deliverable: *SA4 Service Definition and Operation*
- [2] DEISA D-SA4-2 deliverable: *Basic DEISA Infrastructure Documentation*
- [3] DEISA FAQ: <http://www.deisa.org/userscorner/faq.php>
- [4] DEISA Primer Documentation: <http://www.deisa.org/userscorner/primer.php>
- [5] Globus: <http://www.globus.org/>
- [6] INCA (Test Harness and Reporting Framework): <http://inca.sdsc.edu/>
- [7] Modules: <http://modules.sourceforge.net/>
- [8] Nagios: <http://nagios.org/>
- [9] SoftEnv: <http://www-unix.mcs.anl.gov/systems/software/msys>
- [10] TeraGrid: <http://www.teragrid.org/>
- [11] UNICORE (UNiform Interface to COmputing REsources):
<http://unicore.sourceforge.net/>

11. List of Acronyms and Abbreviations

CPE	Common Production Environment
DCRMP	DEISA Cluster Resource Management Package
DECI	DEISA Extreme Computing Initiative
Home site	The site where a user usually works, logs in and submits jobs
IS	Information System
JRA	Joint Research Activity
Reference site	Same as <i>Home site</i>
TTS	Trouble Ticket System