



CONTRACT NUMBER 508830

DEISA
**DISTRIBUTED EUROPEAN INFRASTRUCTURE FOR
SUPERCOMPUTING APPLICATIONS**

European Community Sixth Framework Programme
RESEARCH INFRASTRUCTURES
Integrated Infrastructure Initiative

SA4 Second Annual Activity Report

Deliverable ID: DEISA-DSA4-4
Due date: April, 30, 2006
Actual delivery date: May 17, 2006
Lead contractor for this deliverable: IDRIS-CNRS, France

Project start date: May 1st, 2004
Duration: 4 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 Services)	
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. Common Production Environment.....	4
3.1 CPE software stack.....	4
3.2 Interface to access the software.....	4
4. Software monitoring for the Common Production Environment.....	6
5. User Documentation.....	8
6. Support of scientific Joint Research Activities.....	9
6.1 JRA 1 (Material Sciences).....	9
6.2 JRA 2 (Cosmology Applications).....	9
6.3 JRA 3 (Fusion Research).....	9
6.4 JRA 4 (Life Sciences).....	10
6.5 JRA 5 (Industrial CFD).....	10
6.6 JRA 6 (Coupled Applications).....	10
6.7 DECI projects.....	11
7. Other activities.....	12
7.1 Training sessions.....	12
7.2 Trouble Ticket System.....	12
7.3 DEISA Cluster Resource Management Package.....	12
7.4 DEISA Applications Test Suite.....	13
8. References and Applicable Documents.....	14
9. List of Acronyms and Abbreviations.....	15

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 also providing documentation on its usage, specialised support to help the porting and optimisation of 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)* during the reporting period (PM 19 to PM 24). This constitutes the report of the second annual activities, in complement to the D-SA4-3.2 deliverable which covered the first six months.

This document is publicly available.

2. Introduction

During the last six months, the *Applications and User Support Service* Activity has mainly concentrated its efforts on the following tasks:

- Maintenance of the *Common Production Environment* (CPE). This has included the update of the AIX CPE on the five sites which had already deployed it, according to the evolutions of the software; the installation and customisation of the CPE on the two remaining AIX sites of the seven which had not yet installed it; and, for the main effort, the installation, adaptation and customisation of the CPE on the non-AIX sites.
- The deployment of the framework to monitor this *Common Production Environment* on all the sites (only the core sites were running it before the reported period).
- Updates of the DEISA user documentation, according to both changes made in the user interface and to feedback received.
- Provision of help to the users of the scientific *Joint Research Activities*, and, in conjunction with the *Applications Task Force*, to the users of the first DECI projects.

3. Common Production Environment

The unified *Common Production Environment* (CPE) is a major feature in DEISA, as in all distributed infrastructures of this kind, which defines a coherent set of software accessible on the various sites of the infrastructure. It offers both a common interface to the users, independent of the target platform really used, and the ability to migrate jobs between different supercomputers of the same architecture. Of course, the level of coherence is not the same everywhere in the infrastructure, ranging from very high inside each subgroup of homogeneous computers to a lower level across the other subgroups.

Three main components define the CPE:

- a coherent set of software packages divided into six categories: environment (which does not include any software, but defines the environment needed by a DEISA user), shells, compilers, libraries, tools and applications;
- a uniform interface to access the software;
- a framework to monitor the software (this will be described in section 4).

3.1 CPE software stack

As reported in the deliverable D-SA4-3.2 [6], this initial CPE has already been installed on the five AIX core sites which first joined the infrastructure (CINECA, CSC, FZJ, IDRIS, and RZG). It was used in full production mode on the first four sites since last spring, and since last summer also at CSC which joined the AIX super-cluster later.

During the last six months, the two remaining AIX partners (ECMWF and EPCC) also installed the software packages included in the AIX CPE which were not already available on their computers. The AIX environment is now complete for the seven related partners. In addition, the major step of the integration of the non-AIX sites was mostly achieved during the reported period. The two sites LRZ and SARA, which will both operate an SGI Altix system (SARA have already run one for several years but LRZ will install its own system only this spring) have defined by mutual agreement the common software basis to include in their own CPE. (Many are common with the AIX set, but some are obviously specific to their systems, in particular all the compilers and the scientific libraries provided by the manufacturer). On its side, BSC has also defined its own software stack for their machine.

As initially planned in the two layer infrastructure of DEISA, there is, as required, a very strong homogeneity of the software stack inside each common set of platforms (IBM AIX and SGI Altix), with only minor discrepancies on the patch level of the software included, and a weaker one across the different set of platforms. Nevertheless, there is a general coherence and many common items.

3.2 Interface to access the software

Having the required software installed and making them available to the users is not enough. The local system administration policies define different places to install the software, especially those not included in the standard system installation, and a major requirement of DEISA is that a common interface hides these differences. Additionally, this is also a critical requirement for job migration between homogeneous computers.

As reported earlier [3, 5, 6], the *Modules* [17] public domain tool was chosen at the beginning of the project to fulfil such a requirement, allowing the provision of a dedicated portable interface to each component of the CPE, based on the so-called *modulefiles*. As explained before ([5, 6]), various subsets of these *modulefiles* for the initial CPE on AIX systems have been developed and distributed by IDRIS, who were in charge of this task, then evaluated and tested by the partners, followed by the release of the complete set of the *modulefiles* to interface the full initial AIX software stack. Nevertheless, an important work has been achieved last summer in this area (see [6]), to take into account the various feedback received from the partners and to comply with some of their new expectations, which have lead to important internal rewriting. The major reason for all these changes and efforts was to improve greatly both the generality and portability of the DEISA Modules environment.

As discussed in detail during an internal meeting in October 2005, this work allowed us to base our framework on what we estimate, with all our experience, to be the best technical choices for the future and the best basis from which to extend the environment from the five initial partners to all the rest. In consequence, according to our final design, the two remaining AIX sites (ECMWF and EPCC) have also installed the Modules environment on their systems during these last months, with the required local adaptations, and now the AIX CPE environment is complete for the seven related partners.

In parallel, the work to port the Modules environment to the non-AIX platforms has been mostly completed. The SGI Altix sites have ported this environment to their system and customized it themselves for the software specific to their platform. This is now in production mode at SARA. On their side, LRZ has fully prepared its own environment on a test system and will be able to port it quickly as soon as their new computer has been installed. For their part, BSC has also adapted the current AIX environment to their own Linux system, interfacing their specific software and customizing the whole framework according to their needs. This is now in pre-production mode.

The CVS server which has been set up at IDRIS to distribute the configuration files, allowing each partner to find easily the changes made since their last update and to download each time only the modified files, has been heavily used during the whole of this period.

4. Software monitoring for the Common Production Environment

It is an absolutely necessity that each centre's staff have both an up to date overview and a detailed status of the software environment, especially in a distributed infrastructure like DEISA. The monitoring of such a distributed software environment has become a requirement in all important grid projects, even if the requirements are rather high and sometimes difficult to handle, especially in a heterogeneous context, meaning that the development of such a general framework is a huge task by itself.

The first feature required in such tools is the ability to verify the accessibility of the various software packages installed on the different computers, the expected version level of these packages and also, when possible, their current behaviour. It is also important that these tools allow the verification of the status after software upgrades and offer the administrators and the user support services an updated view of what is installed and available on the various computers of the distributed infrastructure, alerting them of possible deficient components.

As explained in the deliverable D-SA4-3 [5], the INCA tool (*Test Harness and Reporting Framework*) [16], developed by the San Diego Supercomputer Center and the Argonne National Laboratory for the TeraGrid project [20], has been chosen after an evaluation process done at CINECA. This framework was deployed on the first five AIX core sites starting last summer, to monitor the availability of the software of the AIX CPE with the required version levels. This deployment was led by LRZ, which took responsibility for the set up and operation of the centralized server (especially the management of the depot populated by the local clients), and, with the help of several partners, the development of the version reporters.

During the reported period, the work in this task was concentrated on three areas:

1. Preparation to the migration of the initial 0.10 version of INCA to the new version 2.0

The INCA development team has developed during the past year a completely new version of this tool, following both the requirements of the TeraGrid project and the feedback of the other projects which have tested or already deployed this framework, and especially from DEISA. As reported in the deliverable D-SA4-3.2 [6], an INCA SA4 meeting in July 2005 invited two of the main developers at SDSC to allow us to explain in detail all our experiments as well as our suggestions for the upcoming versions, and also to allow the developers to explain to us their future development roadmap.

This new version 2 was rewritten from scratch and has some major architecture changes:

- introduction of a new *Reporter Manager* to send the results of the *Reporters* to the *Depot*, which replaced the previous *Distributed Controller*,
- removal of the *Centralized Controller*,
- introduction of a new *Reporter Agent* to ease the deployment and maintenance of the *Reporters*,
- possibility to create some *Reporter Repositories*,
- introduction of a graphical user interface (*incat* tool) to allow the administrators to configure the INCA resources and suites,
- change of the database type required for the *Depot*, from a round robin to any SQL one,

- changes in the data display mechanism and techniques to define the *Data Consumers*, which display the results of the *Reporters*,
- improvement of the security model,
- change of the API for the reporters (as there is no backward compatibility, the reporters defined for the version 0.10 must be all adapted to this new API), which can be written in an alternative language to Perl, as long as their input conforms to the specifications.

A workshop was organized at SDSC in February 2006, in the same period as the availability of the first beta release of the new version, with the participation of two DEISA representatives.

All these major changes explain that the migration to the new INCA 2.0 version required major work, which we decided to split in several phases:

- a testbed between CINECA, IDRIS and LRZ, under the responsibility of LRZ, was set up at the end of March to experiment with the new version and allow the migration of our old version reporters and to test the new ones,
- outside the five core partners which already run the version 0.10 in production mode, several other partners started recently to operate the client part of the version 2 with, when required, the adapted version reporters for their CPE,
- the migration of these sites to the new version is planned during the summer,
- the integration of the initial unit reporters (see below) in the version 2 is also planned at the same time.

2. Integration in the framework of several sites other than the first five AIX sites

As stated above, several additional sites were included in the initial framework, so as not to delay their integration for too long a period of time. They will migrate to the new version in the same way as the other sites.

3. Development and integration of the first unit reporters

As explained above, the interest to monitor a distributed software environment is first to check the availability of the required components at the expected version levels, using what INCA call the *version reporters*, as was done initially for the first AIX core sites. However, another very important need is also to be able to check periodically the correct behaviour of these software components, for or at least the critical ones. For this, INCA provides a framework of *unit reporters*. During these last months, the tests defined in the initial release of the *DEISA Applications Test Suite* (see the paragraph 7.4) were used as the initial basis of unit reporters. They have been included in our monitoring framework and are now operated by most of the sites (still using the version 0.10 of INCA). It is planned to migrate these unit reporters to the new DEISA INCA version 2 framework during the summer.

5. User Documentation

The preliminary DEISA documentation, prepared at the beginning of 2005, as expected for the D-SA4-2 deliverable [4], was first fully revised at the end of the summer, with the release of the version 1.1 of the *Primer* [15] in October 2005. Then the version 1.2 was released in February 2006, mainly to update the description of the way for a user to get a certificate, to add some clarifications in the usage of the various file systems and in the submission of jobs (both for the usage of UNICORE and the use of the IBM batch manager for the AIX systems), and to update the description of the usage of the Modules environment for the minor changes introduced.

The next version 1.3, scheduled in May, will mainly change two aspects. Firstly, despite the improvements already made in version 1.2 to better describe the initial customization of the UNICORE environment, additional explanations and clarifications will be made. These will include a precise example of the complex procedure that users must follow, especially the initial configuration of the certificates required to use this tool. Secondly, the case of the so-called DEISA implicit users will be added, containing the particular information that this category of users must know. These users are not involved in the JRAs or DECI projects and remain “normal” local users, but will allow the jobs that they submit locally to be rerouted to another homogeneous platform of the DEISA infrastructure if the global operations management of the infrastructure consider it as an efficient procedure. For this job migration to take place, their jobs must fulfil some additional requirements, mainly to put the files used during the execution of their jobs in the global file systems defined across the DEISA sites and to use only software included in the DEISA CPE or accessible in their own private environment.

Each time that it was needed, the FAQ [14] was updated, to introduce all the changed informations and to keep it accurate according to the modifications made.

6. Support of scientific Joint Research Activities

All the projects included in these scientific 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 has in fact also now carried out work and experiments on the systems of many partners (BSC, HLRS, HPCx, IDRIS, RZG, SARA).

6.1 JRA 1 (*Material Sciences*)

During the reporting period, JRA1 has mainly focused on DEISA enabling of the Quickstep code from the CP2K package. At the new DEISA site HLRS, CPMD code has been supported for optimized usage on the vector architecture (see deliverable D-JRA1-4 [7]).

SA4 has started giving first level support for the materials science portal and continued to maintain the codes supported therein. Especially the corresponding *modulefiles* and the CPE have been maintained and newly deployed for CPMD at HLRS.

6.2 JRA 2 (*Cosmology Applications*)

During the last six months, JRA 2 has concentrated on work package WP3, *Grid enablement of FLASH through code migration* (see deliverable D-JRA2-3.1 [8]). During this period, WP3 has been porting and profiling FLASH, a Fortran 90 astrophysical code used by the Virgo Consortium, to several of the DEISA platforms, specifically three Power 4/4+ clusters: HPCx, IDRIS and RZG, the SGI at SARA, the IBM cluster at BSC (MareNostrum) and the NEC SX-8 at HLRS.

The SA4 team members of each of these sites (EPCC, IDRIS, RZG, SARA, BSC and HLRS) have all been instrumental in the success of JRA2 WP3 and have secured accounts on the machines in a timely manner for WP3 staff. Further, HLRS has also granted access to a non-DEISA staff member, namely Dan Sheeler, who is a member of the FLASH support team in Chicago, US, to help with porting issues.

At EPCC, SA4 have assisted JRA2 to install the UNICORE client and the ssh plugin on a local workstation, enabling access to the SGI system at SARA.

The SA4 teams at BSC, SARA and HLRS have also responded rapidly and helpfully to a number of porting issues which arose during the last 6 months, specifically errors encountered during compilation, linking and execution which required local knowledge of the machine under investigation (such assistance was not required at IDRIS nor RZG, due to the familiarity of the JRA 2 users with their platforms.)

6.3 JRA 3 (*Fusion Research*)

During the reporting period, the work of JRA 3 (see deliverable D-JRA3-4 [9]) was concentrated on the ORB5 and GENE codes, which are two complementary approaches to solve the gyrokinetic equations. SA4 will be involved as soon as the target architecture for the related forthcoming DECI project has been decided.

6.4 JRA 4 (Life Sciences)

During the last six months, JRA 4, following the users needs, maintained its efforts into making genomic codes optimized and available to the community, but it also pointed its focus towards a new direction: phylogeny (see deliverable D-JRA4-5 [10]).

On the first topic, the user support group at IDRIS is currently working with an INSERM team in Toulouse (France) whose human genetic project is to locate the genes accountable for the strong familial myopia, proceeding to a large scale analysis using an MPI parallelized code: Genehunter-TwoLocus. The major contribution of the IDRIS user support group to this project has been for the moment the porting of the code to the IBM AIX platform and the rationalisation of the MPI calls. Tests and speedup benchmarks have been conducted with satisfactory results, in fact excellent comparing to those obtained by the users on their own local machines. The next step will be to handle larger scale tests.

On the second topic, a first code has been submitted in the field of phylogeny. A user from a CNRS laboratory in Montpellier (France) wants to be able to conduct large experiments with MrBayes (Bayesian estimation of phylogeny) [18], a phylogenetic software which has a parallelized MPI version. At the moment, the IDRIS user support team achieved the porting of this code and running a first series of tests. JRA 4 intends to browse the phylogenetic domain in order to determine which other codes fulfil most of the users demands.

6.5 JRA 5 (Industrial CFD)

JRA 5 is involved in modelling for aero-acoustic (CAA) and aerodynamic (CFD) simulations for some relevant automotive applications. The software used is CFD++ from Metacomp, a commercial package. The support activities to this JRA during the last semester of the reported period are mainly related to (see deliverable D-JRA5-5 [12]):

1. Set up of the DEISA multi-cluster software environment at some relevant DEISA sites (CINECA and IDRIS) in order to support the CFD++ software.
2. Testing and tuning of a new CFD++ solver release at the CINECA site, in order to assess and resolve some scalability issues and bugs previously found (see deliverable D-JRA5-4 [11]).
3. Involvement of the software developers from Metacomp into the DEISA architecture in order to explore some licensing simplification mechanisms.
4. Submission of “medium” production cases for relevant simulations.
5. Analysis for the development of an off-the-shelf optimisation chain for aerodynamic simulation across the DEISA infrastructure.

6.6 JRA 6 (Coupled Applications)

During the last six months, JRA 6 mainly started new coupling projects: an aeroacoustic project (KOP3D), a Natural Convection / Radiation project and a new Combustion / Radiation project (FOCUS) in the DECI context (see deliverable D-JRA6-4 [13]).

The set-up of the first project was done with the help of the HLRS engineers involved in the DEISA user support activities and the set-up of the second project with the collaboration of the user support group at IDRIS. In particular, they dealt with:

- Porting and optimizing the KOP3D application, which was not initially dedicated to vector platforms, on the NEC vector machines at HLRS; a speed-up of 2 has been obtained after these first optimization steps.
- Migrating the coupled project Natural Convection / Radiation to the DEISA environment. The application has been numerically validated for several configurations. In addition, the layer based on the *Pthread* technology, used to manage asynchronous mechanisms, has been changed to use the OpenMP API which is much easier and convenient to implement. This tends to prove that the OpenMP API and the multi-threaded implementation of CORBA used can coexist without problems.

Concerning the FOCUS project, a particular attention has been paid to provide more flexibility to deploy and load-balance the 2 coupled codes. Furthermore, several code optimizations have been achieved (manually subroutine inlining, reducing memory accesses and I/O improvements) and a memory space minimization has been led. All these enhancements have been made in collaboration with the user support group at IDRIS.

It can be also noticed that, answering to a requirement of SA4, a small test to validate the CORBA implementation availability and correct behaviour on the different sites of the DEISA infrastructure has been developed. It has been integrated in the *DEISA Applications Tests Suite* (see the paragraph 7.4).

6.7 DECI projects

Under the responsibility of the *Applications Task Force*, the support of the DECI projects selected in the first call deeply involved members of the user support groups of the *Home* and *Execution Sites* used by these projects. In order to support the initial DECI projects, the first of which were launched in October 2005, the *Applications Task Force* has set up a personalised support constituted, for each of the selected projects, by an *Ataskf* member of the *Home site*, a member of another one of the *Execution sites* (in most of the sites they belong to the user support groups) and a member of the operations group. As for the JRAs, the work of the people of the user support teams was mainly to help these users to port and tune their codes when necessary, as well as to solve the various technical problems they encountered.

7. Other activities

7.1 Training sessions

Six training sessions will be organized between mid-2006 and early 2008 in different European countries (Finland, France, Italy, Germany and Spain), to enable the fast development of skills and knowledge needed for efficient utilisation of the DEISA infrastructure by the users. Each of these sessions will be divided between a general part, which will be repeated each time, to present a global description of the infrastructure and of the usage of its major software components, and a dedicated part on a specialized topic.

The first of these sessions, which will be held in Paris in July 3-6, has been prepared during the spring. The general part, extended to one day, will be devoted to the presentation of the organisation, infrastructure and major components of DEISA:

- DEISA project and organisation,
- DEISA infrastructure and middleware,
- DEISA Common Production Environment,
- UNICORE.

The dedicated part will be devoted this time to the *Highly scalable parallel applications*, itself divided in two parts:

- one technical, to give information about the methodologies, tools and optimisation techniques which are relevant for all kind of applications, to achieve optimal scaling on a large cluster of shared-memory nodes,
- and one based on *use cases*, to give pedagogical summaries of experiments in different scientific fields (presentations in climate modelling, cosmology, fluid dynamic and plasma physics are planned), in which the focus will obviously not be on the scientific results achieved, but on the problems encountered and the various techniques used to reach a high scalability of these applications.

7.2 Trouble Ticket System

As previously mentioned in the D-SA4-3.2 deliverable [6], a centralised *Trouble Ticket System* has been chosen in the summer of 2005 (*Request Tracker* [19]), according to the conclusion of a previous analysis of the tools actually used locally by the partners. It has been set-up and configured for the DEISA purposes at RZG, which operates this centralised service for the DEISA infrastructure. This is obviously a very important tool for the user support and operations teams, but this is now currently used too by all the service activities.

7.3 DEISA Cluster Resource Management Package

The *DEISA Cluster Resource Management Package* (DCRMP) will help the administrators of the DEISA sites in the installation and maintenance of their software environments, both for system and user oriented components. It will also allow the implementation of a powerful checking mechanism to guarantee the coherence of the installations (checking for dependencies between components, detection of

incoherencies between expected versions of packages, etc.) This activity, led by CINECA as an SA3 one, has already started and the first DCRMP distribution was released at PM 18, including UNICORE [21] and the *Resource Management Information System* (see deliverable D-SA3-3 [1]).

According to the work already achieved in SA3 to distribute some middleware components, and the potential interest of it for SA4 activities, it was decided to experiment it on a first limited set of application oriented packages. The objective is to be able to distribute in this way some public domain software related to applications (mainly tools and graphical and numerical libraries), to simplify and accelerate the installation process by providing distributions already packaged, which are easier to install for the supported dedicated platforms than by using the usual generic processes. It was decided to use two rather large and complex software packages of the CPE for this testbed: Python and the omniORB CORBA distribution. To handle the characteristics of this kind of software, which have some important differences from the ones used in SA3, it appeared necessary to extend the initial framework designed to build the first release of DCRMP. For this, a new tool was defined and built, called *DCRMP Package Creator* (DPC) (see deliverable D-SA3-5 [2]). The second version of DCRMP, including the two described components, was released at PM 24.

7.4 DEISA Applications Test Suite

As part of the various test suites developed inside DEISA to be able to check the behaviour of various components (global file systems, middleware, batch systems), an *Applications Test Suite* (ATS) has been developed in the SA4 activity to check the behaviour of some major software components (compilers, linkers, some scientific libraries, parallel libraries and execution framework).

The initial release of the DEISA ATS is composed of:

- a set of tests on the DEISA environment (Modules environment, environment variables for file systems, shell),
- a test to check the compilation of C and Fortran codes, and their linkage with an external scientific library,
- a test to check the compilation of a C++ code and of an IDL (interface definition language for CORBA) one, and their linkage with the CORBA libraries,
- a test of execution of an OpenMP code,
- two tests of execution of MPI codes,
- a test of execution of a CORBA client / server application,
- a test of execution of a packaged application (CPMD).

As already explained in section 4, most of these tests have been integrated inside the INCA framework as *unit reporters*, to enable their automatic launch and to control on a regular basis some major aspects of the software infrastructure.

8. References and Applicable Documents

- [1] DEISA D-SA3-3 deliverable: *First release of DCRMP*
- [2] DEISA D-SA3-5 deliverable: *Second release of DCRMP*
- [3] DEISA D-SA4-1 deliverable: *SA4 Service Definition and Operation*
- [4] DEISA D-SA4-2 deliverable: *Basic DEISA Infrastructure Documentation*
- [5] DEISA D-SA4-3 deliverable: *First SA4 Annual Report*
- [6] DEISA D-SA4-3.2 deliverable: *SA4 Mid-term Activity Report*
- [7] DEISA D-JRA1-4 deliverable: *Quickstep code from CP2K package for adequate usage in DEISA*
- [8] DEISA D-JRA2-3.1 deliverable: *Specification document for D-JRA2-3.2 (“FLASH on the DEISA infrastructure”)*
- [9] DEISA D-JRA3-4 deliverable: *Turbulence code GENE prepared for adequate usage in DEISA; final report on new ORB5 code for extreme computing*
- [10] DEISA D-JRA4-5 deliverable: *Report on production status of second set of applications*
- [11] DEISA D-JRA5-4 deliverable: *“Light” production cases results*
- [12] DEISA D-JRA5-5 deliverable: *“Medium” production cases results*
- [13] DEISA D-JRA6-4 deliverable: *Grid enabling of second set of projects*
- [14] DEISA FAQ: <http://www.deisa.org/userscorner/faq.php>
- [15] DEISA Primer Documentation: <http://www.deisa.org/userscorner/primer.php>
- [16] INCA (Test Harness and Reporting Framework): <http://inca.sdsc.edu/>
- [17] Modules: <http://modules.sourceforge.net/>
- [18] MrBayes (Bayesian Inference of Phylogeny): <http://mrbayes.csit.fsu.edu/>
- [19] Request Tracker: <http://www.bestpractical.com/rt>
- [20] TeraGrid: <http://www.teragrid.org/>
- [21] UNICORE (UNiform Interface to COmputing REsources): <http://unicore.sourceforge.net/>

9. List of Acronyms and Abbreviations

Ataskf	Applications Task Force
ATS	Applications Test Suite
CNRS	<i>Centre National de la Recherche Scientifique</i>
CORBA	Common Object Request Broker Architecture
CPE	Common Production Environment
DCRMP	DEISA Cluster Resource Management Package
DECI	DEISA Extreme Computing Initiative
DPC	DCRMP Package Creator
Execution site	The site where the jobs of a DECI user are executed
Home site	The site where a user usually works, logs in and submits jobs
IDL	Interface Definition Language
JRA	Joint Research Activity
Reference site	Same as <i>Home site</i>
RMIS	Resource Management Information System
SDSC	San Diego Supercomputer Center
TTS	Trouble Ticket System