



CONTRACT NUMBER 508830

DEISA
**DISTRIBUTED EUROPEAN INFRASTRUCTURE FOR
SUPERCOMPUTING APPLICATIONS**

European Community Sixth Framework Programme
RESEARCH INFRASTRUCTURES
Integrated Infrastructure Initiative

Operation of the full CP2K package

Deliverable ID: DEISA_D-JRA1-5

Due date: October, 30th, 2006

Actual delivery date: November 22, 2006

Lead contractor for this deliverable: Organisation, Country

Project start date: May 1st, 2004

Duration: 4 years

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2008)		
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 Contents

Table of Contents.....	1
1. Introduction.....	2
1.1 Executive Summary.....	2
1.2 References and Applicable Documents.....	2
1.3 Document Amendment Procedure.....	2
1.4 List of Acronyms and Abbreviations.....	2
2. Operation of the full CP2K package.....	4
2.1 CP2K code package update.....	4
2.2 CP2K code package tests.....	4
2.3 CP2K modules.....	4
2.4 CP2K integration into the DEISA CPE.....	4
2.5 Extension of the CP2K application plug-in.....	4
3. Additional work.....	5
3.1 Introduction.....	6
3.2 Analysis of the Functionality of AHE.....	6
3.3 Options for supporting AHE.....	7
3.4 Implementation of an AHE interface.....	9
3.5 Successful tests of the AHE interface.....	9
4. Contributions by HLRS.....	9
5. Workplan Update.....	10
5.1 Enabling work.....	10
5.2 Improvements of the materials science gateway.....	11
5.3 Future deliverables.....	11

1. Introduction

1.1 Executive Summary

Work in months 25-30 was focussed on the full integration of the CP2K package. This includes support of the remaining components, mainly the classical molecular dynamics simulation code FIST.

The previously built QUICKSTEP application plug-in has been extended to also support FIST and the full CP2K package with all existing keywords. The input file syntax and semantics validator has been re-built to support newly introduced keywords.

In addition to the work on fully supporting CP2K, the materials science gateway has been enhanced by designing and implementing an interface to the so-called *Application Hosting Environment (AHE)* used in the materials science Extreme Computing project LIAMS of Prof. Peter Coveney, UK, upon recommendation received at the DEISA Symposium 2006.

The work of HLRS has been focused on the provision and optimization of the newly available versions of CPMD.

The workplan for the next period has also been updated.

1.2 References and Applicable Documents

- [1] <http://www.deisa.org>
- [2] <http://cp2k.berlios.de/>
- [3] <http://www.realitygrid.org/>
- [4] http://www.realitygrid.org/publications/NeSC_paper_edit.doc
- [5] <http://gridsam.sourgeforge.net>
- [6] <http://www.omii.ac.uk/>
- [7] Condor Project Homepage <http://www.cs.wisc.edu/condor/>
- [8] P.V. Coveney, M.J. Harvey, L. Pedesseau, D. Mason, A. Sutton, M. McKeown and S. Pickles, "*Development and deployment of an application hosting environment for Grid based computational science*", *Proceedings of the UK e-Science All Hands Meeting, 19-22 September 2005*.
<http://www.allhands.org.uk/2005/proceedings/papers/366.pdf>
- [9] <http://www.omii.ac.uk/>

1.3 Document Amendment Procedure

1.4 List of Acronyms and Abbreviations

API	Application Programming Interface
CP2K	Car-Parrinello molecular dynamics 2000
CPE	Common Production Environment
DESHL	DEisa Services for Heterogeneous management Layer

EJB	Enterprise Java Bean
GPFS	General Parallel File System
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
HTTPS	Secured version of HTTP
IE	Microsoft Internet Explorer
Java EE	Java Enterprise Edition
JCP	Java Community Process
JRA	Joint Research Activity
JSDL	Job Services Description Language
JSR	Java Specification Request
MDA	Model Driven Architecture
MVC	Model-View-Controller pattern
NJS	Network Job Supervisor
O/R Mapping	Object/Relational Mapping
OASIS	Organisation for the Advancement of Structured Information Standards
OMG	Object Modelling Group
PDA	Personal Digital Agent
POJO	Plain Old Java Object
SA	Service Activity
SOA	Service Oriented Architecture
UML	Unified Modelling Language
UNICORE	UNiform Interface to COmputing REsources
WS	Web Service
XHTML	XML syntax compliant HTML
XLST	XML Style Sheet
XMI	XML container syntax for UML
XML	eXtensible Markup Language
XSD	XML Schema Definition

2. Operation of the full CP2K package

After supporting the QUICKSTEP code in DEISA, work in months 25-30 was focussed on full integration of the CP2K package. This includes support of the remaining components, mainly the classical molecular dynamics simulation code FIST, as well as full integration of CP2K into the materials science portal via the DEISA Common Production Environment, DCPE.

2.1 CP2K code package update

The CP2K package evolution has been closely followed. It has reached a state where the transition from individually maintained versions to a centrally maintained version is now sensible. The most recent stable CP2K version has been installed, on which further work has been based.

2.2 CP2K code package tests

The new version of the complete package has now been tested on various target architectures. Tests both on QUICKSTEP and the new component FIST were successfully conducted.

2.3 CP2K modules

For both QUICKSTEP and FIST, a module has been provided for standardized use. The CP2K package environment can now easily be set-up.

2.4 CP2K integration into the DEISA CPE

The integration of the complete CP2K suite into the DEISA Common Production Environment (DCPE) has been launched, for DEISA wide standardized usability, in close cooperation with the SA4 team.

2.5 Extension of the CP2K application plug-in

The previously built QUICKSTEP application plug-in has been extended to also support FIST and the full CP2K package with all existing keywords. The input file syntax and semantics validator has been re-built to support newly introduced keywords.

In order to achieve this goal, the stylesheets generating the java code as well as cocoon forms had to be extended. For example, so far only parameters with a single value or an unlimited list of values had been used. With FIST there are now parameters which require an exact number of values, e.g. the ATOMS parameter in the LENNARD_JONES section for the nonbonded interactions:

```
&LENNARD_JONES
```

```

ATOMS      O O
RCUT       13.4
EPSILON    3.166
SIGMA      78.198
&END LENNARD_JONES

```

Here there are exactly two atom types for which the interaction can be defined. The style sheets can now also handle any type of parameter with any fixed number of values. A possibility to check the input file for the right number of parameters is also generated,.

Furthermore, work has been started on providing the user with a snapshot of the given conformation based on the information on atom positions and connectivity via chemical bonds, as well as showing the user the a graphical view of the effect of the choice of values for the different parameters in the empirical potentials.

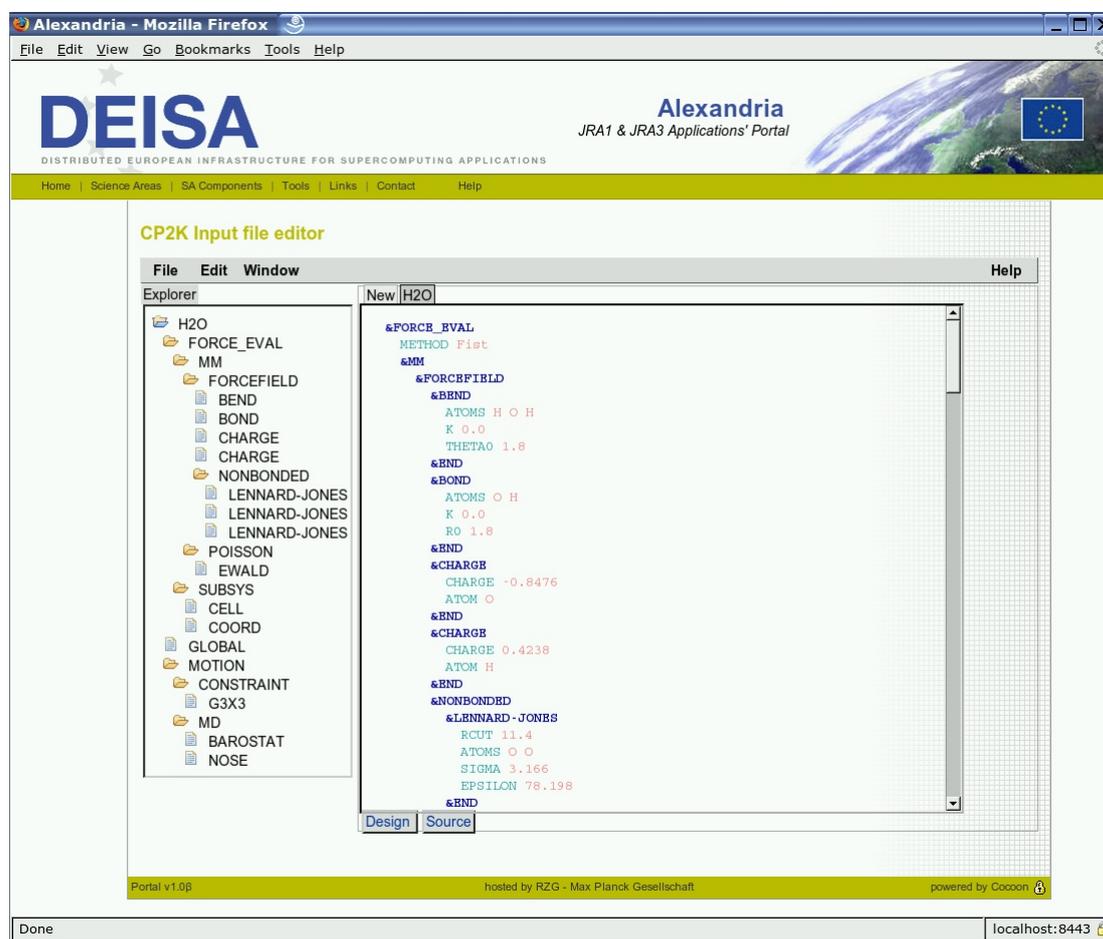


Figure 1: Snapshot of the CP2K input file editor in the materials science gateway, now fully supporting CP2K, also the Monte Carlo and classical molecular dynamics part FIST.

3. Additional work

In addition to the work on fully supporting CP2K, the materials science gateway has been further enhanced by designing and implementing an interface to the so-called *Application Hosting Environment (AHE)*.

3.1 Introduction

The Application Hosting Environment (AHE) is an end-user grid-tool developed in the UK eScience grid for usage in the area of materials science. There is a DEISA Extreme computing project called LIAMS by Prof. Peter Coveney, Imperial College London, one of the initial scientific project advisers of JRA1. During the DEISA Symposium 2006, Prof. Coveney emphasized the usefulness of AHE for materials scientists and recommended its usability in DEISA.

3.2 Analysis of the Functionality of AHE

In [8] AHE is briefly described as follows:

The Application Hosting Environment (AHE) is an extension designed to provide the simplest possible service interface to a client for submitting jobs to highly complex grids.

The security implications of the AHE operating over multiple administrative domains are dealt with by using transport layer security (TLS) and mutual authentication between the client and AHE using X509 digital certificates. Each time an application is started a WS-Resource is created to represent that instance of the application's execution. This WS-Resource provides an interface for the user to interact with that instance of the application; the WS-Resource is provided with a proxy certificate which allows authentication with any services it needs to communicate with.

The AHE interfaces with GridSAM, an OMII funded job submission and monitoring service. GridSAM provides an abstraction of resources, such as Globus Toolkit and Condor on the resource side. Therefore the AHE interacts in a clear transparent way with complex grid architecture and submission systems, such as queuing systems, without interaction from the client. The need for the user to install "heavyweight" middleware such as Globus Toolkit has been removed. The AHE design assumes that the client is behind a firewall allowing only outgoing connections and NAT (Network Address Translation). All the AHE requires the client to support is HTTP, HTTPS and SOAP. The lightweight client could be accessed by the user via a PC, a PDA [17] or even by mobile phone.

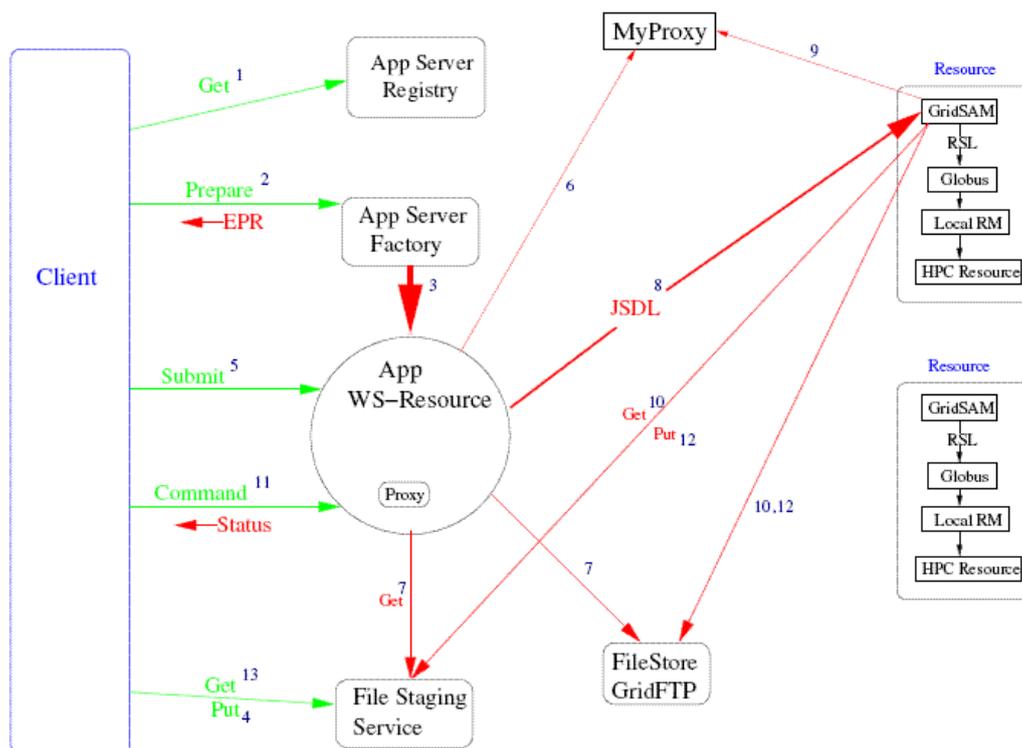


Figure 2: Components and communication paths in AHE. The client uses a previously created Application Web Service Resource in order to submit a job (1-5). That service creates the appropriate JSDL document which is then dispatched to GridSAM. GridSAM takes care of file staging and forwards the job to the underlying Grid Infrastructure such as Globus. During its lifetime the WS Resource can be used to retrieve job status information. Results can be fetched from a File Staging service after completion of the job when files have been staged out.

The AHE interacts in a transparent way with complex grid architecture and submission systems. AHE uses GridSAM for job submission, but GridSAM lacks support for UNICORE, the main grid job submission middleware in DEISA.

Although such a connection to UNICORE is currently missing, it seems to be straightforward to fill the gap by employing an appropriate interface.

Being able to use the AHE in the LIAMS project would greatly lighten the ease of use for the researchers of this project and facilitate the usage of an interface which has become a standard for them. So JRA1 decided to analyse the possibility of building this bridge.

3.3 Options for supporting AHE

In a second step, options for interfacing AHE to the materials science gateway and the underlying UNICORE middleware were explored.

Earlier (see JRA1-2), the JRA1 activity created a Job Submission Enterprise Application (JMEA) which can now easily be reused in order to serve as the missing link between the component GridSAM and UNICORE. If it is possible to integrate that application, only a connector between GridSAM and this application would have to be developed, which essentially converts JSDL documents to the UNICORE AJOs.

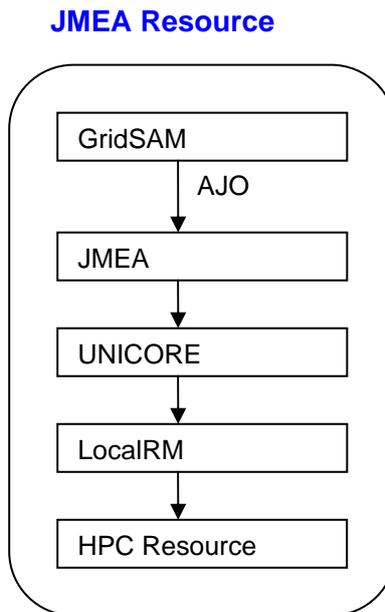


Figure 3: The JMEA Resource from the perspective of AHE. The JDSL document is forwarded to GridSAM as in all other cases. The JMEA connector converts the JSDL document into the UNICORE AJO format and dispatches it to JMEA which in turn takes care of the submission to UNICORE and hence to the underlying UNICORE and HPC infrastructure.

Figure 3 shows the principal structure of the JMEA Resource from an AHE point of view as sketched in Figure 2. A job submitted to GridSAM will be handled by the component extending GridSAM which is to be developed. This component mainly converts the job to the AJO format used in UNICORE and dispatches it to JMEA. JMEA acts then as described earlier.

A job submitted to GridSAM goes through various stages internally. In the case of JMEA the following stages are necessary and needed to be implemented:

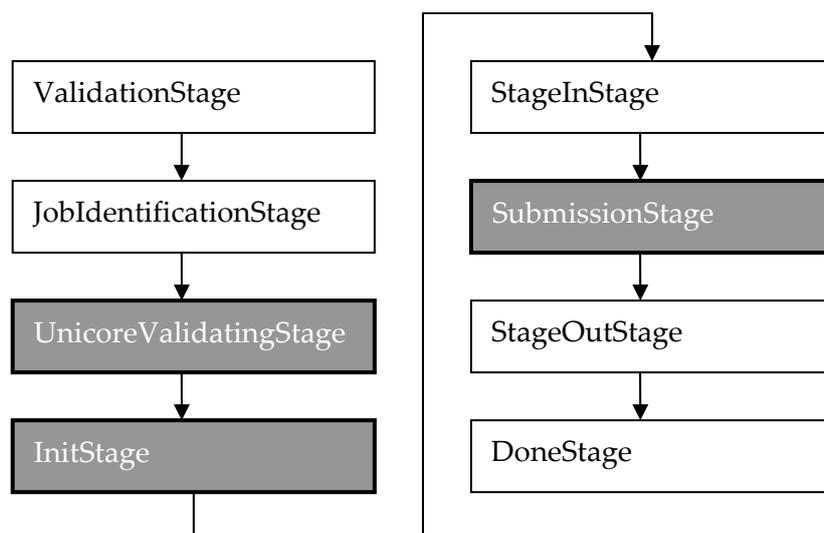


Figure 4: The stages a job goes through assuming success. The shaded stages are the ones which have to be specific to JMEA/UNICORE.

Fortunately, one can make use of GridSAM's standard components and so it turns out that only three components have to be developed more or less from scratch. Those are the `UnicoreValidatingStage` which checks if all necessary conditions are fulfilled and information is given in order to successfully convert this job to an AJO and submit it to JMEA and hence UNICORE.

The `InitStage` performs some initialization steps necessary to have a working environment.

Later, the `SubmissionStage` actually takes care of the job conversion and submission to JMEA and hence UNICORE.

All components for the other stages can be directly used from the ordinary GridSAM distribution.

3.4 Implementation of an AHE interface

The components mentioned above have been developed, resulting in only six classes extending the GridSAM application. For the JSDL to AJO conversion the `jsdl2ajo` library developed by the UNIGRIDS project has been used. Slight modifications were necessary to adjust settings to match the DEISA environment peculiarities as well as support of the JSDL MPI extension.

3.5 Successful tests of the AHE interface

In close collaboration with users from the LIAMS DECI project, the whole instrumentation and processing chain from job setup to submission, to start, to execution and transfer of output data could be successfully tested.

In the LIAMS project, the AHE can now be used via GridSAM with UNICORE.

4. Contributions by HLRS

In summary, HLRS has continued and extended the support for CPMD on the NEC SX-8 system at HLRS, up to the newest version of CPMD.

For CPMD v3.9.1, the testing for large cases has been continued. Extensive tests have been performed in preparation of a large simulation project to find proper parameters for problem size and computer resources.

For CPMD v3.9.2, extensive tests have been performed in preparation of a large simulation project to find proper parameters for problem size and computer resources. The optimization was related to the improvement of the scalability. Some tests have been conducted with the hybrid parallel version of CPMD using MPI and OpenMP. For the moderate number of nodes targeted for production no performance advantage was found.

In the meantime CPMD v3.11.1 has become available and will now be used. For this very new CPMD version installation and tests were performed. The performance of a large run (with 315 GB memory requirement) was assessed:
on 16 nodes NEC SX-8 with 128 cpus, 1.1 TFlop/s sustained were achieved, what corresponds to an overall efficiency of 53%.

During the tests - with only a few iterations - some bottlenecks were identified and were removed. This has improved the scalability, especially for the tests.

5. Workplan Update

The work of JRA1 will be continued at the following two levels:

- Enabling work on further important materials science simulation codes
- Improvements of the materials science gateway:
Updates, enhancements and integration into the DEISA production infrastructure

5.1 Enabling work

A first screening for further important materials science codes of superior relevance had already been carried out at month 24, taking into account the demands for application enabling formulated in DECI proposals from 2005. As candidates the following codes had been marked:

- GROMACS (<http://www.gromacs.org>) (DECI project SNARE)
- ESPResSo (<http://www.espresso.mpg.de>) (DECI project POLYRES)
- LAMMPS (<http://www.cs.sandia.gov/~sjplimp/lammps.html>) (DECI project LIAMS)
- NAMD (<http://www.ks.uiuc.edu/Research/namd/>) (DECI project LIAMS)
- NWChem (<http://www.emsl.pnl.gov/docs/nwchem/nwchem.html>)

An extended screening was planned to also include applications from the second DECI call (which ended on July 15, 2006) in the area of materials science.

A look at the requests shows:

Highest number of requests have been for usage of CPMD (4 requests) and CP2K (3 requests) which are already fully supported by JRA1.

Further requests include usage of the codes GROMACS, NAMD, LAMMPS, ESPRESSO, CASTEP, PWscf/Quantum-Espresso, AMBER, YASP.

A decision on the work plan of making applications generally available was promised for month 30.

In order to be able to distinguish what type of enabling work will be necessary especially from the point of view of JRA1, codes have been looked into with respect of the following categories:

- Enabling work that facilitates code instrumentation via an application plug-in
- Enabling work that requires code optimization and eventually restructuring
- Further enabling work, eventually in the domain of workflows

Naturally code integration into the materials science portal would be preferred tasks for JRA1, while code optimizations and workflow set-ups should be preferentially addressed by eDEISA eSA4-T1+T2.

So the further workplan for JRA1 will include:

Work on NAMD, a highly scalable, well maintained code:

→ development of an application-plugin for the materials science gateway

work on LAMPPS, a well maintained, scalable code:

→ development of an application-plugin for the materials science gateway

ESPRESSO: here a list of tasks has been identified. Priorities will have to be established during the next 6 months. Potential tasks include:

- decoupling of instrumentation and number crunching parts
- code tuning and optimizations, including scalability improvements
- development of an application plug-in

GROMACS, highly efficient code with limited scalability

→ investigation of how an application plug-in for the materials science gateway can be realized;
provision a job submission service for GROMACS in the materials science gateway.

PWscf (<http://www.pwscf.org>)

→ investigation of how an application plug-in for the materials science gateway can be realized;
provision a job submission service for PWscf in the materials science gateway.

5.2 Improvements of the materials science gateway

infrastructure, improvements of the materials science gateway will include an extension of the functionality for supporting online data analysis in terms of standard tasks including 2D plotting of simulation results, standard post processing tasks, case dependent automatic creation of restart jobs (until month 42)

The JRA1 partner HLRS will continue to enable, adapt and optimize relevant applications for the NEC SX-8 vector architecture at Stuttgart.

5.3 Future deliverables

36 month deliverable: DEISA_D-JRA1-6:
Support of a series of important materials science codes with grid-enabled features

42 month deliverable: DEISA_D-JRA1-7
Progress report on enabling work of further important materials science codes and integration into the materials science gateway