

CONTRACT NUMBER 508830

**DEISA**  
**DISTRIBUTED EUROPEAN INFRASTRUCTURE FOR  
SUPERCOMPUTING APPLICATIONS**

**European Community Sixth Framework Programme**  
**RESEARCH INFRASTRUCTURES**  
Integrated Infrastructure Initiative

JRA4 – Production status of Treatment Planning System

Deliverable ID: D-JRA4-2b

**Due date: April 31st, 2005**

**Actual delivery date: May 20, 2005**

**Lead contractor for this deliverable: IDRIS – CNRS, France**

**Project start date : May 1<sup>st</sup>, 2004**

**Duration: 5 years**

<b>Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)</b>		
<b>Dissemination Level</b>		
<b>PU</b>	Public	
<b>PP</b>	Restricted to other programme participants (including the Commission Services)	X
<b>RE</b>	Restricted to a group specified by the consortium (including the Commission	
<b>CO</b>	Confidential, only for members of the consortium (including the Commission Services)	

## Table of Content

Table of Content .....	2
1. Executive Summary .....	3
2. Introduction .....	4
3. Production status of Treatment Planning System .....	5
4. Conclusions .....	6

## 1. Executive Summary

This document is one of the two PM12 deliverables of the Joint Research Activity in Life Sciences. Initially scheduled for PM6, it is actually delivered at PM9. This application was initially because DEISA and EGEE decided to jointly deploy the Radiotherapy Treatment Planning. It describes the status of the two initial "early users" applications in the Genomics area that are planned in the work program. It also describes the activity being carried to deploy a software bio-informatics environment in the DEISA research infrastructure.

This document is publicly available.

## 2. Introduction

The area of Life Sciences is one of the most challenging ones in the context of high performance computing, because in most applications the important raw computing power or the data management facilities provided by the DEISA platforms has to be interfaced and integrated with external lightweight elements (Web interfaces, lightweight servers, etc) that are the ones that are accessed directly by the end users.

This is the reason why particular attention is being paid in this activity to portals that hide the DEISA environment from end users. The Radiation Therapy Planning application belongs to this class, and is jointly deployed with EGEE. By "joint deployment" we mean applications that, according to the specific problem considered, can be operated and run on either platform. In this case, a common portal provides a single point of access which directs the application to the EGEE or DEISA Grids according to some predetermined criteria.

## 3. The Radiation Therapy Planning computational environment

### *The GATE Application*

GATE (Geant4 application for Tomographic emission) is a C++ Monte-Carlo simulation platform based on the Geant4 toolkit. GATE was initially designed for nuclear medical imaging modalities such as Positron Emission Tomography and Single Photon Emission Computerized Tomography. Its functionalities combined with its ease of use make it particularly relevant for radiotherapy and brachytherapy treatment planning.

One of the key challenges related to the deployment of GATE in a grid environment is the perspective of offering services for radiotherapy treatment planning to medical physicists and physicians. Indeed, accuracy of Monte Carlo (MC) dose computation is awesome, provided that the computing power is sufficient to allow for enough run to reduce the statistical noise. The grid may, in the case of the GATE software, be a natural alternative to parallel computers, because GATE is a single processor application and the parallel nature of the Radiation Therapy Planning system relies on the possibility of executing several independent computations simultaneously on different processors. In this way, MC dose computations could become standard for radiotherapy quality assurance (QA), planning.

The GATE application relies on Monte Carlo methods to simulate the propagation of energetic particles in human tissues. Each particle trajectory is an independent event that is determined by some random initial condition and the nature of the tissues in which the particle propagate (an input for the computation). As we said before, GATE is not a parallel application itself; it runs on only one processor. But the whole simulation can be parallelized with respect to the initial condition, and belongs to the well known "embarrassingly parallel" class. Different trajectories can be run on different processors, and the complete results can be collected at the end of all runs for statistical analysis. Each trajectory must of course be able to access the input data, but the grid portal developed by EGEE handles the dispatching of the input data to all the computing

platforms involved in the simulation (as well as the recovery of the final results).

The global Radiation therapy computational environment is depicted in the Figure below. The main point in the DEISA-EGEE collaboration is to interface the DEISA supercomputing environment – in particular, the AIX super-cluster – to the front end GATE environment developed by EGEE, in such a way that this portal can also be used to submit GATE computations to the DEISA Grid. In a previous deliverable (D-JRA2-1b) we have reported on the work done to migrate and validate the GATE application to the DEISA IBM AIX platforms.

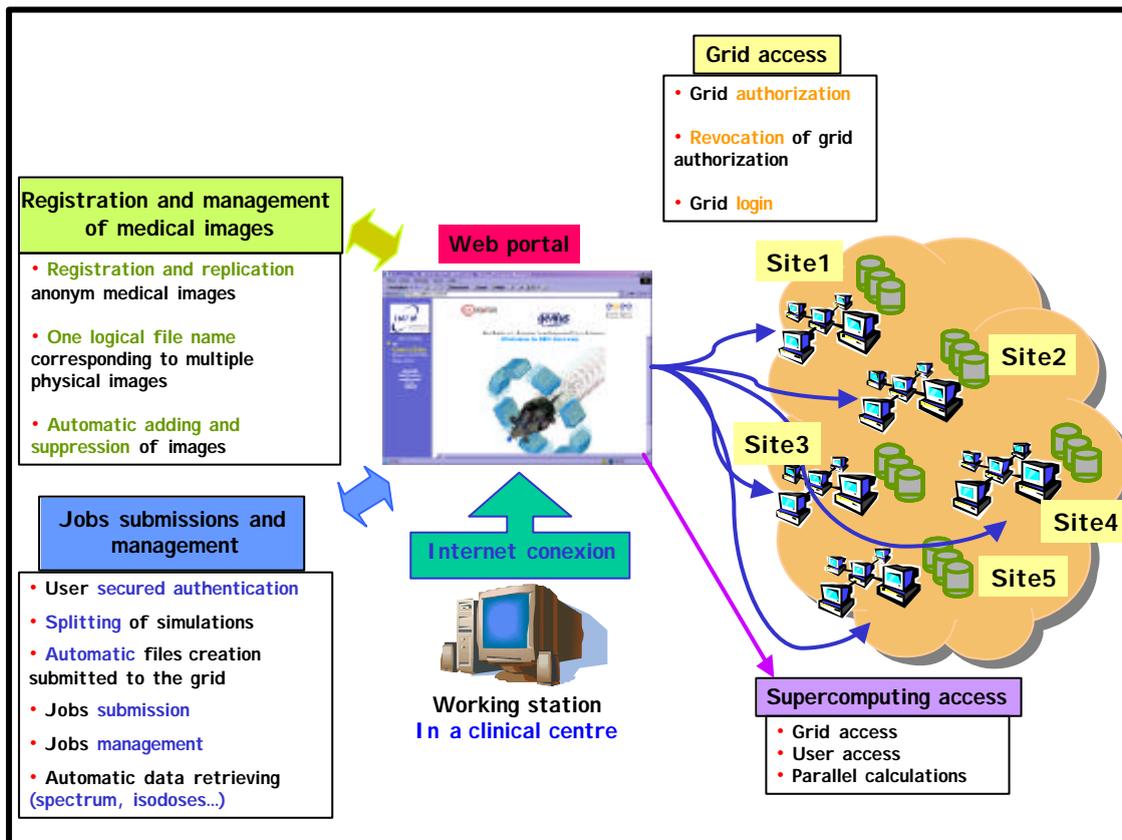


Figure 1 : schematic representation of the grid interface

## 4. GATE in a supercomputing environment

Since the “embarrassingly parallel” nature of GATE makes it very well adapted to the EGEE Grid environment, one may wonder about the interest of rerouting it to a supercomputer. Indeed, GATE is a single processor application, and from the point of view of single processor performance, the DEISA platforms are probably more efficient than PC processors, but the difference is not overwhelming. Single processor performance alone does not justify the deployment on this application on the DEISA platform. This point was clearly recognized by the EGEE-DEISA technical teams involved in the collaboration.

The reason why this application has nevertheless been migrated to the DEISA platform is that the end user does not care about single processor performance; he cares about turnover times, namely, how much he waits to get the end result. Therefore, for some “mission critical” cases where turnover times must be as short as possible, the DEISA platform can make a difference. In the EGEE Grid turnover times are not fully guaranteed if the embarrassingly parallel application is run asynchronously in several scattered computing platforms. On the DEISA platform, a large number of processors can be synchronously allocated to an application. Moreover, DEISA expects to provide in 2006 an “advanced reservation” and “co-allocation” service that will be able to pre-allocate substantial resources to GATE applications. The strategy is therefore meaningful. The DEISA resources will be reserved to some special, mission critical cases.

## 5. Status of GATE on the DEISA platform

The problem that has to be solved before the DEISA platform is aligned behind the EGEE portal for production is the single processor performance of GATE on the 64-bit IBP Power4 AIX architecture. It turns out that up to now these performances are mediocre, because GATE on any Linux PC outperforms the current IBM Power4 code by a factor that can go up to 1.8. We have said before that single processor performance was not the main motivation for migrating GATE to a supercomputer, but DEISA is not willing to engage its expensive supercomputers in inefficient code that can run much better on other platforms.

Let us make it very clear that this was the first migration of GATE to a 64 bit architecture. GATE has always been operated on 32 bit PC architectures and the present situation only means that this code is validated, but not yet well optimized for the DEISA IBM platform.

The most efficient strategy for code optimization is finding the right spot that is producing the most important performance bottleneck. It is not unusual to increase in this way the code performance in a significant way (factor 2 to 3). GATE is a complex application, improved and modified over the years, and eventual code modifications require careful assessment.

We have tried to analyze the possible origins of performance bottlenecks. As we said before, GATE is a complex C++ code using extensively the Standard Template Library (STL), written at a time before the emergence of the final STL standards (this was one of the difficulties with the migration, because our 64 bit compilers are STL standard compliant). The code modifications introduced at IDRIS and reported in the PM6 deliverable make the current 64 bit code in principle portable to any other 64 bit platform.

Performance bottlenecks can be produced by structural reasons, or by implementation reasons:

- ? Structural reasons: GATE is a code that has an important dynamic memory allocation activity. The code systematically creates and destroys a substantial number of small objects. This is, of course, dictated by the initial design that aims at running the code on lightweight platforms with a restricted memory address space. But this strategy is very inefficient on modern supercomputers, and we believe that important performance factors could be gained by modifying the dynamic memory allocation strategy.
- ? Implementation reasons: As we said before, GATE relies heavily on the STL, and it could very well be the case that the particular implementation of the STL we are using is responsible of the moderate single processor performance.

In order to decide if STL implementations are relevant for this performance problem, we have decided to migrate and test GATE on all the DEISA platforms: IBM Power5, SGI AILTIX based on Itanium processors, Cray systems based on Opteron processors, etc. This work is under way.

If we find a particular DEISA platform where performance is acceptable, then we will immediately align the platform behind the EGEE portal and complete the deployment of the GATE application on the DEISA platform. Otherwise, the structural properties of the code will have to be re-examined, but it is not clear to us that it is worth pushing the human investments too far in this direction.

The present situation is the following:

- ? The performance of GATE on the other IBM Power4 platforms of the AIX super-cluster is the same as IDRIS: the performance ratio with respect to our reference standard (& 32 bit PC architecture) is of the order of 1.8.
- ? The performance of GATE on the 64-bit Itanium SGI ALTIX systems of the DEISA Grid is much better. The same performance ratio is of the order of 0.7.

More detailed analysis on other platforms are under way.

-----

