APPLYING GRID TECHNOLOGY TO GAIA DATA PROCESSING

S.G. Ansari[1], Y. Balague-Jordan[1], X. Luri[2], M. ter Linden[3]

[1]Directorate of Science, ESA-ESTEC, 2200 AG Noordwijk, The Netherlands
[2]Astronomy and Meteorology Dept., University of Barcelona, E-08028 Barcelona, Spain
[3]Dept. AS&E/SWE Dutch Space BV, 2303 DB Leiden, The Netherlands

ABSTRACT

In recent years the Grid technology has matured to a degree, where it has become reliable enough to begin experimentation with. The Gaia data processing problem is complex and time-consuming. It requires the effort of about 20 European institutes involved in proposing and maintaining algorithms to process Gaia's data, right from the moment it arrives on Earth. In this talk, we review the technology being used and the testbeds that have been deployed so far to help increase the computing power to overcome performance issues and increase the efficiency of data reduction.

Key words: Gaia; Grid; Data Processing.

1. INTRODUCTION

Gaia data processing can be subdivided into two categories:

- Core Processing: After the science telemetry is received from the spacecraft, the data will need to go through a process in order to establish positions, proper motions, parallaxes and radial velocities. This process is known as core processing. This in turn is composed of the Initial Data Treatment step, where centroiding and cross-matching takes place; and later the Global Iterative Solution procedure is initiated (see Torra et al. 2005).
- Shell Processing: The second category implies the actual analysis of the data. Typically classification tasks, analysis of multiple star systems, planet detection, photometric calibration, etc are all examples of this category.

Due to the intense computational complexity of core processing, which we expect will require little modifications once processing commences on a regular basis, we assume that these tasks should be centralised, due to the fact that they act upon the entirety of the data.

Shell tasks on the other hand involve the community as a whole. These are the algorithms that will be required to analyse the data. They can work with parts of the data and do not require the entire Gaia data base all at once.

2. SHELL TASKS

The definition of the shell tasks are as follows:

- Shell Tasks may be developed by autonomous groups, independent of a core team
- Shell Tasks deliver *derived* data
- Shell Tasks can be collaborative tools
- Shell Tasks are building blocks for data analysis. They may be combined to address more complex processing tasks. Typically, a detection of variability may lead to further analysis of binarity of a multiple star system as other astrometric properties are taken into consideration.

Shell tasks would involve scientific research and deeper collaboration amongst the various groups working on the Gaia data analysis. These groups would typically belong to a *virtual organisation* that shares the same computational environment. The data and code within this group would be accessible to all members.

3. THE GRID

The Grid is based on three major concepts within the context of Gaia computing requirements:

- It is a resource sharing environment. This can be a set of machines, or data storage or algorithms themselves.
- The Grid's main advantage is the capability of the protocol to augment computational performance whenever and wherever this is needed.

- It is an ideal collaboration tool, where a workflow concept can be applied to initiate logical analytical paths for data analysis.

Based on these factors, the Grid environment is an ideal vehicle to use for shell task development.

4. WORKFLOW

The concept of workflow is to allow compatible components to seamlessly interact with one another, provided the output of one is accepted as the input of the other. Workflow *services* may be runnable algorithms, data extractor routines, or simply data storage units, see Figure 1.

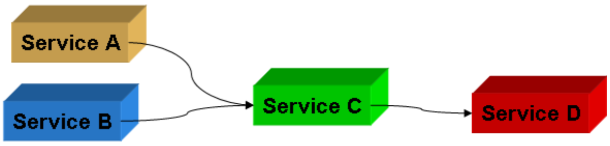

Figure 1. Services can be considered either as algorithms or data storage. This generic diagram describes the creation of workflows that describe components of a larger data processing pipeline.

4.1. GridAssist

We have identified a *workflow* tool that allows the combination of shell tasks to define pipeline processing. GridAssist, developed by DutchSpace allows users to declare individual algorithms as well as share computational resources. Based on the emerging Globus standard Grid protocol, the application manages distributed computational resources and furnishes a workflow tool as a client for users to access each individual algorithm, distribute them accordingly and access the data storage.

The GridAssist Workflow Tool provides the user with a graphical user interface that makes it easy to construct workflows using a drag and drop mechanism as seen in many modelling tools. Services can be selected from a registry and dragged to a canvas without the necessity of having any prior knowledge about their location or capacity. Data flows can be created by connecting inputs and outputs of the services. Checks are performed whether output of one service can be used as input to another service. The workflow tool is a java application running on Windows, Linux and most Unix versions.

When the workflow is completed it can be submitted to the controller. The controller hosts the workflow engine that will execute the workflow. For each service a suitable resource will be selected, the input files will be transferred to the resource and the service is executed. When possible, services will be executed in parallel on several resources. By using wildcards in the specification of inputs and outputs, very large jobs can be specified with small workflows.

The controller is implemented as a standard Web Service. This means that the service is described using WSDL (Web Service Description Language), and that it can be accessed using SOAP (Simple Object Access Protocol). This allows people to write their own clients for the controller, like command line tools or web based interfaces and also firewalls are usually no problem when using SOAP, see Figure 2.

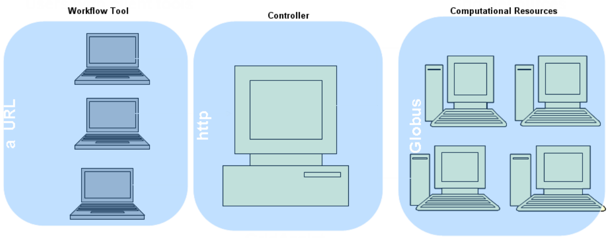

Figure 2. The GridAssist environment is composed of a Workflow tool that acts as a client to access a controller that in turn manages access to Grid resources. While the client uses the http protocol to access the controller, the controller accesses the Grid resources using the Globus protocol.

4.2. The Gaia Virtual Organisation

There are several groups currently working on different aspects of Gaia data analysis. They range from the photometry groups in Cambridge and Leiden to the RVS groups in Paris and Padua as well as Quick Look analysis in Heidelberg (see Figure 3.)

The objective of the current experiment being carried out is two-fold:

- Create a testbed to determine the feasibility of a distributed computational environment, based on standard Grid protocols. This includes access to core data upon which an algorithm is used to carry out an analytical task.
- Create a high-performance computing environment to carry out simulations. In this scenario, a single task, such as the Gaia Simulator is run over a wide range of computational infrastructures all connected using the standard Grid protocol.

4.3. The Experiments

For the purpose of demonstrating the feasibility of the Grid approach, two typical shell algorithms have been selected: the Astrometric Binary Stars (Pourbaix) and Variability Star Analysis (Eyer). They both rely on core data to carry out further analysis of detecting binary or variable stars respectively. In the first example, a data extractor script is prepared in order to query the data base, cur-

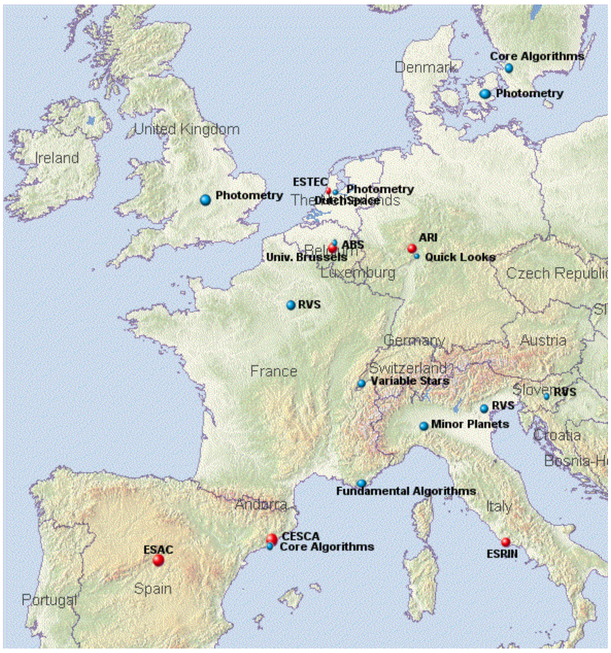

Figure 3. The current status of the Gaia Virtual Organisation (denoted by red dots) and algorithm providers (denoted by blue dots.)

rently accessible on a node at the Super Computing Centre (CESCA) in Barcelona. The result of the data is then passed on to the analysis code, which is run on a node in Brussels. The storage for the result is arbitrary and can be placed on either a disk in ESTEC or in Barcelona. Similarly, the Variable Star Analysis would extract data from the Gaia Results Data Base, run the code on a machine in Geneva and store the result on a Gaia disk. Depending on the amount of storage locally available at the site where the algorithm is run, large or smaller amounts of data can be extracted and worked on. Most of the data extraction for these purposes would be based on celestial coordinate search.

5. SIMULATION OF BINARY STARS: A TESTBED FOR GRID COMPUTING

As described in the previous sections, the Astrometric Binary Star Analysis is one of the algorithms selected to be run in the GaiaGrid environment. As its name clearly shows, this algorithm requires an instance of the GDAAS data base containing data for astrometric binaries to be run. Therefore, the testing of this algorithm needs the generation of realistic simulations of telemetry for this type of stars to be ingested into the GDAAS system.

The generation of simulated data in the context of the Gaia mission is under the responsibility of the Gaia Simulation Working Group (see Luri et al. 2005). Specifically, the simulation of Gaia telemetry is carried on using the GASS simulator (see Masana et al. 2005). GASS has been designed and built to generate realistic simulations of Gaia telemetry to be used by the GDAAS system. This simulated telemetry is ingested into the GDAAS system to populate the test data bases.

In our case, GASS was specifically adapted to generate simulated telemetry for astrometric binary stars, the ones needed by the ABS algorithm. The requirement for the initial tests was to generate observations for about 1000 of these objects for the five years of mission.

In spite of the small number of objects, the generation of these simulations requires a non-negligible computational effort because several aspects of the simulation are not related to the number of objects but to the simulated time span. Therefore, GASS was deployed in the GaiaGrid environment to exploit the computational power it provides. In the case of GASS this deployment is particularly effective because thanks to its design the simulation can be completely split into several smaller pieces that can be simultaneously sent to different nodes of GaiaGrid, taking full advantage of the distributed computing environment provided.

The simulations were run in GaiaGrid using the Grid-Assist workflow tool during a week in October 2004. After some initial testing and fine-tuning, 183 independent jobs were launched to the GaiaGrid environment, covering the full period of five years of simulation. These jobs were automatically distributed to the GaiaGrid nodes by Gridassist and the results were collected in a central repository located at CESCA, accessible through grid. A short summary of the experiment follows:

- The simulation was run in 23 nodes distributed in 8 institutes of 5 countries (see Figure 3)
- A total of 3.8 million CPU seconds were spent
- A total of 16.5 GB of data were produced and automatically transferred to the central repository at CESCA

The simulation was completed in about 4 days (including some stop times devoted to checking of the partial results). Although the simulation could have been certainly carried on without GaiaGrid, the use of this tool made the effort both much simpler and shorter.

6. CONCLUSIONS

It is expected that by the beginning of 2005 the necessary experience of running these experiments would have been established. It is clear that this approach will provide the Gaia community not only with necessary computing power to analyse and implement the algorithms for the operational phase, but also a *collaborative* environment, where various groups can share and together implement the necessary modules to analyse the data. The expectation is that more groups will join GaiaGrid in order to cover all aspects of shell computation. Once this is complete the final versions of the algorithms will have been finely tuned and ready to be implemented in the analysis pipelines.

REFERENCES

Ansari, S.G. et al. 2003, Gaia Spectroscopy, Science and Technology,Ed. U. Munari, ASP Conf. Series Vol. 298, P. 97

Eyer, L. 2005, ESA SP-576, this volume

Luri, X. et al. 2005, ESA SP-576, this volume

Masana, E. et al. 2005, ESA SP-576, this volume

Pourbaix, D. 2005, ESA SP-576, this volume

Torra, J. et al. 2005, ESA Sp-576, this volume