GPS@ Bioinformatic Portal: from Network to EGEE Grid

Christophe Blanchet¹, Christophe Combet¹ and Gilbert Deléage¹

¹ Institut de Biologie et Chimie des Protéines, IBCP UMR5086 CNRS/UCBL, IFR 128 BioSciences Lyon-Gerland
7 passage du Vercors 69007 Lyon France
carol.christophe.blanchet@ibcp.fr

Abstract: Bioinformatic analysis of data produced by complete genome sequencing projects is one of the major challenge of the current years. Integrating up-to-date databanks and relevant algorithms is a clear requirement of such analysis. Grid computing would be a viable solution to distribute data, algorithms, computing and storage resources for Genomics. Providing the bioinformaticians with a good interface to grid infrastructure, such as the one provided by the EGEE European project, is also a challenge to take up. The GPS® web portal, “Grid Protein Sequence Analysis”, aims to provide such an user-friendly interface for these grid genomic resources on the EGEE grid.

Keywords: Genomics, Grid computing, Tool integration, Web portal

1 Introduction

Bioinformatic analysis of data produced by high-throughput biology is one of the major challenge for the next years. Some of the requirements of this analysis are to access up-to-date databanks (sequences, patterns, structures,….) and relevant algorithms of different types (sequence similarity, multiple alignment, pattern scanning,…). Since 1998, we are developing a Web server called NPS@ ([3], Network Protein Sequence Analysis). This server provides the biologist with many of the most common resources for protein sequence analysis integrated into a common workflow. These methods and data can be accessed through the HTTP protocol with a web browser, a client software like MPSA [4] or AnthProt [5].

Today, our server as well as other genomi/post-genomic web portals are limited by the computing resources available in the backoffice of the CGI or servlets interfaces. Indeed some methods are very computing-time and memory consuming. All these web portals have to face to an increasing demand of CPU and disk resources and to the management of the bioinformatic resources (algorithms, databanks). That is why, most of the time, the portal administrators put some restrictions on the available methods and databanks.

Grid computing concept [6], as deployed in the EGEE european project [7], may be a viable solution to foresee these resources limitations [8] [9]. EGEE’s goals are to build an European grid infrastructure. After one year of work, the platform set-up contains 120 sites, for a total of 12 000 CPUs. These resources are available to users through specific components of this middleware: the user interface (UI), the job description language (JDL) and job submission mechanisms. Nevertheless EGEE user interface and usage are still raw and hardly accessible to non-computer scientist.

2 EGEE: European grid infrastructure

2.1 European project EGEE

The Enabling Grids for E-sciencE (EGEE [7]) project is funded by the European Commission and aims to build on recent advances in grid technology and develop a service grid infrastructure. EGEE aims to integrate current national, regional and thematic computing and data Grids to create a European Grid-empowered infrastructure for the support of the European Research Area, exploiting unique expertise generated by previous EU projects (DataGrid, CrossGrid, DataTAG, etc.) and national Grid initiatives (UK e-Science, INFN Grid, Nordugrid, GridIreland, etc.). The EGEE consortium involves 70 leading institutions in 27 countries, federated in regional Grids, with a combined capacity of over 20000 CPUs, the largest international Grid infrastructure ever assembled (Fig. 1).
2.2 EGEE infrastructure

The project EGEE is building a grid computing platform as it usually defined [6]: a grid is a set of information resources (computers, databases, networks, instruments, etc.) that are integrated to provide users with tools and applications that treat those resources as components within a «virtual» system.

EGEE middleware provides the underlying mechanisms necessary to create such systems, including authentication and authorization, resource discovery, network connections, and other kind of components. The platform is built on the LCG-2 middleware (Large Collisioner Grid release 2), which has been inherited from the EDG middleware developed by the European DataGrid Project ([10], FP5 2001-2003). The middleware LCG-2 is based upon the Globus toolkit release 2([11], GT2).

2.3 EGEE use, workload and data management

The EGEE middleware should allow a grid user to launch a job on the EDG grid through an User Interface (UI). Then the job experience the workload management done by the Resource Broker (RB) which determines where and when this job have to be computed: using one computing element (CE) and one storage element (SE) in case of single job, several CEs an SEs in case of large jobs. A CE is a cluster of several CPUs managed by a scheduler system using batch mechanisms. A SE is a hardware component providing a storage space for the applications. The resource broker knows the current state of the grid by querying the information system that centralize all parameters raised by the grid components (cluster, storage, network,…). When available resources have been chosen, the job is transferred to these components and launched. Once executed, the resource broker is informed and gets it back to the user interface.

Nevertheless, the usage of the EGEE middleware is still raw and hardly accessible to non-computer scientist. Firstly, the user has to connect to an EDG user interface (UI) machine by getting an account on an existing UI or installing one in its lab. The UI needs a dedicated linux machine, and the installation of the LCG-2 middleware is manual, difficult to perform and need to be complete for a fine operation. Secondly, when the UI is up and ready, grid user has then to deal with the middleware command line interface (CLI) containing shell script and C binary, and the syntax of several kind of parameter file such as the job description language (JDL). The LCG-2 CLI [] provides the grid user with workload management of its jobs and data management through files replication mechanisms. The principal actions needed to run a job are the following: job submission (edg-job-submit) getting status (edg-job-status) and downloading results (edg-job-get-output); and for data management they are: data registration (lcg-cr), replication (lcg-rep) and data suppression (lcg-del). All these command are not integrated and need to be executed manually by the user.

3 Bioinformatic portal on the grid

To solve the fact that post-genomic web portals are limited by their local computing resources, two solutions
exist: high-performance computing with cluster or looking at the grid concept [8] [9]. Because we think that grid may be a viable solution to foresee these limitations and to bring computing resources suitable to the genomic research field, we decided to use grid computing, as provided by the project EGEE. Nevertheless, the current job submission process on the EGEE platform is complex and hardly usable by non-computer scientist such as biologist. That’s why we decide to propose to biologists a user-friendly interface to the EGEE grid by adapting our NPS@ web site into the GPS@ “Grid Protein Sequence Analysis” web portal.

3.1 Classification of bioinformatic algorithms

Main aims of web portal for Genomics are to integrate bioinformatic methods and databanks to provide users with an easy way to submit their analysis such as we do it on NPS@ since 1998 (40 methods with 7 databanks).

The bioinformatic algorithms integrated in our protein portal can be classified into 4 categories according to a simple model based on CPU resources they consume and according to the input/output data they wait for or produce (Table 1 and Fig. 2). For example, most of times one single BLAST [12] query needs few CPU resources (for protein analysis) but work on large databanks. And on the other hand, a multiple alignment with CLUSTAL W [13] needs large CPU resources with an input file greatly smaller than a databank.

<table>
<thead>
<tr>
<th>CPU Moderate</th>
<th>Small I/O</th>
<th>Large I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSSP (GOR, DPM, Levin…).</td>
<td>BLAST</td>
<td>FASTA, SSEARCH,</td>
</tr>
<tr>
<td>Physicochemical profiles …</td>
<td>ProScan (protein pattern)</td>
<td>BLAST (genomes), PSI-BLAST</td>
</tr>
<tr>
<td></td>
<td>PattInProt (protein pattern)</td>
<td>PSSP (SOPMA, PHD,… )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CLUSTAL W (genomes) …</td>
</tr>
<tr>
<td>CPU Intensive</td>
<td>CLUSTAL W, Multalin …</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Classification of the bioinformatic algorithms used in GPS@ according to their input dataset and CPU requirements (PSSP: Protein secondary structure prediction).

This classification of the bioinformatic algorithms is essential for their adaptation to the grid. Indeed, according to the class of an algorithm, its adaptation to the grid can be predicted. According to its CPU requirement, an algorithm should be sent to the grid in one unique job or split into several smaller sub-processes. On the same way, small I/O could be send during the job submission with the “sandbox”
mechanism, when big I/O should be managed with the Replica Manager System (RMS).

3.2 Gridification of bioinformatic data and algorithms.

In fact, the major problem with a grid computing infrastructure is the distribution of the data and their synchronization according to the last release [9]. Sending a binary of the algorithm to a node on the grid is quite simple because of its size, few kilobytes, and can be done at each execution, although it isn’t very efficient. But sending to the grid a databank, from tens of megabytes (as SwissProt [14]) to gigabytes (as EMBL [15]), consumes a large part of network bandwidth and greatly enlarge the execution time if done each time a BLAST is submitted to the grid.

One simple solution can be to split databanks into subsets sent in parallel to several node of the grid, in order to run the same query on each subset but it needs a synchronization of all the result files at the end. A more efficient solution is to maintain used databanks on several referenced nodes of the grid and to launch the algorithm on computing resources closer to these nodes.

According to this model of bioinformatic algorithms (Table 1), the submission process has to be different. The algorithm submission processes implemented in our GPS@ portal have been adapted to the EGEE grid context. The algorithms and short datasets are sent at submission time through the grid sandbox process. While the other ones, algorithms analyzing large dataset are executed on grid nodes close to the related databanks, that have been replicated earlier or on demand through the RMS.

3.3 GPS@ - Grid Protein Sequence Analysis.

The grid portal GPS@ (“Grid Protein Sequence Analysis”, URL http://gpsa.ibcp.fr) simplify and automated the EGEE grid job submission and data management mechanisms with XML descriptions of available bioinformatic resources: algorithms and databanks (Fig. 3).

![Figure 3. GPSA architecture and interface to the EGEE grid.](image-url)

GPS@ portal runs its own EGEE low-level interface and provides biologists with a web interface to the whole grid platform. They only have to paste their protein input data into the corresponding fields of the submission web page. Then, the execution of the jobs on the EGEE resources can be launched by simply pressing the “submit” button. All the EGEE job management is encapsulated into the GPS@ backoffice: e.g. scheduling and status of the submitted jobs (Fig. 3 and Fig. 4). Finally, the results of the biological grid jobs are displayed into a new web page, ready for other analysis or for download (Fig. 4).
4 Conclusion

GPS@ grid web portal (Grid Protein Sequence Analysis, http://gpsa.ibcp.fr) is a bioinformatic integrated portal such as the current NPS@ protein portal, and would provide the biologist with a user-friendly interface for the GRID resources (computing and storage) made available by the project EGEE (2004-2005).

This genomic grid user interface hides the mechanisms involved for the execution of bioinformatic analyses on the grid infrastructure. The bioinformatics algorithms and databanks have been distributed and registered on the EGEE grid and GPS@ runs its own EGEE interface to the grid. In this way, GPS@ portal simplify the bioinformatic grid submission, and provide biologist with the benefit of the EGEE grid infrastructure to analyze large biological dataset: e.g. including several protein secondary structure predictions into a multiple alignment, or clustering a sequence set by analyzing, with BLAST or SSEARCH, each sequence against the others, …

In the future, main efforts should be focused on taking bioinformatics specific constraints and requirements into account on the EGEE grid. That means, for example, including ontology and semantic parameters into the gridified data with the replica manager system. An other effort should concern the security of the bioinformatics data and methods on the grid: encryption of data, network isolation and algorithm execution sandboxing, fine grain access to data, monitoring private data transfer and replication, …

Acknowledgements

This work is supported by GriPPS project (ACI GRID PPL02-05) and EGEE project (EU FP6)

References


