High Throughput Protein Similarity Searches in the LIBI Grid Problem Solving Environment

Maria Mirto[1], Ivan Rossi[2,3], Italo Epicoco[1], Sandro Fiore[1], Piero Fariselli[2], Rita Casadio[2], and Giovanni Aloisio[1]

[1] SPACI Consortium & ISUFI University of Salento, Lecce & CACT of NNL/CNR-INFM
{maria.mirto,italo.epicoco,sandro.fiore,
giovanni.aloisio}@unile.it
[2] Biocomputing Group, University of Bologna, Italy
piero@biocomp.unibo.it, casadio@alma.unibo.it
[3] BioDec Srl, Casalecchio di Reno, Bologna, Italy
ivan@biodec.com

Abstract. Bioinformatics applications are naturally distributed, due to distribution of involved data sets, experimental data and biological databases. They require high computing power, owing to the large size of data sets and the complexity of basic computations, may access heterogeneous data, where heterogeneity is in data format, access policy, distribution, etc., and require a secure infrastructure, because they could access private data owned by different organizations. The Problem Solving Environment (PSE) is an approach and a technology that can fulfil such bioinformatics requirements. The PSE can be used for the definition and composition of complex applications, hiding programming and configuration details to the user that can concentrate only on the specific problem. Moreover, Grids can be used for building geographically distributed collaborative problem solving environments and Grid aware PSEs can search and use dispersed high performance computing, networking, and data resources. In this work, the PSE solution has been chosen as the integration platform of bioinformatics tools and data sources. In particular an experiment of multiple sequence alignment on large scale, supported by the LIBI PSE, is presented.

1 Introduction

Biological data needs to be reusable, shareable and suitable for "in silico" experiments. In bioinformatics, an experiment is generally characterized by a search in huge biological databases and by the execution of tools that need to access such data intensively. A complete and integrated software environment to execute biological applications is needed, in order to assist scientists and researchers during management and coordination of all of the tasks of an "in silico" experiment, requiring large computational power. Grid Problem Solving Environments (GPSEs) can offer a solution for handling and analyzing so much disparate data connecting many computers within and among institutions through middleware software.

Indeed, according to the definition given by Gallopoulos, Houstis, and Rice: "A PSE is a computer system that provides all the computational features necessary to solve a

P. Thulasiraman et al. (Eds.): ISPA 2007 Workshops, LNCS 4743, pp. 414–423, 2007.

target class of problems.... PSEs use the language of the target class of problems." [5]. Moreover, Grid computing [6] represents an opportunity for PSE designers and users. It can provide an high-performance infrastructure for running PSEs and, at the same time, a valuable source of resources that can be integrated in PSEs. Grids can be used for building geographically distributed collaborative problem solving environments, and Grid aware PSEs (G-PSE) [12] can search and use dispersed high performance computing, networking and data resources.

An user can compose her experiment by using a graphical user interface, accessible via Web, through a Grid Portal, that is an access point to the Grid Problem Solving Environment, based on heterogeneous grid resources exploiting several Grid Middleware such as Globus [14] and gLite [13]. The Grid Portal provides several bioinformatics tools, both sequential and parallel, allowing transparent usage of the underlying grid resources.

This work presents an experiment of multiple sequence alignment (MSA) of human proteins on large scale, by using Workflow technology [16] in a Grid environment. In particular, we have used the Position Specific Iterative (PSI)-BLAST [18], a sensitive sequence similarity search tool, that uses an iterative searching method and a unique scoring scheme to detect weakly related homologues.

Our goal has been the multiple alignment of a large number of human proteins, stored in the UniProt [1] and TRemBL data banks, against those present in the Uniref90 data bank. Taking into account that the number of proteins is very huge, about 70.000, Grid resources are needed in order to reduce the computational time as well as to automate several steps needed for performing the experiment and obtaining the result. In order to run this experiment, a Grid PSE has been developed. It is based on efficient mechanisms for extracting the data and solutions for running several applications composed in a given order, i.e. a workflow. In particular a Workflow engine and a workflow editor have been developed.

The workflow is a component developed inside the Italian LIBI (International Laboratory of BioInformatics) Project [17], supported by the MIUR (Ministry for Education, University and Research) which aims at the creation of a virtual laboratory where e-scientists can share data and bioinformatics tools.

The remainder of the paper is organized as follows. Section 2 introduces the LIBI project whereas Section 3 describes the experiment with the implementation details. Section 4 describes the LIBI Grid Problem Solving Environment with the experimental results and finally, Section 5 draws the conclusions and highlights future work.

2 The LIBI Project

The Italian FIRB 2003 LIBI project, International Laboratory of BioInformatics, funded by the MIUR (Italian Ministry for Education, University and Research) is active since 2005 until 2009.

Main goal of this project is the setting up of an advanced Bioinformatics and Computational Biology Laboratory, focusing on the central activities of basic and applied research in modern Biology and Biotechnologies.

Project activities involve:

- the construction and the maintenance of genomic, proteomic and transcriptomic databases (such as MitoRes, UTRdb, UTRefdb, UTRSite, ENSEMBL, etc.);
- the development of new databases of pathogens relevant for humans, animals and plants;
- the design, development and maintenance of a cell cycle database;
- the design and implementation of new algorithms and software for the analysis of genomes and their expression products.

Two kind of actors have equal responsibility in the LIBI: technological and bioinformaticians partners.

Technological Research Units (URs) are CINECA in Bologna, INFN of the Padova, Bari and Bologna Sections, SPACI & ISUFI University of Salento, Lecce & CACT-NNL, IBM Semea Sud, that is the industrial partner, whereas bioinformaticians RUs are CNRBA, Biomedical Technologies Institute, CNR, Section of Bari, UNIBO of the University of Bologna, UNIMI of the University of Milan, CBMTS of the Center of Molecular Biomedicine, Trieste in Italy.

3 Workflow Experiment

The biggest step in the prediction of protein structures (secondary or tertiary) was obtained by adopting evolutionary information, usually in the form of protein profiles [2]. This is achieved by aligning to a query sequence all the retrieved similar chains detected using a similarity search algorithm. Routinely the most widely used program is PSI-BLAST, because of its speed and its accuracy [18]. In practice, any modern state of the art tool used to predict protein structures and features (such as secondary structures, membrane protein topology, protein solvent accessibility, protein-protein interactions protein stability changes etc.) takes in input some form of evolutionary information to achieve an accuracy compatible with real-world applications. However, the similarity search and the compilation of the sequence profiles is the most time consuming step for the prediction, but it is a necessary constituent of the majority of the prediction tools. For this reason, a system that can speed up the similarity search step can be profitable both for accelerating the prediction phase and for testing more ideas to improve the current state of the art methods.

In this optic an experiment related to the MSA of about seventy thousand human proteins against the data bank of UniProt NREF Uniref90 has been supported.

The data flow (see Fig. 1) consists in the extraction of the sequences by several files, for each sequence a run of a MSA tool is carried out and then an optimization of the results is made.
Involved tools are:

- a library for the sequence extraction;
- PSI-Blast of the NCBI for the multiple sequence alignment;
- a tool for the result adjustment.

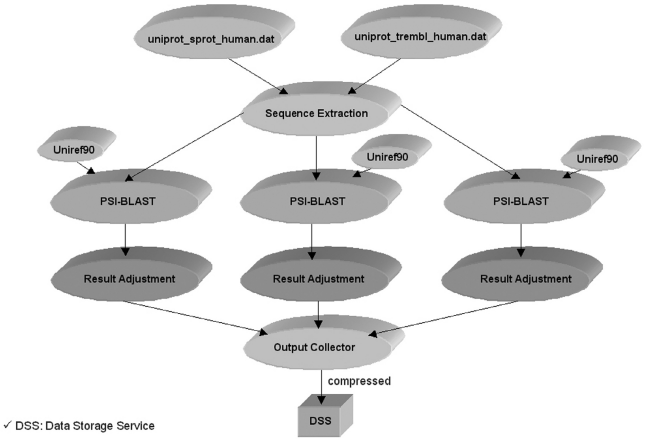


Fig. 1. Experiment Data Flow

A library has been developed for extracting the sequences by annotated input files, i.e. that contains both sequences and other information about the specie, organism, bibliographic references, etc., in EMBL format.

Indeed, dynamic libraries for accessing biological flat files are available inside the library in order to simplify the access to flat files and to provide seamless access. Some features of this library are: i) connection to flat files; ii) data manipulation; iii) information extraction; iv) printing the result in various formats such as Fasta and XML; v) creation of an XML dump of flat files. Moreover, other functions support previous APIs.

PSI-BLAST is a sensitive sequence similarity search tool that uses an iterative searching method and unique scoring scheme to detect weakly related homologues [18].

Finally, in order to reduce the redundancy of the results, taking into account that each result file contains several iterations and last is the most important, a module for reducing the dimension of output files has been developed. It parses output files and deletes the intermediate iterations.

In order to support this experiment, several requirements have been met:

- Access to flat file data bank (UniProt NREF Uniref90) with dimension about 800MB;
- Extraction of 70.845 sequences by annotated input files (human protein - UniProtKB database);
- For each run, the application produces the result of several iterations, specified by the user (three iterations in this experiment). Last iteration is the most important so output files must be updated;
- Management of produced results;
- Need to reduce the total computing time.

In order to satisfy the access to the data bank, it has been installed on grid nodes, where the application is run, and hence indexed. Indeed PSI-Blast runs just on indexed data banks.

The sequences are extracted by using above cited library and the parsing of the results allows reducing redundancy.

Regarding the management of produced result, taking into account that each result file has a dimension that ranges from 200 KB to 2 MB and that these results are on grid nodes it is important to use efficient mechanisms for the optimization of the file transfer time. Indeed, GridFTP protocol has been used so all of the produced files are retrieved on a storage grid node. Finally in order to reduce the total computing time, dynamic scheduling algorithms have been used to allow load balancing in a distributed environment.

For supporting this experiment a Workflow Management System (WMS) has been implemented. It is composed by an editor for modelling the experiment and an engine for scheduling and monitoring the applications execution. Workflow editor allows the discovery of the bioinformatics applications, mapped as graph nodes.

4 The LIBI Grid Problem Solving Environment

PSEs are typically designed for a specific application domain: this simplifies the designer task and generally produces an environment that is particularly tailored for a particular application class. Moreover, PSE users must have transparent access to dispersed and de-coupled components and resources. Thus, managing distributed environments is another main issue in PSEs design and use. Distributed and parallel computing systems are used for running PSEs both to get high performance and for facing distribution of data, machines, and software components.

The Grid is an high-performance distributed infrastructure that combines parallel and distributed computing systems: it is a distributed computing infrastructure whose main goal is resource sharing and coordinated problem solving in "dynamic, multi-institutional virtual organizations".

Thus, the role of the Grid is fundamental to build PSEs, since it provides an enormous number of hardware and software resources that can be transparently accessed by a PSE.

The designed G-PSE has a layered architecture built on a network infrastructure for information exchange (see Fig. 2).

Starting from the top, the first layer includes the services that can be used by the final user (*Application Services*). A graphical user interface allows the user defining a problem and contains the logic to guide the user in the choice of the applications needed for solving a given problem. While a single component of an application is a service, a more complex application is obtained by composing more services using workflows.

At the second level (*Programming Tools and Environment*) are the design tools to support the applications such as digital libraries, software components, etc. These contain the description of each single application and mechanisms for searching the programs and validating their composition. Moreover, the Grid Portal contains the application logic in order to access distributed systems through simple interfaces.

At the third and fourth level (*Grid Middleware Services*) are Grid services that constitute the middleware and, in particular, Data Management, Resource Management and Information Services represent their high-level services.

The Data Management (*Data Mng*) incorporates the grid logic of advanced data management, allowing a dynamic data management, a resource reuse and optimization mechanisms on computational grids such as to improve performances. This service also includes data transfer mechanisms on Grid offered through advanced protocols also able to assure high performances without neglecting security. An information service allows discovering the resources and to know their availability.

Resource Management (*Resource Mng*) functionalities are thought to receive user requirements, planning the operations to be run in the Grid, contacting monitoring and scheduling services for optimal resource allocation in a distributed environment.

Finally, the Security service (*Security*) is transversal to the previous indicated services and its invocation is the first step to be followed in a distributed environment, in order to access services and resources. Finally, at the bottom layer there are local services, containing physical Grid resources and infrastructures (e.g. computers, clusters, networks, instruments, etc.).

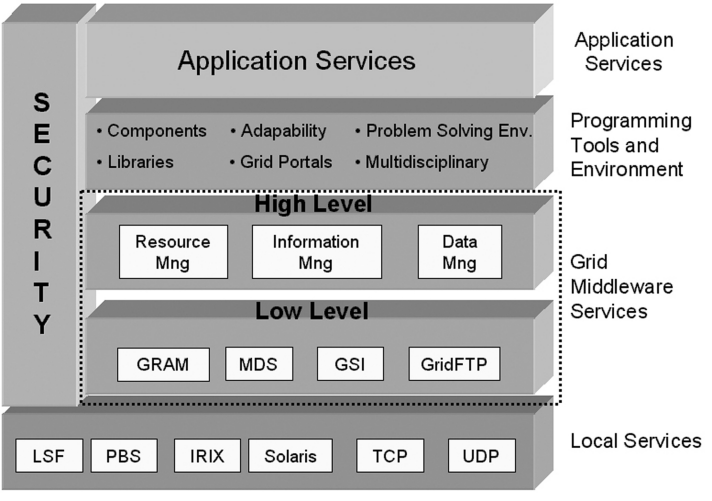


Fig. 2. Grid PSE Architecture

LIBI G-PSE reflects this architecture and several services have been built. Regarding the Application Services layer, today are available the following applications:

- PSI-Blast, used in the experiment;
- MrBayes [7] for the Bayesian estimation of phylogeny;
- Gromacs [3] for performing molecular dynamics, i.e. simulate the Newtonian equations of motion for systems with hundreds to millions of particles;
- PatSearch [11] for pattern matching in order to find a well defined pattern against a given sequence(s) or database (primary or specialized) divisions.

Regarding the Data Management, the LIBI exploits the federated database approach, accessible through the DB2 Integrator system. Several data banks have been already federated such as MitoRes, Mitonuc, GenBank and other. In order to allow the access in Grid, a driver for DB2 Integrator has been developed inside the GRelC (Grid Relational Catalog) toolkit [10]. This toolkit allows the access to distributed resources by using Web Services technology and allowing dynamically and transparently access to the relational and not relational databases.

Regarding the Resource Management, the LIBI exploits an extension of the GRB (Grid Resource Broker) [9] Meta Scheduler in order to schedule the jobs in a Grid environment. Indeed, GRB has been extended for submitting and checking jobs on Globus-based as well as gLite-based machines. Today, the extension at the Unicore-based machines is previous [4].

These services are built on top of basic services offered by Globus and gLite middleware. GRB takes the burden to guarantee the interoperability among heterogeneous middleware by means of different drivers each one in charge to translate the user request in the opportune formalism used on the selected target machine. This means that if the scheduler selects a Globus-based machine, the user request for job submission will be translated in a RSL (Resource Specification Language) statement or if the scheduler selects a gLite based machine the submission will be specified using JDL (Job Description Language) formalism. Analogous mechanisms have been implemented to guarantee the interoperability for file exchange, information access and job tracking.

Finally, for guaranting the execution of workflow job, an editor and an engine have been developed. The editor, implemented in the Java language, discovers at run time the applications that are available in the Grid. Such applications are registered through the LIBI Portal, as it is possible to register also available biological data banks. The user can compose own graph, save it and submit it. The job will be sent at the engine for the submission. Moreover, by querying the monitoring service, it is possible to know the status of the job and finally visualize the results.
The engine has been designed taking into account the following requirements:

1. ability to handle simple workflows described dy directed acyclic graphs (DAGs);
2. ability to handle complex workflows described by arbitrary graphs, supporting cycles and conditions;
3. support for recursive composition i.e., the possibility to define a workflow vertex;
4. scheduling and monitoring of jobs on different grid middleware: Globus, gLite and Unicore.

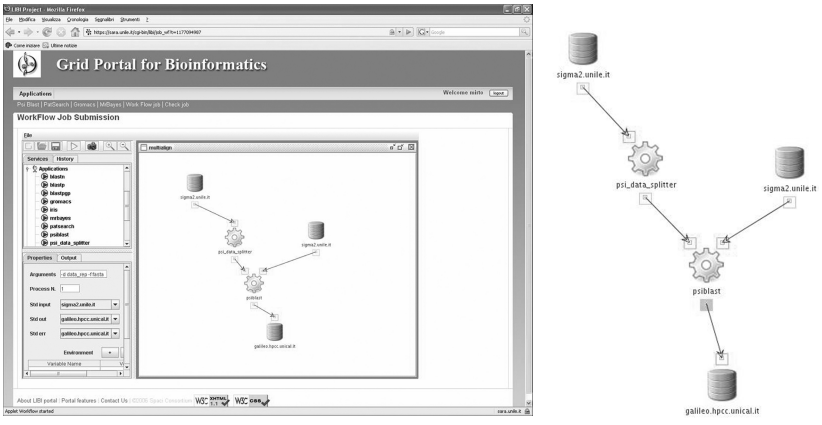
Major details about this service are in [15].

4.1 Implementation

As cited above, a Grid Portal for accessing to the G-PSE has been built.

The implementation is based on CGI written in C using g-SOAP with GSI-plugin [8] in order to contact the GRB scheduler service. Moreover the portal uses a MySQL

DBMS in order to store configuration data as well as user specific environment data. In particular for simulating the experiment, it is possible to use two graphical interface: a web page that contains all of the parameters in input at the algorithm and the workflow editor (Fig. 3(a)) for composing the graph (Fig. 3(b)). Moreover it is possible to monitor the status of the execution of a job and visualize/download the results by using a simple interface.



(a) Workflow Editor (b) MSA graph

Fig. 3. LIBI Grid Portal

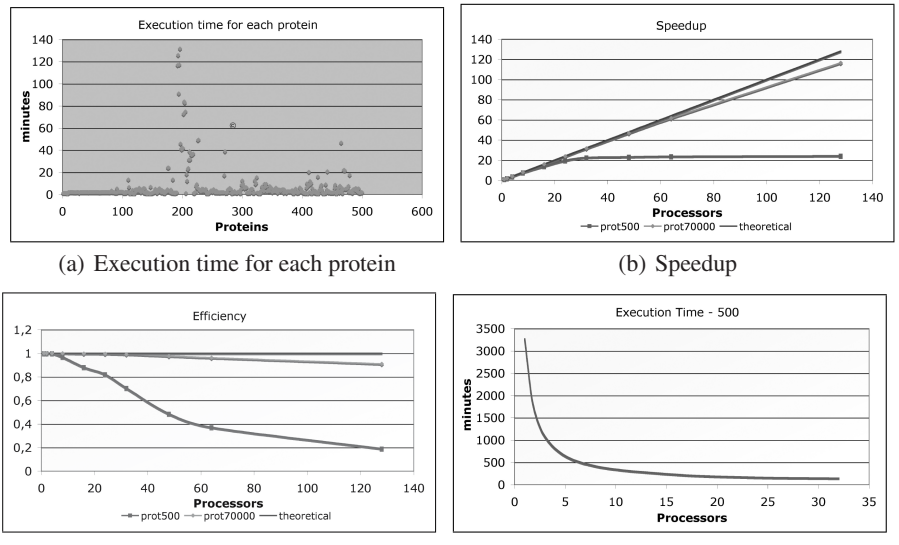
4.2 Results

The experiment has been run on the SPACI Grid (Italian Southern Partnership for Advanced Computational Infrastructures), a partnership among University of Salento, University of Calabria and University of Naples "Federico II". The SPACI Grid is based on three geographically spread High Performance Computing centers located in southern Italy, namely, the MIUR/HPC Center of Excellence of University of Calabria, the CACT/NNL (Center for Advanced Computational Technologies) of University of Salento, and the Naples/DMA (Dept. of Mathematics and Applications) of University of Naples "Federico II".

The parallel application is characterized by a not uniform distribution of tasks to be distributed among the available processes, this implies that the best choice is a dynamic scheduling policy based on a on-demand distribution of tasks. The performance has been evaluated on 500 proteins ranging the number of processes from 1 to 128. This experiment highlighted that the algorithm achieves the best efficiency up to 32 processors. Beyond this limit the execution time can not be reduced due to the existence of a task that takes 130 minutes (Fig. 4(a)). Running the algorithm with 70845 proteins we improved the scalability of the application maintaining an high efficiency. MSA result has been obtained taking a time of about 65 hours, using 128 processors against an estimated 96 days of computing on a single CPU. Moreover, taking into account that

each result file is 1,7 MB, the disk storage occupancy is about 120 Gigabytes; with the adjustment of the results, the storage has been reduced at 20 GB.

The preliminary analysis of the application performance shows that the scalability (Fig. 4(b)) and efficiency (Fig. 4(c)) are pretty good. This is due to the fact that the application is embarrassing parallel with no communication among the parallel tasks. Indeed the Fig. 4(d) shows a great decrease of the execution time when the number of processors grows. This experiment has been an interesting testbed for our platform also for testing the workflow engine.



(a) Execution time for each protein

(b) Speedup

(c) Efficiency

(d) Execution time for 500 proteins

Fig. 4. Computational Time

5 Conclusions

This paper presented an experiment of multiple sequence alignment of human proteins, simulated into a Grid Problem Solving Environment. By using this system has been possible to compose several applications, obtaining a reduction of the computational time. The simulation has been carried by using a workflow, both for designing the graph and for scheduling it into a Computational Grid. Used Middleware are Globus and gLite and this allows using a large amount of resource involved into the SPACI and Egee Projects. Future work involves the design of other experiments to cover the integration in this system of several bioinformatics applications and the full integration of the Unicore middleware into the environment. This work was supported by the MIUR (Italian Ministry for Education, University and Research) in the LIBI (International Laboratory of BioInformatics) project, F.I.R.B. 2003, under grant RBLA039M7M.

References

1. Bairoch, A., Apweiler, R., Wu, C.H., Barker, W.C., Boeckmann, B., Ferro, S., et al.: The Universal Protein Resource (UniProt). Nucleic Acids Res. 33, 154–159 (2005), http://www.uniprot.org
2. Rost, B., Sander, C.: Progress of 1D protein structure prediction at last. Proteins 23, 295–300 (1995)
3. Van der Spoel, D., Lindahl, E., Hess, B., Groenhof, G., Mark, A.E., Berendsen, H.J.C.: GROMACS: Fast, flexible, and free. J. Comput. Chem. 26, 1701–1718 (2005)
4. Erwin, D.W., Snelling, D.: UNICORE: A Grid Computing Environment. In: Sakellariou, R., Keane, J.A., Gurd, J.R., Freeman, L. (eds.) Euro-Par 2001. LNCS, vol. 2150, Springer, Heidelberg (2001)
5. Houstis, E., Gallopoulos, E., Bramley, R., Rice, J.: Problem-Solving Environments for Computational Science. IEEE Comput. Sci. Eng. 4(3), 18–21 (1997)
6. Berman, F., Hey, A.J.G., Fox, G.: Grid Computing: Making The Global Infrastructure a Reality. Wiley & Sons, Chichester (2003)
7. Ronquist, F., Huelsenbeck, J.: MrBayes 3: Bayesian phylogenetic inference under mixed models. Bioinformatics 19, 1572–1574 (2003)
8. Aloisio, G., Cafaro, M., Lezzi, D., Van Engelen, R.: The GSI plug-in for gSOAP: Enhanced Security, Performance, and Reliability. In: Proceedings of Information Technology Coding and Computing (ITCC 2005), vol. I, pp. 304–309. IEEE Computer Society Press, Los Alamitos (2005)
9. Aloisio, G., Cafaro, M., Carteni, G., Epicoco, I., Fiore, S., Lezzi, D., Mirto, M., Mocavero, S.: The Grid Resource Broker Portal. In: Concurrency and Computation: Practice and Experience, Special Issue on Grid Computing Environments. (to appear, 2006)
10. Aloisio, G., Cafaro, M., Fiore, S., Mirto, M.: The Grid Relational Catalog Project. In: Grandinetti, L. (ed.) Advances in Parallel Computing, Grid Computing: The New Frontiers of High Performance Computing, pp. 129–155 (2005)
11. Grillo, G., Licciulli, F., Liuni, S., Sbisà, E., Pesole, G.: PatSearch: A program for the detection of patterns and structural motifs in nucleotide sequences. Nucleic Acids Res. 31, 3608–3612 (2003)
12. Von Laszewski, G., Foster, I., Gawor, J., Lane, P., Rehn, N., Russell, M.: Designing Grid-based Problem Solving Environments and Portals. In: Proceedings of International Conference on System Sciences (HICSS-34) (2001)
13. gLite Project: http://glite.web.cern.ch/glite/documentation/
14. Foster, I., Kesselman, C.: Globus Toolkit Version 4: Software for Service-Oriented Systems. In: Jin, H., Reed, D., Jiang, W. (eds.) NPC 2005. LNCS, vol. 3779, Springer, Heidelberg (2005)
15. Cafaro, M., Epicoco, I., Mirto, M., Lezzi, D., Aloisio, G.: The Grid Resource Broker Workflow Engine. In: IEEE Proceedings of The 6th International Conference on Grid and Cooperative Computing, Urumchi, Xinjiang, China, August 16-18, IEEE Computer Society Press, Los Alamitos (to appear, 2007)
16. Workflow management coalition reference model: http://www.wfmc.org/
17. The LIBI Project: http://www.libi.it
18. Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 17(25), 3389–3402 (1997), http://www.ncbi.nlm.nih.gov/cgi-bin/BLAST/nph-psi