Replication and update of molecular biology databases in a grid environment

J. Salzemann, N. Jacq, Gaël Le Mahec, Vincent Breton

Replication and Update of Molecular Biology Databases in a Grid Environment

Jean Salzemann, Nicolas Jacq, Gaël Le Mahec, Vincent Breton

Laboratoire de Physique Corpusculaire de Clermont-Ferrand, Université Blaise Pascal/IN2P3-CNRS, UMR6533, 24 av. des Landais, 63177 Aubière Cedex, France
{salzemann, jacq, lemahec, breton}@clermont.in2p3.fr

Abstract. Update of molecular biology databases is a growing burden on the biomedical research community. As the grid allows to share and replicate data, we propose a service to automatically update the biology databases from a single changing reference using web services. In this paper we report the components, the architecture and the deployment of the update service on the french RUGBI grid infrastructure. RUGBI is a computing grid infrastructure based on existing middleware and technologies for the community of scientists in bioinformatics.

Keywords: Grid computing, databases replication and update on grid, biological databases, grid services, web services.

1 Introduction

One of the main challenges in molecular biology is the management of data and databases. A large fraction of the biological data produced is publicly available on web sites or by ftp protocols. These public databases are internationally known and play a key role in the majority of the public and private researches. But their exponential growth raises an exploitation problem. Indeed scientists have to access easily to the last update of the databases in order to apply algorithms. The frequent and regular update of the databases is a recurrent issue for all host or mirror centres, and also for scientists using locally the databases for confidentiality reasons.
Grid technology opens perspectives for data access by providing dedicated services for data and meta-data management. In this paper, we will use the word grid to describe a system that coordinates resources which are not subject to centralized control, using standard, open, general-purpose protocols and interfaces to deliver non-trivial quality of service. Different production infrastructures are now available on which applications in the field of molecular biology are deploying access computing and storage resources at a large scale [1] [2]. Grid is also an opportunity to reduce human cost for distributing updated data. An update can be propagated automatically in the different centres with the replication and information services. Deploying a service for automatic database update is a requirement within a grid for bioinformatics.

Our goal is to give the most up to date version of each database for a job in a grid environment. Hence there is a need for a service that will update each site storing the databases through the grid with their last modifications. Here the real challenges are to optimize the use of network bandwidth and have a scalable system that can support easily a large number of storage elements. The system must also be light and transparent enough not to disrupt job execution and to automate the procedure to achieve a minimum of human intervention. At the end the service should be a black box delivering up to date databases on the grid that should prevent users from wondering which is the version deployed. After defining the components and the architecture of the service, we will describe its deployment on the RUGBI grid.

2 Service Components

RUGBI [3] is a French project financed by the Gen'Homme Network whose goal is to build a computing grid infrastructure on the basis of existing middleware and technologies for the community of scientists in bioinformatics. A set of applications has been selected to define the requirements for the RUGBI grid. Most of the applications are fed with third party databases.
The databases used by the biologists are constituted of several flat files, organized in directories. They are just handled as file systems: files can be added, removed or modified and are available directly on ftp servers anonymously. The users have shown interest for the update of SWISSPROT [4], TREMBL [5], EMBL [6], PDB [7], KEGG [8] and NCI [9], which are all falling into this category of file system databases. The RUGBI grid considers databases as resources, which are all registered in the information system of the grid, such as the applications, computing elements and storage elements. This information system is based on XML (Extensible Markup Language) native databases, and all the resources are described in XML sheets stored into it. XML is well adapted to store textual information and metadata on the databases such as information on their ftp repositories, their version, size or the date of the last update. While the description sheets are held on a central service, the databases are stored physically on the grid on storage elements (SE). There are some storage elements of reference (SER) (one per database), which are repositories of the databases on the grid. These grid repositories should be synchronized with the ftp servers.
The RUGBI architecture also includes a grid service called the database finder, used mainly to find the best location of a given database on the grid for a given user. The service manages locks put on the databases by the running jobs to prevent its modification while the job is being executed. Moreover it can forbid a job to use a given database location, hence allowing the update service to perform its operations without disrupting job execution. Locks are needed to ensure the safety of jobs and the integrity of the databases replica. For security issues, locks are automatically removed after a given period of time.
The service is intended to be a direct interface between the databases ftp servers and the SEs of the grid. Its operations will be governed by the information in XML sheets. Our main developments will focus on the RUGBI grid infrastructure but the whole

service is designed to work as a standalone tool that we can adapt easily to any middleware architecture and with any database.

3 Service Architecture

The update process can be achieved through three steps presented in figure 1.

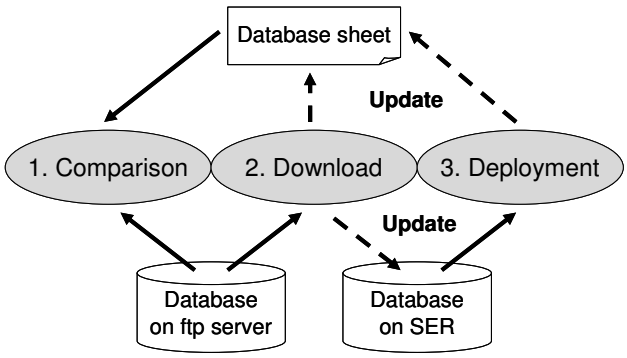

Fig. 1. Database update process on grid. SER is the storage element of reference. Step 1 is done regularly. Steps 2 and 3 are executed for update deployment.

The update service as a client/server application is conceived in two parts communicating with each other. The server, deployed on the storage elements of reference, just regularly compares the versions available on the ftp with the one deployed on the grid using the XML description sheet of the databases given by the information system. If necessary, it rebuilds the databases by downloading the necessary files or directory on repository spaces on the storage element. After that first step it queries the information system to know which are the SEs that host the database and notifies the SE clients to update the database. When an update notification is received by a client, it pulls the data from the SER. To do that, it simply downloads the differences on the SER in a new local working space, closed for jobs, using the specified transfer protocol. As soon as the number of SEs (clients) that successfully deployed the updates reaches a given threshold, the RUGBI Database Finder service is notified to register the new database, this implies that the working space will open and accept new jobs. If a database version is already on the SE (in a working space), the same service is notified to unregister the old version. New jobs won't be allowed to run on the old version, since it is unregistered, but will use the new registered version. When the grid has registered the new deployed database, the XML description sheet of the database is updated on the information system and the old one deleted. As soon as no more jobs are using the old version, the server is notified by the Database Finder Service about it, and deletes the old version. That notification is propagated through the grid so that any SE hosting this version will delete it. The whole process can be repeated through time to keep the databases updated.

4 Service Development and Deployment

Developments were made in JAVA for portability and modularity. A raw version, including the client/server subservices was developed using basic ip socket programming for communication. This enables simple networks to host databases and synchronize them with ftp servers. The internal data transfers between SEs can be achieved through: GridFTP [10] (ftp protocol implementing secured and reliable multichanneled transfers using GSI (Globus Security Infrastructure)), and Rsync (a protocol that optimizes transfer by comparing the sources with the destinations and copy just the differing parts). As the RUGBI grid is based on GT4 [11], the sub services were also implemented as web services, messaging with SOAP protocol, in order to achieve better connectivity with the others services, and to avoid grid sites to handle exotic firewall configurations: they should just allow inbound and outbound connectivities for web services (8080 for Apache tomcat) and gridftp (2811 most of the time). The developments were done under an Apache Axis environment, with the GridFTP and GSI API provided by the globus commodity grid kits.
The RUGBI grid has currently height sites in Clermont-Ferrand, Lyon and Grenoble. The client Update Service is deployed on each SE of the Grid. Moreover the Clermont-Ferrand and Grenoble sites are hosting the SER of the grid, so the master service was deployed there. The following bases are deployed and updated regularly: SWISSPROT (700 MB), TREMBL (2.4 GB), EMBL (release without annotations: 180 GB), KEGG (13 GB), PDB (2.9 GB), NCI (900 MB), representing a total of 200 GB. Once the update of a database is initiated from the portal by the database administrator, its reference site runs the update process each time it is configured. The volume of the transfers required by each update varies from several KB to several GB which depends of the database and their activity. Performance of the complete update process depends of the network bandwidth.

5 Conclusion and Perspectives

The aim of the update service is to provide the grid users the most up to date version of any biological database, to do it transparently and without disturbing the running jobs. The service, packaged as a middleware component, is installed and deployed on the RUGBI grid by the site administrators as a grid plug-in. In the continuity of the RUGBI project, the service will be deployed over several grid middlewares:

- The regional Auvergrid grid [12], based on the EGEE [13] middleware LCG-2, will require some interfacing with the job submission system and to allow jobs to put locks on site databases to secure updates.
- The DIET middleware [14], which follows the gridRPC API defined within the Global Grid Forum [15], presents a hierarchical architecture where agents manage scheduling and some persistence mechanisms. Used in conjunction with a database declaration server, the update service could be easily implemented.
- We finally aim at a deployment on the new GLITE middleware [16], which should be theorically quite similar to RUGBI with its web service-friendly architecture providing APIs for new service integration.

It would be possible to offer this service as an application, but it would mean that its use is not mandatory, and should become a part of a grid workflow. There are also future plans to add some optimisation on deployments of the databases: for example, being able to split databases and lock only parts of it, or add the ability to offer several synchronized SER per databases. The service will mature through its deployments on grid middlewares in production environments.

Acknowledgements. The authors acknowledge the contributions of Alexandre Mula, Matthieu Reichstadt, Yannick Legré and all the RUGBI collaboration. Part of this work has been funded within the framework of the RUGBI project (Ministère de la Recherche, Réseau Gen'Homme)

References

[1] Jacq, N., Blanchet, C., Combet, C., Cornillot, E., Duret, L., Kurata, K., Nakamura, H., Silvestre, T., Breton, V., Grid as a bioinformatic tool, Parallel Computing 30 (2004) 1093-1107.
[2] Breton, V., Jacq, N., Hofmann, M., Grid added value to address malaria, Proceedings of the 6-th IEEE/ACM CCGrid conference (2006)
[3] RUGBI, http://rugbi.in2p3.fr
[4] Bairoch, A., Apweiler, R., The SWISS-PROT protein sequence data bank and its supplement TrEMBL in 1999, Nucliec Acids Res. 27 (1999) 49-54.
[5] Stoesser, G., Tuli, M.A., Lopez, R., Sterk, P., The EMBL nucleotide sequence database, Nucleic Acids Res. 27 (1999) 18-24.
[6] The European Bioinformatics Institute, http://www.ebi.ac.uk
[7] Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E., The Protein Data Bank. Nucleic Acids Research, 28 (2000) 235-242
[8] Kanehisa, M., Goto, S., KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic Acids Res. 28 (2000), 27-30.
[9] NCI and NCBI's SKY/M-FISH and CGH Database (2001), http://www.ncbi.nlm.nih.gov/sky/skyweb.cgi
[10] Allcock, W., Bester, J., Bresnahan, J., Cervenak, A., Liming, L., Tuecke, S., GridFTP: Protocol extensions to ftp for the grid. Tech. rep., Argonne National Laboratory, (2001).
[11] Foster, I., Globus Toolkit Version 4: Software for Service-Oriented Systems, IFIP International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779, pp 2-13, (2005).
[12] Auvergrid, http://www.auvergrid.fr
[13] Enabling Grids for E-sciencE, http://public.eu-egee.org
[14] Caron, E., Desprez, F., DIET: A Scalable Toolbox to Build Network Enabled Servers on the Grid. Technical report RR-5601, INRIA, (2005)
[15] GridRPC Working Group, https://forge.gridforum.org/projects/gridrpc-wg
[16] Lightweight Middleware for Grid Computing, http://glite.web.cern.ch/glite