Grid Computing Solutions for Distributed Repositories of Protein Folding and Unfolding Simulations

Martin Swain[1], Vitaliy Ostropytskyy[1], Cândida G. Silva[2], Frederic Stahl[1], Olivier Riche[1], Rui M.M. Brito[2], and Werner Dubitzky[1]

[1] School of Biomedical Sciences, University of Ulster, Coleraine BT52 1SA, Northern Ireland, UK
mt.swain@ulster.ac.uk
[2] Chemistry Department, Faculty of Science and Technology, and Center for Neuroscience and Cell Biology, University of Coimbra, 3004-535 Coimbra, Portugal
brito@ci.uc.pt

Abstract. The P-found protein folding and unfolding simulation repository is designed to allow scientists to perform analyses across large, distributed simulation data sets. There are two storage components in P-found: a primary repository of simulation data and a data warehouse. Here we demonstrate how grid technologies can support multiple, distributed P-found installations. In particular we look at two aspects, first how grid data management technologies can be used to access the distributed data warehouses; and secondly, how the grid can be used to transfer analysis programs to the primary repositories – this is an important and challenging aspect of P-found because the data volumes involved are too large to be centralised. The grid technologies we are developing with the P-found system will allow new large data sets of protein folding simulations to be accessed and analysed in novel ways, with significant potential for enabling new scientific discoveries.

1 Introduction

Protein folding is one of the unsolved paradigms of molecular biology, the understanding of which would provide essential insight into areas as diverse as the therapeutics of neurogenerative diseases or bio-catalysis in organic solvents. Protein folding simulations are time-consuming, data intensive and require access to supercomputing facilities. Once completed, few simulations are made publicly available – this hinders scientists from accessing data reported in publications, performing detailed comparisons, and developing new analytical approaches. A platform to allow access to protein folding and unfolding simulations, and to support data mining functionalities would clearly benefit the field.

The P-found project [1] aims to create a distributed public repository for storing molecular dynamics simulations, particularly those concerned with protein folding and unfolding. It aims to provide the tools needed to support the comparison and analysis of the simulations and thus enable new scientific knowledge to be discovered and shared.

M. Bubak et al. (Eds.): ICCS 2008, Part III, LNCS 5103, pp. 70–79, 2008.

P-found is designed to be a distributed system. This is because protein folding and unfolding simulations require large volumes of storage space: a simulation can easily comprise 1 to 10 Gigabytes of data, with simulation archives consisting of Terabytes. In addition, the groups who created the simulations have often already carefully stored and archived them. Ideally a distributed system would allow scientists to share their data, without undue interference in their existing practices, and without causing them to lose control over their own data.

There are two data storage components to the P-found system: a *primary data store or file repository* for the unaltered protein folding and unfolding simulations; and a *secondary data store or data warehouse* containing, for instance, information about how the simulations were created and derived physical properties of the simulations that are typically used in scientific analysis [1]. The secondary data is approximately 10 to 100 times smaller in volume than the primary data and consists of local and global molecular properties that summarise the simulations.

Grid technology is appropriate for P-found mainly because the user groups involved are based in independent organisations, each with their own IT infrastructures and security policies. Grid technology provides a set of tools whereby these user groups can come together in a virtual organisation, with security components that enable both data and compute servers to be shared across the independent administrative domains [2]. Data grids have been reviewed by Finkelstein *et al.* (2004) [3] and the efficiency of their design has been discussed by Laure *et al.* (2005) [4].

In this paper we consider the use of grid and distributed computing technologies to manage all aspects of data storage and analysis in the P-found system. Many of the results we present here have been adapted from tools and services developed by the DataMiningGrid project [5], [6]. However, P-found is an ongoing project, independent from the DataMiningGrid, and this is the first time that grid solutions for P-found have been presented in detail.

P-found is designed so that the data warehouse facilitates most comparative analyses and data mining operations, with the result that access to the primary data store can be avoided except for particularly novel or complex queries. We assume that every location with a primary data store has an associated data warehouse. However, in practice it is likely that while there will be many primary stores, there may be only a few, centralised data warehouses. This is due to the smaller data volume stored in the data warehouse and the greater administrative effort involved with maintaining such a facility. It is essential that the P-found system supports direct access to the primary data as novel analyses of this data may lead to important scientific discoveries. Therefore, we are investigating the use of grid technologies to solve two aspects of the P-found system:

1. *Distributed data management*: functionality to federate a number of distributed data warehouses so that they may be accessed and managed as if they were a single resource.
2. *Distributed analysis*: functionality for shipping or transferring analysis programs to the distributed primary data stores so that analysis takes place at the data store and thus by-passes the need to transfer large data volumes.

This paper is structured as follows: we briefly describe the DataMiningGrid, and then give more detailed descriptions of how its associated technology can be used to provide solutions to the distributed data management and analysis aspects of P-found. We then discuss the security implications involved with allowing scientists to distribute and execute potentially any kind of program over the P-found infrastructure. Before concluding the paper we outline our on-going and future development plans.

2 Grid Solutions for P-Found

The DataMiningGrid project took P-found's technical requirements into its design, and initial solutions for the problems associated with P-found were developed by the DataMiningGrid. The DataMiningGrid is described in detail elsewhere [5]. In summary, the DataMiningGrid was built using the Globus Toolkit [2], which contains various data management tools, including GridFTP for file transfers and OGSA-DAI version WSRF-2.2 [7], which was used to provide data management functionality for databases. A test-bed was created, based at three sites in the UK, Slovenia and Germany and at each site Globus components were used to interface with local Condor clusters. Condor is an opportunistic, high throughput computing system that uses cycle-stealing mechanisms to create a cluster out of idle computing resources [8].

The P-found system used with the DataMiningGrid is a prototype, and consists of just one installation [1]. This prototype consists of both the file repository and the data warehouse, which is implemented using the Oracle 10g database system. With just one installation it was not possible to fully demonstrate the distributed aspects. However, general grid data management solutions were developed and tested with other distributed database applications in the DataMiningGrid and these are fully applicable to the P-found system. More technical details of components comprising the P-found system are given in Sec. 2.2.

2.1 Data Mining Distributed Data Warehouses

Here we show how OGSA-DAI can be used to federate P-found data warehouses and calculate a summary of the data sets accessed. This summary is then used by different data preprocessing operations, resulting in formatted data sets that are ready for processing with data mining algorithms.

Figure 1 shows how the data summary was developed with the OGSA-DAI APIs. The process begins when the user selects a data service and defines an SQL query to be executed against one of the service's data resources. Then:

1. The client sends the query to the data service and after execution the meta-data associated with the query is retrieved.
2. The meta-data returned by the query is processed by the client, which then sends a request to the data service to create a temporary database table according to the meta-data values.

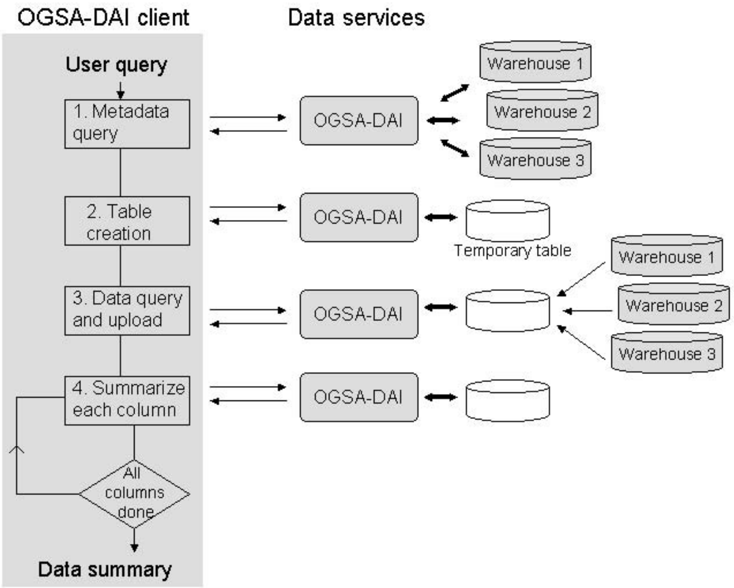

Fig. 1. A P-found application using the OGSA-DAI client API and OGSA-DAI services to calculate a data summary

3. The client retrieves the query data from the data service and loads them into the temporary table.
4. Now it is possible to calculate statistical values of this data set, with the particular values to be calculated depending on the application. Typically, for numeric data the average, standard deviation, maximum, minimum and variance are calculated, while for character data this involves finding all the distinct values in the column. The client performs this by issuing a series of requests, consisting of SQL queries, to summarize the values in each column. The resulting data set summary is stored at the client for use in subsequent requests or routines.

Using software from the DataMiningGrid, OGSA-DAI clients may be used to integrate distributed P-found databases: the clients first access data from a number of repositories, store the data in a temporary table, and then calculate a data summary. In the DataMiningGrid these data were streamed from the temporary table and a number of data filtering operations and data transformations were applied before the data were delivered to a file. The process is shown in Fig. 2:

1. A data summary is calculated, based on data from multiple databases that have already been uploaded to a temporary table. All data are now selected from the table.

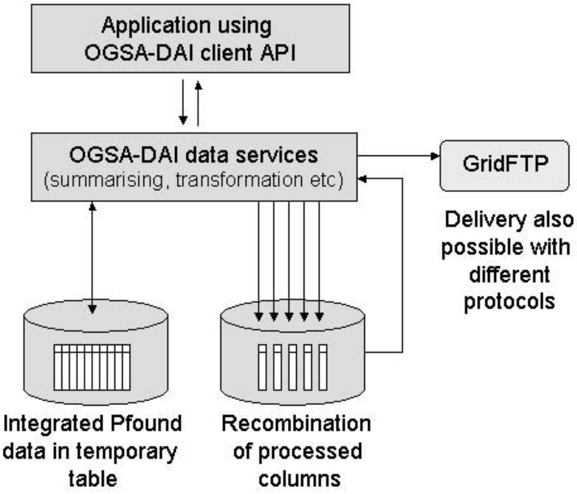
Application using OGSA-DAI client API
OGSA-DAI data services (summarising, transformation etc)
GridFTP
Delivery also possible with different protocols
Integrated Pfound data in temporary table
Recombination of processed columns

Fig. 2. A P-found application performing data transformation on values generated by a data summary

2. The OGSA-DAI projection operations are used to extract table columns as data are streamed from their source (i.e. the temporary table) to their destination (in this case another temporary database table). As the data is streamed at the data service, columns are extracted and the data in each column is transformed, for example by aggregating values or discretizing continuous numeric values. In this case returning the data to a database table is convenient, as it allows the process to be repeated with a series of different transformations.
3. Now the transformed data is retrieved from the table.
4. Additional functionality may now be applied to convert this transformed data into different formats.
5. In the final step the data are deposited in one or more files on a GridFTP server. These are then available for data mining.

The DataMiningGrid provided some enhancements to OGSA-DAI, for example to perform data mining preprocessing operations such as cross-validation, data transformation, and data formatting i.e. formatting data according to Weka's ARFF structure [9], or the format of the CAREN association algorithm [10]. Such tasks usually required the data summary calculated earlier, and could be used to deliver data transformed and formatted to servers ready for processing with data mining algorithms.

OGSA-DAI supports a variety of data delivery operations, including delivery using GridFTP, HTTP, email or simply returning results from the web service to the client using SOAP messages. There is therefore only a small overhead required to use the clients we have presented here outside of the DataMiningGrid infrastructure.

2.2 Transferring Data Analysis Programs to Distributed File Repositories

Here we present a simplified version of the process for shipping algorithms to multiple, distributed data locations, as implemented by the DataMiningGrid. The main difference is that the DataMiningGrid developed a specialised Resource Broker Service that interfaced with the WS-GRAM utilities from the Globus Toolkit 4 to submit, monitor and cancel jobs on computational resources and local scheduling systems, and to coordinate the input and output staging of data. This Resource Broker Service could ship both data and programs to any location on the grid. It could handle the heterogeneity of grid resources and it provided additional functionality to support data mining applications [5]. Here we assume that the data is never moved from the primary storage repositories, and that there are dedicated compute resources available which are local to the storage repositories.

The system we propose for P-found is shown in Fig. 3 and uses the following components:

1. A client application to analyse, process or data mine the simulations stored in the primary repositories.
2. A system registry: this is a centralised component, listing all the analysis software, simulation repositories, and data warehouses available in the system. It would also contain information to tell either human users or software how to interact with those components.
3. A software repository: there can be one or more of these, and it is here that analysis programs and their associated libraries are stored. This does not require a sophisticated solution: executable files and libraries can be simply stored on a file system and accessed using GridFTP. The software repository could be merged with the client application.
4. Multiple, distributed P-found installations, each installation containing both a primary data store for the simulation files and a secondary data warehouse for computed simulation properties. Compute resources will be available at each installation: these may be a few dedicated processors or a local Condor cluster, connected to the data repositories via a networked file system or high-speed network. There is also a GridFTP server at each installation for performing fast file transfers.

The process for shipping analysis programs to data is shown in Fig. 3 and is described here. Note that there may be a preceding step, not shown in the figure, in which the client application performs a distributed query on the P-found data warehouses, similar to that described in Sec. 2.1. The data warehouses contain all technical information about the simulations i.e. the name of the protein simulated, the techniques used, and so on. In the preceding step this information is queried in order to find a set of interesting simulation files. Then:

1. The client application queries the registry to discover available analysis software. The result will be a list of programs, with a URI giving their location,

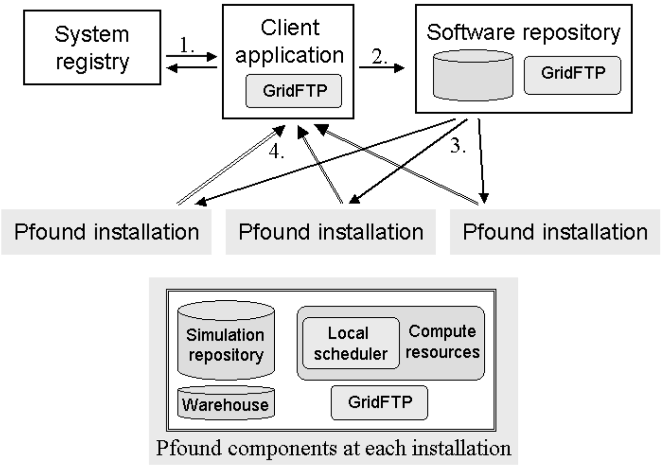

Fig. 3. Shipping programs to the primary P-found data repositories

along with the technical information required to execute these programs, such as how much CPU and memory they need, operating system preferences, etc. This technical information is very important for grid systems which are composed of heterogeneous resources.

2. A program is selected, the correct software repository is contacted, and the program is transferred to the P-found repositories that contain the simulations of interest.
3. Once transferred to the repositories, the programs are executed. This may be quite a complicated procedure and may require a sophisticated component such as the DataMiningGrid's Resource Broker Service, as mentioned earlier in this section. For example, such a component would ensure programs are executed in the correct environment and on machines with the correct specification.
4. Once the analysis is completed, the results are transferred using GridFTP, either back to the client application or to some other location e.g. another P-found repository where they are used in subsequent calculations.

In this scenario there is no explicit support for workflows, unless they are constructed using a scripting language (e.g. Perl or Python). This should be adequate for P-found as many scientists use Perl and Python to coordinate analysis tasks. It does mean, however, that all the executables and associated libraries should be transferred to the primary repositories along with the scripts.

3 Discussion

Grid data management technology is maturing and is able to support relatively sophisticated data analysis scenarios, as has been demonstrated by projects such as Gridminer [11] and the KnowledgeGrid [12], as well as the DataMiningGrid

[5]. Unfortunately, the shipping of programs to data locations has not been investigated in as much detail, even though this is an important approach for distributed data sets that cannot be transferred to centralised processing facilities (due to either their large volume or their confidential nature). There is therefore little “off-the-shelf” software that we can use to provide this functionality.

Allowing users to freely execute arbitrary code on machines outside of their own administrative domain does raise some serious security issues. For example, there is great potential for malicious users to create havoc on the P-found systems by destroying the data storage facilities. To persuade new groups to participate in the P-found system, it is essential for them to be fully confident that their data will remain secure. The Globus Toolkit provides tools for security based on public key cryptography [2], which is sufficient for most purposes. This may also be reinforced by only allowing software that is completely trustworthy to be registered on the system and transferred for execution on different administrative domains. However, this does limit scientists from performing arbitrary analysis on the repositories, one important aim of the P-found system. In the end, the security framework needs to be agreed on by the community, and it must be flexible enough to persuade the most cautious users to participate.

As an alternative to using executables and grid scheduling mechanism, we investigated the Dynasoar framework [13]. Dynasoar provides an architecture for dynamically deploying Web services remotely on a grid or the Internet. A potential use, motivating its design, was to move Web services that access data and perform analysis on it, closer to the data storage facilities, which fits with the requirements of the P-found system. Web services require that an interface is specified, describing the functionality of the Web service and how it should be used – this approach makes a little harder to hide malicious software that is able to damage the system. However, while the Dynasoar has some nice features, it is still a prototype with limited functionality. Moreover, as most scientists from the molecular simulation community typically use scripts (Perl or Python) or executables (C/C++ and Fortran) for analysis and are therefore not familiar with Web service technology, we decided that this approach was currently unsuitable for this project.

The BioSimGrid [14] is a project with similar aims to P-found. It enables data sharing of molecular simulation data between different research groups and universities. A Python scripting environment is used to pre-process, deposit and retrieve data, and a general purpose Python analysis toolkit is available for manipulating deposited data. The authors do not discuss security issues, presumably because the system was designed for use with six UK universities – a relatively small and localised community where all users may easily be held accountable for their actions. This is different to P-found, which we plan to open up to the wider international community.

4 Future Work

P-found is still under development, and project partners are investigating additional approaches to implement the final system. These include using grid services available through the EGEE project (Enabling Grids for E-sciencE), and comparing the approach given here (based on multiple, distributed data warehouses) with a single, centralised data warehouse and multiple storage elements. The P-found partners plan to develop these three prototypes and make them available to the user community. Feedback from the user community will be essential in choosing the final design.

5 Conclusions

To further scientific discoveries there is a need to enable scientists to share and analyse protein folding and unfolding simulations. The P-found protein folding and unfolding simulation repository is designed to fulfill this need, and right now consists of two data storage components: a repository of unprocessed simulation data and a data warehouse containing detailed information regarding how the simulations were generated as well summaries of the simulations in the form of calculated local and global physical properties.

While a centralised version of P-found (`www.p-found.org`) is currently available, P-found is ultimately envisioned as a distributed system due to the massive data volumes involved. Here we have described how grid technologies can be used to realise that vision. We have demonstrated how OGSA-DAI, a grid data management tool can be used to federate distributed P-found data warehouses and prepare data for analysis and data mining tasks, we have also presented a mechanism to allow scientists to ship arbitrary programs to the distributed simulation repositories in order to process this primary data in novel and sophisticated ways. This is an important aspect of P-found, with the potential to generate new scientific discoveries as protein folding and unfolding data sets become more available.

Acknowledgements. This work was supported in part by the European Commission FP6 grants No. 004475 (the DataMiningGrid[1] project), and No. 033437 (the Chemomentum[2] project) to WD, MS and VO; and in part by grant GRID/GRI/81809/2006 (FCT and FEDER, Portugal) to RMMB, and Doctoral Fellowship SFRH/BD/16888/2004 to CGS.

References

1. Silva, C.G., Ostropytsky, V., Loureiro-Ferreira, N., et al.: P-found: The Protein Folding and Unfolding Simulation Repository. In: Proc. 2006 IEEE Symp. on Computational Intelligence in Bioinformatics and Computational Biology, pp. 101–108 (2006)
2. Foster, I.T.: Globus Toolkit Version 4: Software for Service-Oriented systems. J. Comput. Sci. Technol. 21, 513–520 (2006)

[1] `http://www.DataMiningGrid.org`

[2] `http://www.chemomentum.org/`

3. Finkelstein, A., Gryce, C., Lewis-Bowen, J.: Relating Requirements and Architectures: A Study of Data-Grids. J. Grid Comput. 2, 207–222 (2004)
4. Laure, E., Stockinger, H., Stockinger, K.: Performance Engineering in Data Grids. Concurrency - Practice and Experience 17, 171–191 (2005)
5. Stankovski, V., Swain, M., Kravtsov, V., et al.: Grid-Enabling Data Mining Applications with DataMiningGrid: An Architectural Perspective. Future Gener. Comput. Syst. 24, 259–279 (2008)
6. Swain, M., Hong, N.P.C.: Data Preprocessing using OGSA-DAI. In: Dubitzky, W. (ed.) Data Mining Techniques in Grid Computing Environments, Wiley, Chichester (in press)
7. Antonioletti, M., Atkinson, M., Baxter, R., et al.: The Design and Implementation of Grid Database Services in OGSA-DAI. Concurr. Comput.: Pract. Exper. 17, 357–376 (2005)
8. Litzkow, M., Livny, M.: Experience with the Condor Distributed Batch System. In: Proc. IEEE Workshop on Experimental Distributed Systems, pp. 97–100 (1990)
9. Witten, I.H., Frank, E.: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
10. Azevedo, P.J., Silva, C.G., Rodrigues, J.R., Loureiro-Ferreira, N., Brito, R.M.M.: Detection of Hydrophobic Clusters in Molecular Dynamics Protein Unfolding Simulations Using Association Rules. In: Oliveira, J.L., Maojo, V., Martín-Sánchez, F., Pereira, A.S. (eds.) ISBMDA 2005. LNCS (LNBI), vol. 3745, pp. 329–337. Springer, Heidelberg (2005)
11. Fiser, B., Onan, U., Elsayed, I., Brezany, P., Tjoa, A.: On-Line Analytical Processing on Large Databases Managed by Computational Grids. In: Proc. 15th Int. Workshop on Database and Expert Systems Applications (2004)
12. Congiusta, A., Talia, D., Trunfio, P.: Distributed Data Mining Services Leveraging WSRF. Future Gener. Comput. Syst. 23, 34–41 (2007)
13. Watson, P., Fowler, C.P., Kubicek, C., et al.: Dynamically Deploying Web Services on a Grid using Dynasoar. In: Proc. 9th IEEE Int. Symp. on Object and Component-Oriented Real-Time Distributed Computing (ISORC 2006), Gyeongju, Korea, pp. 151–158. IEEE Computer Society Press, Los Alamitos (2006)
14. Ng, M.H., Johnston, S., Wu, B., et al.: BioSimGrid: Grid-Enabled Biomolecular Simulation Data Storage and Analysis. Future Gener. Comput. Syst. 22, 657–664 (2006)