Adaptation of a Multi-Resolution Docking Bioinformatics Application to the Grid

J.I. Garzón
Centro de Investigaciones Biológicas-CSIC
28040 Madrid (Spain)
garzon@cib.csic.es
E. Huedo
Facultad de Informática, Universidad Complutense
28040 Madrid (Spain)
ehuedo@fdi.ucm.es
R.S. Montero
Facultad de Informática, Universidad Complutense
28040 Madrid (Spain)
rubensm@dacya.ucm.es
I.M. Llorente
Facultad de Informática, Universidad Complutense
28040 Madrid (Spain)
llorente@dacya.ucm.es
P. Chacón
Centro de Investigaciones Biológicas-CSIC
28040 Madrid (Spain)
pablo@cib.csic.es

Abstract— **Rigid body fitting is the common way to interpret the 3D information contained in a electron microscopy (3DEM) low resolution density map in terms of its available 3D atomic resolution structural components. This fitting process, termed multi-resolution docking, consists in localizing atomic resolution structures into the 3D EM map by means of an exhaustive search of all possible relative rotations and translations.**

In addition to the cost of a single search, the necessity to carry out multiple searches with many different structures makes this problem appropriate for high performance computing (HPC). The Grid Computing paradigm provides such computing power for this type of resource-intensive scientific applications allowing the access to large resource pools conformed from shared assets of different centres or administration entities.

Here, we present an efficient Grid approach for performing the multi-resolution docking searches. This approach has been designed over the GridWay Metascheduler. We show the suitability of the adaptation of the problem to the Grid paradigm. Results showing the high efficiency achieved are discussed together with the analysis of the performance obtained over the Grid testbed employed.

Index Terms: Multi-resolution docking; Grid computing; Grid application.

I. Introduction

Detailed knowledge of macromolecular structure is essential for the understanding of how the cellular machines work. Despite of the explosive growth of research in structural biology in last decades, the atomic resolution access to large macromolecular complexes involved in the main cellular functions is still rather limited. Electron microscopy (EM) techniques are able to capture such large macromolecules in diverse near-physiological conditions [1]. Unfortunately, the resolution that can be obtained by EM is limited to low medium resolutions (10-20Å). However, it is possible to achieve the atomic detail of the structure by localizing available atomic resolution components into the 3D EM low resolution map of a macromolecule. This is a complicated jigsaw puzzle in which the low resolution 3D EM density map of a macromolecule acts as a fuzzy frame to guide the assemblage of interlocking atomic-resolution pieces. When complete, this jigsaw puzzle produces a near-atomic detail picture of the entire macromolecule. Thus, by solving this puzzle we can have access to a better understanding of the inner

Based on "Grid Multi-Resolution Docking", by Garzon, J.I., Huedo, E., Montero, R.S., et al. which appeared in the proceedings of the Parallel, Distributed and Network-Based Processing, 2007, PDP 07, 15th Euromicro International Conference, Naples, Italy, Feb. 2007. ©2007. This research was supported by Ministerio de Ciencia y Tecnología through the research grants TIC2003-01321 and BPU2004-01282, and also by the Fundación BBVA. This work makes use of results produced by the Enabling Grids for E-sciencE project, a project co-funded by the European Commission (under contract number INFSO-RI-031688) through the Sixth Framework Programme. EGEE brings together 91 partners in 32 countries to provide a seamless Grid infrastructure available to the European research community 24 hours a day. Full information is available at http://www.eu-egee.org.

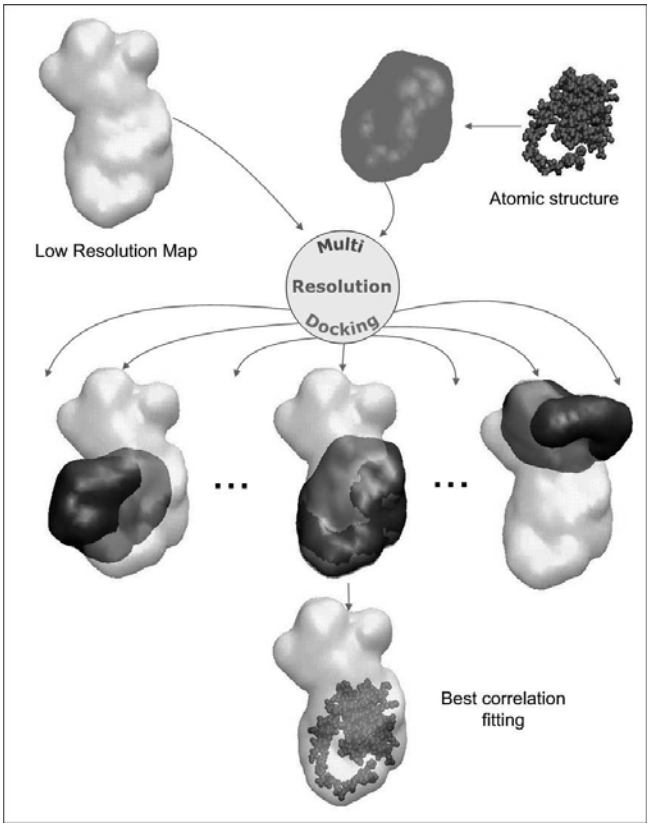

Figure 1. Basic Multi-Resolution docking process.

workings of the central actors in the principal cellular processes.

This puzzle, termed multi-resolution docking, can be reduced to register geometrically two 3D electron density maps: the experimental EM map with a simulated map obtained by lowering the resolution of the atomic structure to be docked (for reviews see [2], [3], [4]). In practical terms, the multi-resolution docking process consists in estimating the 3D rotation matrix and the translational vector that maximizes the density overlap, *i.e.* maximizes a simple density correlation function (scalar product of the densities). This correlation is typically calculated between the target experimental EM map and the simulated probe map (Figure 1). To explore all possible solutions, the docking is performed by a full 6D rigid-body search. This exhaustive exploration is needed to avoid any missing valid registration. Note that we are confronting a non-trivial problem and several docking alternative poses can be obtained because of the resolution differences, the EM low signal to noise ratio or small changes between atomic and EM structures (eg. missing regions, disorder or conformational changes).

The atomic structures of the components to be docked can be obtained from a variety of sources. In many cases, X-ray crystallography can provide the atomic structure of small components. However crystallization of the components is a difficult task, and in many times impossible. In these cases, homology modelling strategies can be applied to develop atomic models based on known structures of homologous proteins (i.e evolutionary related proteins). It has been shown that comparative modeling provides useful model structures for fitting into EM [5]. Thus, it is frequent that the simple docking puzzle illustrated in figure 1 becomes more complex since multiple atomic models need to be fitted.

In this context, taking into account that the exhaustive docking is a highly computational demanding process together with the fact that the atomic structures to be aligned are of the order of few thousands, the use of both efficient algorithms and suitable computing platforms is essential to get correct and fast multi-resolution docking solutions.

Problems like this one have permitted the evolution to a new paradigm called Grid Computing. The ability to have applications draw computing power from a global resource pool to achieve high performance has become a new challenge for distributed-computing and Internet technologies. Several research centres share their computing assets in grids, which dramatically increase the number of accessible processing and storage resources. Grids enable efficient and secure sharing of a large variety of computational resources scattered across several administrative domains [6]. This new computational infrastructure provides a promising platform to carry out loosely coupled, high throughput computing applications. In general, these applications comprise the execution of a high number of tasks, each of which performs a given calculation over a subset of input values.

However, despite the rather simple structure of these applications, their efficient execution on computational Grids involves challenging issues [7], mainly because of the nature of the Grid itself, namely: dynamic resource availability and load, heterogeneity and a high fault rate. Among the crucial elements of a computational grid, the Metascheduler is gathering most attention as a way to meet the challenging needs of several application domains. The term Metascheduler can be defined as a grid middleware that discovers, evaluates and allocates resources for grid jobs by coordinating activities between multiple heterogeneous schedulers that operate at local or cluster level [8]. In general, the scheduling process includes the following phases: resource discovery and selection, job preparation, submission, monitoring, migration and termination [9].

Although several philosophies for the Grid and implementations of the Metascheduler can be found, here we have employed the GridWay Metascheduler [10] due to the fact that it provides a fast, easy and adaptive mechanisms for the use of Grid resources.

In this work, we combine a novel rigid-body registration docking tool based on spherical harmonics, termed FRM (Fast Rotational Matching), with the computing power provided by Grid infrastructures and the employ of the GridWay Metascheduler [11]. We analyze the adaptation to the Grid of a multi-resolution docking application. In particular, we consider a highly heterogeneous Grid infrastructure, which comprises resources

from the EGEE [1](Enabling Grids for E-sciencE) production testbed. In this way, we will asses the suitability of this Grid environment to execute this large-scale Bioinformatics application.

The rest of this paper is organized as follows. In Section II, we briefly describe the multi-resolution docking problem considered. Characteristics of the Grid paradigm used in this research and the adjustments introduced in the application to adapt its execution to the Grid are introduced in Section III. The experimental results obtained are then analyzed in Section IV. Finally, Section V presents a discussion of our results and hints of our future work.

II. Problem Description

Here, as a benchmark test case, we centre our study in a concrete multi-resolution docking case where the atomic structure to dock is unknown. In this case, one can appeal to homology modelling Bioinformatics tools which can give us an extensive set of possible atomic models. Homology modelling [12] is based on the reasonable assumption that two proteins that have a good similarity in their sequence of amino acids will share very similar structures. Predictions of the structure of a target protein can be done finding one or more related proteins whose structure are known, aligning the target sequence to the sequences of the related proteins and building structure models based on the previous sequence alignments. The amount of related proteins and possible sequence aligning can be very wide, so many different models can be constructed. Also different homology model algorithms can be used increasing the number of possible docking candidates.

In summary our computational challenging experiment will consist in performing an exhaustive docking search over a big set of homology models and then select those that better fit into the EM map, *i.e.* select those with higher density correlations.

The computational cost of this problem is due to:

- the exhaustive docking of an atomic model into a density map by a 6D (3 translational + 3 rotational) search is by itself a high computing demanding process.
- This docking operation must be repeated over a large collection of different models obtained from modelling techniques.

The combination of these two aspects can increase significantly the computing demand, making the docking process even unapproachable. Therefore, both aspects must be tackled in an efficient way. First, several methods have been developed to speed up the exhaustive search for computing correlations [2], [3]. If we use the standard multi-resolution docking tool COLORES [13] which accelerates the translational search by the use of the convolution theorem and fast Fourier transform, a single docking can take from many minutes to several hours. Here we employ a fast approach based on spherical harmonics, termed FRM (Fast Rotational Matching), detailed elsewhere [14], [4]. Briefly, FRM accelerates the rotational search by expressing the density objects as spherical harmonics representations. This harmonic representation together with a convenient representation of the rotational group permits a fast computation of the rotational correlation function. The translational space is simply uniformly scanned. Previous work has been done to develop an optimized version of this method that is able to reduce the docking time to few minutes [4]. For example, a rotational search with sampling step of 5.6, which means more than 130 000 rotations will be explored, will take 4 minutes for a typical size EM map in a standard linux PC box. Thus, this method offers an adaptable and fine rotational screening.

Second, to address the multiple executions, we employ Grid technology. This technology has been proved to be an efficient platform to perform High Throughput Computing (HTC) applications where it is comprised the execution of a set of independent tasks each of which performs the same calculation over a different set of data. Next section introduces the basics of the Grid paradigm and its adaptation to our application.

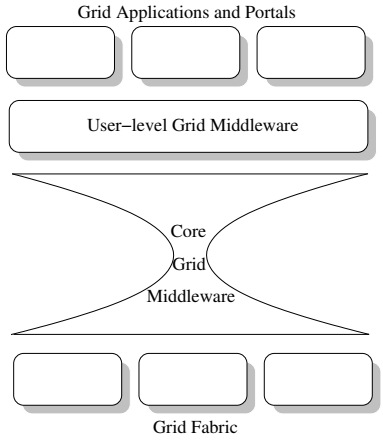

Figure 2. Grid layers.

III. Grid Platform and adaptation of the Multi-docking Algorithm

A. Grid and GridWay characteristics

A Grid infrastructure is usually decomposed into the following layers [15]: Grid applications and portals; user-level Grid middleware; core Grid middleware; and Grid fabric. The two internal layers are called *the middleware*, since they connect applications with resources (or Grid fabric). These layers should be separate and independent, communicated with a limited and well defined set of interfaces and protocols. This is especially important for the user and core Grid middleware. This way, clients have access to a wide range of resources provided through a

[1]http://www.eu-egee.org

limited and standardized set of protocols and interfaces, *e.g.* those provided by Globus [16], as core Grid middleware (Figure 2).

User-level middleware, such as GridWay, is required in the client side to make it easier and more efficient the execution of applications. GridWay works on top of Globus services, performing job execution management and resource brokering, allowing unattended, reliable, and efficient execution of jobs, array jobs, or complex jobs on heterogeneous, dynamic and *loosely-coupled* Grids formed by Globus resources. GridWay offers several advantages that make it suitable for performing efficient executions of computing demanding applications like the one described above. GridWay allows array jobs and jobs with dependencies. Also, GridWay offers C and Java implementations of the DRMAA Application Programming Interface, which is a Open Grid Forum (OGF) standard [17].

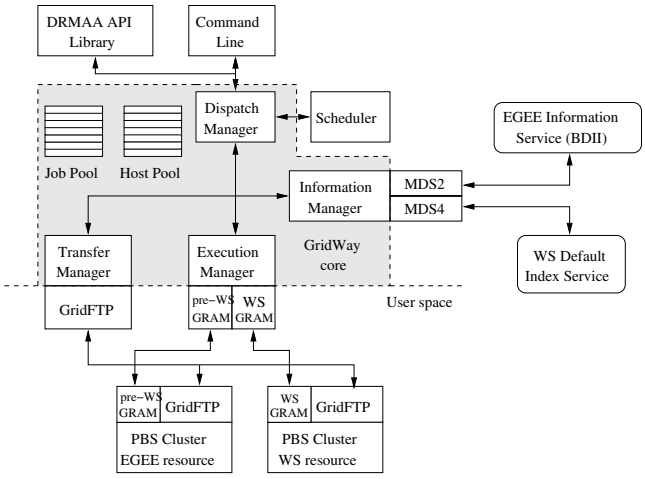

Figure 3. GridWay architecture and its interaction with Grid Services.

Figure 3 shows the modular architecture of GridWay [18]. It is conformed by the GridWay Daemon (GWD) and different Middleware Access Drivers (MADs) to access different Grid services (information, execution and transfer). GridWay can be installed to implement several Grid architectures, namely: enterprise grids, partner grids (like the EGEE infrastructure used here) and utility grids [19].

GridWay supports dynamic scheduling, providing a way to filter and evaluate resources based on dynamic attributes, by means of different policies. These dynamic attributes are obtained from different Grid information services. With GridWay, an application can take decisions about resource selection as its execution evolves by modifying its requirement and rank expressions. Also, it takes count of the suspension time in remote batch systems and requests a migration when it exceeds a given threshold. Moreover, jobs are submitted together with a light-weight self monitoring system. The job will migrate when it doesn't receive as much CPU as the user expected.

Regarding fault tolerance, GridWay detects job cancellation (when the job exit code is not specified), remote system crash and network disconnection (both when the polling of the job fails). In all of these cases, GridWay requests a migration for the job [20]. With GridWay, user-level checkpointing or architecture independent restart files managed by the programmer can be implemented. Migration is implemented by restarting the job on the new candidate host. If the checkpointing files are not provided, the job should be restarted from the beginning. These checkpoints are periodically retrieved to the client machine or a checkpoint server. Also the system running the scheduler could fail. GridWay persistently saves its state in order to recover or restart the jobs when the system is restarted.

B. Implementation of the Multi-docking Algorithm

We focused our work on the challenging docking case where multiple atomic resolution structures or models must be localized into a given target EM density map. In this case, each model can be independently docked and the searches can be performed in independent jobs. Thus, by using the Grid framework, all of the dockings (jobs) can run concurrently making use of different computing resources. After all the jobs have been fulfilled the outputs must be merged and sorted to bring out the best fitting models. With GridWay this scheme can be follow without a complex design process. With the tools provided by GridWay for launching jobs in a Grid environment transparently to the user, simple sequential executables can be employed. Following this scheme the Grid version of our multidocking tool was divided in the next three phases (see also Figure 4).

1) Pre-computation Phase: All the jobs perform several calculations that only depend on the common target EM density map. To save computing time these common calculations can be pre-computed. The pre-computations are related to:

- The translational search space limits. The shape and dimensions of the target density map constrain the possible positions that any atomic structure could occupy. Based on these geometric properties a mask of valid translational positions can be pre-established for all the 6D searches.
- FRM pre-computations. Since the EM map is always fixed, several calculations of the FRM algorithm can be also pre-computed.

These operations are locally performed to generate all the pre-computed data from the density map.

2) Correlation Phase: Independent jobs are launched through the Grid environment. This can be performed in an easy way using the jobs array launch capacity provided by GridWay. Each job performs the docking between the density map and a different assigned atomic model. To this end, a specific script is called which takes the binary docking tool (FRM) and two input files. These input files correspond to the pre-computation files and the atomic coordinates of the model to be docked. The final output of this phase will be a list of possible poses (position and rotations) sorted by higher correlation values. Using

TABLE I.
SUMMARY OF THE GRID RESOURCE'S CHARACTERISTICS.

Resource Name	Architecture	Mhz	Nodes	DRMS	Location
gridgate.cs.tcd.ie	i686	2600	54	jobmanager-pbs	Ireland
lcg02.ciemat.es	i686	1001	202	jobmanager-lcgpbs	Spain
ce01.ariagni.hellasgrid.gr	i686	3400	116	jobmanager-lcgpbs	Greece
ce02.tier2.hep.manchester.ac.uk	i686	2800	836	jobmanager-lcgpbs	U.K.
marseillece01.mrs.grid.cnrs.fr	i686	2400	200	jobmanager-pbs	France
t2ce02.physics.ox.ac.uk	i686	2800	74	jobmanager-lcgpbs	U.K.
ce.epcc.ed.ac.uk	i686	2000	7	jobmanager-lcgpbs	U.K.

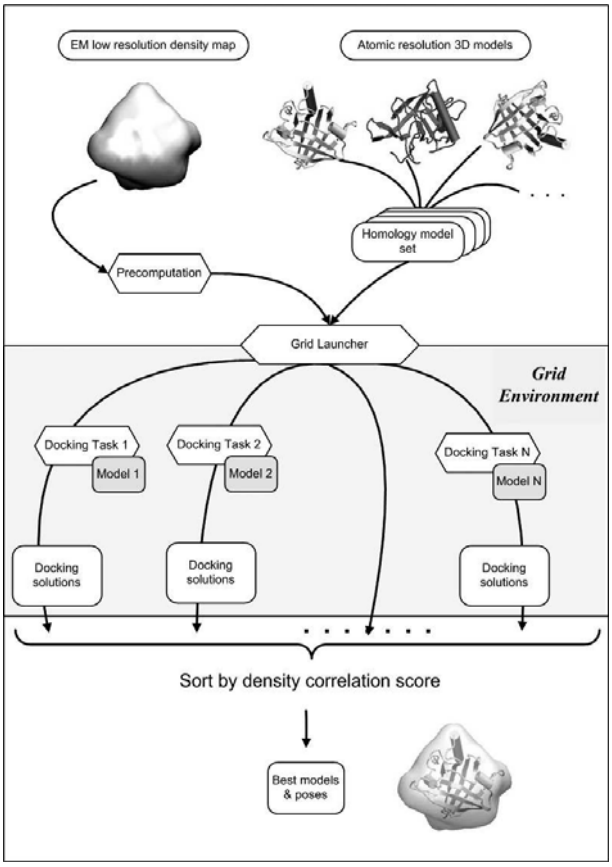

Figure 4. Scheme of the Grid Multi-Docking application.

GridWay allows synchronizing the endings of all the jobs to combine their results.

3) Combination Phase: The best fittings of each atomic model are merged in a single file which is subsequently sorted by the correlation value. This task is performed by other script subroutine. At the end, the first solutions will correspond to the best fitting models that can be used to correctly locate the structure inside the EM map.

C. Grid Infrastructure

The grid infrastructure used for testing the new application is part of the Biomed Virtual Organization of the EGEE project (See Table I). In EGEE, the Globus behaviour has been slightly modify, although it does not loose its main protocols and interfaces, so GridWay, which relies on Globus services, can be used in a standard way. The whole infrastructure is composed by 7 sites localized in different European countries and 1489 processors. However, the accessibility to the sites changes dynamically and there are other processes that contend for their usage so the performance of an application can be variable for different executions. Also, to avoid the saturation of the infrastructure, limitations in the job launching are established. In this way the maximum number of jobs launched at the same time by an user is limited to 15 and no more than 30 jobs can run concurrently in the system with a maximum of 10jobs per host. So, from the user's point of view, the Grid infrastructure apparently has no more than 30 processors available.

TABLE II.
CORRELATION VALUES OF THE BEST FITTING RESULTS.

Model	Normalized correlation
Model 0	0.9947
Model 6	0.9504
Model 3	0.9493
Model 291	0.9492
Model 288	0.9492
Model 298	0.9490
Model 21	0.9485
Model 241	0.9478
Model 263	0.9476
Model 260	0.9474

IV. RESULT ANALYSIS

As an illustrative example, here we show the results of a docking of 300 atomic homology models into a single simulated map of the protein rodent urinary (PDB entry 1mup). By simplicity, we restrict the experiments to only 300 homology models but in real applications the number of models could be of the order of few thousands. The resolution of this map was of 12Å. Gaussian noise was added to this map for simulating real experimental conditions. MODELLER [12] was used to generate the alternative comparative models from distant homologous if the 1mup protein (<30% of sequence identity). These homology models were fitted by our FRM docking tool through the Grid environment. For validation purposes, we included in the data set the original protein structure

used to generate the map. Consequently, this real structure is expected to be the model with the highest correlation value.

In order to analyze the efficiency of the application in the dynamic Grid environment, the execution of the docking process over the 300 models was repeated ten times. Finally, for comparative purposes, the operation for all the models was performed in a single job over a local AMD Sempron(tm) Processor with 3207Mhz.

A. Validation and Efficiency of the Solution

In Table II, the scoring correlation of the ten best fitting models is shown. As expected, the Model 0 which corresponds to the original structure of the target EM map is on the top of the list (*i.e.* perfect fit). The next model (model 6) corresponds to the best homology model obtained. As it can be seen in figure 5B, the structure of this model (light grey) fits very well into the EM map. This correspondence can be also observed by comparing the best model obtained with the original atomic structure (dark grey. Figure 5C). The great resemblance of both structures validates the developed Grid base multi-resolution docking approach. In real world, the similarity of the best fitting comparative model found ensures a proper atomic resolution interpretation of the EM even if the original underlying atomic structure is not available. It is important to notice that in the entire test carried out we obtain the same results, demonstrating the robustness of the fitting algorithm used.

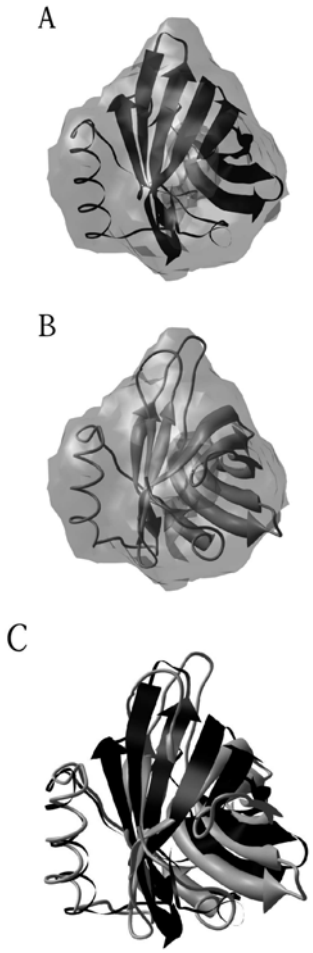

Figure 5. Docking results. Panel A) Original atomic structure (ribbons diagram) and its corresponding target EM density map (transparent isosurface). Panel B) The best docking structure obtained is superposed into the target EM density map. This structure corresponds to model 6. Panel C) Structural comparison between the best fitting model (light ribbons) and the original structure underlying the target EM map (dark ribbons). Note the high structural similarity.

B. Performance Analysis of the Grid environment

Figure 6 shows the average execution time for the tests performed together with the time spent in the single sequential execution for all the models in a local processor. As can be observed, the Grid implementation employed to fit all the models is 20 times faster than the sequential implementation, reducing from 30 hours to just 1.5 hours the time necessary to obtain the results. Figure 7 presents the average dynamic throughput, defined as the number of jobs completed per second:

$$r(t) = \frac{N(t)}{t} \tag{1}$$

Based on this parameter we can characterize the employed Grid environment [11]. The asymptotic performance (r_x) defined as the maximum rate of performance in jobs per second is approximately 0.055 in our tests. The half-performance length ($n_{1/2}$) defined as the number of jobs required to obtain half of the asymptotic performance is around 12. These values are useful to create an idealized representation of the Grid environment, so it can be determined that the performance of the Grid environment is equivalent to the one obtained by a homogeneous array of 24 processors ($2n_{1/2}$) with an execution time per job of 436 seconds ($2n_{1/2}/r_x$). Accordingly, it can be inferred that the Grid environment's performance (in terms of throughput) will stabilize if more than 24 jobs are launched, being this value the saturation point of the system. New tests have been done executing the application with different numbers of models (and a different number of jobs) to prove this assertion and their results are shown in figure 8. As expected, time spent in the overall executions grows faster when there are more than 24 jobs, while with less jobs the time keeps around 600 seconds. This can be checked in the dynamic throughput curve as well, where a valley is found over 24 jobs in where the curve is prone to stabilization. As we commented before, the limitations in the number of jobs concurrently running through the testbed (30 jobs in the full testbed, 10 per each host) produce this apparent low number of processors.

On the other hand, the half performance length ($n_{1/2}$) provides a quantitative measure of the heterogeneity on a Grid environment. The degree of heterogeneity (v) can be defined as:

$$v(t) = \frac{2n_{1/2}}{N} \tag{2}$$

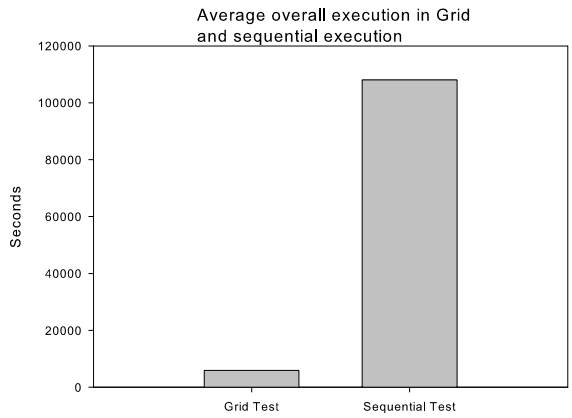

Figure 6. Average execution time of the application over the Grid and over a single processor.

Where N is the total number of processors of the Grid environment. The degree of heterogeneity varies from $v = 1$ when the environment is homogeneous (all the processes present the same behaviour) to $v \approx 0$ when there is a great degree of heterogeneity (performances of the different processes vary in high degree). In the last case, the apparent number of processors of the Grid environment, from the application's point of view, will be lower than the total number of processors (N). In our experiments, taking into account that the real accessible number of concurrent processors is 30, the degree of heterogeneity (v) obtained is approximately 0.8, showing that the system has an small degree of heterogeneity.

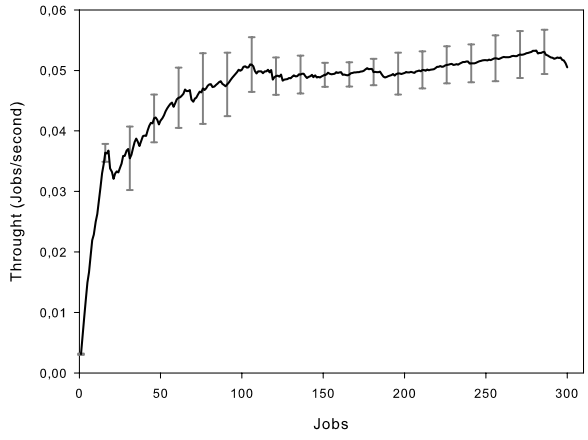

Figure 7. Average Grid throughput (jobs per second), in the execution of the multi-resolution docking application.

In summary, the employed Grid environment has proved to be significantly efficient, being 20 faster and providing 95% reduction in the overall execution respect to the sequential application. This reduction is highly remarkable taking present that, although a high number of resources conform the testbed, from the application point of view the full testbed can be represented as a homogeneous system of only 24 processors. This apparent low number of processors in the system is due to the limitations in the number of concurrent jobs by user.

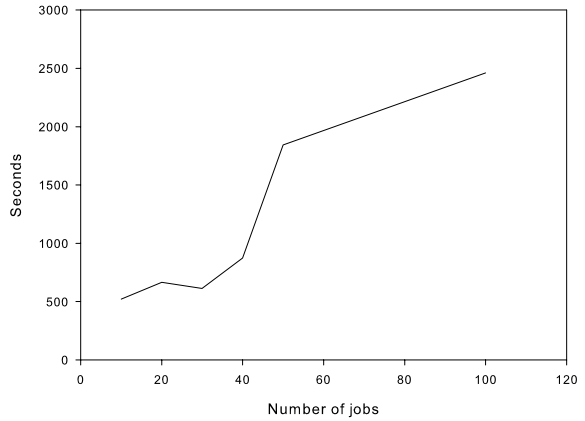

Figure 8. Execution time progression as a function of the number of jobs to perform.

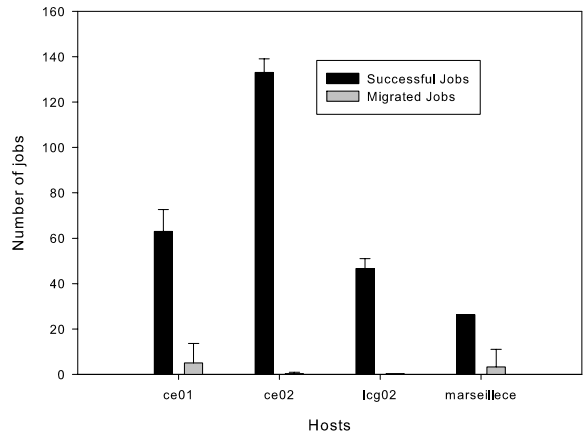

Figure 9. Average per host of successful and migrated jobs. Dark columns show successful jobs in each test, white columns show migrated jobs. Only hosts with successful executions are shown.

Let's now study the behaviour of the Grid environment's hosts. Figure 9 shows the average number of jobs correctly executed and migrated for each host. The migration of jobs can be the result of an incorrect execution of the job or the expiration of the queue time in the host. Notice that some of the hosts never get jobs assigned. The existence of non-contributing hosts could be explained by the dynamic availability of the resources together with the existence of other competing processes. gridgate is a special case, jobs are assigned to this host but they always migrate to another one, probably because of a high queue time. This non-contributive host could also increase the overall time, keeping jobs vainly waiting and increasing the transfer traffic through the Grid. Finally, it can be observed a correlation of the number of processors per host with the jobs successfully performed. For example, ce02 produces the higher amount of correct executions because it has the higher number of processors.

Figure 10 presents the average job queue, file transfer

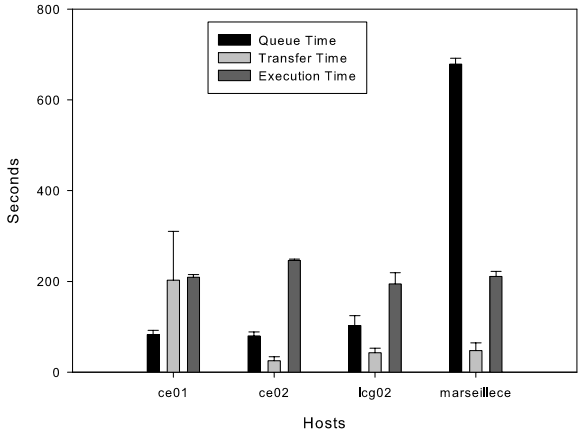

Figure 10. Average times for each host.

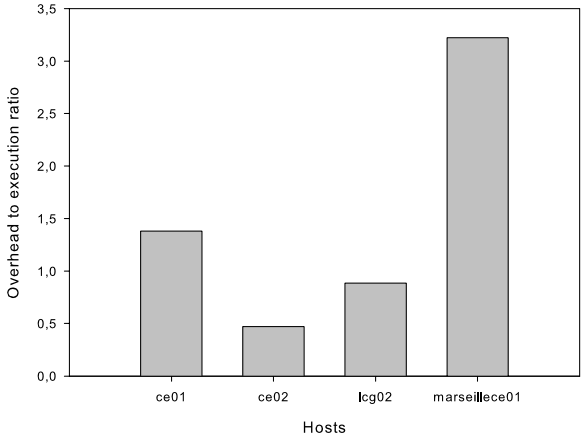

Figure 11. Execution overhead per host.

and execution times on each host. As it can be observed, such times are different for each host. For example, marseillece01's queue time is significantly higher. These variations are related to the different architectures of the hosts, operative systems and Local Resource Management Systems (LRMS). The degree of heterogeneity observed above is a consequence of these variations. For a more detailed study of the hosts' performance the execution ratio overhead (s) for each host is defined as:

$$s = \frac{T_{xfr} + T_{queue}}{T_{exec}} \tag{3}$$

Where T_{exec} is the execution time, T_{xfr} is the transfer time and T_{queue} is the queue wait time. This ratio determines the balance between the real active time and the schedule time per host. Values of s close to 0 means a favourable ratio where the execution time is bigger. On the contrary, $s > 1$ denotes a higher time in the schedule respect to the execution, showing that the computation weight of the jobs is too low. In Figure 11 are shown the average ratios for the different hosts. The ratio ranges from 0.5 in ce02 to 3.22 in marseillece01. The relative high ratios obtained indicate a low computational weight in the jobs, so resources are not being correctly exploited. This can be improved using a coarser grain distribution. For example, instead of only one fitting operation per job, multiple fitting operations could be performed. This strategy will increase the execution time with a low cost in the schedule.

V. Conclusions and Future Work

In the present work we report the employ of the GridWay Metascheduler over a Grid environment to a HTC Bioinformatics docking application. The challenging problem of finding the best fitting atomic model into a 3D low resolution map of a macromolecule has been successfully solved in a Grid environment. This new approach greatly simplifies the large-scale merging of 3D information data coming from diverse structural sources including bioinformatics modeling. This will effectively contribute to obtain accurate atomic interpretations of large macromolecular complexes.

This adaptation has been greatly facilitated by the resources provided by the GridWay Metascheduler. The obtained results show a 95% reduction in the overall execution time of the Grid approach respect to the sequential single job case. Having into account the restrictions in the use of the Grid environment that limit the concurrent jobs, this a very promising result. In fact, the current grid implementation can be routinely used for docking thousands of atomic resolution models. The timings obtained permit the large scale interpretation of low resolution EM experimental maps at atomic detail.

A deeper analysis of the hosts' characteristics has revealed a low computational weight in the job distribution. This observation suggests the suitability of a coarser grain distribution for increasing efficiency. To this end, the computational charge of the jobs should follow a dynamic grain schedule. This improvement can be obtained performing a variable number of fittings of different atomic structures in the same job, instead of only one fitting per job. Depending on the constitution of the Grid environment, the number of fittings per job could be balanced from a large number (providing a heavy grain distribution with few jobs demanding large computation resources) to a low number (providing a light grain distribution with many jobs demanding few computation resources). This capacity is not directly supported by Gridway's functionality and must be implemented by means of the DRMAA Application Programming Interface.

Current work is pursuing this research line. In addition, we are particularly interested in extending the use of this GridWay based rigid body search to other existing problems where 3D matching is needed. These problems can be found in a diverse range of fields such as Structural Biology [21], [22] or image processing [23], [24]. In concrete, Protein-Protein and Protein-Ligand dockings are promising candidates to this adaptation. There are already successful Grid applications to Protein-Ligand docking such as [25]. Protein-Protein docking is a very important

and complex problem in the area of structural biology that currently requires days or even weeks with high computational resources. To determine potential contact regions between two proteins requires a similar 6D search to the one reported here, but with a much wider space to explore. Thus, the adaptation of the presented approach to this problem will be extremely profitable by splitting the translational search for a single docking in parallel jobs over a Grid. This research line is actually under study and promising results have been obtained.

REFERENCES

[1] J. Frank, *Three-Dimensional Electron Microscopy of Macromolecular Assemblies*. San Diego, EEUU: Academic Press, 1996.

[2] W. Wriggers and P. Chacon, "Modeling tricks and fitting techniques for multiresolution structures," *Structure (Camb)*, vol. 9, pp. 779–788, 2001.

[3] F. Fabiola and M. Chapman, "Fitting of high-resolution structures into electron microscopy reconstruction images," *Structure (Camb)*, vol. 13, pp. 389–400, 2005.

[4] J. Garzon, J. Kovacs, R. Abagyan, and P. Chacon, "Adp_em: fast exhaustive multi-resolution docking for high-throughput coverage," *Bioinformatics*, vol. 23, no. 4, 2007.

[5] M. Topf, M. Baker, B. John, W. Chiu, and A. Sali, "Structural characterization of components of protein assemblies by comparative modelling and electron cryo-microscopy," *Journal of Structural Biology*, vol. 149, pp. 191–203, 2005.

[6] I. Foster, "What Is the Grid? A Three Point Checklist," *GRIDtoday*, vol. 1, no. 6, 2002.

[7] E. Huedo, R. S. Montero, and I. M. Llorente, "Experiences on Adaptive Grid Scheduling of Parameter Sweep Applications," in *Proc. 12th Euromicro Conf. Parallel, Distributed and Network-based Processing (PDP2004)*. IEEE CS, 2004, pp. 28–33.

[8] J. Yu and R. Buyya, "A Taxonomy of Workflow Management Systems for Grid Computing," *Journal of Grid Computing*, vol. 3, no. 3–4, pp. 171–200, 2005.

[9] J. M. Schopf, "Ten Actions when Superscheduling," The Open Grid Forum, Tech. Rep. WD8.5, 2001, scheduling Working Group.

[10] E. Huedo, R. S. Montero, and I. M. Llorente, "The Gridway Framework for Adaptive Scheduling and Execution on Grids," *Scalable Computing: Practice and Experience Journal*, vol. 6, no. 3, pp. 1–8, 2005.

[11] ——, "A Framework for Adaptive Execution on Grids," *Software – Practice and Experience (SPE)*, vol. 34, no. 7, pp. 631–651, 2004.

[12] M. Marti-Renom, A. Stuart, A. Fiser, R. Sanchez, F. Melo, and A. Sali, "Comparative protein structure modeling of genes and genomes," *Annu. Rev. Biophys. Biomol. Structl*, vol. 29, pp. 291–325, 2000.

[13] P. Chacon and W. Wriggers, "Multi-resolution contour-based fitting of macromolecular structures," *J Mol Biol*, vol. 317, pp. 375–384, 2002.

[14] J. Kovacs, P. Chacon, Y. Cong, E. Metwally, and W. Wriggers, "Fast rotational matching of rigid bodies by fast fourier transform acceleration of five degrees of freedom," *Acta Crystallogr D Biol Crystallogr*, vol. 59, pp. 1371–1376, 2003.

[15] M. Baker, R. Buyya, and D. Laforenza, "Grids and Grid Technologies for Wide-Area Distributed Computing," *Software – Practice and Experience*, vol. 32, no. 15, pp. 1437–1466, 2002.

[16] I. Foster and C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit," *Intl. J. Supercomputer Applications*, vol. 11, no. 2, pp. 115–128, 1997.

[17] J. Herrera, E. Huedo, R. Montero, and I. Llorente, "Developing Grid-Aware Applications with DRMAA on Globus-based Grids," in *Proc. 10th International Euro-Par Conference*, ser. Lecture Notes in Computer Science, vol. 3149, 2004, pp. 429–435.

[18] E. Huedo, R. S. Montero, and I. M. Llorente, "A Modular Meta-Scheduling Architecture for Interfacing with Pre-WS and WS Grid Resource Management Services," *Future Generation Computer Systems*, vol. 23, no. 2, pp. 252–261, Feb. 2007.

[19] I. Llorente, R. Montero, E. Huedo, and K. Leal, "A Grid Infrastructure for Utility Computing," in *Proc. 3rd International Workshop on Emerging Technologies for Next-generation GRID (ETNGRID 2006), 15th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE 2006)*, ser. IEEE Computer Society Press, 2006, pp. 163–168.

[20] E. Huedo, R. S. Montero, and I. M. Llorente, "Evaluating the Reliability of Computational Grids from the End User's Point of View," *Journal of Systems Architecture,*, vol. 52, no. 12, pp. 727–736, 2006.

[21] M. Rossmann and E. Arnold, *Crystallography of Biological Macromolecules.* Dordrecht, The Netherlands: Kluwer Academic Publishers, 2001.

[22] G. Smith and M. Sternberg, "Prediction of protein-protein interactions by docking methods," *Curr Opin Struct Biol*, vol. 12, pp. 28–35, 2002.

[23] G. Papaioannou, E. Karabassi, and T. Theoharis, "Reconstruction of three-dimensional objects through matching of their parts," *IEEE Transactions on Pattern Analysis and Machine In-telligence*, vol. 24, pp. 114–124, 2002.

[24] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. New Jersey: Prentice-Hall, 1998.

[25] M. Chang, W. Lindstrom, A. Olson, and R.K.Belew, "Analysis of hiv wild-type and mutant structures via in silico docking against ligand libraries," *J Chem Inf Model Biol*, vol. 43, no. 3, 2007.

José Ignacio Garzón received his M.E. in Computer Science (2001) from the Universidad de Granada (UGR). He is doing his Ph.D. in the Structural Bioinformatics Group at the Centro de Investigaciones Biológicas, Spanish National Research Council (CSIC), in Madrid. His research interests lie mainly in Structural Biology fitting algorithms and Grid Technology.

Eduardo Huedo received his M.E. in Computer Science (1999) and Ph.D. in Computer Architecture (2004) from Universidad Complutense de Madrid (UCM). He is an Assistant Professor of Computer Architecture and Technology in the Department of Computer Architecture and System Engineering at UCM since 2006. Previously, he was Postdoctoral Researcher in the Advanced Computing Laboratory at Centro de Astrobiología (CSIC-INTA), associated to NASA Astrobiology Institute. His research interests include Performance Management and Tuning, Parallel and Distributed Computing and Grid Technology.

Rubén S. Montero received his B.S. in Physics (1996), M.S. in Computer Science (1998) and Ph.D. in Computer Architecture (2002) from the Universidad Complutense de Madrid (UCM). He is an Associate Professor of Computer Architecture and Technology in the Department of Computer Architecture and System Engineering at UCM since 2006. He has held several

research appointments at ICASE (NASA Langley Research Center), here he worked on computational fluid dynamics, parallel multigrid algorithms and Cluster computing. Nowadays, his research interests lie mainly in Grid Technology, in particular in adaptive scheduling, adaptive execution and distributed algorithms.

Ignacio M. Llorente received his B.S. in Physics (1990), M.S. in Computer Science (1992) and Ph.D. in Computer Architecture (1995) from the Universidad Complutense de Madrid (UCM). He is Executive M.B.A. by Instituto de Empresa since 2003. He is Full Professor of Computer Architecture and Technology in the Department of Computer Architecture and System Engineering at UCM and Senior Scientist at Centro de Astrobiología (CSIC-INTA), associated to NASA Astrobiology Institute. He has held several appointments since 1997 as a Consultant in High Performance Computing and Applied Mathematics at ICASE (NASA Langley Research Center). His research areas are Information Security, High-Performance Computing and Grid Technology.

Pablo Chacón received his Ph.D. degree in Biochemistry from the Universidad Complutense de Madrid, Spain, in 1999. From 2000 to 2003 he was research associate at The Scripps Research Institute (TSRI) in La Jolla, CA where he developed software for image processing and biomolecular docking. He is currently staff scientist at the Centro de Investigaciones Biológicas, Spanish National Research Council (CSIC), in Madrid. His structural bioinformatics group is interested in the development of efficient algorithms to solve biophysical problems.