The Development of a Drug Discovery Virtual Screening Application on Taiwan Unigrid

Li-Yung Ho[1,2], Pangfeng Liu[2,3], Chien-Min Wang[1], and Jan-Jan Wu[1]

[1] Institute of Information Science, Academia Sinica, Taipei, Taiwan, R.O.C.
{lyho,cmwang,wuj}@iis.sinica.edu.tw
[2] Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.
{d96025,pangfeng}@csie.ntu.edu.tw
[3] Graduated Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan,R.O.C
{pangfeng}@csie.ncu.edu.tw

Abstract. This paper describes the development of an *in silico virtual screening* application on Taiwan Unigrid. In silico virtual screening is one of the most promising approach to accelerate the drug development process. This pilot application implementation demonstrates the power of grid, which provides reliable service via shared computing resources. This development illustrates not only a new collaboration pattern to challenge difficult research topics, but also a promising perspective for developing future grid infrastructure.

1 Introduction

Grid is an infrastructure for people to share computing and storage resources. Scientists in various disciplines may use grid to solve problems that cannot be solved efficiently by traditional clusters and workstations. On the other hand, grid systems provide computing capability beyond clusters by combinging clusters that are geogrpahically dispersed into a collective computing enviroment.

Despite that a large number of applications are already running on grids, many of these developments are just in the level of proof of concept [1], and few of them are ready for large scale production with experimental data. For example, Large Hadron Collider experiments at CERN is one of the largest data production system on grid infrastructure [2]. Approximatively, it foresaw to produce more than 1 PB/year of raw data and 200 TB of event summary data. The computational capacity required to analyze the data is roughly the equivalent of 200,000 of today's fastest PC processors [3].

The motivation of this work has been to develop large scale applications on grid systems, so that we can pinpoint the essential technology to run such applications. The reason is that it requires new grid environment capability, grid operation methodology, and grid services to run large scale applications in very stressful conditions, instaed of just the level of proof of concept. Consequently, with the experience of running serious applications on grid, we can improve the infrastructure by strengthening the performance of grid service, job scheduling algorithms on grids, and distribution efficiency.

In silico virtual screening is one of the most promising approach to accelerate the drug development process. In silico virtual screening is to select the best candidate drugs that bind on a given target protein by the use of computer [4]. It reduces the number of compounds required for *in vitro* and then *in vivo* testing from a few millions to a few hundreds, which saves the time and money for drug developers. Moreover, virtual screening fosters the collaboration of laboratories and has important societal impact by lowering the barrier to develop new drugs for rare or threatening diseases [5].

In silico virtual screening requires intensive computing, which is in the range of a few TFlops per day on one target protein. As a result it is a serious large scale application that requires a large amount of computing power. In addition, the grid is an excellent platform that provides enormous computing resources to sustain high throughput virtual screening [6]. For example, Taiwan unigrid provides access to more than a hundred CPUs, so that we can perform in silico virtual screening by simulating the docking process for millions of compounds.

This paper describes the development of an in silico virtual screening application on Taiwan Unigrid. This pilot application implementation demonstrates the power of grid, which provides reliable service via shared computing resources. This development illustrates not only a new collaboration pattern to challenge difficult research topics, but also a promising perspective for developing future grid infrastructure.

We organize the paper as follows. Section 2 presents Taiwan unigrid environment and the Autodock [7] docking engine we used in the development of virtual screening. Section 3 describes our virtual screening implementation on Taiwan Unigrid and the experiments on testing the throughput of production runs. Section 4 summaries the results and conlcudes with possible future works.

2 Grid Environment and Virtual Screening

This section introduces the infrastructure of Taiwan Unigrid and describes important issues in virtual screening applications. We will introduce the goals of Taiwan Unigrid project and its key grid technologies, and then discuss the essential issues to devlop virtual screening applications on grids.

2.1 Taiwan Unigrid

The goal of Taiwan Unigrid [8] project is to connect clusters at universities in Taiwan to build an experimental national computing platform, and to popularize the concept of grid computing to academic research and IT industry. Right now, Taiwan Unigrid has accumulated more than a hundred CPUs and Tera-byte level storage capacity from more than 20 sites in Taiwan. Fig 1 shows the topology of Taiwan Unigrid [8]. So far, there are seven universities and two institutes in Taiwan that participate in the Taiwan Unigrid project.

Globus toolkit [9] and SRB(Storage Resource Broker) [10] are two key components of Taiwean Unigrid. Globus toolkit provides the protocols to handle remote resources and SRB provides a storage sharing solution. SRB was developed at San Diego Supercomputing Center as a client-server middle-ware for managing file collections in a heterogeneous and distributed environment. With these two key technologies, Taiwan Unigrid provides reliable and automated services for sharing remote resources.

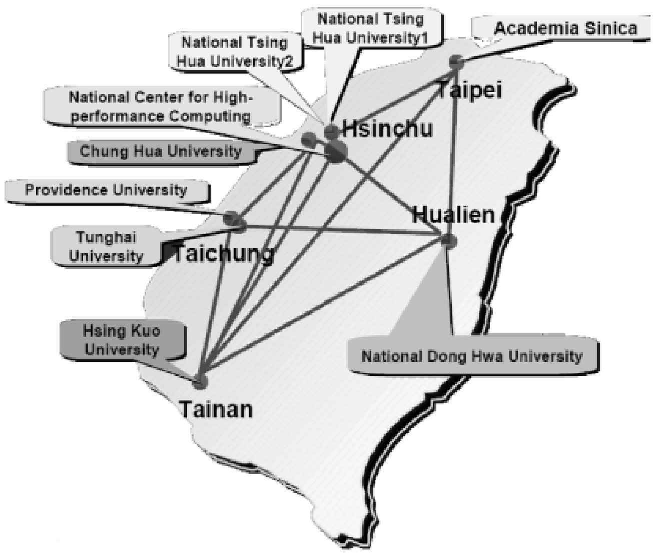
National Tsing Hua University1
National Tsing Hua University2
Academia Sinica
Taipei
National Center for High-performance Computing
Hsinchu
Chung Hua University
Providence University
Hualien
Tunghai University
Taichung
Hsing Kuo University
National Dong Hwa University
Tainan

Fig. 1. The topology of Taiwan Unigrid

2.2 Virtual Screening

Screening is a procedure in drug development to find the candidate compounds for vitro and vivo testing. There are two screening methods. The first method is to use real chemical compounds (in vitro), which is very expensive and subject to labortory contaimination. The second method is virtual screening; which uses a docking program to simulate the binding reaction. As long as we set the parameters of docking program correctly, the program can predict the binding position accurately.

The goal of virtual screening is to reduce the number of molecules required in both vitro and vivo testing from a few millions to a few hundreds [11], and to find out the lowest energy candidates in a compound database. Consequently, a large amount of resources are required in order to test a family of targets against a significant amount of possible drug candidates using a docking engine with different parameters settings. In addition, the dynamic conformation of the binding site (rigid or flexible) of a protein and the diversity of compound database will decide the complexity of the modeling system. Therefore, it induces a computing and storage resource intensive challenge problem to distribute millions of docking simulations with millions of small compounds. It is characterized by the computing and the storage load, which pose a great challenge to resources that a single institute can afford. We may estimate that a database with several hundred thousand compounds will take hundreds of years to screen. With the help of grid systems, the time may be reduced to several weeks.

A large-scale virtual screening deployment requires the development of a job submission and output data collection environment. A number of issues need to be addressed in order to achieve significant acceleration from a grid deployment. These issues are summarized in the following.

– The volume of data movement during job execution has a significant impact on grid performance. Therefore, the files providing the three-dimensional structure of the target protein and chemical compounds should preferably be stored on grid storage elements where the test runs will be executed.
– The jobs submission rate to grid computing elements must be carefully monitored in order to avoid system overloading.
– After submitting a large number of tasks, it is expected that the system should deliver fast turnaround of feedback results from these tasks, so that the grid resources are utilized efficiently.
– We should store the docking simulation results on storage elements close to the site where the job is executed, instead of sending them back to where the job is submitted, which may cause heavy I/O traffic and degrade the performance of job submission.

AutoDock docking engine Protein-ligand docking is to estimate the binding energy of a protein target to a potential drug using a scoring function. The target is typically a protein, which plays a pivotal role in a pathological process, e.g. the biological cycle of a given pathogen (like parasite, virus, bacteria, etc). The goal is to identify which molecules could dock on the active sites of a protein in order to inhibit cell infection, and therefore interfere the molecular process essential for the pathogen.

Every screening task is a simulation of docking a ligand to a target, and most of the time these collective tasks are embarrassing parallel with independent calculation from one another. As a result, a high-throughput computing environment is esstential to the overall performance of virtual screening.

The computational tool for screening is based on molecular docking engines, such as AutoDock, to carry out a quick conformation search of small compounds in the binding sites. Moreover, the docking engine calculates the binding energy of possible binding poses, and effeciently selects the most probable binding poses. The docking engine uses the 3D structures of compounds openly available and are produced by chemistry companies to calculate the binding energy.

The docking engine ranks binding poses accroding to the binding energy after a simulation. Consequently, we can keep good binders accroding to a given engergy threshold. These good binders serve as good cadinates for the most possible drug-like compounds.

Many docking software are available either as open-source or through a proprietary license. For example, we use AutoDock, an automated docking tool developed by Scripps Research Institute [12], in our development on Taiwan Unigrid. AutoDock models the binding process by treating the ligand and selected parts of the targets as conformationally flexible. AutoDock uses a AMBER force field based energy scoring function to estimate the free energy of binding. Hybrid global-local evolutionary algorithms are used to search the phase space of the ligand-macromolecule complex [7].

There are two major functions in the AutoDock package. The first function *autogrid* generates energy map files for each atom of the target protein. The second function *autodock* uses the energy map files generated by autogrid to calculate the final total binding energy. The version we used in this paper is AutoDock 3.0.5.

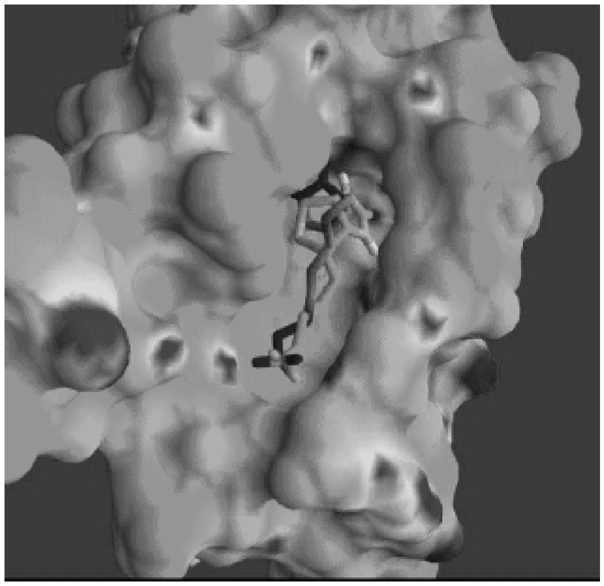

Fig. 2. A docking simulation result

Figure 2 compares AutoDock simulation results with actual crystallization results from docking the same biotin ligand on a Streptavidin [13, 7]. The lighter binding in Figure 2 is from AutoDock simulation, and the darker binding is from crystallization experiments. By comparing the pose and position between these two results, we conclude that Autodock accurately simulates the interaction between the lignad and the active site. The simulation result is very similar to those produced by actual crystallization. Therefore, we are confident that Autodock predicts binding ligand poses and positions with very high accuracy.

3 Throughput Test

Throughput rate is one of the most important measurement to estimate the performance of the in silico virtual screening platform on grid systems. The objective of this experiment is to investigate the throughput rate of the virtual screening environment. The investigation gives us the information about the system overhead and data latency. We can improve the system performance and data transfer mechanism according to the measurement result. The improvement will reduce the time and cost of initial investment on structure-based drug design, and will also demonstrate the feasibility of high performance screening application on Taiwan Unigrid infrastructure. After illustrating the effectiveness of the applications, we can foster an uptake of grid technologies in scientific area.

3.1 Model

We select three computing elements and two storage elements in Taiwan unigrid to be the test environment. The total number of CPUs is 52 and each storage element has the same copy of the input data. The preliminary test of virtual screening is to screen a 100 ligands database to avian influenza virus (H5N1) [14]. According to the characteristics of *autogrid*, the map file of a protein in a virtual screening job is applicable to all kind of ligands. Therefore, we generate the map files of a target protein first by the *autogrid* program, and then store these map files on the two storage element. The *autodock* tasks screening a compound database could request those map files directly without running the *autogrid* program to the same protein over and over again. This approach reduces the execution time of a single pipeline simulation.

We implement the mechanism of data transfer and resource allocation via Globus toolkit [9]. In the following, we describe the procedures to develop our testing environment. First, we prestage the grid map files of the target protein which are generated by the *autogrid* program on one grid storage element. Second, we submit the 100 *autodock* tasks to the grid environment. In order to balance the workload among sites, we distribute the tasks according to the CPU speed on each site, i.e. the number of tasks a site receives depends on the number of CPUs and their speed on that site. Finally, the output of each task is collected and saved in the same storage element with input data. Figure 3 illustrates the relationship between the components of our computation. The user receives the information of ligand database from the storage element, and then submits tasks to the grid computing elements. The computing elements receives input data and the executable program from the storage element. Once the task finishes, it returns the result to the storage element.

We consider two factors that may influence the throughput efficiency of the *autodock* screening: the number of storage elements and the starting time to transfer input data. The number of storage elements we use in our experiments is one and two. The input data is staged in either at task submission or during runtime. Therefore, totally we have four different strategies for a task to stage input data: from one storage element at task submission, from one storage element during task runtime, from two storage elements at task submission, and from two storage elements during task runtime. The reason for considering the starting time of staging input data is the concern about the loading of the gridftp server. For batch submission of virtual screning tasks, staging input data at task submission time will cause overloading of the storage elements, and thus result in task failure. To solve this problem, we balance the workload of the storage elements by transferring input data during runtime of the task. In the following section, we will compare the throughput rate of the four strategies for transferring input data.

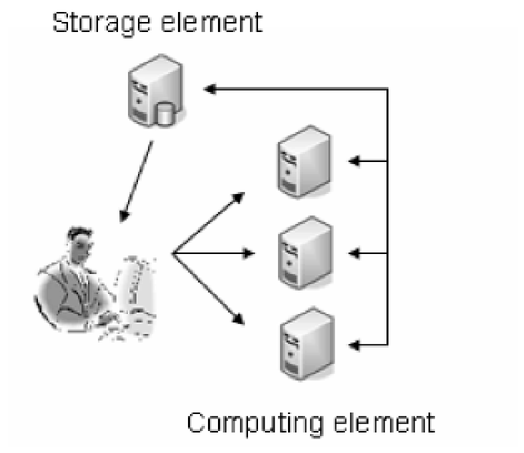

Fig. 3. The model of the test

We implement the test with a small database containing 100 compounds. A single docking simulation takes about 6 minutes to finish on a 2.4 GHz, 2G RAM machine, and produces a 300 KB. log file. The size of a prestaged input file for each task is about 5.5 megabyte. Therefore, the 100 tasks of the virtual screening application take 10 hours of CPU time and produce 30 megabyte of data. Our test case is not particularly large, but it can help us discover the bottleneck of the grid system so that we can devise effective methods to improve the efficiency in the future.

3.2 Preliminary result

Our preliminary result indicate that in the best case of the four strategies, we can finish at most 72 tasks in 25 minutes on Taiwan Unigrid, and it returns a result per 20.8 seconds in average. The successful rate of successfully submitted task is up to 100% and the crunching factor is between 8.18 and 8.78. Table 1 summarizes the results of the four strategies in our testing. We use "SS" to denote "single storage element with staging input at submission time","SR" to denote "single storage element with staging input at runtime","MS" to denote "multiple storage elements with staging input at submission time", and "MR" to denote "multiple storage elements with staging input at runtime".

We found that the *SS* case (staging input at submission time to single storage element) experiences a significant failure rate of task submission. It is caused by the heavy loading of simultaneous accesses to a single storage element. Figure 4 illustrates the duration of completed tasks which are successfully submitted. All curves are similar except for the SS case, whcih is due to the high failure rate of task submission. Furthermore, almost all submitted tasks are completed within 50% of the total duration time.

	SS	SR	MS	MR
successful submission rate of tasks	47.10%	70.59%	71.57%	70.59%
Successful rate of submitted task	100%	95.80%	94.50%	93.10%
Estimated duration on 1 CPU(hours)	4.8	6.9	6.9	6.7
Duration of the experience(minutes)	29	44	41	41
Crunching factor	8.57	8.18	8.90	8.78

Table 1. Summary of the prelimlnary test.

During the deployment of the screening application on Taiwan Unigrid, we encountered several problems that need to be solved. We also encountered difficulties that require new mechanism that should be incorporated into Taiwan Unigrid in order to develop a high throughput virtual screening platform. In the following, we point out two issues that need to be addressed.

– In the model, we prestage only one docking executable on a storage element. This approach causes simultaneous accessing to the same storage element and therefore incurs significant I/O bottleneck. Moreover, submitting task to grid will fail because GRAM jobs cannot stage in the executable program due to I/O bottleneck. This problem also occurred when tasks read the input file. Instead of simultaneous accessing, a GRAM job needs to wait for seconds to get the input file due to the heavy loading of storage element. This causes the GRAM job to fail because it needs to stage the input file in the computing element at the time when the job is submitted, but the input data is unavailable due to heavy loading of the storage element. Therefore, either

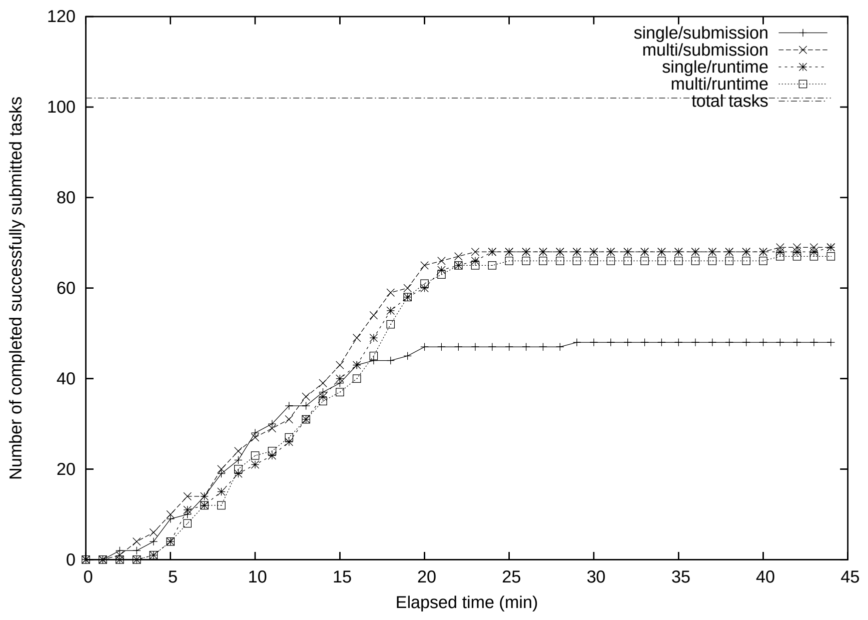

Fig. 4. Duration time of completed tasks

the unreachable executable or input files will cause failure of job submitted to the grid. This failure caused by heavy I/O bottleneck might be avoided by distributing multiple copies of the executable and input files over the grid to prevent simultaneous accesses to a single storage element.

– We balance the workload among computing elements by submitting tasks sequentially to each site. This kind of resource assignment would not utilize the grid environment very well. In order to resolve this problem, we need a resource broker to help us choose the computing element to submit our task. Moreover, the time of accessing data on storage element has significant impact on the duration time of respondence. Therefore, a more advanced resource broker that takes the cost of accessing storage element into account should be also considered. By developing a resource broker, we may utilize the grid environment more efficiently and reduce the possibility of job submission failure due to heavy loading of the computing elements.

The advantage of using Globus toolkit directly in *out test* is the high successful rate of job submission. Moreover, without the overhead of grid middle-ware such as LCG or gLite [15], the overhead of resource allocation is negligible, and the returning time of the result is much shorter. However, with the Globus toolkit, management of resources cannot be controlled very well and utilized efficiently. That is the reason of high successful rate of task submission but low resource utilization in out test. Based on our experiment results, we suggest that a light weight middle-ware is desirable to control data transfer and job assignment without incuring much overhead. Furthermore, a more powerful and flexible inferface is important for the user community. The functions of bulk tasks submission and collection of large amount of result data should be considered when developing the interface. The Globus toolkit lacks these support to facilitate efficient development of a virtual screening platform. Instead of using Globus as the main interface and middle-ware, it is desirble to incorporate a more friendly interface and a light weight middle-ware in the Taiwan Unigrid.

4 Conclusion

In the paper, we demonstrate the deployment of in silico virtual screening developed on Taiwan Unigrid. We investigate the throughput rate of screening and discover the issues about data transfer and computing element assignment. In conclusion, an easy-to-use interface for user and a lite weight middle-ware should be introduced to establish a reliable virtual screening platform on Taiwan Unigrid. This pilot application on Taiwan Unigrid could play a role of navigating the direction for developing grid environment, and also provide an opportunity for scientists to cooperate more closely.

As the complexity of applications grows rapidly, scientists encounter the challenge of requiring more and more computing and storage resources. Grid environment provides the abilities to conquer them and achieve the goal of unifying the global computing resources. However, this paper illustrates the inefficiency of the current grid environment in deploying a high throughput productive applications. A middle-ware to assign, manage resources, distribute tasks and an easy-to-use interfaces should be introduced for more reliable and versatile grid services.

References

1. Jacq, N., et al: Grid as a bioinformatics tool. In: Parallel Computing 30. Elsevier (2004) 1093–1107
2. Campana, S., et al: Analysis of the atlas rome production experience on the lhc computing grid. In: IEEE International Conference on e-Science and Grid Computing. (2005)
3. Rueden, W.V., Mondardini, R.: The Large Hadron Collider (LHC) Data Challenge. (2003) IEEE Technical Committee on Scalable Computing.
4. Lyne: Structure-based virtual screening:an overview. In: Drug Discov. Today 7. Elsevier (2002)
5. Nwaka, S., Ridley, R.G.: Virtual drug discovery and development for neglected diseases through public-private partnerships. In: Nat Rev Drug Discov 2. Elsevier (2003) 919–28
6. Chien, A., et al: Grid technologies empowering drug discovery. In: Drug Discov. Today 7. Elsevier (2002) 176–180
7. ADT: AutoDock docking tools. (http://autodock.scripps.edu)
8. Taiwan Unigrid: Taiwan Unigrid Project. (http://www.unigrid.org.tw)
9. The Globus Alliance: Globus Toolkit. (http://www.globus.org)

10. SRB: Storage Resource Broker. (http://www.sdsc.edu/srb/index.php)
11. Spencer, R.W.: Highthroughput virtual screeing of historic collection on the file size, biological targets, and file diversity. In: Biotechnology and bioengineering. Wiley (1998) 61–67
12. Morris, G.M., et al: Automated docking using a lamarckian genetic algorithm and empirical binding free energy function. J. Computational Chemistry **19** (1998) 1639–1662
13. Freitag, S., Trong, I.L., Klumb, L., Stayton, P., Stenkamp, R.: Structural studies of the streptavidin binding loop. Protein Sci **6** (1997) 1157–1166
14. Russell, R., Haire, L., Stevens, D., Collins, P., Lin, Y., Blackburn, G., Hay, A., Gamblin, S., Skehel, J.: Structural biology: antiviral drugs fit for a purpose. Nature **443** (2006) 37–38
15. LCG2: LCG-2 middleware overview. (https://edms.cern.ch/file/498079/0.1/LCG-mw.pdf)