

Drug Discovery Grid

Wenju Zhang¹, Jun Zhang³, Yan Chang³, Shudong Chen¹, Xuefeng Du¹, Fei Liu¹,
Fanyuan Ma¹, Jianhua Shen^{2*}

¹ Shanghai Jiao Tong University, Shanghai, 200030

² Shanghai Institute of Materia Medica, Chinese Academy of Sciences, Shanghai, 201203

³ JIANGNAN Institute of Computing Technology, Wuxi, 214083

* Corresponding author: email address jhshen@mail.shcnc.ac.cn

Abstract

This paper discusses the design and implementation of drug discovery grid in details. The purpose of the Drug Discovery Grid (DDGrid) is to set up a grid environment with aggregated compute and data resources to provide the drug virtual screening services and the pharmaceutical chemistry information services. Then, emphasizes the analysis about the architecture and key technology of implementation of the DDGrid, and gives a particular description of the components of the DDGrid. Finally, the system evaluation of the DDGrid is conducted, and the job scheduling policy is analyzed.

1. Introduction

Grid computing joins together many individual computers, creating a large system with massive computational power that far surpasses the power of a handful of supercomputers [3]. Because the work is split into small pieces that can be processed simultaneously, research time is reduced from years to months. The technology is also more cost-effective, enabling better use of geographically distributed resources.

P2P (Peer-to-Peer) [4] environments are characterized by large scale, intermittent user and resource participation, lack of trust, and lack of implicit incentives for good behavior with the potential of harnessing millions of computers is tremendous. Grid computing focuses on distributing computation on wide-area collections of shared resources. Rooted in the high-performance scientific computing community, the driving force for Grid computing is supporting computation-, data-, and network-intensive applications.

While P2P and Grid systems share the

same focus on harnessing resources across multiple administrative domains, they differ in many respects: Grids address support for a variety of applications and hence focus on providing infrastructure with quality-of-service guarantees to moderate-sized, homogeneous, and partially trusted communities. In contrast, P2P systems concentrate on providing support for intermittent participation in vertically integrated applications for significantly larger communities of untrusted, anonymous individuals. However, the convergence of the two systems is increasingly visible: the two research communities started to acknowledge each other by forming multiple research groups that study the potential lessons that can be exchanged; P2P research focuses more and more on providing infrastructure and diversifying the set of applications; Grid research is starting to pay particular attention to increasing scalability.

Drug discovery is an extended process that can take as many as 15 years from the

first compound synthesis in the laboratory until the therapeutic agent or drug, is brought to market. Reducing the research timeline in the discovery stage is a key priority for pharmaceutical companies worldwide. Many such companies are trying to achieve this goal through the application and integration of advanced technologies such as computational biology, chemistry, computer graphics and high performance computing (HPC). Molecular modeling has emerged as a popular methodology for drug design--it can combine computational chemistry and computer graphics. Molecular modeling can be implemented as a master-worker parallel application, which can take advantage of HPC technologies such as clusters and Grids for large-scale data exploration.

Drug design using molecular modeling techniques involve screening a very large number (of the order of a million) of ligand records or molecules of compounds in a chemical database (CDB) to identify those that are potential drugs. This process is called molecular docking [7]. It helps scientists to predict how small molecules, such as substrates or drug candidates, bind to an enzyme or a protein receptor of known three-dimensional (3D) structure. Docking each molecule in the target chemical database is both a compute and data intensive task. It is our goal to use P2P and Grid technologies to provide cheap and efficient solutions for the execution of molecular docking tasks on large-scale, wide area parallel and distributed systems.

The rest of this paper is organized as follows. An overview of the architecture of DDGrid is presented in Section 2. The implementation and application in reality are presented in Section 3. The experimental results and job scheduling policy are analyzed in Section 4. Section 5 presents the related work. The final section summarizes the paper.

2. Architecture

DDGrid (Drug Discovery Grid) project aims to build a collaboration platform for drug discovery using the state-of-the-art grid computing technology. The project intends to solve large-scale computation and data intensive scientific applications in the fields of medicine chemistry and molecular biology with the help of grid middleware developed by our team.

2.1 P2P inspired Grid platform

Grid computing involves organizationally-owned resources: supercomputers, clusters, and PCs owned by universities, research labs, and companies. These resources are centrally managed by IT professionals, are powered on most of the time, and are connected by full-time, high-bandwidth network links. But these resources usually are underutilized. How to harness the full power of those geographically distributed cluster and supercomputing systems? There comes the Drug Discovery Grid. We only use the idle cycle the cluster and supercomputing system through process yield. The compute client will yield to the system when the system is busy. The overall framework of the DDGrid is shown in Fig. 1.

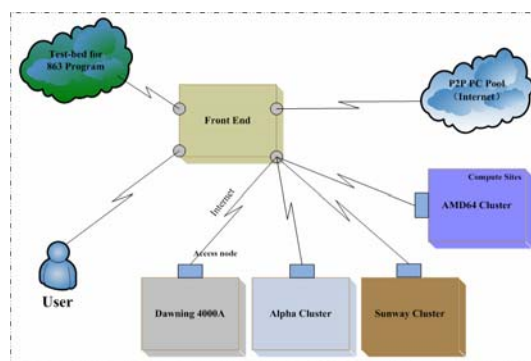


Figure 1. Overall Framework of the DDGrid

2.2 Software Stack and Resources of the DDGrid

DDGrid is build on the existing P2P and Grid

technologies and tools for performing compute and data intensive computing on distributed resources. A layered architecture and the software stack essential for performing molecular modeling on distributed resources is depicted in Figure 2. The components of the DDGrid software stack and resources are:

- 1) The Dock [4] and gsDock software for molecular modeling.
- 2) Toolkits for Drug Discovery such as CDB maintain software, preprocess and security-related tools.
- 3) Web portal for grid administrator, resources provider and consumer.
- 4) Grid middleware based on BOINC [2].

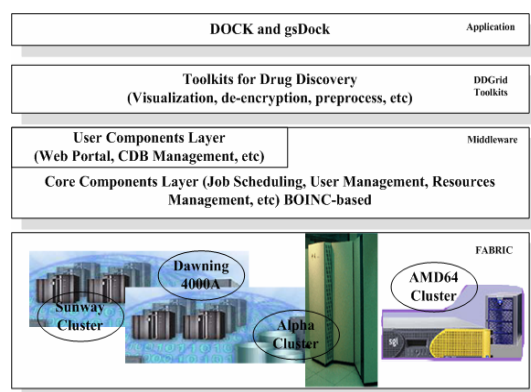


Figure 2. Layered Architecture of the DDGrid

2.3 Modules of the DDGrid

DDGrid mainly contains four components: Front End, Access Node Agent, Compute Client and Applications. The Front End, also known as global main server, consists of the following modules: job management module (scheduling, work generator, distributed work pool management, result assimilator and result valuator, etc.), user management module, resources management module, data service and web portal. Those modules are depicted in Figure 3.

The Access Node Agent has the similarity function with the Front End. It consists of the following modules: local job management module, local resources

management module and local data service. The default implementation of the local job management module is OS-based fork process on every compute client. The resource management plug-in for LSF [10] is also deployed on some sites with LSF faculty. Other plug-in for PBS [11], Condor [12] and SGE [13] is also considering.

The Compute Client software resides on every node of site. When idle, it will request data on a specific project from slave server. It will then perform computations on this data, send the results back to the slave server, and ask the slave server for a new piece of work. The slave server maintains a local work pool. It will request more work from global work pool on demand.

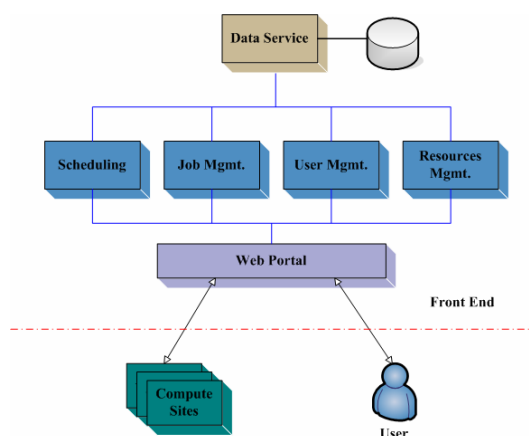


Figure 3. the Components of Front End

2.4 Security Consideration

All the protocols we used for the connection of the global main server and slave server are http/https-based. The scheduling procedure is client-initiative. The scheduling server authenticates the sites by password they provided through XML message. So this security mechanism can prevent from the unauthorized access.

3. Implementation and Application of DDGrid

3.1 Wrapping application, computation and

data

DDGrid uses a simple but rich set of abstractions for files, applications, and data. A project defines application versions for various platforms (Alpha, Linux/x86, Linux/AMD64, etc.). An application can consist of an arbitrary set of files. A workunit represents the inputs to a computation: the application contains a set of references input files, and sets of command-line arguments and environment variables. Each workunit has parameters such as compute, memory and storage requirements and a soft deadline for completion.

A result represents the result of a computation: it consists of a reference to a workunit and a list of references to output files. Files (associated with application versions, workunits, or results) have project-wide unique names and are immutable. Files can be replicated: the description of a file includes a list of URLs from which it may be downloaded or uploaded. Files can have associated attributes indicating, for example, that they should remain resident on a host after their initial use, that they must be validated with a digital signature, or that they must be compressed before network transfer. When the compute client communicates with a scheduling server it reports completed work, and receives an XML document describing a collection of the above entities. The client then downloads and uploads files and runs applications; it maximizes concurrency, using multiple CPUs when possible and overlapping communication and computation.

The server complex of a project is centered around a relational database that stores descriptions of DDGrid applications, platforms, versions, workunits, results, accounts, teams, and so on. Server functions are performed by a set of web services and daemon processes: Scheduling servers handles RPCs from clients; it issues work and handles

reports of completed results. Data servers handles file uploads using a certificate-based mechanism to ensure that only legitimate files, with prescribed size limits, can be uploaded. File downloads are handled by plain HTTP.

3.2 Global and local scheduling policy

DDGrid maintains two level's work pool. One is global work pool at the Front End and another is local work pool at the slave server. The global work pool is managed dynamically. We use HexStr to identify the number and location of the CDB to dock. For example, there are 10 molecules to dock in the CDB and the scheduler allocates four nodes to complete it. The four nodes screen molecule No. [1,2], [3,5,6], [4,7,8], [9,10] respectively. So the HexStr of those four nodes is "3", "34", "C8" and "400" respectively. The maximal number and location of molecules can be identified is 256K for a 64K-length HexStr.

The DDGrid compute client and local scheduling server (slave server), in its decisions of when to get work and from what project, and what tasks to execute at a given point, implements a "local scheduling policy". This policy has several goals:

- 1) To maximize resource usage (i.e. to keep all processors busy);
- 2) To satisfy result deadlines;
- 3) To respect the participant's resource share allocation.

3.3 Resources aggregated and applications deployed

We have collected several sites which provide geographically distributed heterogeneous resources. Those resource sites include 1) SIMM (Shanghai Institute of Materia Medica) Alpha Cluster, allocated 16 CPUs with 8 nodes. 2) HKU (the University of Hong Kong) Gideon 300 Cluster, allocated 16 CPUs with 16 nodes. 3) Beijing Molecule Ltd. Sunway 256P Cluster, allocated 256 CPUs with 128

nodes. 4) SSC (Shanghai Supercomputing Center) Dawning 4000A, allocated 32 CPUs with 16 nodes. 5) SJTU (Shanghai Jiao Tong University) PC Cluster, allocated 16 CPUs with 8 nodes. There are also some sites will be joined by Sept. 2005, which are Singapore Poly-tech University and Dalian University of Technology.

There are two versions of Docking applications deployed on this grid platform. One is Dock v4.0.1 developed by UCSF, another is gsDock v2.0 developed by SIMM which is optimized by generic algorithm. Some preprocess software and toolkits are also deployed on this grid platform such as autogrid/autodock, combimark, combilib, etc.

3.4 Chemical databases

The chemical databases contain records of a large number of molecules from self-made, free or commercially available organic synthesis libraries and natural product databases. The molecules in the CDB are represented in MOL2 file (.mol2) format, which is a portable representation of a SYBYL molecule. The MOL2 file is an ASCII file that contains all the information needed to reconstruct a SYBYL molecule. Each ligand record in a chemical database represents the 3D structural information of a compound. The numbers of compounds in each CDB can be in the order of tens of thousands and the database size be anywhere from tens of megabytes to gigabytes and even terabytes.

There are three types of chemical databases available on this grid platform. 1) free available or commercially purchased, 2) shared self-made, 3) private self-made. The first type of chemical database is deployed in advance on the every site involved. The last two types of chemical database are deployed on demand. All the chemical databases are encrypted before deployment.

Over four million compounds database

with 3-D structure and physicochemical properties are also provided to identify potential drug candidates. Users can build and maintain their own customized ligand database to share in this grid platform. The chemical databases deployed on this grid platform are as follows.

Specs, provides about 230,000 compounds.

CMC-3D, provides 3D models and important biochemical properties (including drug class, logP, and pKa values) for over 8,400 pharmaceutical compounds.

ACD-3D, provides 200,000 3D compounds commercial available.

NCI-3D, provides 213,000 compounds with 2D information from the National Cancer Institute.

CNPD (China Natural Products Database), Collected 12,000 Chinese natural products with chemical structure.

TCMD (Traditional Chinese Medicine Database), provides 9127 compounds and 3922 herbs.

ZINC, a free database of commercially-available compounds for virtual screening, ZINC contains over 3.3 million compounds in ready-to-dock, 3D formats, provided by the Shoichet Laboratory in the Department of Pharmaceutical Chemistry at the University of California, San Francisco (UCSF).

3.5 Application in reality

The farnesoid X receptor (FXR) belongs to a family of ligand-inducible transcription factors. FXR has been implicated in the regulation of cholesterol metabolism and enterohepatic circulation of bile acid. Due to its role in the metabolism of cholesterol and the regulation of bile acid biosynthesis and transportation, FXR has been shown to be a potential drug target for treatment of hyperlipidemia, cholelithiasis, or cholestasis. Based on the crystal structure of

FXR/fexaramine complex, DDGrid virtual screening approach was employed to search the SPECS database (<http://www.specs.net>, including 120,000 small molecules). Compounds were ranked according to the relative binding energy, favorable shape complementarity and potential of forming hydrogen bonds with FXR. The top 5,000 candidate molecules were obtained with the best scores by a shape complementarity scoring function in DDGrid.

These compounds were re-estimated using CScore and the binding energies of the top 2,200 molecules with FXR were calculated and modeled using FlexX program. Accordingly, the candidate molecules for bioassay were selected according to the binding energy. Moreover, the shape complementarity and drug-likeness were considered in the molecule selection. Finally,

Table 1. The software, configuration and network environments of involved nodes

No.	Location	Configuration	Software Env.	Speed of Conn. to Front end
0	SIMM (Front end)	CPU 2×P4 2.0G/512M Memory	RedHat EL AS 3.0/ddgrid/Dock	N/A
1	SJTU PC Cluster	CPU 1×P4 1.7G × 8/512M Memory	Gentoo Linux 2004.3/ddg_client	5Mbps
2	SIMM Alpha Cluster	CPU 2×200M×8/2G Memory	Compaq Tru64 UNIX V5.0A (Rev. 1094)/ddg_client	100Mbps
3	HKU Gideon 300 Cluster	CPU 1×P4 2.0G× 16/512M Memory	RedHat Linux 9.0/ddg_client	5Mbps
4	Beijing Molecule Ltd.	CPU 2×P4 2.0G× 128/512M Memory	RedHat Linux 7.3/ddg_client	5Mbps

In this experiment, we use 200 protein molecules to dock in order to evaluate the effect of varying the task granularity and network on performance. The web portal is implemented with PHP and the other modules are implemented with C++. We use Apache HTTPD as our web server.

Figure 4 shows the performance is best when each task receive 4 protein molecules. We think the cost of time used to compute is

73 compounds were selected based on the score by virtual screening and drug-likeness analysis for biological assay. Fluorescence resonance energy transfer (FRET) measurements were used to determine the binding affinity of these 73 candidate molecules against FXR. After FRET-based binding affinity assay, five compounds were determined to show high binding affinities against FXR at micro-molar level.

4. Experimental results and evaluation

The performance of our system is evaluated by protein molecules docking experiment which is used to analyze the similarity between protein molecules provided by end-users and those in chemical/biological databases. Experiments were carried out on five nodes which are as follows:

related to network and database. The cost of the communication decrease with the task granularity increases. So the total cost of time decrease. On the other hand, load balance is worse with the task granularity increases. The bad load balance can cause the total computing time increase.

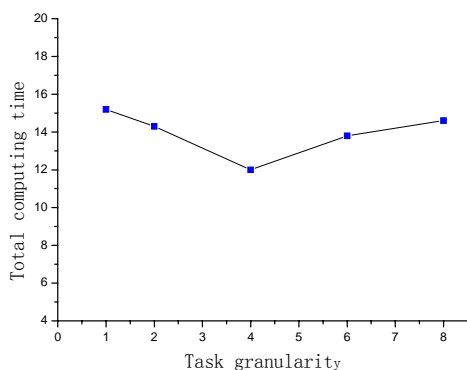


Figure 4. The effect of varying task granularity on total computing time

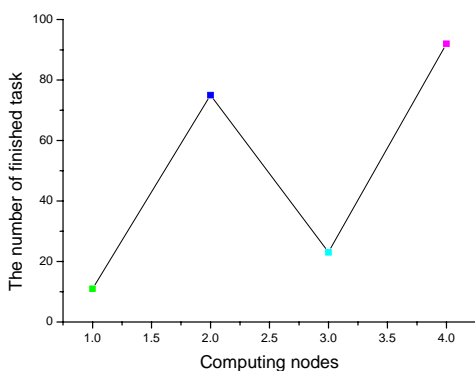


Figure 5. The load balance of four computing nodes

Figure 5 shows the load balance of four computing nodes. In this experiment we set each task process 4 protein molecules. From Fig 7, we find the load of node 2 and 4 is larger than node 1 and node 3 which is consistent with the capability of four nodes. So we can see our system has good load balance.

5. Related Work

Many researchers have explored the use of parallel computing techniques in molecular docking for drug design, but with the limitation in the local network domain. The effort of leveraging P2P and Grid technologies have made some progress, the examples including SETI@HOME [1] project, Folding@Home [6], distributed.net, D2OL

(Drug Design Optimization Lab) [5] and World Community Grid [15] launched by IBM.

Most of the efforts explicitly develop docking application as a parallel application using a special purpose, legacy or standard, parallel programming languages and interfaces such as PVM and MPI, which requires extra development effort and time. The scalability of such applications and runtime systems is limited to resources available in a single domain and they need powerful computers and networks for faster processing.

The team of Grid Computing and Distributed Systems Laboratory in the University of Melbourne are also exploring molecule modeling using grid technologies [8], whose application is based on Globus [9] and Nimrod-G. The Drug Discovery Grid leverages the P2P with grid technologies, so it can harness the full potential power of cluster and supercomputing systems owned by different organizations.

6. Conclusions

Grid computing are emerging as a new paradigm for sharing and aggregation of geographically distributed resources for solving large-scale compute and data intensive problems in science, engineering and commerce. However, application development, resource management and scheduling in these environments are a complex undertaking. In this paper, we illustrate the development of the DDGrid environment by leveraging existing P2P and Grid technologies to enable molecular modelling for drug design on geographically distributed resources.

By means of a highly effective communication model adapting to the current Internet conditions and the efficient load balancing technology both in-bound to and out-bound from the grid nodes, DDGrid

enables the maximum usage of idle computation resources within the China National Grid. DDGrid can provide high throughput capacity and secure novel drug virtual screening services for the end users.

Several supercomputers and computer clusters located in Shanghai, Beijing and Hong Kong have been integrated into this application grid platform, forming a computing power of higher than 1TFlops. Also, this platform provides databases containing three dimensional structures and the drug information of more than four million compounds. DDGrid has been used in drug screening for anti-SARS, anti-diabetic, anti-arthritis drug research projects. Based on the correlation research results four new drugs invention patents have been applied.

Acknowledgement

This research work is supported in part by the National High Technology Research and Development Program of China (863 Program), under Grant No. 2004AA104270.

Reference

1. D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer. SETI@home: An experiment in public-resource computing. *Communications of the ACM*, Nov. 2002, Vol. 45 No. 11, pp. 56-61. See also <http://setiathome.berkeley.edu>
2. Berkeley Open Infrastructure for Network Computing. 2004. <http://boinc.berkeley.edu/>
3. Foster I, Kesselman C, Tuecke S. The anatomy of the Grid: Enabling scalable virtual organization. *International Journal of High*

Performance Computing Applications 2001; 15(3):200–222.

4. Geoffrey Fox, Dennis Gannon, Sung-Hoon Ko, Sangmi-Lee, et al. *Peer-to-peer Grids. Grid Computing – Making the Global Infrastructure a Reality*. John Wiley & Sons, Ltd. 2003.

5. Drug Design Optimization Lab. <http://www.d2ol.com/>

6. Folding@Home distributed computing project. <http://folding.stanford.edu/>

7. Ewing A, et al. DOCK Version 4.0 Reference Manual. University of California at San Francisco (UCSF), U.S.A., 1998. <http://dock.compbio.ucsf.edu/>.

8. Rajkumar Buyya, Kim Branson, Jon Giddy and David Abramson. *The Virtual Laboratory: a toolset to enable distributed molecular modelling for drug design on the World-Wide Grid. Concurrency and Computation: Practice and Experience* 2003; 15:1–25.

9. Globus Grid Project. <http://www.globus.org/>

10. Zhou S. LSF: Load sharing in large-scale heterogeneous distributed systems. *Proceedings of Workshop on Cluster Computing*. December 1992.

11. OpenPBS. <http://www.openpbs.org/>

12. The Condor Project. <http://www.cs.wisc.edu/condor/>

13. Sun Grid Engine. <http://gridengine.sunsource.net/>

14. Foster I, Kesselman C (eds.). *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 1999.

15. World Community Grid. <http://www.worldcommunitygrid.org>