Performance of CFD Application on a Grid Platform between USA, China and Germany

Weizhe Zhang, Bin Li, Hui He, Mingzeng Hu
School of Computer Science and Technology
Harbin Institute of Technology University
Room C410, Dorm 10, Harbin Institute of Technology, 150001
P.R.China
weizhe.zhang@gmail.com

Abstract: - A computational grid platform “Rnet Grid” across all the 31 provinces of China, University of Houston in USA and University of Stuttgart in Germany is built and deployed, which provides a heterogeneous hardware infrastructure to enable scientific and engineering applications to run in a China-centric grid environment. Based on the widely distributed hardware and software resources, some preliminary results on an air quality grid (CFD) application are acquired and verify the stability of the platform and the scalability of latency-tolerance algorithm for the application.

Key-Words: - **CFD, computational grid, fault tolerance, load balance, scheduling,**

1 Introduction

The computational grid is getting popular day by day with the emergence of the Internet as a ubiquitous media and the wide spread availability of powerful computers and networks as low-cost commodity components. The computing resources are located across various regions around the globe. A number of countries and orginizations put great efforts on the contruct and deployment of computational grid platform. In America, National Science Foundation claimed to build and deploy the world's largest, fastest, distributed infrastructure for open scientific research with multi-year effort. When completed, the TeraGrid[1] will include 20 teraflops of computing power distributed at nine sites, facilities capable of managing and storing nearly 1 petabyte of data. In Europe, European Union plans to build European Data Grid (E-sciencE)[2] including five major participators (EONR, ESA, CNRS in France, NIKHEF in Holland and PPARC in UK) and fifteen orginizations. In China, the China National Grid (CNGrid)[3] and the China Education and Research Grid (China Grid)[4] aim to build one of the biggest super grid platform with more than 15 teraflops of computing power comprised of twelve most famous universities. [1]

This paper describes our preliminary efforts to build a new computational grid platform Rnet Grid distributed around all the provinces of China in Asia and other two continents North America and Europe. The cooperation is between China with National Computer Network Emergency Coordination Center of China and Parallel Computing Technology Laboratory of Harbin Institute of Technology and the US with the University of Houston (Computer Science Department), Germany with the University of Erlangen (RRZE HPC Center and Computer Science Department).

The goal of this paper is to show the performance of a Grid that has been set up between CHINA-USA-GERMANY. This Grid will be mainly used to experiment in Meta Computing environment Mechanical Engineering and CFD applications: for example, Fast Parabolic Solver [5-9], Fast Elliptic Solver [10], Fault Tolerant Algorithms [11-13] and Solution Verification Problems [14]. In this paper, we only study the behavior on this Grid of a Fast Parabolic Solver with convection-reaction-diffusion equations modeling the Air Quality problem.

The machines we have access are very heterogeneous with different performances and network specifications. We will see in the first section a description (hardware and software) of the resources involved in this Grid. Then, in the second section we will give an overview of the Benchmark used on the Grid. Also, we will explain the procedure used to get the appropriate load on each machine, which will allow us to verify the performance of latency-tolerance and load-balance algorithms for the Air Quality application.

[1] Supported by the National Natural Science Foundation of China under Grant No.90412001; the National Grand Fundamental Research 973 Program of China under Grant No.G2005CB321806.

2 Architecture of Rnet Grid Platform

2.1 Hardware and Software of the Rnet Grid

Currently, there are about two hundred and forty hosts totally in the Rnet Grid. About one hundred and ninety hosts of them are deployed in China, which are absolutely controllable distributed on all the thirty-one provinces, municipality and city; thirty-two hosts locate at USA and eighteen ones in Germany. All the hosts in China are divided into thirty-two sites connected with Digital Data Network and connect with foreign hosts by the Internet. The network topology of the Rnet Grid is shown in Fig.1.

We give some hardware and software details on each resource part of the Grid as shown in Table.1 and Table.2. The operating systems of all the hosts are Linux, with gcc and Fortran compilers. The major message-passing library is the PACX-MPI (PArallel Computer eXtension), and MPICH-G2 binding with Globus 2.4.

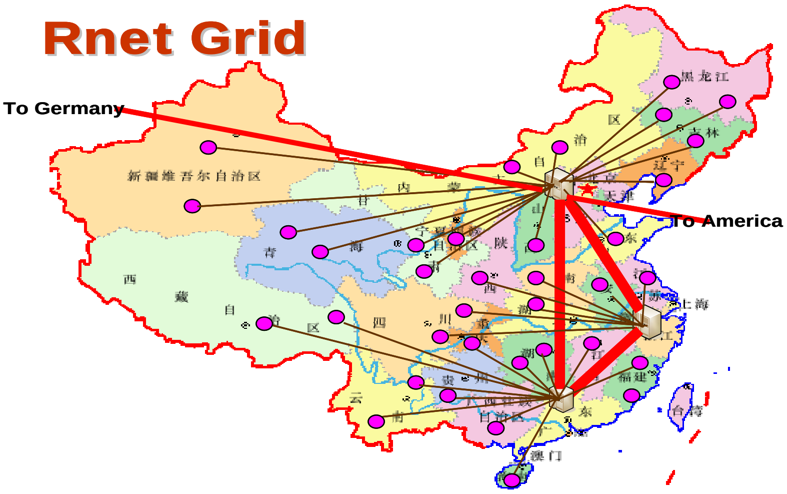

Figure.1. Network topology of the Rnet Grid

We give some hardware and software details on each resource part of the Grid as shown in Table.1 and Table.2. The operating systems of all the hosts are Linux, with gcc and Fortran compilers. The major message-passing library is the PACX-MPI (PArallel Computer eXtension), and MPICH-G2 binding with Globus 2.4.

Table.1. Summary of Rnet Grid resource hardware characteristics

	Hosts Number	CPU	Memory	Network Performance (inside site)		Network Performanc (between site)
				Latency	Bandwidth	
Beijing Center	128	2*Xeon 2.4GHz	2GB	0.2ms	960Mb/s	Average latency 550-600ms; average bandwidth 50-150 Kb/s
31 sub-centers	62 (2 hosts each)	Intel PIII 1.2GHz	512MB	0.1ms	80 Mb/s	
USA	32	2*AMD Athlon 1.5GHz	2GB	0.15 ms	95 Mb/s	
Germany	18	2*AMD Opteron 2.2GHz	2GB	0.05ms	941Mb/s	

Table.2. Summary of Rnet Grid resource software characteristics

	OS	Compiler	Local scheduler	Resource management	Message Passing libraries
China	Linux7.3 2.4.18-3	GCC Intel Fortran	No	Globus 2.4	MPICH1.2.5 PACX-MPI
USA	RedHat	GCC	SGE 5.3	Globus 2.4	MPICH1.2.6

	Linux 3	Intel Fortran	scheduler		PACX-MPI
Germany	Linux 2.6.5-7.	GCC Intel Fortran	PBS Scheduler		MPICH-1.2.5 PACX-MPI

2.2 Message Passing on the Rnet Grid

The Globus project [15, 16] is developing basic software infrastructure needed to build computational Grids across geographically distributed computational and information resources. Grids are persistent environments that enable software applications to integrate instruments, displays, computational and information resources that are managed by diverse organizations in widespread locations.

The library PACX-MPI (PArallel Computer eXtension) [17] allows scientists to gather heterogeneous resources connected through high-speed networks or even Internet in order to run applications requiring a lot of computations. However, with PACX-MPI, we have two processes, called PACX-IN and PACX-OUT, running on each machine part of the Grid, which has to take care of the communication either coming in or coming out the machine.

As a first comparison, regarding these libraries, they have exactly the same goal: connect machines to each other all over the world to create a Grid of resources. Then, allow users to run applications on it just by linking their codes with the wanted library during the compilation. However, the way to set the library up is completely different: For Globus, we need the help of the system administrator for almost all steps during the installation and that can become quickly a nightmare whereas for PACX-MPI, a single user on a machine can set it up easily without having root permissions. Moreover, to run jobs using GLOBUS library, all the parallel machines part of the Grid, even the compute nodes, should have external interface, i.e. public IP address. When we look at our Grid Infrastructure, only some machines have this specificity and it can be hard to ask to all the partners to change their local policy on their machines. However, we do not have this problem with PACX-MPI library because we only need one node with public IP address, in the major cases it is the front-end or login node, which will be used by the two daemons to communicate with the other machines. Therefore, we have decided to link our resources using PACX-MPI library.

3 CFD Application on Rnet Grid Platform

3.1 Air Quality Application Description

Air Quality code is based on reaction-convection-diffusion equations. We remind here the scheme as follow:

$$\frac{\partial C}{\partial t} = \nabla.(K\nabla C) + (\vec{a}.\nabla)C + F(t,x,C),$$

$$C \equiv C(x,t) \in \mathrm{R}^m, K \equiv K(x,t) \in \mathrm{R}^{m\times m}, x \in \Omega \subset \mathrm{R}^3, t > 0$$

This application is a 3D parallel code written in Fortran 90 using MPI library. The code implements also a 2D topology of processors for the communication as shown below in Figure 2. It corresponds to an overlapping domain decomposition that is designed to cope with high latency and low bandwidth for the numerical solution. The code reads first a parameter file containing the number of subdomains, the global size in the Y and Z directions, the local size per subdomain in the X direction, the number of processes in the X and Y directions per subdomain, the number of overlapping points and finally the number of iterations. In fact, the only parameter we have to play with is the size per subdomain in the X direction.

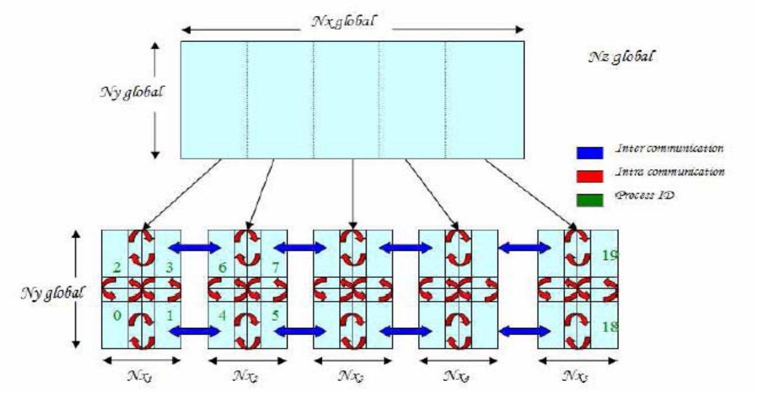

Figure.2. Topology of communications in the Air Quality code with 5 subdomains

3.2 Latency Tolerance Performance of Air Quality Application

To solve the problem (1) efficiently, we use a combination of techniques that consists of the method of characteristics for the convection term and the stabilization with a posteriori filtering, of the explicit treatment of the diffusion term. The salient feature of this method is that the computation per sub-domains is fairly intense, while there are only two local communication between neighboring subdomains per time step that exchange boundaries. Further information about this method could be found in the references at the end of this document.

We have obtained some preliminary results for reaction diffusion solver that was run successfully on the Rnet Grid in China with Professor Garbey on 24th of April 2004. Figure 3 reports the results with only hosts in Beijing Center, 128 dual xeron 2.4GHz; with 2 GB of memory each node and a Gigabit Ethernet network.

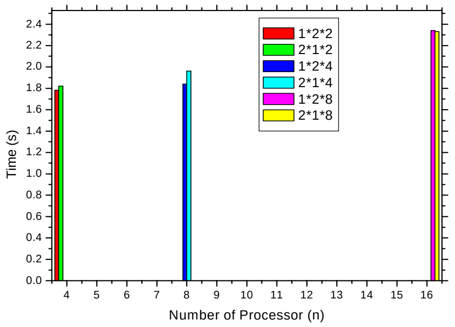

Figure.3. Application performance on Beijing site of Rnet Grid China

Figure.3 gives the elapse time for run with 10 time steps and a fixed size of block per processor that is 0.5*24*56*16, i.e. 43K unknowns. To be more accurate, we have a two-level domain decomposition: the date is partitioned into overlapping macro-bloc first. Each macro bloc itself is decomposed second into two non-overlapping subdomains. The total number of subdomains equals the number of processors, but the algorithm and communications scheme depends on the level of domain decomposition. The novel of the algorithm is to make the parallel scheme highly tolerant to low bandwidth and high latency of communications between macro-bloc. The decomposition into macro bloc is one dimension and the overlap in this space direction is 5 meshes. The global size of the problem is asymptotically equivalent to the number of processors. From Figure 3 we observe that the parallel code scales fairly well. The performance of the code is also independent of the space direction used for splitting the macro-bloc. However the elapsed time varies from one run to the other. The stand deviation is 0.1s up to 8 processors and is 0.4 with 17 processors.

Figure.4 shows the scalability of the algorithm on the Rnet Grid. This first met computing result uses a grid of PCs distributed in eight provinces of China: Si-chuan Province (Cheng du), He-nan Province (Zheng zhou), Shan-xi province (Tai yuan), Shan-dong province, Qing-hai Province (Xi ning), Hai-nan province, Tian-jing and Guang-zhou Province. Each province has two PCs; Intel Pentium III 1.2GHz with cache 512kB, linked by 100 mega Network Card.

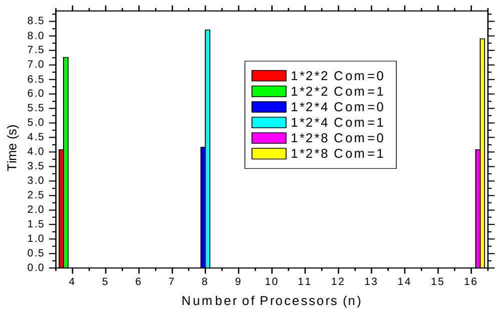

Figure.4. Application performance on 8 sites of Rnet Grid China

The code is set with the exact same geometric configuration. Each macro-domain is processed by two PCs located in the same province. For each group of two vertical bar, the left bar is for the performance of the code without communication between sites, i.e. macro domains. The solution is therefore numerically wrong. The right bar is for the performance with long distance communications. In the second configuration, the numerical solution is second order accurate (in time and space) with a time step proportional to the space step. Each PC of this grid is at least two times slower than the processors of the Beijing site cluster. We observe globally a nice scalability of the code when we increased the number of sites. But we spend almost fifty percent of the time in communications between sites. As opposed to previous runs on the system in Beijing site, we have fairly consistent performance with five successive runs. The Rnet Grid environment is therefore quite stable.

4 Conclusion

In this paper, we present the network topology, hardware and software characteristics of Rnet Grid. On the environment, we get some initial results on an air quality grid application to verify that the platform is stable but also latency-tolerance and load balance algorithms of CFD application are scale well.

References:

[1] The TeraGrid project, http://www.teragrid.org/

[2] The Europe Data Grid, http://egee-intranet.web.cern.ch/egee-intranet/gateway.html

[3] The China National Grid (CNGrid), http://www.cngrid.org/

[4] The China Education and Research Grid, http://www.chinagrid.edu.cn/

[5] F.Dupros, W. E.Fitzgibbon and M. Garbey, A Filtering technique for System of Reaction Diffusion equations, preprint COSC, University of Houston, submitted.

[6] M.Garbey, V.Hamon, R.Keller and H.Ltaief, Fast parallel solver for the metacomputing of reaction-convection-diffusion problems, to appear in *parallel CFD04*.

[7] M. Garbey, H.G.Kaper and N.Romanyukha, A Some Fast Solver for System of Reaction-Diffusion Equations, *13th Int. Conf. on Domain Decomposition DD13*, Domain Decomposition Methods in Science and Engineering, CIMNE, Bracelona, N.Debit et Al edt, pp. 387–394, 2002.

[8] M. Garbey, R. Keller and M. Resch, Toward a Scalable Algorithm for Distributed Computing of Air-Quality Problems, *EuroPVM/MPI03* Venise, 2003.

[9] J. G. Verwer, W. H. Hundsdorfer and J. G. Blom, Numerical Time Integration for Air Pollution Models, MAS-R9825, http://www.cwi.nl, *Int. Conf. on Air Pollution Modeling and Simulation APMS'98*.

[10] M.Garbey, B.Hadri and W.Shyy, A fast elliptic solver for CFD problems on the grid, *43rd Aerospace Sciences Meeting and Exhibit Conference*, Reno January 2005, paper number: AIAA-2005-1386

[11] Gropp and Lusk, Fault Tolerance in Message Passing Interface Programs, *International Journal of High Performance Computing Applications*, No 18, 2004, pp. 363-372.

[12] W.Eckhaus and M.Garbey, Asymptotic analysis on large time scales for singular perturbation problems of hyperbolic type, *SIAM J. Math. Anal*, Vol 21, No 4, pp. 867-883, 1990.

[13] D.A.Murio, *The Mollification Method and the Numerical Solution of Ill posed Problems*, Wiley, New York, 1993

[14] Christophe Picard, Marc Garbey and Venkat Subramaniam Mapping LSE method on a grid: Software architecture and Performance gains, To appear *parcfd2005* at Washington.

[15] I. Foster, C. Kesselman Globus: A Metacomputing Infrastructure Toolkit., *Intl J. Supercomputer Applications*, Vol 11,No 2,1997, pp.115-128.

[16] I.Foster, The Grid: A New Infrastructure for 21st Century Science, *Physics Today*, Vol 55, No 2, 2002, pp. 42-47

[17] HLRS, The library PACX-MPI (PArallel Computer eXtension), http://www.hlrs.de/organization/pds/projects/pacx-mpi/, Germany.