THE 5TH ASIAN COMPUTATIONAL FLUID DYNAMICS
BUSAN, KOREA, OCTOBER 27 ~ 30, 2003

Flow Calculations on GRID Environment

***D. Kim*[1], *Y. Kim*[2], *S.H. Park*[3], *J.H. Kwon*[4]**
Korea Advanced Institute of Science and Technology,
373-1, Guseong-dong, Yuseong-gu, Daejeon, 305-701, Korea
1. dehee@kaist.ac.kr; 2. arcturus@kaist.ac.kr; 3. pish@kaist.ac.kr; 4. jhkwon@kaist.ac.kr
***J.S. Yoon*[5]**
NEUROS, 461-35, Jeonmin-dong, Yuseong-gu, Daejeon, 305-811, Korea
5. jsyoon@neuros.co.kr
***K.W. Cho*[6]**
Korea Institute of Science and Technology Information,
52, Oeun-dong, Yuseong-gu, Daejeon, 305-806, Korea
6. ckw@hpcnet.ne.kr

Corresponding author ***J.H. Kwon***

This paper presents some flow calculations on GRID computing environment. Through several experiments on GRID, we shall review merits, defects and requirements of GRID computing. Though CFD users can be provided with tremendous resources, they could wait for a long time to get their solutions. Since GRID technology is implemented on WAN, the state of network is essential for efficient computation. In this work, we show how much bandwidth is necessary for general CFD calculation and whether parallel efficiency on GRID is endurable to users. Several test cases present the practical points of view on GRID computing for CFD users.

Advancement of IT technology has following two characteristics: Firstly, the growth of CPU performance is getting saturated. Secondly, the growth rate of network speed is far faster than that of CPU and storage capacity. During 1986-2000, computer speed became 500 times faster, at the same time, the networks were improved by 340,000 times.

Therefore proper construction of the network connecting distributed resources could be more efficient than waiting for a new high performance CPU and high capacity of storage media. This is the motivation of GRID technology.

Resources for GRID computing include vast, heterogeneous, and geographically-distributed resources. So, middlewares such as Globus have been developed for computations that integrate those computational resources. On the middleware, we can work in high level programming environment and transparent computational environment even between different computer architectures. Since difficulties related to different characteristics such as OS, location, size, connectivity, name, and so on, can be hid by using interface such as Globus, users can utilize the system as a unified machine.

GRID computing is an important and popular new field. It is distinguished from conventional distributed computing in that it focuses on large-scale resources sharing, innovative applications. Large-scale science and engineering is typically done through the interaction of people, heterogeneous computing resources, multiple information systems, and instruments, all of which are geographically and organizationally dispersed. The overall motivation for "GRIDs" is to enable the routine interactions of these resources to facilitate this type of large-scale science and engineering.

The GRIDs can be classified as COMPUTAIONAL GRID, DATA GRID, and ACCESS GRID. COMPUTATIONAL GRID is the conventional metacomputing infrastructure, DATA GRID is for the purpose of large scale data processing and management that require the participation of world wide researchers, and ACCESS GRID provides human interface for COMPUTAIONAL GRID and DATA GRID(e.g. real time video conference with P2P based remote control).

As the problem size becomes huge, we need more computing resources. Large size of computing resources could be provided for computing by GRID. Tremendous resources can be connected by using GRID technology and unlimited number of processors can be utilized for computing.

Though GRID computing is attractive, it has some defects compared with conventional local parallel systems. Since GRID is connected through WAN, cumbersome problems related with network can arise unlike local systems on private network.

In this work, numerical analyses about several test cases on GRID computing environment were tried. The experiments were performed on a simple GRID testbed consisting of Linux PC clusters. Of course, various platforms could be connected to testbed. Our testbed has been composed of two clusters at KAIST and KISTI. The bandwidth of local system is 100Mbps fast Ethernet ideally, but the network isn't on private network any more, so its real bandwidth is influenced by the network traffic. The real bandwidth between two institutes is about 2~70Mpbs.

In some test cases to check network performance, resources at Soongsil Univ.(Seoul) and Tongmyung Institute of Technology(Busan) are connected over GRID. For data communications, MPICH-G2 is used and PBS(Portable Batch System) is used as a jobmanager.

For numerical tests, four cases are presented. Firstly, the flow over the DFVLR axial fan was computed to show the effect of network performance. The second case is the RAE2822 airfoil for the performance comparison of data communications. The third one is 3-dimensional wing design optimization to review the parallel efficiency on GRID. Lastly a large-scale computing was performed about the DLR-F6 configuration with pylon and nacelle.

Effect of Network Bandwidth

Computing time in the flow calculation can be influenced seriously by the state of network. To show the effect of network bandwidth some experiments have been performed between two cities (Seoul ⇔ Daejon). 2 nodes(1 node at KAIST and 1 node at Soongsil Univ.) are used. Test case is the DFVLR axial fan with 28 blades and grid system used is 45x19x19. The network bandwidth was controlled by QoS. As the results show, in the case of 2Mbps overhead from GRID is nearly 70%. If the bandwidth becomes greater than 10Mbps, the overhead decreases within 10%. So, we can conclude that the numerical analysis can be performed without any severe degradation of performance if the bandwidth is more than 5 Mbps in this case. Of course, the data size of communications could be large, but it may not be so large if the domain is decomposed properly. From this experiment we can see approximate requirements of GRID computing.

Varied network latency in a day can influence the efficiency of computation, too. This experiment has been performed between two cities (Busan ⇔ Daejon) to this effect. 2 nodes are used from two institutes. The ratio of the smallest computing time to the largest computing time is greater than 4 times.

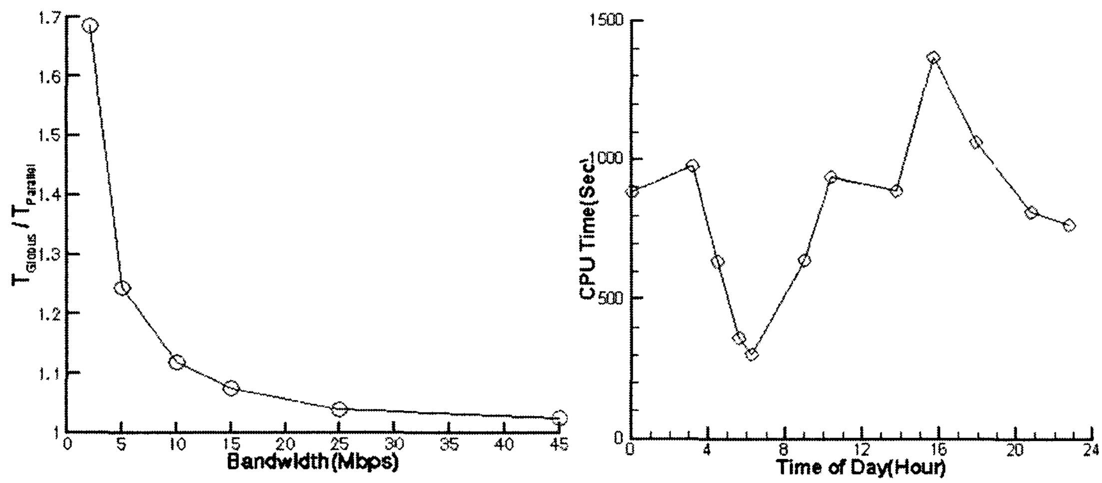

Fig. 1 Effect of bandwidth(left), computation time in a day(right)

Flow Calculations

Numerical analysis of RAE 2822 airfoil under a transonic turbulent flow condition was performed for the comparison between a conventional local parallel environment and the GRID computing environment.

8 nodes at KAIST, 8 nodes at KISTI and 8 nodes at KAIST-KISTI are used for the test. From the results, we can see that the communication overhead of GRID computing increase over 20% compared with the conventional local parallel computing in the daytime, while nearly negligible at night.

Besides the effects of bandwidth and varying latency, MPICH-G2 has properties such as reduced latency for intramachine messaging and increased latency for intermachine messaging, so the performance between intermachines could be poor a little.

Another test case is the drag minimization of the ONERA M6 wing. By using 4, 8, 24 CPUs with same number of nodes from each institute, we could compare the parallel efficiency between results with MPICH and MPICH-G2. Conventionally parallel efficiency shows about 100% using up to 24 CPUs, but the result doesn't show the same trend because Globus is working only on the public network. One of several defects about Globus is to use the public network even within a local parallel system, so the network traffic on LAN influences the computing. This can be seen apparently from Fig. 2.

The DLR-F6 wing-body-pylon-nacelle(WBPN) configuration is the last test case. For the WBPN configuration, the grid system is composed of 11 blocks and has 6 million grid points. For calculation, 21 nodes are used over GRID and the computing time is about 35 hours. Compared with the results with MPICH, the overhead is nearly 20%.

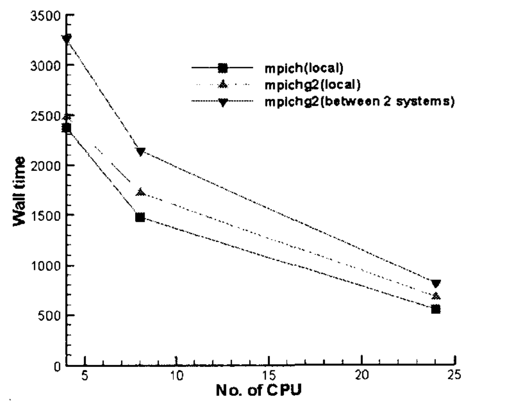

Fig. 2 Computing time according to number of CPUs

Flow calculations have been performed on the GRID environment to show that general CFD calculations could be applicable on GRID. Characteristics of GRID computing were reviewed. On the GRID computing environment, to accomplish calculations without severe loss of efficiency, a proper network bandwidth should be provided, but the required bandwidth isn't unreasonable under present state of network. Furthermore, we must consider the varying latency. Since the network latency varies seriously within a day and the CPU performance used for computations is different from each other, it is necessary to select resources within GRID resource pool more cautiously and reasonably to consider load balancing.

If many processors from several geographically-distributed institutes are used for computation, then the overhead caused by using this GRID would be increased, but we could still perform our work in a practical computing time.

HTC(High-Throughput Computing) can be a promising strategy for GRID computing. Methods such as the genetic algorithms, the response surface methods are good examples of HTC.