

그리드 컴퓨팅을 이용한 열유동 해석 기법에 관한 기초 연구

홍승도* · 하만영† · 조금원**

(2004년 1월 30일 접수, 2004년 3월 16일 심사완료)

A Basic Study of Thermal-Fluid Flow Analysis Using Grid Computing

Seung Do Hong, Man Yeong Ha and Kum Won Cho

Key Words : Grid(그리드), CFD(전산유체역학), Cluster(클러스터), Parallel Computing(병렬 컴퓨팅), MPI(메시지 패싱 인터페이스)

Abstract

Simulation of three-dimensional turbulent flow with LES and DNS takes much time and expense with currently available computing resources and requires big computing resources especially for high Reynolds number. The emerging alternative to provide the required computing power and working environment is the Grid computing technology. We developed the CFD code which carries out the parallel computing under the Grid environment. We constructed the Grid environment by connecting different PC-cluster systems located at two different institutes of Pusan National University in Busan and KISTI in Daejeon. The specification of PC-cluster located at two different institutes is not uniform. We run our parallelized computer code under the Grid environment and compared its performance with that obtained using the homogeneous computing environment. When we run our code under the Grid environment, the communication time between different computer nodes takes much larger time than the real computation time. Thus the Grid computing requires the highly fast network speed.

1. 서론

시대의 흐름에 따라, 컴퓨터의 성능은 급격히 증가해 왔으며 이러한 컴퓨터를 사용하여 순수과학 및 응용공학 분야가 크게 발전하였다. 전산 유체역학 분야에 있어서도 1970년대 초반의 컴퓨터로는 압축성 포텐셜 유동 방정식에 대한 수치해를 구하여 3차원 날개 주위의 유동을 해석할 수 있었다. 난류 유동은 2차원 익형 주위의 유동 해석이 가능한 정도였다. 1980년대 초반 3차원 항공기 전체의 포텐셜 유동과 3차원 날개 주위의 난류 유동을 해석할 수 있을 정도까지 컴퓨터 하드웨어

기술이 발전하였다. 1990년대에 들어오면서 3차원 항공기 전체 형상 주위의 난류 유동을 시뮬레이션할 수 있는 단계까지 컴퓨터가 발전하였다.⁽¹⁾ 특히 난류 유동 해석을 위하여 LES(Large Eddy Simulation) 및 DNS(Direct Numerical Simulation) 기법을 사용하여 계산을 수행할 때 경험하는 가장 어려운 점의 하나는 난류의 가장 작은 스케일인 Kolmogorov 스케일의 와동 구조를 해석할 수 있어야 한다. 이를 위하여 엄청나게 많은 수의 계산 격자가 요구되어진다. DNS 기법을 사용하여 3차원 계산을 수행할 시 $Re^{4/3}$ 의 계산 격자수가 필요하다. 여기서 Re 는 유동의 관성력과 점성력의 상대적인 비인 무차원 레이놀즈 수를 나타낸다. 그러므로 3차원 난류 유동 계산을 수행하기 위하여 엄청나게 큰 계산 시간뿐만 아니라, 대용량의 메모리와 저장 장치가 필요하다. 이러한 컴퓨팅 파워의 한계를 극복하기 위하여 다수의 CPU를 병렬로 연결하여 하나의 주어린 문제를 협동적으로 계산하는 병렬 컴퓨팅 기법이 최근에 개발되었다.

+ 책임저자, 회원, 부산대학교 기계공학부
E-mail : myha@pusan.ac.kr
TEL : (051)510-2440 FAX : (051)512-9835
* 부산대학교 대학원 기계공학과
** 한국과학기술정보연구원 슈퍼컴퓨팅센터

Table 1 Specification of system

Location		PNU-RIMT	KISTI	PNU-HPSC
Computing environment		-	LAN(With Globus)	LAN(Without Globus)
		WAN(With Globus)		
System name		Wake	Jupiter & Venus	Supreme
System architecture		PC-Cluster	PC-Cluster	PC-Cluster
No. of nodes		24	80	64
Specification	Process	P4 2.0GHz	P4 2.0GHz	P3 550MHz
	Memory	512MB	512MB	1GB
	OS	Redhat Linux 7.3	Redhat Linux 7.3	Redhat Linux 7.1
Network		FastEthernet	FastEthernet	Myrinet

이러한 개념 하에 개발된 병렬 컴퓨터를 ‘Cluster (클러스터)’라고 부른다. 최근에 단일 CPU의 성능 및 네트워크의 속도가 급격하게 발전함에 따라 클러스터를 이용한 병렬 컴퓨팅 기법도 빠르게 발전하고 있다. 그러나 현재의 기술로는 수천, 수만 대 이상의 컴퓨터를 고성능의 네트워크로 연결하여 단일 병렬 컴퓨터를 구성하는 기술은 간단하지 않으며, 아직 풀어야 하는 문제가 많이 남아있다. 그리드 컴퓨팅 기술은 이러한 컴퓨팅 파워의 한계를 극복할 수 있는 좋은 대안으로 평가 받고 있으며, 최근 많은 연구자들이 많은 관심을 보이고 있다. 그리드란 지역적으로 분산되어 있는 고성능 컴퓨터, 대용량 저장 장치, 첨단 실험 장비 등을 고속의 네트워크로 연동한 통합 환경을 말한다. 그리드라는 용어는 1990년대 중반 미국의 슈퍼 컴퓨팅 센터를 중심으로 고성능의 분산 컴퓨팅 인프라를 구축하는 데서 시작하였다. 고성능 자원의 구축, 대용량 정보의 사용 및 혁신적인 응용에 초점이 맞추어져 있는 그리드 컴퓨팅은 일반적인 분산 컴퓨팅과는 구별된다.⁽²⁻⁴⁾

그리드는 해석하고자 하는 문제와 해석 방법에 따라 계산(Computational) 그리드, 데이터(Data) 그리드, 액세스(Access) 그리드로 구분할 수 있다. 계산 그리드는 고성능의 컴퓨팅 파워를 필요로 하는 고성능(high-performance) 컴퓨팅과 일정 시간 내에 많은 문제를 해결할 수 있는 고처리(high-throughput) 컴퓨팅으로 구분할 수 있다. 데이터 그리드는 대규모의 데이터 처리가 필요한 연구 기관들이 협력하여 데이터의 저장 및 송수신을 가능하게 하는 그리드 환경을 구축하고 이를 활용하는

것을 의미한다. 액세스 그리드는 원거리의 연구자들이 공동의 연구 결과를 공유할 수 있도록 화상 회의 시스템과 같은 협업 환경을 제공하는 것을 의미한다.

본 연구에서는 그리드 환경을 위한 테스트 베드를 구축하였다. 본 연구실에서 개발된 테스트 베드를 타 기관의 클러스터와 연동하여 그리드 환경을 구축하였다. 또한, 본 연구실에서 개발한 열유동 해석 병렬 프로그램을 사용하여, 그리드 환경의 성능을 테스트하였다.

2. 연구 내용

2.1 테스트 베드 구축

Table 1은 그리드 환경의 구축을 위하여 본 연구를 통해 자체 제작한 PC-클러스터(Wake) 시스템 및 여러 계산 환경을 만들기 위해 사용한 PC-클러스터의 사양을 나타내고 있다. 그리드 서비스를 제공하는 그리드 미들웨어로 Globus 2.0⁽⁵⁾를 설치하였으며, 그리드 환경 하에서 병렬 컴퓨팅 및 Job Scheduler의 수행을 위한 소프트웨어로서 MPICH-G2⁽⁶⁾ 및 OpenPBS⁽⁷⁾를 설치하였다.

2.2 그리드 환경 구축

그리드 환경 구축을 위해서는 지역적으로 분산되어 있는 클러스터들이 필요하다. 본 연구에서는 그리드 환경의 구축을 위하여 자체적으로 구축한 PC-클러스터와 KISTI(한국과학기술정보연구원)에서 구축한 테스트 베드를 선도시험망을 통해 연결하였다. Figure 1은 본 연구에서 사용한 서로 다른

3 가지 계산 환경을 보여주고 있다. 각 계산 환경에서 사용한 시스템의 사양은 Table 1 에 정리하였다.

LAN(Without Globus)은 부산대 공동실험실습관에 설치되어 있는 리눅스 클러스터 시스템을 나타낸다. 이 시스템은 일반 병렬 컴퓨팅을 위하여 제작된 것이며, 그리드 미들웨어인 Globus 는 설치되어 있지 않다. 단일 계산 노드의 CPU 는 P3-550MHz 로 성능이 상대적으로 떨어지지만, 네트워크는 전송 속도가 상대적으로 매우 빠른 미리넷(Myrinet)으로 연결되어 있어 노드 간 통신 성능은 대단히 좋다.

LAN(With Globus)은 KISTI 에서 PC-클러스터를 사용하여 구축한 테스트 베드를 의미한다. 그리드 환경의 구축을 위하여 그리드 미들웨어인 Globus 가 설치되어 있다. 단일 계산 노드의 CPU 는 P4-2.0GHz 로 성능이 상대적으로 좋다. 통신을 위한 네트워크는 100Mbps 의 패스트 이더넷(FastEthernet)으로 연결되어 있으므로 LAN(Without Globus)보다는 노드 간 통신 성능은 떨어진다.

WAN(With Globus)은 부산대 기기연에 위치하고 있는 본 연구실에서, 직접 구축한 PC-클러스터와 KISTI 의 테스트베드를 선도 시험망을 통해 연동하여 구축한 시스템이다. LAN(Without Globus) 및 LAN(With Globus)은 단일 기관에서 구축한 병렬 계산 시스템을 나타낸다. 그러나 WAN(With Globus)은 지리적으로 멀리 떨어져있는 두 기관의 시스템을 네트워크망을 통해 연결하여 하나의 클러스터 시스템처럼 사용하기 때문에 이것은 그리드 환경을 나타낸다.

본 연구에서는 이러한 서로 다른 세가지 계산 환경 하에서 계산을 수행하여 그들의 성능을 비교 분석한다. 이를 통하여 그리드 환경을 사용하여 계산을 수행할 때 발생할 수 있는 현상을 조사한다. 각 계산 환경을 Figure 1 에 나타내었으며 외부의 사용자는 화살표로 분류된 각 환경으로 접속하여 해당 환경에서 컴퓨팅을 수행하게 된다.

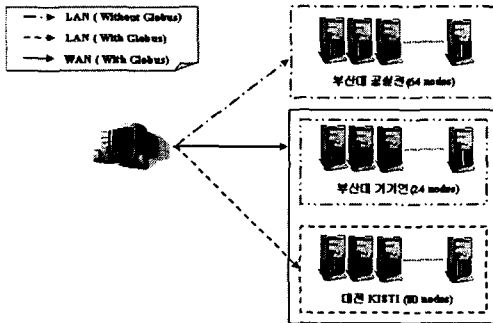


Fig. 1 Different computing environments used in the present study

2.3 계산 그리드

Figure 1 에 보여준 서로 다른 3 가지 계산 환경의 성능을 비교 분석하기 위하여 아래의 경우들에 대한 계산을 수행하였다.

Figure 2 는 레이놀즈 수가 400 일 때 driven cavity 내에서의 유동을 계산하는 첫 번째 계산 예를 보여주고 있다.⁽⁶⁾ 계산에 사용한 격자 수는 40,000~160,000 개 정도이며, 사용한 계산 격자 수로 보아 본 연구에서는 비교적 작은 사이즈의 문제에 해당한다. 통신량 대 한 개의 계산 CPU 노드가 담당하는 문제 (격자 수)의 비를 의미하는 Surface to volume ratio 를 변화시키면서 다음 두 가지 경우에 대한 테스트를 수행하였다. case 1 은 계산에 사용된 총 격자 수를 고정시킨 경우에 해당한다. 그러므로 계산에 사용된 노드 수가 증가할수록 각각의 계산 노드가 담당하는 계산 격자 수는 줄어든다.

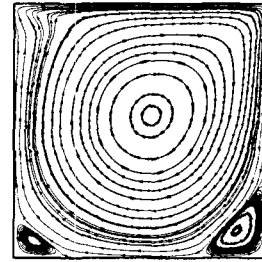


Fig. 2 Flow in the driven cavity

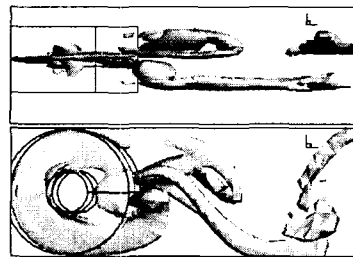


Fig. 3 Flow past a cylinder with circular fins



Fig. 4 Flow around tube banks

Table 2 CPU nodes, total number of grid points and geometry used for different cases

	CPU node	Total number of grid points	Geometry
Case 1	4	70,000	Driven Cavity
	9	70,000	
	16	70,000	
Case 2	4	40,000	Driven Cavity
	9	90,000	
	16	160,000	
Case 3	12	700,000	Cylinder with Circular Fins
Case 4	24	2,000,000	Tube Banks

case2는 한 계산 노드가 담당하는 계산 격자 수를 고정시킨 경우에 해당한다. 그러므로 계산에 사용된 CPU노드 수가 증가할수록 계산에 사용된 총 계산 격자 수는 증가하게 된다.

Figure 3 은 레이놀즈 수가 300 일 때 원형 핀을 가지는 3 차원 실린더 주위의 유동을 계산하는 예를 보여주고 있으며 case 3 에 해당한다. 계산에 사용한 격자 수는 70 만개이며, 사용한 격자 수로 보아 본 연구에서는 중간 사이즈의 문제에 해당한다.

Figure 4 는 레이놀즈 수가 4,000 일 때 3 차원 관군 주위의 유동을 계산하는 예를 보여주고 있으며 case 4 에 해당한다. 계산에 사용한 격자 수는 200 만개이며, 사용한 격자 수로 보아 본 연구에서는 가장 큰 사이즈의 문제에 해당한다.

위에서 설명한 모든 case 를 Table 2 에 정리하였다.

3. 이론적 연구

3.1 지배 방정식

그리드 환경 하에서 Fig. 2 ~ Fig. 4 에 보여준 열유동 문제에 대한 수치해를 구하기 위하여 정의되어진 무차원 질량, 운동량 및 에너지 보존 방정식은 다음과 같다

$$\nabla \cdot \mathbf{u} = 0 \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u} \quad (2)$$

$$\frac{\partial \theta}{\partial t} + \mathbf{u} \cdot \nabla \theta = \frac{1}{Re \cdot Pr} \nabla^2 \theta \quad (3)$$

여기서 무차원 변수들은 다음과 같이 정의되어

진다.

$$t = \frac{U_{ref} t^*}{L_{ref}}, \quad \mathbf{u} = \frac{\mathbf{u}^*}{U_{ref}}, \quad p = \frac{p^*}{\rho U_{ref}^2}, \quad \theta = \frac{T - T_{ref}}{\Delta T_{ref}} \quad (4)$$

여기서 U_{ref} , L_{ref} , ΔT_{ref} 는 각각 무차원화를 위한 특성 속도, 길이 및 온도차를 나타낸다. 식 (4)에서 \mathbf{u} , t , p 및 θ 는 각각 무차원 속도 벡터, 시간, 압력 및 온도를 나타낸다. 또한 상첨자 *는 변수들이 차원을 가지고 있는 것을 나타낸다. 무차원 방정식 (1)~(3)으로부터 레이놀즈 수 ($Re = U_{ref} L_{ref} / \nu$) 및 프란틀 수 ($Pr = \nu / \alpha$)가 중요한 무차원 변수로 정의 되어진다. 여기서 ν 및 α 는 유체의 동점성 계수 및 열확산 계수를 나타낸다.

3.2 수치해석 및 병렬화 기법

무차원 지배방정식 (1)~(3)에 대한 공간적분을 위하여 스펙트럴 방법 (spectral method) 혹은 유한체적법(finite volume method)을 사용하였다. 시간에 대한 적분 방법으로는 2 단계 시간 분할법 (two-step time-splitting method)를 사용하였다. 비선형 방정식인 대류항에 대하여는 Adams-Bashforth 그리고 확산항에 대하여는 Crank-Nicolson 방법을 사용하였다.⁽⁹⁾

Fig. 2 및 Fig. 3 의 case 1 ~3 에 해당하는 계산은 모델없이 직접 계산을 수행하였다. 그러나 Fig. 4 의 case 4 의 경우는 LES(large Eddy Simulation) 기법을 사용하여 계산을 수행하였다.⁽¹⁰⁾

병렬 컴퓨팅 계산을 위하여 MPI(Message Passing Interface) 기법을 사용하였다. MPI 기법은 병렬 컴퓨팅의 기법 중 메시지 패싱에 기본을 둔 프로그

래명 방법이다. 메시지 패싱은 이미 여러 가지의 병렬 컴퓨팅 방법에서 다 년간의 시험을 거쳐 테스트 한 프로그래밍 기법이다. 그러나 MPI 표준이 나오기 이전까지는 모든 병렬 컴퓨팅이 각기 다른 형태의 프로그래밍 인터페이스를 제공하여 사용자 프로그램의 호환성에 문제가 있었다. 이러한 문제점을 인식하고 일정한 문법과 공통 사항을 마련하고자 Message Passing Interface Forum 을 결성하고 수년간 노력한 끝에 1994 년에 제정한 것이 MPI(v1.0)이다.⁽¹¹⁾

본 연구에서 고려하는 모든 계산 경우들에 대한 병렬화를 위하여 영역 분할법(DDT: Domain Decomposition Technique)이 사용되었다. 영역 분할법은 계산 영역을 분할하여 계산을 수행한다. 또한 이 방법을 사용하면 영역을 소 영역으로 분할하는 것이 가능하기 때문에 복잡한 형상을 가진 영역에 대한 계산도 용이하다. 이 방법을 사용할 때 프로그램의 확장 및 이식 성능이 아주 우수하다. 본 병렬 계산에서는 Virtual topology 방법을 사용하였다.⁽¹²⁾ 이때 병렬 계산에 참여하는 프로세서들은 2 차원 혹은 3 차원 배열로 가상 배치되어지고, 각각의 프로세서들은 고유의 독립된 좌표계를 가진다.

4. 결과 및 고찰

Fig. 5 는 Fig. 2 의 case 1 에 대하여 계산에 사용한 노드 수의 증가에 따른 노드 간 통신에 소요된 시간을 비교한 결과를 보여주고 있다. 통신에 소요된 시간이란 프로그램의 실행 중 병렬계산에 사용되는 라이브러리를 호출하는데 사용된 시간을 나타낸다. 계산에 사용된 모든 계산 노드들에서 측정된 뒤 이들의 평균을 취하였다. 계산에서 고려한 3 가지의 모든 환경에 대하여 노드 수의 증가에 따라 전송 속도는 전체적으로 증가하고 있다. 그리드 환경인 원거리에 위치한 두 기관 (부산의 부산대학교 및 대전의 KISTI)의 서로 다른 이종의 시스템 (heterogeneous system)으로 구성된 WAN(With Globus) 환경에서의 전송 속도는 단일 기관에서 동일한 시스템 (homogeneous system)으로 구성된 LAN(Without Globus) 및 LAN(With Globus) 환경에서의 전송 속도에 비하여, 4 개의 노드를 사용할 경우 약 8~40 배 정도 더 느리고 16 개의 노드를 사용할 경우 약 2~13 배 정도 더 느리다. 이것은 원거리에 있는 이종 시스템을 네트워크로 연결하여 그리드 환경을 구축하여 계산을 수행할 때 전송 속도가 전체 계산의 성능에 지대한 영향을

미칠 수 있다는 것을 보여주고 있다. 그러므로 그리드 환경을 구축할 때 초고속의 네트워크를 사용하는 것이 대단히 중요하다는 것을 알 수 있다. LAN(Without Globus)의 전송 속도는 LAN(With Globus)의 전송 속도에 비하여 전송 속도가 빠르다. 이것은 동일 기관에서 동일한 시스템으로 컴퓨팅 환경을 구축할 때, 미리넷이 패스트 이더넷에 비하여 전송 속도가 빠름을 보여주고 있다.

Fig. 6 은 Fig. 2 의 case 1 에 대하여 계산에 사용한 노드 수의 증가에 따른 계산에 소요된 시간을 비교한 결과를 보여주고 있다.

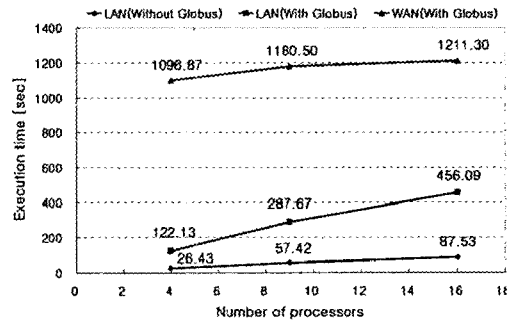


Fig. 5 Variation of communication time in Case 1

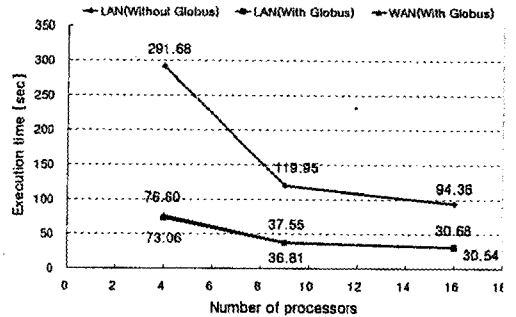


Fig. 6 Variation of computation time in Case 1

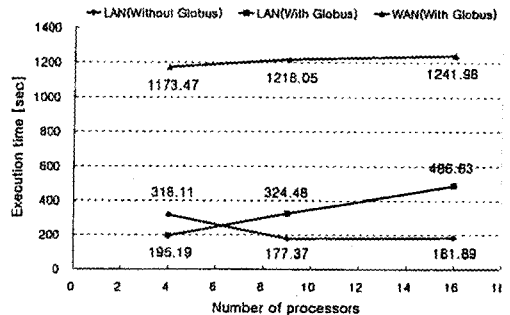


Fig. 7 Variation of total run time in Case 1

계산에 소요된 시간이란 전체 프로그램을 실행한 시간에서 통신에 사용된 시간을 뺀 것이다. 계산에 사용된 모든 계산 노드들에서 측정된 뒤 이들의 평균을 취하였다. 계산에서 고려한 3 가지의 모든 환경에 대하여, 노드 수의 증가에 따라 계산 시간이 전체적으로 감소하고 있다. 즉 여러 노드를 사용하여 병렬 컴퓨팅을 수행할 때 사용하는 노드 수가 증가함에 따라 계산에 소요되는 시간이 감소하는 것을 알 수 있다. LAN(With Globus) 및 WAN(With Globus) 환경을 사용할 때 소요된 계산 시간은 노드 수가 증가함에 따라 유사하게 감소한다. 그러나 LAN(Without Globus)을 사용하여 계산을 수행할 때 소요된 계산 시간은 LAN(With Globus) 및 WAN(With Globus) 환경을 사용할 때에 비하여 보다 많은 계산 시간이 소요된다. 이것은 Table 1 에 보는 것처럼 LAN(Without Globus)을 구성하는 각 노드 CPU 의 성능이 LAN(With Globus) 및 WAN(With Globus)에 비하여 훨씬 떨어지기 때문이다.

Fig. 7 은 Fig. 2 의 case 1 에 대하여 계산에 사용한 노드 수의 증가에 따른 총 계산에 소요된 시간을 비교한 결과를 보여주고 있다. 여기서 총 계산 시간은 통신 및 계산에 소요된 시간의 합을 나타내고 있다. 미리넷을 통신 장비로 사용하는 LAN(Without Globus) 환경에서는 좋은 병렬 성능을 보여주며, 노드 수의 증가에 따라 총 계산 시간이 감소하고 있다. 그러나 패스트 이더넷을 통신 장비로 사용하는 LAN(With Globus) 및 WAN(With Globus)에서는 노드 수가 증가함에 따라 총 계산 시간이 전체적으로 증가한다. 이것은 노드 수가 증가함에 따라 계산 시간은 감소하지만 통신 시간이 상대적으로 많이 증가하기 때문에 노드 수가 증가함에 따라 LAN(With Globus) 및 WAN(With Globus) 환경에서의 총 계산 시간이 증가하게 된다. 이러한 문제는 case 1 처럼 계산하는 문제의 사이즈 (사용하는 격자의 수)가 작은 문제에서는 통신 장비의 통신 속도가 상대적으로 느린 경우 발생할 가능성이 있다. 그리드 환경인 WAN(With Globus)에서의 총 계산 시간은 LAN(Without Globus) 및 LAN(With Globus) 환경에서의 총 계산 시간에 비하여 4 개의 노드를 사용할 경우 약 4~6 배 정도 더 느리고, 16 개의 노드를 사용할 경우 약 3~7 배 정도 더 느리다. 이것은 앞에서 설명한 것처럼 그리드 환경에서 전송 속도를 향상시키는 것이 그리드 컴퓨팅에서 대단히 중요하다는 것을 보여주고 있다.

Fig. 8 은 Fig. 2 의 case 2 에 대하여 계산에 사용한 노드 수의 증가에 따른 노드 간 통신에 소요된 시간을 비교한 결과를 보여주고 있다. case 2 의 경

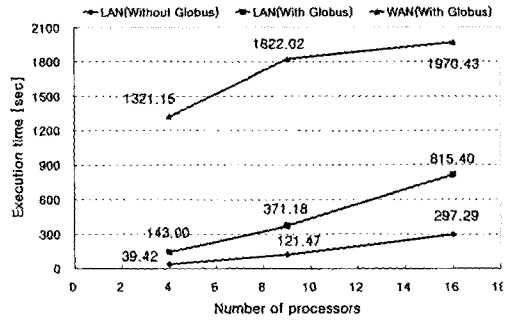


Fig. 8 Variation of communication time in Case 2

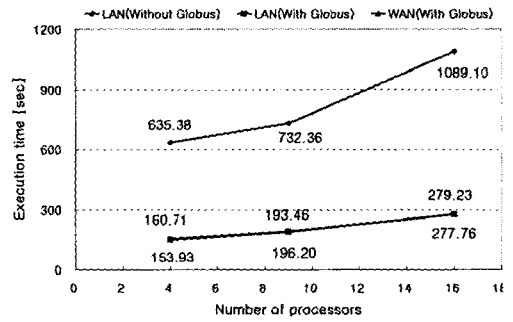


Fig. 9 Variation of computation time in Case 2

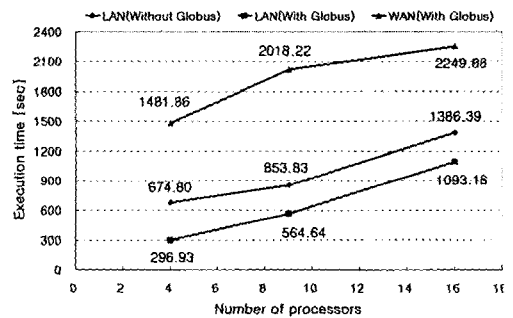


Fig. 10 Variation of total run time in Case 2

우 단일 노드에 할당된 격자의 수가 동일하기 때문에, 계산에 사용한 노드 수에 상관없이 이론적으로는 통신 및 계산에 소요된 시간과 이들을 합한 총 계산 시간이 동일하여야 한다. 그러나 Fig. 8 에 보여준 것처럼 통신 시간은 노드 수가 증가함에 따라 증가하고 있다. 이것은 case 2 의 경우도 역시 계산하는 문제의 사이즈가 작고, 병렬 계산에 소요되는 오버헤드(overhead) 때문에 발생한다. case 1 과 유사하게 case 2 의 경우도 그리드 환경인

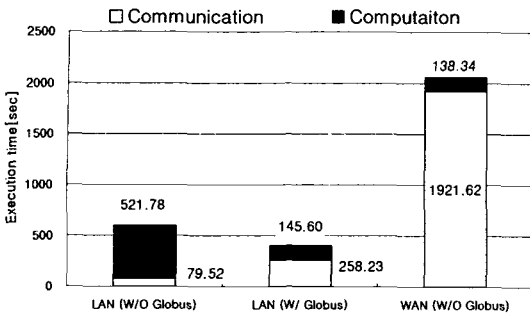


Fig. 11 Variation of run time in Case 3

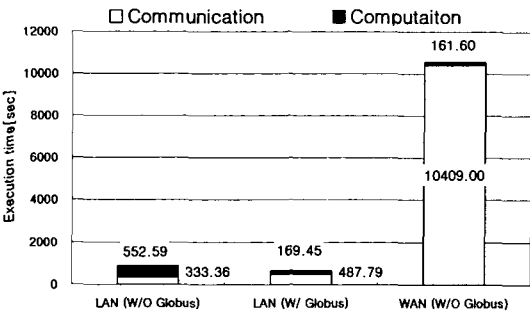


Fig. 12 Variation of run time in Case 4

WAN에서의 전송 속도가 LAN 환경들에 비하여 4개의 노드를 사용할 경우 약 10~30 배 정도 더 느리고, 16개의 노드를 사용할 경우 약 3~7 배 정도 더 느리다.

Fig. 9 및 10은 Fig. 2의 case 2에 대하여 계산에 사용한 노드 수의 증가에 따른 계산에 소요된 시간 및 통신과 계산에 소요된 총 계산 시간을 비교한 결과를 보여주고 있다. WAN 및 LAN 환경에서 사용한 노드 수의 변화에 관계없이 이론적으로는 동일한 성능이 나와야 하나, 앞에서 설명한 이유에 의하여 노드 수의 증가에 따라 이들의 속도가 3가지 환경 모두에서 증가하고 있다. Fig. 10에 보여준 것처럼 case 2의 경우 case 1과 유사하게 그리드 환경인 WAN에서의 총 계산 시간은 LAN 환경에 비하여 4개의 노드를 사용할 경우 약 2~5 배 정도 더 느리고, 16개의 노드를 사용할 경우 약 2~3 배 정도 더 느리다.

Fig. 11은 Fig. 3의 case 3에 대하여 3가지 다른 환경에서 통신 및 계산에 소요된 시간 및 이들의 합인 총 계산 시간을 보여주고 있다. 여기서 사용된 계산 노드 수는 12개로 고정되었다. 앞의 case

1 및 case 2와 유사하게 계산 시간은 사용하는 환경에 관계없이 LAN 및 WAN 환경을 구성하는 각 CPU의 성능에 따라 결정된다.

통신에 소요되는 시간은 WAN 환경은 LAN (Without Globus)에 비하여 약 24 배 그리고 LAN (With Globus)에 비하여 약 8 배 정도 더 많이 걸린다. 이로 인하여 WAN을 사용할 때 총 계산 시간은 LAN(Without Globus)에 비하여 약 3 배 그리고 LAN(With Globus)에 비하여 약 5 배 정도 더 많이 걸린다.

Fig. 12는 Fig. 4의 case 4에 대하여 3가지 다른 환경에서 통신 및 계산에 소요된 시간 및 이들의 합인 총 계산 시간을 보여주고 있다. 여기서 사용된 계산 노드 수는 24개로 고정되었다. 전체적으로 Fig. 11의 case 3과 유사한 결과를 보여주고 있다. Case 4는 계산하는 문제의 사이즈(사용하는 격자의 수)가 가장 큰 문제에 해당하고 사용하는 노드의 수가 24개이므로, 그리드 환경인 WAN에서 통신에 소요되는 시간이 LAN 환경에 비하여 대단히 크다. 그러므로 WAN을 사용할 때 총 계산 시간은 LAN(Without Globus)에 비하여 약 12 배 그리고 LAN(With Globus)에 비하여 약 16 배 정도 더 많이 걸린다.

5. 결론

- 본 연구에서는 최근에 국내외에서 연구가 활발히 진행되고 있는 그리드 컴퓨팅과 관련하여, 소규모의 그리드 환경을 직접 구축하였다.
- 그리드 컴퓨터 환경 하에서 계산이 가능한 병렬 컴퓨팅 프로그램을 개발하여 적용하였다.
- 다양한 컴퓨팅 환경 하에서 개발된 병렬 컴퓨터 프로그램의 수행을 통하여 여러 컴퓨팅 환경의 성능을 비교하였다.
- 그리드 환경 하에서 계산을 수행할 때 계산보다는 통신에 보다 많은 시간이 소요된다. 이러한 통신에 소요되는 시간을 줄이기 위하여 초고속의 네트워크망이 요구되어진다. 아울러 이러한 환경에서 통신 시간을 줄일 수 있도록 계산에 사용하는 프로그램의 병렬 및 계산 알고리즘을 개선할 필요성이 있다.

후기

본 연구는 NRL의 지원에 의해 이루어 졌으며 이에 감사 드립니다.

참고문헌

- (1) Park, W., 1998, *Lecture Note on the Computational Fluid Dynamics*, Anda, Vol. 1, pp. 5~7.
- (2) Czajkowski, K., Fitzgerald, S., Foster, I. and Kesselman C., 2001, "Grid Information Services for Distributed Resource Sharing," *10th IEEE International Symposium on High-Performance Distributed Computing*, IEEE Press.
- (3) Cho, K. W., Park, H. W. and Lee, S. S., 2002, "Information Communication Technology and Computational Fluid Dynamics," *Korean Society of Computational Fluids Engineering*, Vol. 6, No. 3, pp. 51~56.
- (4) Kim, D. K., Lee, P. W. and Hwang, I. S., 2002, "Grid Computing," *Korea Information Science Society*, Vol. 20, No. 02, pp. 5~19.
- (5) The Globus Project, <http://www.globus.org>
- (6) MPICH-G2, Grid enabled implementation Message Passing Interface(MPI), <http://www3.niu.edu/mpi>
- (7) Portable Batch System, <http://www.openpbs.org>
- (8) Ghia, U., Ghia, K. N. and Shin, C. T., 1982, "High-Re Solution for Incompressible Flow Using the Navier-Stokes Equation and Multigrid Method," *J. of Computational Physics*, Vol. 48, pp. 387~410.
- (9) Kim, J. and Moin, P., 1985, "Application of a Fractional-Step Method to Incompressible Navier-Stokes Equations," *J. of Computational Physics*, Vol. 59, pp. 308~323.
- (10) Kim, S. H., Ha, M. Y. and Lee, S. S., 2002, "A Large Eddy Simulation for the Three-Dimensional Fluid Flow Past Tube Banks," *5th KSME-JSME Fluid Engineering Conference*, Nagoya, Japan.
- (11) The Message Passing Interface(MPI) standard, <http://www-unix.mcs.anl.gov/mpi/>
- (12) Marc, S., Steve, O., Steven, H., David, W. and Jack, D., 1999, *MPI: The Complete Reference*, Massachusetts Institute of Technology, Vol. 1, pp. 319~340.