전산유체역학 병렬해석을 위한 클러스터 네트웍 장치 성능분석

Performance Analysis of Cluster Network Interfaces for Parallel Computing of Computational Fluid Dynamics

*이보성[1], 홍정우[2], 이상산[2], 이동호[3]
Bo-sung Lee, Jeong-Woo Hong, Sangsan Lee, Dong Ho Lee

요 약

전산유체역학분야의 고속 연산을 위해서 병렬처리가 보편화되고 있으며 이러한 병렬해석은 주로 클러스터에서 저렴한 비용으로 수행되고 있다. 전산유체역학을 위한 클러스터 컴퓨터에서의 해석프로그램의 성능은 클러스터에 사용되는 프로세서의 성능 뿐만 아니라 클러스터 내부의 통신 장비의 성능에 크게 좌우된다. 본 논문에서는 클러스터 컴퓨터의 구축에 널리 사용되고 있는 Myrinet2000, Gigabit Ethernet, Fast Ethernet 등의 네트웍 장치에 대해서 Netpipe, Linpack, NAS NPB, 그리고 MPINS2D Navier-Stokes 해석프로그램을 사용하여 성능을 비교하였다. 이를 통해서 향후 전산유체역학을 위한 클러스터 구축시 최대의 가격대 성능비를 얻을 수 있는 방법을 제시하고자 한다.

1. 서 론

최근의 컴퓨팅 환경의 변화로 인하여 전산유체역학해석 시스템으로 리눅스 기반의 병렬 클러스터가 보편화되고 있다. 리눅스 기반의 병렬 클러스터는 COTS(Commodity Off The Shelf) 부품을 사용하여 제작되므로 시장상황에 따라서 쇠퇴주기가 대단히 빠른 편이다. 따라서 한정된 예산 범위에서 전산유체역학 분야에 가장 적합한 가격대 성능비를 보장하는 리눅스 클러스터를 구성하기 위해서는 전산유체역학분야의 문제 특성에 적합한 하드웨어를 채택할 필요가 있다. 이를 위해서는 다양한 형태의 벤치마크 테스트가 선행되어야만 한다.

본 논문에서는 리눅스 기반의 병렬 클러스터의 성능에 큰 영향을 미치는 요소인 계산노드의 프로세서, 각 계산노드간의 연결네트웍구성, 사용된 컴파일러 등에 따라서 전체 시스템의 성능이 어떠한 영향을 받는지를 Linpack[1], Netpipe[2], NAS Parallel Benchmark Suite[3], 그리고 이차원 압축성 Navier-Stokes 병렬 해석프로그램인 MPINS2D를 사용하여 성능을 비교하고 이중에서 특히 연결네트웍 장비와 컴파일러에 따른 성능을 Pentium4 기반의 리눅스클러스터 상에서 집중적으로 비교하여 전체 클러스터 시스템의 성능에 미치는 영향을 고찰하고자 한다.

2. 단위노드 성능 분석 및 시스템 선정

먼저 클러스터 시스템에서의 성능을 비교하기에 앞서 클러스터 구성을 위한 단위노드 선정을 위해 Linpack 테스트와 이차원 비압축성에 Navier-

Table 1. 단위노드 성능측정에 사용된 시스템

	HB2110R 1GHz	HB2110R 1.26GHz	HB2170R 1.26GHz	HB2180R 1.13GHz	HB2180R 1.26GHz	Pentium4 PC	Athlon MP1500
Processor	P3 1GHz 256KB	P3 1.26GHz 512KB	P3 1.26GHz 512KB	P3 1.13GHz 512KB	P3 1.26GHz 512KB	P4 1.4GHz 256KB	1.33GHz 256KB
Chipset	VIA Apollo Pro	VIA Apollo Pro	ServerWorks SE-LE	ServerWorks HE-SL	ServerWorks HE-SL	Intel 850	AMD 760MP
Memory	1GB 133MHz SDRAM	1GB 133MHz SDRAM	1GB 133MHz SDRAM	2GB 133MHz SDRAM	2GB 133MHz SDRAM	1GB 800MHz RDRAM	1GB 266MHz DDR
Kernel	2.4.2 SMP	2.4.2 SMP	2.4.2 SMP	2.4.2 SMP	2.4.2 SMP	2.4.2	2.4.2 SMP
ATLAS	3.2.1	3.2.1	3.2.1	3.2.1	3.2.1	3.3.0	3.2.1

1) 리눅스원(주) 2), 3) 한국과학기술정보연구원, 4)서울대학교

Stokes 해석프로그램을 단일 프로세서에 대해서 수행하였다.

Fig. 1 Single Processor LINPACK Results

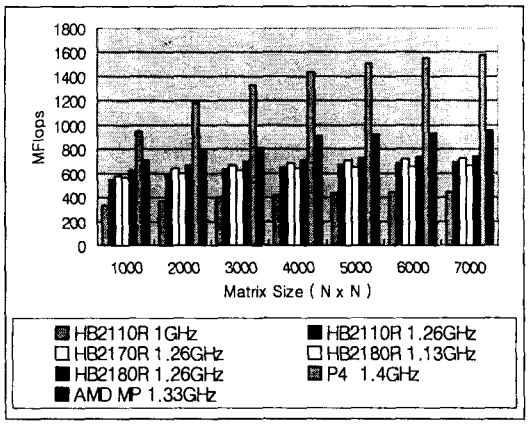


Fig. 2 2차원 비압축성 NS Solver 수행시간

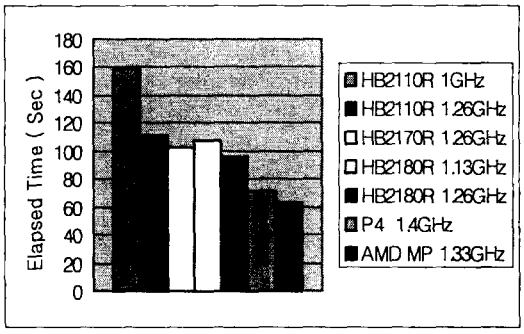


Figure 1과 Figure 2의 결과를 보면 Linpack 테스트의 경우 Pentium4 1.4 GHz에서의 수행결과가 가장 좋게 나타났는데, Pentium3와 AMD 프로세서에서는 SSE1만 지원하지만 Pentium4에서는 SSE2 인스트럭션을 지원하므로 Linpack 성능이 월등함을 보인다. Fortran77을 이용한 2차원 비압축성 Navier-Stokes 해석프로그램의 수행시간은 Pentium4 1.4GHz와 AMD Athlon MP1500의 수행시간이 Pentium3 시스템들보다는 뛰어난 성능을 보임을 알 수 있다.

단일 노드 성능테스트 결과를 바탕으로 실수 연산에서는 Pentium4와 AMD Athlon 프로세서가 Pentium3시스템에 비해서는 성능이 탁월함을 알 수 있었는데 현재 CPU의 클럭 속도 증가추세를 감안할 때 Pentium4가 AMD에 비해서 클럭 향상이 월등함으로 클러스터 시스템의 단위노드로 본 연구에서는 Pentium4를 채택하였으며 이후의 클러스터에서의 성능측정은 Pentium4 클러스터에서 수행되었다.

Table 2. 클러스터 단위노드 사양

Processor	Intel Pentium4 1.7GHz/256KB
Chipset	Intel 850
Mainboard	Intel D850MV
Memory	1GB(256MB ECC RDRAM * 4)
Kernel	2.4.2
확장 슬롯	32bit/33MHz PCI

이상을 바탕으로 성능 측정을 위해 선정된 클러스터 시스템 단위노드의 사양은 Table 2와 같다. Table 3에는 본 논문에서 수행한 성능 측정을 위해 사용한 컴파일러와 네트웍 장비 및 사용된 Message Passing Library를 정리하였다.

Table 3. 클러스터 시스템 네트웍 구성 및 사용 컴파일러, 메시지패싱 라이브러리

네트웍장비	단위시스템구성	테스트 규모	컴파일러 / 메시지패싱라이브러리		
			Linpack	NPB2.3	MPINS2D
Fast Ethernet	Table 1	1,2,4,8,16	gcc 2.96 mpich 1.2.3	icc 5.0 lammpi 6.5.6	
Gigabit Ethernet			gcc 2.96 mpich 1.2.3	gcc 2.96 icc 5.0 mpich 1.2.3 lammpi 6.5.6	icc 5.0 lammpi 6.5.6
Myrinet2000			gcc 2.96 mpich-gm 1.5.1	gcc 2.96 mpich-gm 1.5.1	icc 5.0 lammpi 6.5.6
CrayT3E Interconnection Network	Alpha 21164 450MHz, 128MB		Cray Compiler	Cray Compiler	

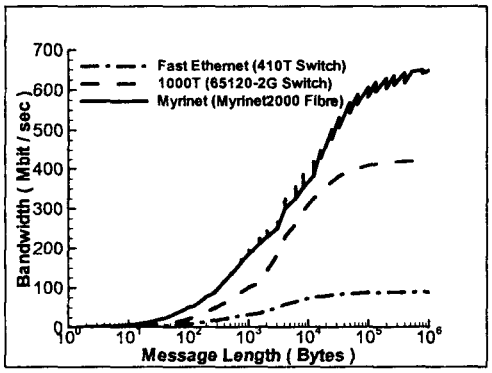
3. 성능측정 결과

3.1 Netpipe

Netpipe(Network Protocol Independent Perfor -mance Evaluator)는 두 시스템간의 point-to-point 통신에서의 latency 및 대역폭을 측정해주는 프로그램을 통신망의 속도를 측정하는데 널리 사용되는 프로그램이다. 본 논문에서는 Netpipe를 사용하여 Table2 에 기술된 단위노드간의 Fast Ethernet, Gigabit Ethernet, Myrinet 2000 네트웍 연결 형태의 latency와 대역폭을 측정하였다.

Figure 3은 세 가지 네트웍 장비에서의 대역폭을 보여주고 있다. 그림에서 알 수 있듯이 Myrinet 2000의 대역폭은 최대 640Mbps, Gigabit Ethernet의 경우 최대 410Mbps, Fast Ethernet의 경우 약 90Mbps 정도의 대역폭을 나타내었다.

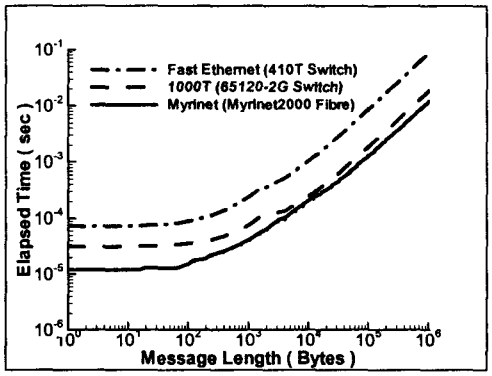
Fig. 3 Netpipe Bandwidth Result



주목할 점은 실제 이론 성능치인 Myrinet2000의 2Gbps, Gigabit Ethernet10의 1Gbps에는 미치지 못하는 성능을 보이는데 이는 성능 측정에 사용된 시스템의 PCI 슬롯이 32bit/33MHz이기 때문일 것으로 판단된다. 이 부분에 대해서는 향후 64bit/66MHz 슬롯이 있는 시스템에서 다시 Netpipe 성능을 측정할 필요가 있음을 뜻한다.

Figure 5는 세 가지 네트웍 장비에서의 latency를 나타내는 것으로 예상대로 Myrinet2000이 가장 latency가 작음을 알 수 있다. 하지만 Gigabit Ethernet도 예상과 달리 상당히 좋은 성능을 보일 뿐 아니라 메시지 크기가 10KByte를 넘어서면서부터는 Myrinet2000과의 차이가 많이 줄어듦을 볼 수 있다. 이러한 결과를 바탕으로 볼 때 작은 메시지를 빈번하게 보내는 프로그램의 경우는 Myrinet2000이 좋은 해석성능을 나타낼 것으로 예상되며 Gigabit Ethernet을 사용하여 클러스터를 구성할 경우 전송되는 메시지의 사이즈를 증가시켜서 Myrinet2000에 근접하는 해석성능을 얻을 수 있을 것으로 보인다.

Fig. 4 Netpipe Latency Results



3.2 Linpack

Fig. 5 16 Nodes Linpack Result

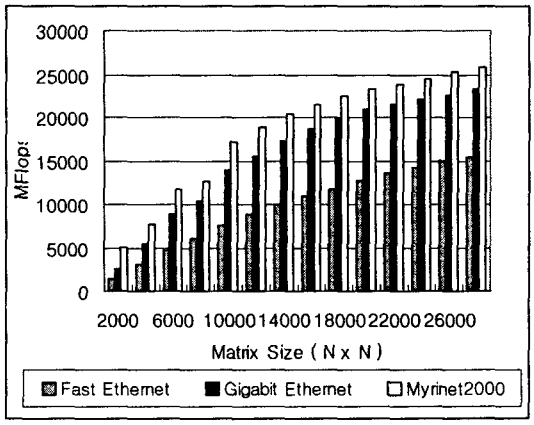


Figure 5는 16노드에서의 Linpack 성능측정을 세 가지 네트웍 장치에 대해서 수행한 결과이다. 그림에서 볼 수 있듯이 작은 사이즈의 문제의 경우에는 latency가 좋은 Myrinet2000이 Gigabit Ethernet이나 Fast Ethernet에 비해서 성능이 훨씬 좋게 나타나지만 문제의 사이즈가 커짐에 따라서 Gigabit Ethernet과 Myrinet2000의 성능차이가 10% 대로 줄어듦을 알 수 있다. 따라서 클러스터 구성에서 Gigabit Ethernet을 채용할 경우 풀고자 하는 문제 사이즈를 증가시켜서 노드간 메시지 교환 시 메시지 크기를 일정 크기 이상이 되도록 프로그래밍 함으로써 성능 향상을

기할 수 있을 것으로 생각된다.

3.3 NPB(NAS Parallel Benchmark)

NPB 프로그램은 NASA의 NAS(Numerical Aerodynamics Simulations)에서 개발된 것으로 실제 전산유체역학 코드에서 출발한 벤치마크 프로그램이다. 이는 NASA에서 도입하고자하는 슈퍼컴퓨터의 성능을 나타내기에 적합한 표준으로 개발되었다. NPB 1.0은 공기역학 문제로부터 개발된 8개의 벤치마크 문제로 구성되어 있는데 이는 다시 5개의 커널과 3개의 CFD 응용문제로 나누어진다.

NPB 2.0에는 NPB 1.0의 8개의 프로그램 중에서 5개의 커널이 포함되었다. 이 중에서 FT는 3차원 FFT 기반의 spectral 코드이며 MG는 3차원 스칼라 Poisson 방정식의 해를 구하기 위해 다격자(multigrid) 기법을 사용하고 있다. 그리고 LU는 3차원 Navier-Stokes 방정식을 unfactored implicit finite-difference 기법을 사용하여 이산화할 때 나타나는 block lower triangular-block upper triangular 연립방정식의 해를 SSOR(Symmetric Successive Over Relaxztion) 기법을 사용하는 구하는 프로그램이다. SP와 BT는 Navier-Stokes 방정식을 approximately factored implicit finite-difference 기법을 사용하여 이산화할 때 나타나는 연립방정식의 해를 구하는 프로그램이다. BT는 5 × 5의 block-tridiagonal 연립방정식을 푸는 프로그램이며 SP는 완전 대각화 시에 나타나는 scalar pentadiagonal 연립방정식을 풀게 된다. 5개의 벤치마크 커널은 실제 전산유체역학 해석 프로그램에서 사용되고 있는 가장 핵심적인 부분을 바탕으로 작성되었기 때문에 이들을 이용한 성능 비교 결과가 실제 전산유체역학 해석 프로그램을 사용한 결과를 대표하기에 충분하다고 생각된다.

본 논문에서는 NPB 버전 2.3의 커널 5개중 BT, SP, MG, LU와 통신망의 속도에 가장 큰 영향을 받는 CG(Conjugate Gradient) 등 5개의 프로그램을 Fast Ethernet, Gigabit Ethernet, Myrinet2000 등 세 가지 네트웍장비에 대해서 각각 MPICH, LAM MPI 등 두가지 MPI 버전을 이용하여 성능을 비교하였다. 그리고 각각의 경우에 대해서 리눅스에 내장된 gcc 2.96 버전과 Intel의 5.0 컴파일러를 사용하여 성능비교를 수행하였다.

Fig. 6 NBP CG Class B

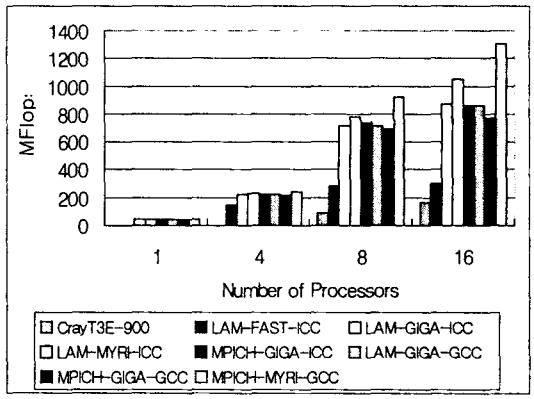


Figure 6에서 Figure 10까지에 NPB Class B의 수행결과가 나타나있는데 여기서는 다음과 같이 비교를 수행할 수 있다.

그림에서 세 번째와 네 번째 항목은 LAM MPI, Intel Compiler를 사용하여 Gigabit Ethernet과 Myrinet2000에서 수행한 결과이다. 일곱 번째와 여덟 번째 항목은 MPICH, GCC를 사용하여 Gigabit Ethernet과 Myrinet2000에서의 수행결과이다. 즉, 두 가지 네트웍 장비의 성능차이를 볼 수 있다.

그리고, 세 번째와 다섯 번째는 Gigabit Ethernet에서 LAM MPI와 MPICH 등 MPI Implementation만 달리하여 수행한 결과이므로 MPI Implementation의 성능차이를 볼 수 있다.

마찬가지로, 세 번째 항목과 여섯 번째 항목은 Gigabit Ethernet에서 LAM MPI를 고정시키고 컴파일러만 GCC와 Intel Compiler로 바꿔서 수행한 결과이며 다섯 번째와 일곱 번째는 Gigabit Ethernet에서 MPICH를 사용하여 컴파일러만 바꿔서 수행한 결과이다.

Figure 6은 CG의 수행결과를 보이는 것으로 CG의 경우 MPI_ALL_REDUCE 함수를 호출하게 되는데 이는 통신망의 latency와 대역폭에 상당한 영향을 받는다. CG의 경우 컴파일러의 성능차이는 그다지 생기지 않지만 Myrinet2000과 Gigabit Ethernet의 성능차이는 크게 나타나며 Myrinet2000을 사용하는 것이 가장 좋은 성능을 얻을 수 있다.

Fig. 7 NPB BT Class B

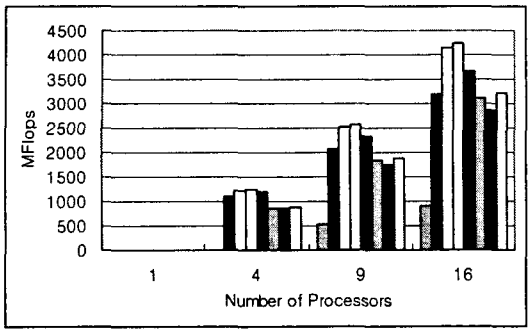


Fig. 8 NPB SP Class B

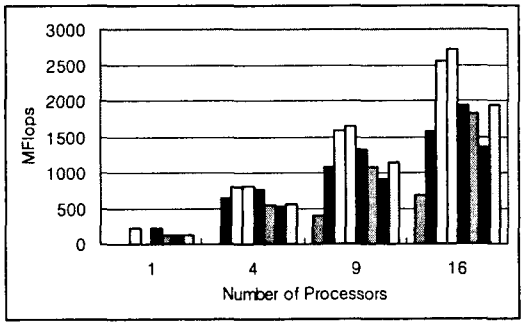


Fig. 9 NPB MG Class B

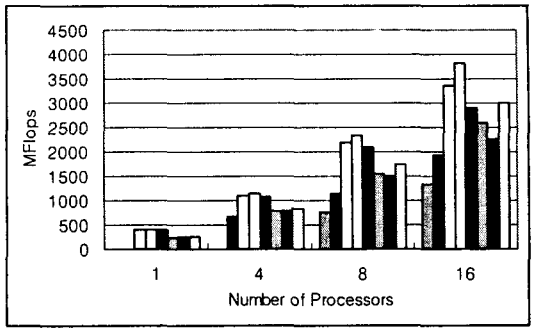


Fig. 10 NPB LU Class B

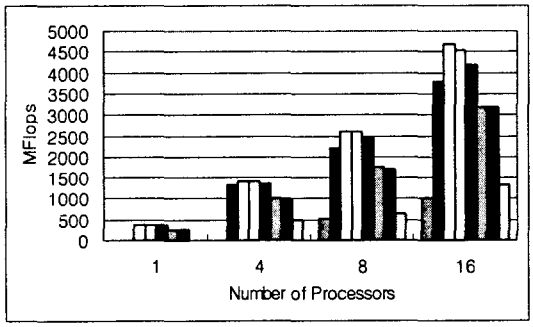


Figure 7과 Figure 8은 BT와 SP를 수행한 결과로서 LAM MPI가 MPICH보다는 좋은 성능을 보이며 Intel Compiler를 사용한 것이 GCC를 사용한 것보다는 훨씬 좋은 성능을 보임을 알 수 있다. Myrinet2000과 Gigabit Ethernet의 성능차이는 예상과 달리 크게 나타나지는 않음을 알 수 있다. BT에 비해서 SP가 통신량이 약 25배 많은 프로그램[4]으로 Myrinet2000과 Gigabit Ethernet의 성능차이가 SP에서 BT에 비해 조금 크게 나타남을 볼 수 있다.

Figure 9는 MG를 수행한 결과로 BT와 SP에서의 결과와 유사한 경향을 보이고 있다.

Figure 10은 다른 경우와는 달리 Myrinet2000에서 Gigabit Ethernet보다 성능이 나쁘게 나타났으며 특히 Fast Ethernet에서의 수행결과보다 나쁘게 나타났다. Myrinet2000에서의 NPB LU의 성능저하는 널리 알려져 있으나 그 원인에 대해서는 아직까지 심도 있게 분석된 바 없다. 향후 어떠한 통신 패턴이 MPICH-GM에서 성능저하를 유발했는지를 규명한다면 프로토콜의 개선 혹은 특정 분야에서의 알고리즘 개발에 도움이 될 것으로 보인다.

이상의 NPB 테스트를 수행한 결과를 요약하면 다음과 같은 결론에 도달할 수 있다.

(a) Pentium4 프로세서 클러스터에서는 GNU Compiler보다는 Intel Compiler의 성능이 우수함
(b) LAM MPI가 MPICH보다는 성능이 우수함
(c) Myrinet2000이 Gigabit Ethernet에 비해서 통신량이 많은 경우 성능이 우수하나 예상보다는 크게 성능차이가 발생하지 않음

본 연구에서는 Myrinet2000에서 MPICH-GM을 설치할 때 Intel Compiler를 사용한 결과를 함께 비교하지 못하였다. 향후 이 부분에 대한 실험결과가 추가된다면 최종적인 성능 비교가 완료될 수 있을 것이라 본다.

3.4 MPINS2D

마지막으로 이차원 압축성 Navier-Stokes 방정식 병렬 해석프로그램인 MPINS2D 코드를 Gigabit Ethernet과 Myrinet2000 상에서 수행하여 두 가지 네트웍 장비의 성능을 비교하였다. 3.3절의 NPB 수행결과에서 가장 좋은 성능을

보인 LAM MPI, Intel Compiler 조합에 대해서 두 가지 네트웍 장비의 성능을 비교하였다.
해석은 DP-SGS[5] 알고리즘을 사용하여 병렬화 했으며 281 x 65의 C-Type격자에서 RAE2822 익형에 대해서 받음각 2.79도, Re=6,500,000, M_{inf} = 0.73의 유동조건이 적용되었다.

Fig. 11 Error History of MPINS2D

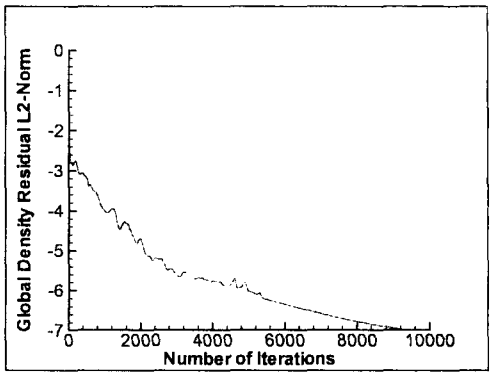


Fig. 12 MPINS2D Results

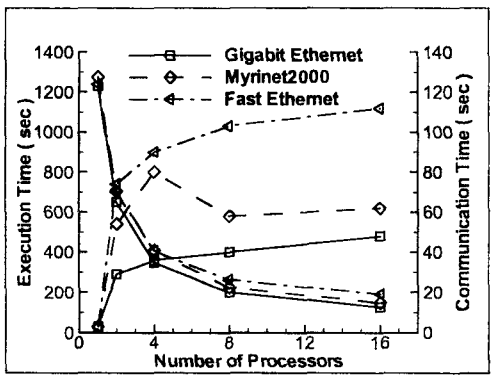


Figure 11은 수렴곡선으로 Global Density Residual L2-Norm이 -7이 될 때까지의 소요시간을 비교하였다.
Figure 12에 MPINS2D에서의 수행 결과를 나타내었다. 그림에서 알 수 있듯이 전체적인 수행시간은 Gigabit Ethernet이 Myrinet2000에서의 수행시간보다 다소 작게 걸림을 알 수 있는데 그 원인으로는 전체 수행시간에서 MPI 통신에 사용되는 수행시간이 Gigabit Ethernet에 비해서 Myrinet2000에서 더 크게 나타났기 때문이다. Myrinet2000에서 LAM MPI를 사용할 경우 GM Protocal 상에서 TCP/IP를 구동하고 이 기반 위에서 LAM MPI가 수행되기 때문에 TCP/IP 통신을 위해서는 Gigabit Ethernet에 비해서 하나의 layer를 더 거치게 되므로 통신 시간이 더 늘어난 것으로 해석된다.
앞에서 언급한 바와 같이 Myrinet2000에서는 MPICH-GM을 사용할 것을 권장하고 있으므로 향후 MPICH-GM상에서 Intel Compiler를 사용하여 수행시간을 비교할 필요가 있다.

4. 결 론

본 논문에서는 전산유체역학 병렬 해석에 적합한 고성능 리눅스 클러스터를 구축하는데 있어서 단위 노드 성능, 연결 네트웍 장비, 컴파일러 등이 전체적인 해석 프로그램의 성능에 어떠한 영향을 끼치는지를 Linpack, Netpipe, NAS NPB, 그리고 MPINS2D 코드를 사용하여 분석하였다.
성능 분석 결과 클러스터 시스템의 성능에 네트웍 장비의 영향이 상당히 크다는 것을 확인할 수 있었으며 MPI Implementation, 컴파일러 또한 성능에 많은 영향을 끼침을 확인하였다.
한편 Myrinet2000과 같은 고가의 고성능 네트웍 장비에 비해서 절반이하의 가격인 Gigabit Ethernet을 사용하여 클러스터를 구축할 경우에도 적정 규모의 클러스터 시스템에서는 만족스러운 성능을 얻을 수 있음을 알 수 있다. 현재와 같은 Gigabit Ethernet의 지속적인 가격 인하를 고려한다면 Gigabit Ethernet 기반의 클러스터가 전산유체역학 분야에서는 보편화 될 수 있을 것으로 보인다.

참고문헌

[1] Jack Dongarra, Jim Bunch, Cleve Moler and Pete Stewart, "LINPACK", http://www.netlib.org/linpack
[2] Quinn O. Snell, Armin R. Mikler and John L. Gustafson, "NetPIPE: A Network Protocol Independent Performance Evaluator", http://www.scl.ameslab.gov/netpipe/paper/full.html
[3] Bailey, D. H. *et al*, "The NAS Parallel Benchmarks", NASA TM-103863, NASA Research Center, Moffett Field, CA, 94035-1000, July 1993, http://www.nas.nasa.gov/NAS/NPB
[4] 권오영, "고성능 컴퓨터 성능 측정 도구 NPB의 소스코드 구성 및 병렬화 방법 분석", 슈퍼컴퓨팅소식 Vol. 4, 2001. 3. ISSN 1339-7838, 한국과학기술정보연구원, 슈퍼컴퓨팅센터
[5] 이보성, 이동호, "MPP에서의 효율적 분산처리를 위한 Data Parallel Symmetric Gauss-Seidel 알고리즘," 한국항공우주학회지 제 26권 2호, pp. 60-72