리눅스 클러스터 시스템 기반의 고성능 연결 망 벤치마크

홍인표
고려대학교 컴퓨터정보통신대학원
iphong72@korea.ac.kr

High Performance Network Benchmarking Based on Linux Cluster System

InPyo Hong
Graduate School of Computer and Information Technology, Korea University

요 약

여러 대의 컴퓨팅 시스템을 네트워크로 연결한 클러스터 시스템의 경우, 뛰어난 가격대비 성능 특성 때문에 많은 공학분야의 연구에 널리 활용 되고 있다. 최근에는 뛰어난 확장성과 안정성을 요구하는 기업체 업무에도 널리 활용 되고 있다. 이러한 클러스터 시스템의 성능은 네트워크 시스템의 성능에 크게 좌우되므로 고성능 네트워크 시스템에 대한 연구는 지속적으로 수행되고 있다. 하지만 새로운 네트워크 시스템의 성능이 실제 응용 소프트웨어들의 성능에 어떠한 영향을 주는지 파악하기란 쉽지가 않다. 따라서 여러 개의 응용 소프트웨어 및 계산 소프트웨어에 대한 서비스를 제공해야 하는 기업업무 환경하에서 클러스터 시스템 기반의 새로운 고성능 네트워크 시스템 선택 시 벤치마크는 필수적이다. 본 연구에서는 최근에 출시된 고성능 네트워크 시스템 (Infiniband, Myrinet)들에 대해서 효율적인 노드들간 데이터 통신의 성능을 벤치마크 툴을 통하여 그 결과를 비교 분석하고자 한다.

1. 서 론

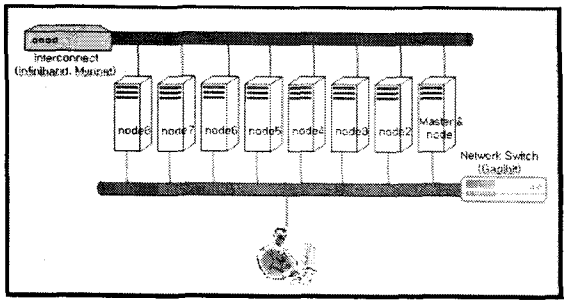
리눅스 클러스터 시스템의 계산능력을 결정하는 주요 구성요소로는 실행 노드와 고속 네트워크 그리고 파일 시스템으로 나누어 볼 수 있다. 기존의 리눅스 클러스터 시스템은 고속의 네트워크에 있어서 독자적인 디자인으로 구성되어 프로세서간 통신이 매우 고속으로 이루어지는데 반하여 일반적인 노드들 간의 통신수단은 100Mbps(Ethetnet), 1Gbps(Gigabit), 2Gbps(Myrinet, Qudrics) 정도의 속도를 넘어설 수 없었다. 그러나 10Gbps(Infiniband, Myrinet, Qudrics)등의 기술이 빠르게 성장하고 그 이용분야가 넓어짐에 따라 네트워크 트래픽은 매년 평균 성장률이 증가하고 있으며 이 고속 근거리 네트워크 장비들이 리눅스 클러스터 계산성능에 적합한지 여부를 판별하기 위해 시기 적절히 벤치마크 도구를 사용하여 테스트할 필요성이 있다.[1],[2] 고속의 네트워크 장비 성능과 실행 노드 성능이 리눅스 클러스터 시스템의 전체적인 계산능력을 결정하는 중요 요인이 되기 때문에 고속 네트워크의 벤치마크는 단일 노드로 부터 노드 수를 증가시키면서 진행하며 파일시스템은 제외하고자 한다. 이에 본 논문에서는 리눅스 클러스터 시스템 기반 상에서 Infiniband (10Gbps), Myrinet (10Gbps)등의 고속 근거리 네트워크 장비들에 대해서 NPB, NetPIPE등 네트워크 벤치마크 도구와 MPI를 활용한 전자구조 계산에 많이 사용하는 병렬프로그램인 VASP등을 사용하여 다양한 환경에서 성능을 측정 평가하고자 한다.[3]~[5] 2장에서는 실험 환경 구성 및 현황에 대해서 설명하고 3장에서는 벤치마크 툴들에 대해 살펴본다. 그리고 4장에서는 구현 환경에서 벤치마크 성능을 평가하고 5장에서는 본 논문의 결론을 맺고 향후 계획을 설명한다.

2. 관련 연구

클러스터는 네트워크로 연결된 컴퓨터들의 그룹이 하나의 공통작업을 나누어 처리하여 대규모의 연산을 빠르게 처리할 수 있도록 구성 되었으며, 이러한 클러스터 구성방식을 Beowulf클러스터라 한다.[6] 1994년 Thomas Sterling과 Don Becker가 16개의 DX2 프로세서를 연결하여 Beowulf라는 리눅스 클러스터 시스템을 구축하면서 주목 받기 시작 하였다. 이후 리눅스 클러스터를 구성하는 컴퓨터 노드 수는 수십 대에서 수백 대 이상으로 점점 규모가 커지고 있는 상황이다.

2.1 리눅스 클러스터 구성

기존 논문들의 리눅스 클러스터 시스템 구성방식과 동일한 Beowulf 시스템 구성 방식으로 적용하였으며, 병렬계산 용도로 사용하기 위하여 하드웨어 자원의 성능을 높이고자 64bit 시스템과 Infiniband 그리고 Myrinet을 사용하여 기본 대역폭을 확대 하는 등의 최신의 하드웨어와 소프트웨어를 사용하여 고성능을 꾀하였다. 보다 자세한 리눅스 클러스터 구축 환경은 (그림 1) 및 <표 1> 에 제시하였다.



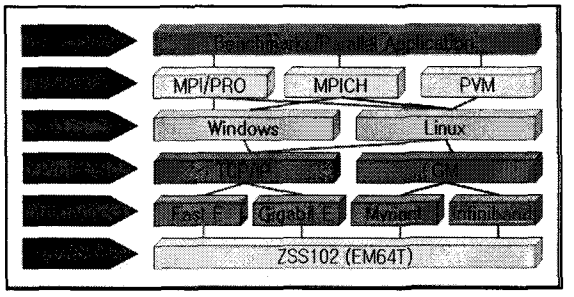
(그림 1) 리눅스 클러스터 구성환경

<표 1> 리눅스 클러스터 플랫폼

Master/Cluster node 8	
Processor	Intel Pentium4 3.8GHz EM64T(Hyper Threading: HT) (Front Side Bus 800MHz L1 cache 16KB L2 cache 1MB)
MainBoard	Intel 7221BA1-E
Memory	DDR2 SDRAM 4GB (1GB x 4) 533MHz
HardDisk	SATAII 250GB (7200rpm/8M)
Common	
Network	Foundry GigabIt Ethernet Switch HUB, Marvel 88E8050 Gigabit Ethernet Card
	MTPDK08 Infiniband Switch, InfiniHost III Ex Card
	M3F-SW8 Myrinet Switch, 10GBase-CX4 Card
MPI	MPICH-MX, OSUMVPICH-1.7.0
OS	REDHAT Enterprise WS ver.4 Update 4 64bit
Compiler	Intel C++/Fortran version 9.3 CMKL (Cluster Math Kernel Library) version 5.2

2.2 Interconnect 및 Middleware 환경

기존 소켓기반의 TCP/IP를 이용한 통신 모듈은 사용자 영역이나 커널 영역에 존재하는 여러 네트워크 계층으로 인하여 하부 네트워크가 제공하는 대역폭을 충분히 활용하지 못하고 메시지를 송수신 하는 과정 또한 큰 지연시간을 가져오게 된다. 이러한 대역폭 및 지연시간을 없애고 고성능 통신 서브시스템을 제공하기 위해서 사용자 수준 통신 계층이 제안되었다. 사용자 수준 통신 계층의 기본 아이디어는 메시지의 송수신 과정에서 커널의 관여를 배제하고 사용자와 고속의 네트워크 장비 사이에 직접적으로 메시지를 송수신하는 것이다. 현재 제안된 많은 사용자 수준 통신 프로토콜로는 VMMC, VMMC-II, GM, U-NET, BIP, FM, AM등이 있다. 이들 프로토콜들은 각각의 특성과 기능성을 가지고 있고 나타내는 성능 또한 다양하다. 이런 다양한 사용자 수준 통신 프로토콜을 사용한 고속 근거리 네트워크 장비(Myrinet, Infiniband)는 병렬처리가 가능하도록 하여주는 소프트웨어 이며 특히 최고의 성능을 발휘하기 위해서는 높은 대역폭과 낮은 응답시간이 중요하므로 그러한 구현을 가능하게 해주었다. 병렬처리는 프로그램을 동시에 실행할 수 있는 여러 조각으로 나누어 각자 자신의 프로세서에서 실행함으로써 프로그램 수행 속도를 빠르게 한다는 개념이며 프로그램을 N개의 프로세서에서 실행하면 하나의 프로세서가 실행하는 것보다 최대 N배까지 빠른 성능을 보여준다. 이러한 병렬처리 프로그램을 구현하기 위해서 미들웨어 (MPI, PVM)등이 사용된다.[7] MPI는 표준화되어 있기 때문에 서로 다른 시스템환경에서 구현된 프로그램일지라도 코드의 변경 없이 수행할 수 있는 이점이 있다. 또한 MPI는 이식성을 높일 수 있도록 시스템의 네트워크 구조에 종속적이지 않는 인터페이스로 개발되었으며 시스템의 초기화 및 종료에 관련된 함수와 일대일 통신을 위한 블록화 함수와 비 블록화 함수 그리고 집합으로 통신을 수행할 수 있는 다양한 형태의 메시지 패싱 함수를 지원함으로써 사용자가 프로그램을 쉽게 작성할 수 있도록 지원하고 있다. 본 논문에서는 리눅스 환경하에서 MPICH을 사용하였다.[8] 보다 자세한 환경은 (그림 2) 에 제시 하였다.



(그림 2) 클러스터 하드웨어 및 소프트웨어 환경

3. 벤치마크 툴

벤치마크 테스트 프로그램에 있어서 다양한 처리 방식 및 많은 툴들이 존재 하고 있으나, 본 연구에서는 리눅스 클러스터 구성 환경하에서 근거리 네트워크 장비들의 통신환경을 분석하고자 하는 것이므로 네트워크의 대역폭 및 응답시간 병렬 프로그램 활용 성능 측정 그리고 시스템 성능 측정을 하기 위한 보편적으로 사용되고 있는 벤치마크 툴들에 대해 알아보고자 한다.

3.1 VASP

VASP(Vienna Ab-initio Simulation Package)는 오스트리아 빈 대학의 J. Hafner 교수 그룹에서 개발된 분자모델링 계산 프로그램이며, 양자역학 계산을 통해 원자간 결합을 기술하며, 분자의 평형구조, 분자간 화학반응, 고체의 내부나 표면의 원자구조, 고체표면에서의 화학반응 등을 원자수준에서 정교하게 계산할 수 있다. 이러한 원자 수준의 계산은 High-k 유전체의 불순물, 연료전지의 촉매반응, CNT나 Si 나노 와이어의 전기적 특성 등 다양한 나노 재료 및 나노 소자 연구에 적용되고 있으며 다른 분자 모델링 계산 프로그램과 대비하여 3가지에 대한 큰 장점을 가지고 있다. 첫째로 Ultra-soft 수도 포텐셜 또는 PAW(Projector-Augmented Wave)방법을 이용하여 전자와 이온간의 상호작용을 기술함으로써 계산 시간이 원자 수에 대해 비례한다. 기존의 분자 모델링 프로그램은 N의 3승 또는 N의 6승에 비례하기 때문에 상대적으로 많은 원자를 포함하여 계산을 할 수

있다. 두번째로 Pentium-based Linux Cluster등 실행 노드 수가 많은 시스템에 대해 효율적인 MPI-Parallelization을 지원한다. 세번째로 1995년 초기 안정화 버전이 배포 되어진 이후로 꾸준히 최신 기술이 접목되어 현재 고체물리학 이론 분야에서 사용되는 분자 모델링 프로그램 중에서 정확도, 계산속도, 기능, 사용편이 면에서 성능이 월등하게 나타나고 있다. 이러한 장점들로 인하여 물리, 화학, 재료분야의 많은 연구자들이 VASP를 이용하여 첨단 연구를 진행하고 있으며 앞으로 많은 사용자들이 늘어날 전망인 프로그램이다.

3.2 NPB

NPB는 NASA의 NAS(Numerical Aerodynamics Simulations)에서 개발된 것으로 실제 전산유체역학 코드에서 출발한 벤치마크 프로그램이다. 이는 NASA에서 도입하고자 하는 슈퍼컴퓨터의 성능을 나타내기에 적합한 표준으로 개발되었다. NPB 1.0은 공기역학 문제로부터 개발된 8개의 벤치마크 문제로 구성되어 있는데 이는 다시 5개의 커널과 3개의 CFD응용문제로 나누어지며 NPB2.0에서는 NPB1.0의 8개 프로그램 중에서 5개의 커널이 포함되었다. 이중에서 FT는 3차원 FFT기반의 spectral코드이며 MG는 3차원 스칼라 Poisson 방정식의 해를 구하기 위해 다격자 기법을 사용하고 있다. 그리고 LU는 3차원 Navier-Stokes방정식을 unfactored implicit finite difference기법을 사용하여 이산화할 때 나타나는 block lower triangular-block upper triangular 연립방정식의 해를 SSOR(Symmetric Successive Over Relaxation) 기법을 사용하여 구하는 프로그램이다. SP와 BT는 Navier-Stokes 방정식을 approximately factored implicit finite difference 기법을 사용하여 이산화할 때 나타나는 연립방정식의 해를 구하는 프로그램이다.

3.3 NetPIPE

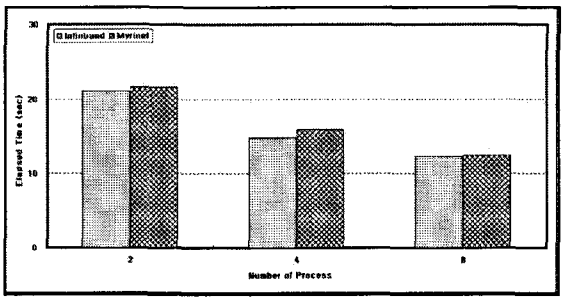
NetPIPE(Network Protocol Independent Performance Evaluator)는 두 시스템간의 메시지를 반복적으로 주고받는 방식(Point-to-Point)의 통신으로 응답속도 및 대역폭을 측정해주는 프로그램이며 최초 1Byte의 작은 메시지를 수 천번 주고 받는 것에서 시작하여 수 Mbyte까지 주고 받으면서 데이터 크기에 대한 통신망의 속도를 측정하는데 널리 사용되고 있다.

4. 성능평가

리눅스 클러스터 기반의 시스템 구축 시 전체적인 계산능력을 결정하는 중요 요인인 고속의 네트워크 장비 성능과 실행 노드의 성능이다. 따라서 다양한 전송 메시지의 크기에 따라 시스템의 성능이 어떻게 달라지는지 구축된 리눅스 클러스터 시스템으로 고속의 네트워크 장비에 대한 벤치마크 테스트 툴을 활용하여 각 10회에 걸쳐 네트워크 성능을 측정하고 그 결과값을 분석하였다.

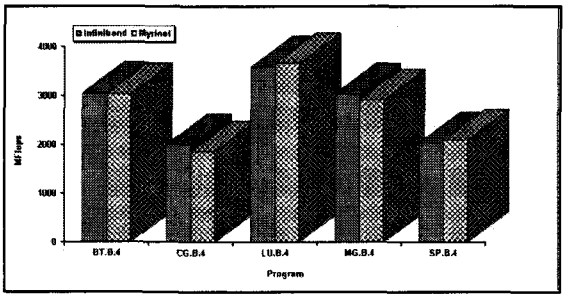
4.1 성능측정 결과

본 논문에서 VASP, NPB, NetPIPE를 사용하여 [표 1]에 기술된 단위 노드간의 Infiniband, Myrinet 네트워크의 연결 형태 성능을 측정하였으며, (그림 3)은 VASP 실행 후 경과시간을 두 가지 네트워크 장비에서 보여주고 있다. 예상대로 Infiniband 경우 2 프로세서, 4 프로세서에서 약간의 성능 우위를 보이고 있지만, 8 프로세서 병렬 실행 시 두 가지 네트워크가 거의 비슷한 수준의 해석 성능을 보이고 있다. 이러한 결과를 볼 때 분자 모델링 병렬 계산 프로그램의 경우 8 프로세서를 사용하여 해석을 한다면 두 가지 네트워크 모두 좋은 해석 성능을 나타낼 것으로 예상되며, 향후 16 프로세서는 측정할 필요성이 있다.

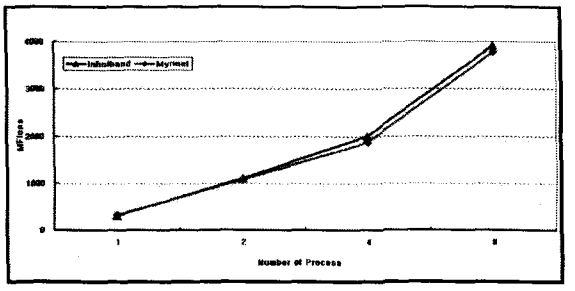


(그림 3) VASP 경과시간 결과

또한 NPB의 5개 벤치마크 프로그램은 실제 전산유체역학 해석프로그램에서 사용되고 있는 가장 핵심적인 부분을 바탕으로 작성되었기 때문에 이들을 이용한 성능 비교 결과가 실제 전산유체 역학 해석프로그램을 사용한 결과를 대표하기에 충분하다고 생각 된다. 본 논문에서 NPB의 BT, SP, MG, LU와 네트워크의 속도에 가장 큰 영향을 받는 CG등 5개의 프로그램을 B클래스에 대해서 (그림 4)처럼 비교하였으며 차이가 많이 나타나지 않고, (그림 5)의 CG 경우도 프로세서가 증가 되면서 선형적으로 증가하는 것을 보여 주고 있다. 따라서 전산유체 역학 해석 프로그램 역시 두 가지 네트워크의 성능은 별 차이가 없음을 알 수 있다.

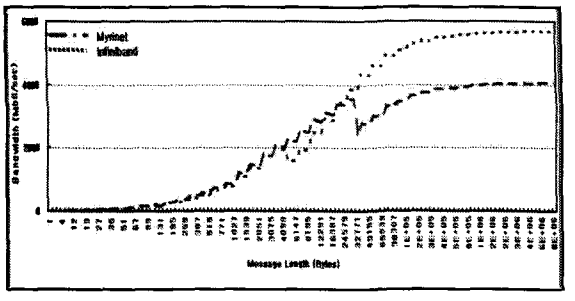


(그림 4) NPB 5개 프로그램 결과

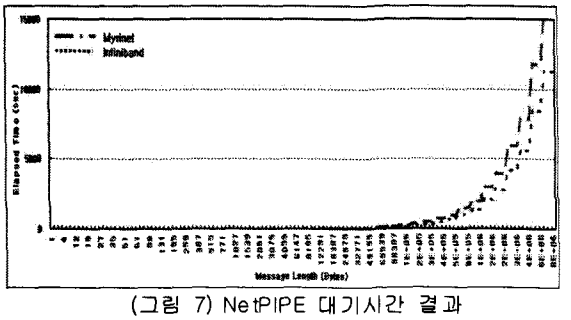


(그림 5) NPB CG.B 프로세서별 결과

(그림 6)에서 알 수 있듯이 NetPIPE 실행 후 Myrinet의 대역폭은 최대 400Mbps, Infiniband 570Mbps 정도의 대역폭을 나타내고 있다. 실제 이론 성능치인 Myrinet의 10Gbps, Infiniband의 10Gbps에는 미치지 못하는 성능을 보이고 있으며, 이는 성능측정에 사용된 시스템의 PCI 슬롯이 64bit/66MHz 슬롯으로 8배속의 성능을 지원하기 때문으로 판단된다. 향후 16배속의 슬롯을 지원하는 시스템으로 측정할 필요가 있음을 뜻한다. (그림 7)는 두 가지 네트워크 장비에서의 대기시간을 나타내는 것으로 예상대로 Infiniband가 대기 시간이 작음을 알 수 있었다. 하지만 Myrinet도 상당히 좋은 성능을 보이고 있으며, 메시지 크기가 10Kbyte를 넘어서면서부터는 Infiniband와의 차이가 많이 넓어짐을 볼 수 있다. 이러한 결과를 바탕으로 볼 때 작은 메시지를 빈번하게 보내는 프로그램의 경우에도 Myrinet도 좋은 해석성능을 나타낼 것으로 예상된다.



(그림 6) NetPIPE 대역폭 결과



(그림 7) NetPIPE 대기시간 결과

5. 결론 및 향후 연구

본 논문에서는 병렬해석에 적합한 리눅스 클러스터를 구축하고 연결 네트워크 장비의 성능이 전체적인 해석 프로그램의 성능에 어떠한 영향을 끼치는지를 VASP, NPB, NetPIPE 프로그램을 사용하여 분석하였다. 성능 분석 결과 리눅스 클러스터 시스템의 성능에 네트워크 장비의 영향이 상당히 크다는 것을 확인 할 수 있었다. 또한 Inifinband, Myrinet 등의 고성능 네트워크 장비를 사용하여 리눅스 클러스터를 구축할 경우에도 적정규모의 클러스터 시스템에서는 만족스러운 성능을 얻을 수 있음을 알 수 있었다. 현재와 같은 Infiniband, Myrinet 등의 지속적인 가격 인하를 고려 한다면 고성능 네트워크를 사용하는 기업업무 하에서 병렬 해석을 하는데 있어 보편화 될 수 있을 것으로 보인다. 따라서 향후 계획으로는 MCAE/ECAE 분야의 병렬어플리케이션 벤치마크와 MIS 시스템 상에서의 고성능 네트워크의 지원 여부를 확인하고 적용시키고자 한다.

6. 참고 문헌

[1] Mellanox Inc, http://www.mellanox.com/.
[2] Myricom Inc, http://www.myri.com/.
[3] NASA, "NAS Parallel Benchmarks", http://www.nas.nasa.gov/Resources/Software/npb.html/.
[4] NetPIPE, "Network Protocol Independent Performance Evaluator", http://www.scl.amesleb.gov/Projects/NetPIPE/.
[5] Computational Materials Science Group, The Vienna University, "Vienna Ab-initio Simulation package", http://cms.mpi.univie.ac.at/vasp/.
[6] http://www.beowulf.org/.
[7] W. Gropp, E. Lusk, N.Doss, and A. Skjellum. "A high-performance, portable implementation of the MPI message passing interface standard", Parallel Computing, 22(6):789-828, 1996.
[8] Network-Based Computing Lab, The Ohio State University, "MPI for Infiniband over VAPI Layer", http://nowlab.cse.ohio-state.edu/projects/mpi-iba/.