리눅스 클러스터 시스템 계산노드용 단일서버 벤치마크

홍태영, 홍정우, 김성호
한국과학기술정보연구원 슈퍼컴퓨팅 센터
{tyhong, jwhong, sungho}@kisti.re.kr

Benchmarking a commodity server as a Compute node of Linux Cluster System

Taeyoung Hong, Jeongwoo Hong, Sungho Kim
Supercomputing Center, KISTI

요 약

Beowulf 타입의 리눅스 클러스터 시스템의 핵심노드인 계산노드는 일반적으로 범용 엔트리급 서버 및 PC 등을 이용하여 구성되며, 이 계산노드의 성능은 전체 클러스터의 계산 성능을 결정하는 가장 중요한 요소 중의 하나이다. 이에 본 논문에서는 현재 시중에서 유통 중인 대표적인 로엔드 플랫폼-Xeon, P-IV, Opteron, Athlon64-들을 대상으로 HPL, NPB, stream등 고성능 컴퓨팅 분야에서 널리 쓰이는 벤치마킹 테스트 도구를 사용하여 개별 노드의 성능을 측정하여 비교 분석하였다.

1. 서 론

리눅스 클러스터 시스템의 계산능력을 결정하는 주요 구성 요소는 네트워크와 파일시스템 그리고 계산노드를 꼽을 수 있다. beowulf 타입[1]의 클러스터 시스템은 일반적으로 시중에 대량 공급되는 PC 혹은 엔트리급 서버를 계산노드로 채택한다. 따라서 이 플랫폼들이 계산노드로 적합한지 여부를 판별하기 위해 시기 적절히 고성능 컴퓨팅 분야의 벤치마킹 도구를 사용하여 테스트할 필요가 있다. 단일 노드의 성능과 네트워크의 성능이 클러스터 시스템의 전체적인 계산능력을 결정하는 요인이 되기 때문에 클러스터 시스템의 벤치마크는 단일 노드로부터 노드수를 증가시키면서 진행하는 것이 일반적이다. 하지만 클러스터 환경에서 사용되는 Gigabit[2], Myrinet[3], Infiniband[4] 등 주요 네트워크 장비의 고성능 계산에서의 interconnect-efficiency는 일반적으로 널리 알려져 있으며[5], 이에 본 논문에서는 HPL(High Performance Linpack)[6], NPB(NAS Parallel Benchmark)[7], Stream[8]등 고성능 컴퓨팅 벤치마크 도구로 널리 쓰이는 툴들을 사용하여 다양한 환경에서 단일 노드의 시스템 성능을 중점적으로 측정하였다. 논문의 구성은 2장에서 HPL, 3장에서 NPB, 4장에서 Stream 벤치마킹 결과를 설명하며 5장에서 결론 및 향후 계획을 설명한다.

2. HPL benchmark

2.1 테스트베드 사양

벤치마크 테스트를 수행하기 위해 4 종류의 서버 및 PC를 테스트 베드로 사용하였다. 테스트베드로 사용한 장비는 i386기반의 intel P-IV와 Xeon DP 그리고, x86_64 기반의 Athlon64와 Opteron 240이다. 또한 테스트에서 공정을 기하기 위해 4개의 시스템 모두 CPU 1개에서만 BMT가 진행되도록 하기위해 단일 쓰레드-단일 프로세스로 BMT 테스트를 진행하였다. 사용된 테스트베드는 [표 1]과 [표 2]에 나타낸 바와 같이 dual 프로세서가 장착된 IBM X325 Opteron 240 및 IBM x335 Xeon DP 서버와 single CPU가 장착된 Athlon64와 P-IV 2.8 GHz PC급 플랫폼이다. 각 시스템의 운영체제는 모두 리눅스를 채택하였으나 배포판별로 약간의 차이를 가지고 있으며, 벤치마킹 프로그램의 버전과 사용한 주요 라이브러리는 모두 동일시켰다. 일반적으로 코드 최적화 과정을 거친 고성능 컴퓨팅 분야의 계산코드의 성능은 거의 전적으로 사용한 수학 라이브러리의 성능에 의해 결정되며, SSE/SSE2를 지원할 경우 컴파일러의 버전 차이에도 큰 영향을 받지 않는 것으로 사전 실험을 통해 얻어냈다. 본 실험에서 Itanium을 테스트 대상에서 제외한 이유는 일반적으로 가격 및 성능 면에서 동급으로 알려진 Xeon 과 Opteron 그리고 Athlon64와 P-IV의 성능 비교를 주요 목적으로 했기 때문이다.

[표 1]

testbed	Xeon DP 2.8GHz	Opteron 240 1.4GHz
nCPUs	2	2
Memory	4GB DDR 266 (Max: 4.2 GB/s)	2GB DDR 333 (Max: 5.3 GB/s)
FSB	533MHz	1.4GHz
L2 Cache	512KB	1MB
OS	Red Hat 7.3 2.4.20-28.7	Red Hat Enterprise Linux AS 3 2.4.21-15.0.4.Elsmp
GCC	2.96	3.2.3
MPICH	MPICH-1.2.5.2 ch_p4	MPICH-1.2.6 ch_p4
NPB	NPB3.0-SER	NPB3.0-SER
Math lib	libgoto_p4_512-r0.96	libgoto_opt64p-r0.96.so

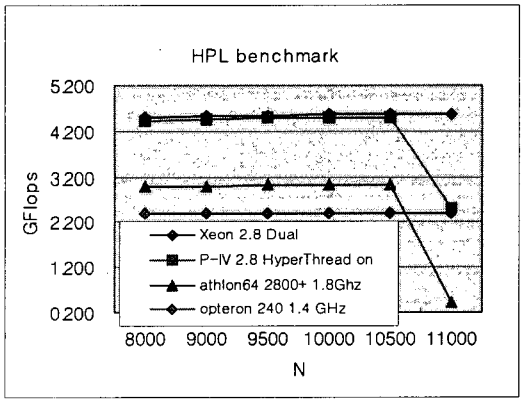
[표 2]

testbed	Athlon64 2800+ 1.8Ghz	P-IV 2.8GHz
nCPUs	1	1
Memory	1GB DDR 400 64 MB shmem for VGA (Max: 3.2 GB/s)	1GB DDR 400 dual channel (Max: 6.4 GB/s)
FSB	800MHz	800MHz
L2 Cache	512KB	512KB
OS	Fedora Core 3 2.6.9-1.667	rocks 3.3.0 (Red Hat Linux AS 3 based) 2.4.21-20.Elsmp

GCC	3.4.2 for x86_64	3.2.3
MPICH	MPICH-1.2.6 ch_p4	MPICH-1.2.6 ch_p4
NPB	NPB3.0-SER	NPB3.0-SER
Math lib	libgoto_p4_512-r0.96	libgoto_p4_512-r0.96

2.2 HPL 성능 비교

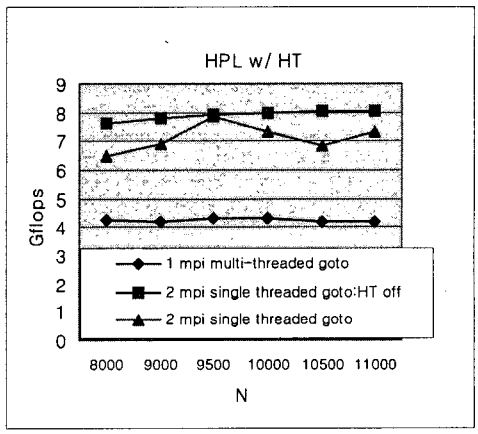
HPL은 배정도 LU 인수분해를 통하여 고밀도 선형 방정식의 해를 구하는 소프트웨어 패키지로서, 프로그램의 수행 시간과 배정도 부동 소수점 연산 횟수를 통해 시스템의 성능을 측정하게 된다. 본 논문에서는 HPL 벤치마킹을 단일 노드 단일 프로세스에 대해서 진행을 했으며, 단일 CPU 시스템의 경우 서버에 기본적으로 걸리는 부하를 최소화 하기위해 불필요한 리눅스의 서비스를 모두 정지시킨 후 테스트를 진행하였다. HPL 테스트 결과, Xeon의 경우 이론 성능의 81%, P-IV의 경우 80%, Athlon64의 경우 84%, Opteron의 경우 85%의 실측 성능이 측정되었다. AMD의 x86_64 기반의 프로세서가 일반적으로 i386 기반 프로세서에 비해 Performance Yield가 상대적으로 높게 나왔으나, 클럭 속도가 인텔 기반의 CPU에 비해 매우 낮은 탓에 HPL의 절대 값은 아래의 [그림 2]의 결과와 같이 상대적으로 큰 격차를 보여준다. 또한 x86_64는 일반적인 64비트 플랫폼과 달리 클럭당 Floating point 연산을 최대 2번 수행할 수 있기 때문에 i386 기반의 CPU에 비해서도 코어 자체의 계산능력이 떨어진다. 아래의 그림에서 P-IV와 athlon64의 경우 HPL 결과가 problem size N=11000에서 급감하는 이유는 어플리케이션이 사용하는 메모리가 물리적인 메모리의 크기를 넘어서 하드 드라이브의 swap 영역을 사용하기 시작하기 때문이다.



[그림 1] HPL benchmark

2.3 hyperthreading의 사용유무에 따른 성능 비교

인텔 Pentium-IV에서부터 지원하기 시작한 hyperthreading Technology(이하 HT)[9]이 HPC 클러스터 장비에서 적합한지 여부를 판별하기 위해서 HT on/off 두 가지 조건에 대해 HPL 벤치마크 테스트를 수행했다. 테스트 장비로는 Xeon DP 2.8 GHz를 사용했으며, 두 개의 프로세서 모두를 사용하도록 테스트를 진행하기 위해 병렬 프로세스 혹은 병렬 쓰레드 방식으로 프로그램을 수행하였다. 일반적으로 HT는 소규모의 데이터처리가 빈번한 다중 쓰레드 어플리케이션에서 최대 30 %의 성능 향상을 가져오는 것으로 알려져 있으나, HPC 분야에서는 대규모의 데이터에 대한 연산이 주류를 이루며, 이것은 하나의 물리적 프로세서에 대해 두 개의 논리 프로세서가 동작할 때 빈번한 cache miss를 야기할 거라는 점에서 HPC 클러스터용 계산노드에서 인텔의 HT을 사용하는 것은 부적절 하다고 예측할 수 있다. 아래의 측정결과는 HPL의 수학라이브러리인 goto 라이브러리를 다중 쓰레드 방식의 바이너리와 단일 쓰레드 방식의 바이너리를 이용하여 HT를 사용할 경우의 성능 향상 여부를 측정한 것이다. 2개의 프로세서에서 단일 쓰레드 goto를 사용하는 2개의 MPI 프로세스를 실행하는 경우에 대해 테스트를 진행한 결과 HT를 사용하지 않는 경우가 HT를 사용하는 경우에 비해 훨씬 안정적인 성능을 보여주었다. 이것은 HT을 사용하면 종종 2개의 MPI 프로세스가 동일한 물리 프로세서에 기반을 둔 2개의 논리 프로세서를 사용하는 경우가 발생하기 때문이다. 또한 cache miss 및 위와 동일한 이유로 다중 Tm 레드 goto 수학라이브러리를 사용한 1개의 프로세스를 수행할 경우에도 성능이 매우 저조하게 나타났다.

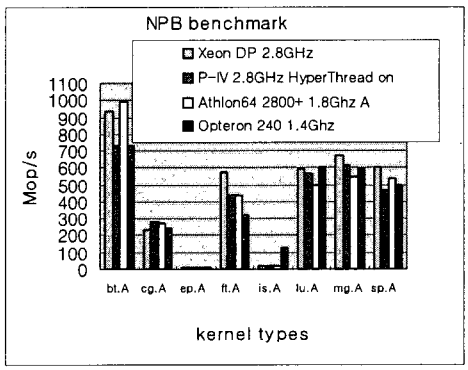


[그림 2] HT 성능 비교

3. NPB Benchmark

3.1 Athlon64, Opteron, P-IV, Xeon의 성능 비교

NPB(NAS Parallel Benchmark) 패키지는 전산유체역학 코드에서 발전된 벤치마크 툴로서 서로 다른 연산 및 통신 패턴을 가진 8개의 벤치마크 프로그램으로 구성되어 있다.



[그림 3] NPB Benchmark

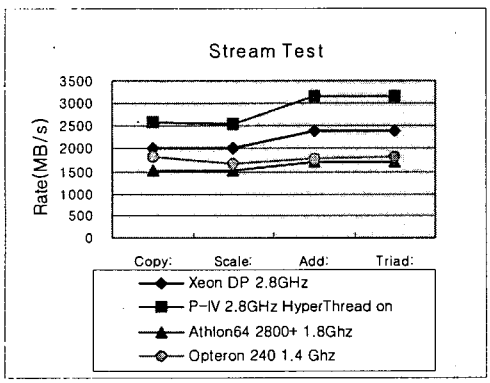
본 측정에서는 순차적인 단일 프로그램으로 구현된 NPB-SER 버전을 class-A의 문제 크기에 대하여 사용하였다. 측정 결과

HPL의 결과와 마찬가지로 Xeon, P-IV, Opteron, 그리고 Athlon64의 순으로 성능이 우수하게 나타났으며 다만, IS(integer sort)문제에서는 L2 cache의 크기가 다른 시스템에 비해 두 배인 Opteron이 최고의 성능을 나타내었고, BT(Block tridiagonal) 연산에서는 Athlon64가 가장 좋은 결과를 보여주었다.

4. Stream Benchmark

4.1 Athlon64, Opteron, P-IV, Xeon의 성능 비교

Stream 벤치마킹 프로그램은 각 성분이 배정도 부동소수로 이루어진 벡터의 복사, 상수곱셈, 덧셈 및 Triad 연산 등을 통해 시스템 메모리의 대역폭을 측정하는 간단한 도구이다. 일반적으로 벡터의 크기가 시스템의 cache 메모리 보다 훨씬 크게 설정되어 있으며, 데이터를 재사용할 수 없도록 코드가 구조화되어있다. 단일 프로세스로 측정한 stream 테스트에서 FSB(Front Side Bus)가 상대적으로 가장 빠르며 메모리 이중 중첩(2-way interleaved memory channel)을 지원하는 P-IV가 가장 우수한 결과를 보여주었으며, Xeon은 FSB의 차이 때문에 그리고 Athlon64 메모리 이중 중첩을 지원하지 않기 때문에 상대적으로 낮은 대역폭 성능을 보였다.



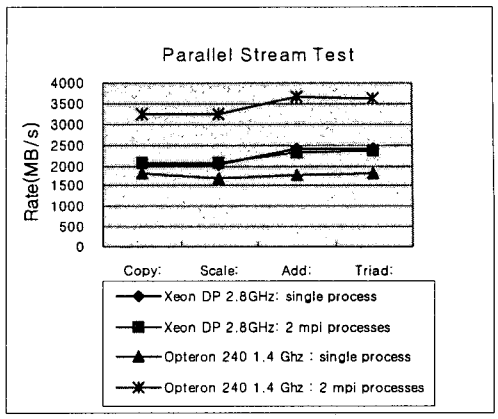
[그림 4] Stream test

4.2 병렬 stream 테스트

메모리 대역폭의 확장성을 측정하기 해서 2개의 MPI 프로세스가 실행되는 병렬 stream 벤치마크를 듀얼 CPU가 장착된 Opteron 및 Xeon 시스템에 대하여 메모리 이중 중첩을 사용하도록 설정한 후 테스트를 진행 하였다. [그림 5]의 결과에서 알 수 있듯이 2개의 CPU가 각각 memory controller를 내장하고 있으며 hypertransport[10] 방식으로 연결되어 있는 Opteron의 경우, 단일 프로세스에 비해 2배 가까운 메모리 대역폭의 향상을 가져왔으나, Xeon의 경우 2개의 MPI 병렬 프로세스로 실행한 것과 단일 프로세스 stream은 거의 동일한 성능을 보였다. 이것은 2개의 Xeon CPU가 공유하여 사용하는 단일 메모리 버스(FSB)의 대역폭의 이미 상위 한계 값에 도달했기 때문으로 판단된다.

5. 결론 및 향후 계획

클러스터 컴퓨팅 환경에서의 계산노드로서의 적합성을 판별하기 위해 다양한 플랫폼에 대하여 HPL과 NPB 그리고 Stream 등의 벤치마크를 수행하였다. HPL 등의 결과에서 알 수 있듯이 일반적으로 단일 CPU환경에서 x386 기반의 플랫폼들이 동



[그림 5] 병렬 Stream Test

급의 x86_64기반의 AMD 장비에 비해 전반적으로 높은 성능을 보였다. 이것은 기본적으로 AMD의 클럭 속도가 인텔 기반의 장비에 비해 상당히 뒤떨어지는 한계 때문으로 보인다. 단 Stream 병렬 테스트에서 볼 수 있듯이 Opteron의 hyperthreading 기술은 다중 CPU의 병렬 환경에서 인텔 기반의 플랫폼에 비해 메모리 대역폭에 관한한 월등한 성능상의 우위를 보였으며, NPB의 정수 정열(Integer sort)연산에서도 타 시스템에 비해 두 배의 cache 크기를 가진 Opteron이 최고의 성능을 보여주었다. 이러한 Opteron의 장점은 대규모 데이터 처리 및 전달이 대부분인 고성능 컴퓨팅 분야에서 매우 중요한 장점이 될 수 있다. 또한 인텔의 현재의 hyperthreading 기술은 일반적인 HPC 환경에서는 오히려 성능을 저하시키는 요인이 될 수 있기 때문에 사용을 하지 않는 것이 적절하다고 본다. 차후 추가적으로 PowerPC, IA64 등 64비트 기반의 플랫폼들에 대해 중점적으로 벤치마킹을 수행할 계획이다.

참고문헌

[1] T. Sterling, D. Becker, D. Savarese, et al. "BEOWULF: A Parallel Workstation for Scientific Computation", Proceedings of the 1995 International Conference on Parallel Processing (ICPP), Vol. 1, pp. 11-14, August 1995.

[2] Guide to Myrinet-2000 Switches and Switch Networks" http://www.myri.com/myrinet/m3switch/guide/myrinet-2000_switch_guide.pdf

[3] InfiniBand Trade Association. InfiniBand Architecture Specification, Release 1.0, October 24 2000.

[4] The Quadrics Network : High-performance Clustering Technology, http://www.c3.lanl.gov/~fabrizio/papers/ieeemicro.pdf

[5] Jack Dongarra, "Present and Future Supercomputer Architectures and their Interconnects", International Supercomputing Conference 2004, http://www.netlib.org/utk/people/JackDongarra/SLIDES/dongarra-isc2004.pdf

[6]http://www.netlib.org/benchmark/hpl/

[7]http://www.nas.nasa.gov/Software/NPB/

[8]http://www.streambench.org

[9]http://developer.intel.com/technology/hyperthread/

[10]http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/Hammer_architecture_WP_2.pdf