# Beowulf Cluster

박정환 (jhpark@nurapt.kaist.ac.kr) 1998년 10월 28일

이 문서는 Beowulf Cluster에 대한 간단한 소개입니다.

## 목 차



## 제 1 절 소개

웹서버, DB 서버, 메일서버, FTP 서버, PROXY 서버 등 전산 업무 환경에서 리눅스 서버의 안정 성, 성능, 보안성 등을 기존의 유닉스 서버나 NT서버와 비교해볼때 리눅스 서버의 가격대 성능비 의 탁월함은 어느 누구도 부인하지 못할 것이며 요즘들어서는 상용 소프트웨어 업계인 넷스케이 프, 오라클, 인포믹스, 코렐 등의 참가로 리눅스는 서버 및 개인 사용자 OS로서 새로운 전기를 맞고 있다. 이에 못지 않게 컴퓨팅 속도를 최우선으로 하는 테크니컬 컴퓨팅 환경에서도 이미 알 파 프로세스를 사용한 리눅스 웍스테이션은 탁월한 부동소수점 연산으로 인해 현재 새로운 관심 을 받고 있으며, SGI, IBM, SUN, HP, DEC에 의해 점유되었던 테크니컬 컴퓨팅 영역을 급속한 속도로 알파 리눅스가 파고 들고 있으며, 또한 리눅스는 단일 프로세서나 SMP에만 머물지 않고, 리눅스의 강력한 네트워킹을 통한 네트웍 클러스터를 구축하여 수퍼 컴퓨터 영역에 도전하고 있 다.

리눅스 네트웍 클러스터의 한 형태인 Beowulf cluster은 1994년 NASA의 CESDIS에서 16 node cluster를 리눅스와 표준 소프트웨어 패키지를 이용하여 개발하였는데, 이런 바탕에는 그 당시 NASA에서 사용중인 CRAY의 임대 기간 종료와 더불어 새로운 병렬처리용 수퍼 컴퓨터 개발을 직접해보자는 분위기가 있었다. 이때 리눅스 네트웍 드라이버의 개발이 CESDIS에 의해 주도적 으로 이루어져 현재 널리 사용되고 있다.

영화 '타이타닉'에서도 128개의 알파 리눅스 클러스트를 이용한 병렬처리를 통하여 리눅스의 우수 성을 성공적으로 보여주었으며 현재 미국 로스 알라모스 국립 연구소에서는 하이텔 하드웨어 통신판매란에서 흔히 볼수 있는 70개의 알파 리눅스 박스로 구성된 Avalon이라는 Beowulf 시스 템을 직접 만들었으며, 현재 Top500 supercomputers list (http://www.top500.org/top500.list.html) 에 315위로 등록 되어 있다. 비슷한 순위의 수퍼 컴퓨터로는 SGI의 64개의 프로세서를 사용하는 ORIGIN 2000 시스템(304위)이다. 참고로 국내의 열악한 수퍼 컴퓨팅 환경을 대변하듯 315순위 이상으로 기록된 국내 수퍼컴퓨터는 6대뿐으로, 이공학분야로는 SERI, KT, 삼성만 보유하고 있 을 뿐, 국내 대학이나 일반 연구소에서 제대로 된 수퍼컴퓨터는 전무한 상태이다.

Avalon을 통해 Beowulf cluster의 특징을 간단히 살펴보면, 우리가 흔히 볼수 있는 알파 리눅스 박스를 이용하였기 때문에 벤더의 특별한 지원 없이 설치가 가능할 정도로 간단하였으며, 리눅스 박스마다 개별적인 파워 서플라이와 쿨링 장치로 인해 항온 항습 장치가 필요치 않는 점도 특징 으로 들 수 있으며, 유지 보수에 대한 비용 또한 크게 절감할수 있다.

본 글에서는 PC hardware를 사용하여 수퍼 컴퓨팅에 도전하고 있는 Beowulf cluster의 특징과 성공 요인들을 살펴보고, Beowulf 클러스터를 구성하기 위해 필요한 하드웨어와 소프트웨어 환경 및 필자가 구성해본 4-node mini beowulf 클러스터에 대해서 소개함으로 수퍼 컴퓨팅 영역에 도전하고 있는 리눅스의 또 다른 모습을 소개할려고 한다.

## 제 2 절 Beowulf Cluster

공학을 하고 있는 필자 뿐만 아니라, 모든 사람들이 "better, faster, cheaper"한 컴퓨터를 원하고 있다. 그러나, 웍스테이션 구입을 해본 사람들은 잘 알겠지만, 이런 바램과는 멀리 하이-엔드 컴 퓨터 이른바 우리가 잘알고 있는 웍크스테이션 벤더인, HP, SGI, SUN, DEC의 제품들을 보면 우 리의 이러한 바램과는 달리 좋은 것은 하나도 없지만, PC보다 약간 빠른 속도로 인해 엄청난 비 용을 지불해야 하는 경우를 만나게 된다. 특히 이러한 하이-엔드 컴퓨팅 환경에서는 제한된 벤더 의 수로 인해 시스템간의 하드웨어와 소프트웨어 호환성의 부족으로 PC와 같이 사용자가 직접 조립하여 적당한 소프트웨어를 설치하여 사용한다는 것은 생각 할수 없는 현실이다. 당연히 이러 한 특성으로 인해 하드웨어와 소프트웨어의 가격은 비쌀수 밖에 없고 유지와 보수에 드는 비용 또한 무시할 수 없는 현실이다. 이러한 고가의 하이-엔드 컴퓨팅 환경의 대안으로 NASA에서 Beowulf 프로젝트라는 이름으로, 시장에서 누구나 살수 있는 PC 하드웨어와 누구나 사용할수 있 는 소프트웨어인 리눅스를 이용한 하이엔드 컴퓨팅 영역에 대한 새로운 시도를 하게 되었다.

Beowulf 프로젝트가 성공하게 된 몇가지 중요한 요인으로 PC 프로세스(Intel x86, DEC Alpha, Power PC)의 급속한 발전이다. 근래에 들어 부동소수점 연산에 획기적인 향상이 있었으며, PC 프로세서의 발전 속도는 이미 기존의 하이-엔드 프로세스보다 3배이상 빠르며, 리눅스가 사용가 능한 알파 프로세서는 이미 그 속도를 앞질렀다. 이와 더불어 고급 유닉스 서버나 웍스테이션에 만 사용되었던 SCSI 장비의 대중화와 그리고, 네트웍 장비의 급속한 발전과 더불어 100Mbps switching hub의 가격하락으로 인해 Beowulf 클러스터를 구축할 수 있는 하드웨어를 PC급으로 도 충분히 구축 가능해졌다. (외국에서는 100Mbps switching hub를 포트당 100불 미만으로 구입 가능하지만, 국내는 여전히 고가이다.)

물론 이러한 하드웨어 발전만으로 Beowulf 프로젝트가 성공할수 있었던 것은 절대 아니다. 이렇 게 다른 목적을 두고 발전해온 하드웨어를 리눅스는 하나의 시스템으로 통합하여 Beowulf 라는 새로운 타입의 병렬컴퓨터를 만들어 낸 것이다. 너무나 잘 아는 얘기지만, 리눅스는 기존의 상용 유닉스보다 더 낳은 환경을 제공하였는데, 바로 소스 코드의 100% 개방으로 Beowulf cluster에 적합하게 자유롭게 소스 코드를 수정하고 새롭게 개발하여 배포할수 있는 환경을 제공하였는데, 이러한 리눅스의 자유함이 없었다면 이 프로젝트는 실패하였을 것이다. 이외에도 MPI, PVM같은 message passing library의 표준화가 이루어져 상용 병렬컴퓨터와 마찬가지로 Beowulf cluster에 서도 자유롭게 사용할수 있게 됨으로 인해 Cray로 시작하는 수퍼 컴퓨터에서 시작해서 Desktop PC까지 일관된 프로그래밍 작업을 할수 있게 되었다.

수퍼 컴퓨터를 구조적으로 살펴보면, RC5/DES 크랙같이 전세계적으로 internet으로 연결된 컴퓨 터들도 하나의 병렬처리 수퍼컴퓨터로 볼수 있으며, 이와 반대로 상용 수퍼 컴퓨터로서 초고속 네트워크로 구성된 MPP, vector, SIMD cluster도 있다. 이러한 상용 수퍼 컴퓨터는 시스템 디 자인과 개발 시간으로 인해 급속히 발전하고 있는 마이크로 프로세서를 즉각적으로 수용하기 힘 든 단점이 있는 반면 Beowulf Cluster는 각각의 node가 개별적인 OS를 가지고 있으며 고속의 local area network를 이루고 있기 때문에 뛰어난 확장성이 있으며, 급속히 발전하고 있는 마이 크로 프로세서를 즉각적으로 수용하기가 용이하다. 물론 SUN같은 대중적인 웍스테이션을 이용하 여 network clustering을 할 수 있지만, 고비용으로 인해 확장성에 한계를 가지게 된다. Beowulf cluster는 각 slave node는 기본적으로 M/B, RAM, NIC, CPU만을 가지고 키보드, 마우스, 모니터는 공유기를 통하여 master node에서 전체 node를 제어하고 전적으로 병렬처리에 사용된다는 점에서 일반적인 workstation cluster와 구분된다. 가장 간단한 beowulf cluster는 switching hub 없이 직접 cross link하여 2 node로 구성할 수 있 는데 각 node에 CPU를 2개씩 설치한다면 간단한 4 processor mini beowulf cluster를 만들수 있 다. 초기 beowulf cluster은 8-node 또는 16-node로 ethernet을 바탕으로 시작하였지만, 현재 제 대로 된 Beowulf cluster을 구성하기 위해서 shared 100MBPS급인 일반 fast ethernet hub 보다 는 각 node간의 network I/O bandwidth를 보장하기 위해서 dedicated 100MBPS를 제공하는 switching hub가 필수적이다. 현재 16포트 switch를 사용할 경우 최대 16 node를 구성할 수 있 으며, network I/O의 bandwidth가 부족할 경우 각 node에 NIC를 추가 설치하여 network channel bonding을 하여 node수는 줄어들지만, 네트웍 대역폭을 증가시킬수도 있다. (여기서 말하 는 node는 M/B의 수로 각각의 node는 dual 또는 quad cpu로도 가능하다.)

100 node이상 설치하여 운영하고 있는 곳은 대개 gigabit uplink가 가능한 100MBPS switch를 사 용하여 각

node를 연결시키고 이러한 switch를 gigabit switch로 연결하여 각 node간의 network 를 통한 I/O bandwidth를 손실을 최소화 한다. 이런 방식으로 네트웍을 구성할 경우 극단적인 경우를 생각하면, beowulf cluster과 네트웍 구성은 틀리지만, Intel의 Pentium Pro CPU를 9152 개를 사용한 ASCI 의 Red System (현재 Top500 list에서 1위)으로 beowulf cluster의 가능성을 충분히 짐작할 수 있을 것이다. 이와 같이, Beowulf cluster의 각 node는 시장에서 쉽게 구할 수 있는 PC 하드웨어로 구축되지만, 네트웍 구성에 사용되는 장비는 예외로서, 전체 Beowulf cluster예산 중에 큰 부분을 차지하고 있으며 기술의 발달로 Myrinet같은 새로운 네트웍 장비가 등장하고 있으나, channel bonding한 네트웍 구성에 비해 가격대 성능비에서 큰 효과를 보이지 못하고 있어 지금은 널리 사용되고 있지 못하다.

## 제 3 절 Beowulf Cluster - GALAXY

위에서는 Beowulf cluster에 대한 일반적인 특징과 구성방식에 대해서 설명을 하였다. 여기서는 필자가 직접 구성해본 Beowulf cluster의 구성방법과 특징에 대해서 간단히 소개를 하겠다. 대부 분의 software적인 셋팅관련은 LINUX HOW-TO 문서에 잘 나와 있으므로 참조하기 바란다.

97년 봄 필자의 연구실에서는 SUN Sparc 20와 HP C180 시스템을 사용하고 있었지만, 실원들 의 수치계산량을 수용하기에는 역부족이어서 그 당시 새로운 시스템을 찾는 중 앞서 말한 바와 같이 펜티엄 프로에 리눅스를 설치하여 사용하는 것이 가격대 성능비가 최적이라는 판단 아래에 펜티엄 프로(200MHz)를 구입하여 리눅스를 설치하고, 전공관련 코드 패키지를 설치하여 적극 활 용하기 시작한 것이 리눅스를 처음으로 테크니컬 컴퓨팅에 이용한 것이었다. 물론 그 전에야 남 들 다 하는 것처럼 리눅스로 부팅했다가 NT로 부팅했다가 하면서 이것 저것 장난 쳐보는 단계였 지, 직접 연구에 활용하기는 처음이었다. 그 이후 97년 겨울 듀얼 펜티엄II 300MHz로 업그레이 드하였으며, 병렬처리를 염두에 두고 98년 봄 싱글 펜티엄II 300MHz를 두대를 더 구입하여 1GB SCSI하드에 OS만 설치하고 NFS를 이용해서 사용자 디렉토리를 시스템간 서로 공유하여 사용 하였다. 이때 전공분야중 Monte Carlo시뮬레이션을 하는 코드가 PVM library를 이용한 병렬처리 를 지원하였기 때문에, g77과 PVM library를 이용하여 컴파일하여 사용하게 되었다. Monte Carlo 시뮬레이션 특성상 프로그램 시작과 끝에만 각 node들간의 데이타 전송이 필요하기 때문 에, shared 10Mbps환경에서도 무난하게 프로세스 갯수만큼, 4배정도의 속도증가를 가져왔으며 상 당히 만족하게 사용하였다. 그러나 shared 10Mbps환경에서 4개의 프로세서를 사용하여 number crunching 작업을 하기에는 너무 열악한 환경이어서 생각지도 못하였으나, 7월에 KAIST Backbone network이 기존의 FDDI에서 ATM으로 이전하면서 enduser에게 dedicated 10MBPS를 제공함에 따라, 연구실내에서 Baynetwork에서 나온 24포트 10/100M Switch인 BayStack 350T-HD를 구입하게 되었다.

기존의 시스템으로 새롭게 workstation cluster가 아닌, Beowulf-class의 시스템으로 구축하기 위 해, 일단 master node에만 하드 디스크를 장착하고 slave node는 diskless로 하여 기존에 있던 OS용 1GB SCSI하드를 제거하고 slave node는 M/B, 1 CPU, 10/100M NIC, 64MB RAM, ISA VGA, keyboard로만 구성 하였다. 실제 VGA는 필요없었지만, 시스템 부팅을 하기위해, 어쩔수 없이 굴러다니는 ISA 비디오 카드를 꽂게 되었고 mini beowulf 시스템이라 키보드 공유기를 구 입하여 master node에서 전체를 관리하는 것은 일단 생각지 않았다. 이와 반면에 master node에 서는 M/B, 2 CPU, 2 10/100M NIC, 512MB RAM, 22GB SCSI HDD, PCI VGA로 구성하였다. 일단 BayStack 350T-HD의 VLAN 기능을 이용하여 24포트중 Beowulf cluster가 사용하는 3포 트를 VLAN2 로 하고, 나머지 포트는 VLAN1에 속하도록 설정하였고, Master node에는 2개의 NIC를 설치하였는데 eth0는 VLAN1 에 연결하여 외부 네트웍에서 접근가능하게 하고 eth1은 VLAN2에 연결하여 node들간에 private network를 구성하였다.

사용한 리눅스 배포판은 RedHat 5.1로서, RedHat사에서 NASA의 CESDIS와 함께 Extreme Linux라는 RedHat 5.0에 기초하여 인텔 X86을 위한 Beowulf cluster용 배포판을 판매하고 있 으며 현재 anonymous ftp server로부터 다운로드 받을수 있다. 필자가 설치할 시기에는 anonymous ftp server로부터 다운로드 받을 수 있는 상황이지만, 그 당시에는 Extreme Linux를 구할수 없었고, 5.0에 기준하고 있어 별다른 메리트가 없어 보여 RedHat 5.1을 설치하고 관련 소 프트웨어를 직접 설치하였다. 그러나, Intel X86환경에서 NIC를 각 노드에 2장씩 사용하여 channel bonding하여 node간의 network 대역폭을 2배로 증가시키거나 소프트웨어 설치의 수고를 덜기 원한다면, Extreme Linux를 사용하는 것이 바람직할 것 같다.

Master node는 diskless slave node를 위해 NFS-ROOT server가 되게 하였고, NIS서버는 설치 하지 않고

각 node들을 위한 /tftpboot 디렉토리의 passwd, group 파일을 master node의 파일과 하드링크 시켜놓았다. slave들이 bootp를 통해 ip address를 설정할수 있게 하였으며, DNS 서 버를 설치하여 private network내에서 node들간의 네트웍 환경에 문제가 없게 하였다. 설치한 공학 관련 소프트웨어는 앞부분에서 말한 ASCI Red 수퍼컴퓨터에서 사용중인 펜티엄 프 로전용 BLAS, FFT math library를 설치하였고, message passing library로, MPICH, LAM MPI, PVM을 설치하여 각 노드간 message passing을 통해 병렬처리를 가능하게 하였으며, 기타 병렬 처리용 math library를 설치하였다. Beowulf cluster 사용자들은 master node로만 네트웍으로 접 근 가능하기 때문에, master node에 job이 집중될 것을 우려하여 비록 소규모 시스템이지만, DQS를 사용하여 사용자들의 batch job을 관리하게 하여 사용자들의 job이 프로세스마다 균등하 게 배분할수 있게 하였고, 최종적으로 이렇게 설치된 Beowulf System에 대해서 NAS Parallel Benchmarks (NPB)을 이용하여 테스트 해보았으며 필자의 전공관련 문제를 PETSc라는 툴킷을 이용해서 프로그래밍하여 테스트 해보았다. 테스트해보면서 몇가지 문제점이 나타났는데, 일단 CPU가 4개인 반면에 NIC는 3개여서 데이타 병목현상이 발견되었으며 약간의 성능 저하가 나타 남을 발견하였다. 정확한 parallel benchmark 결과를 싣지는 않지만, 거의 85% 이상 parallel efficiency를 보여주었으며, NIC를 하나더 추가하고 channel bonding을 한다면 더욱 더 효과적인 시스템이 될 것이라 생각된다.

# 제 4 절 마치면서

짧은 내용이지만, 수퍼 컴퓨팅에 도전하고 있는 리눅스에 대해서 간략하게 소개하였다. 비록 이러 한 Beowulf cluster을 일반 리눅스 이용자가 사용하기에는 거리가 멀어 보이지만, 리눅스는 업무 환경과 관련된 서버 시장은 물론, 고도의 정확성과 초고속의 계산력을 필요로하는 테크니컬 컴퓨 팅 영역에서도 자유함으로 도전하는 리눅스의 무한한 가능성을 엿볼수 있다.

88년부터 시작한 국내 수퍼컴 역사는 올해로 10년이 넘게 되었고, 국내 수퍼컴 수는 몇 십대되는 것 같지만, 대부분 '무늬만' 수퍼컴인 경우가 대부분으로, 규모면에서 미국의 1백11분의 1, 일본 의 40분의 1, EU의 30분의 1수준에 불과한 실정으로 선진국들과 비교해서 열악하기만 하다. IMF로 인한 국내 경제여건악화로 SERI의 수퍼컴퓨터 3호기의 도입이 전면 보류된 상황이며, 올 여름 비난리로 인해 기상청만이 신규로 1천 3백만달러의 100GFLOPS의 기상예보전용 수퍼컴만이 도입예정이다. 이런 국내 상황으로 볼때 앞서 말한 로스알라모스 국립연구소의 20GFLOPS의 계 산능력을 가진, 15만 달러짜리 home made supercomputer, Avalon은 강력한 '펭귄' 군단의 가능 성울 보여준다. 이러다 보니 심지어 어떤 사람은 북한, 이라크, 쿠바 같은 나라가 군사용으로 beowulf cluster을 사용하지 않을까? 걱정하기도 한다.

Beowulf cluster이 기존의 수퍼컴퓨터에 비해 비용은 1/10이하지만, 그래도 비용면에서 부담은 된 다. 그러나, 대학의 학과 PC룸에 깔려 있는 Window를 밀어버리고, 리눅스를 설치한다면 쉽게 소 규모 beowulf cluster를 구축할 수 있으며, 만일 KAIST같은 대단위 연구 환경이라면 기관에서 master node와 network 장비만 초기 지원해준다면, 각 연구실에서는 slave node 용으로 리눅스 박스 하나씩 가져와 추가 설치하여 강력한 수퍼컴퓨팅 환경을 쉽게 구축할 수 있을것으로 생각되 며 또한 언제 든지 해체하여 다시 각 연구실로 가져갈 수도 있을 것이다.

자, 이제 남은 일이라고는 고가의 웍스테이션만 고집하지 말고 눈을 돌려 2대의 Dual PII 리 눅스 박스를 channel bonding하여 cross cable로 직접 연결해보자. 고가의 switching hub 없이도 4 processor beowulf cluster의 성능에 아마 감동할 것이다.