

리눅스 클러스터링

한국정보통신대학원대학교 유찬수

1. 서 론

클러스터는 대량 과학기술 계산을 위한 시스템 또는 웹서버 등과 같은 상용서버를 값싸게 만들 수 있다는 점에서 최근 많은 연구 및 개발이 되고 있다. 본 논문에서는 클러스터에 대한 개괄적인 내용과 특히 리눅스 운영체제를 운용하는 클러스터를 중심으로 알아보도록 하겠다.

2. 클러스터에 대한 열 가지 문답

- Q1: 클러스터란 무엇인가 ?
 Q2: 클러스터를 사용하는 이유는 무엇인가 ?
 Q3: 클러스터가 최근 관심이 집중되는 이유는 무엇인가 ?
 Q4: 클러스터 붐의 또 다른 중요한 이유가 있다면 ?
 Q5: 리눅스를 클러스터에서 사용하는 이유는 무엇인가 ?
 Q6: 클러스터 컴퓨터가 유용한 분야는 ?
 Q7: 가장 널리 알려진 클러스터에는 어떠한 것이 있는가 ?
 Q8: 산업체에서의 개발상황은 어떠한가 ?
 Q9: 이전에는 이러한 클러스터의 개념이 없었는가 ?
 Q10: 클러스터의 연구과제 또는 문제점은 무엇인가 ?

- Q1: 클러스터란 무엇인가 ?
 A1: 다수의 PC 또는 워크스테이션을 연결하

여 하나의 시스템으로서 함께 작동하는 컴퓨터를 말한다[1]. 1998년 12월에 구성된 IEEE TFCC (Task Force on Cluster Computing) [2]에서는 클러스터를 "A type of parallel or distributed system that consists of a collection of interconnected whole computers used as a single, unified computing resource (providing capability, availability, and scalability)"로 정의하고 있다. 그림 1은 전형적인 형태의 클러스터의 모습을 보여주고 있다.

Q2: 클러스터를 사용하는 이유는 무엇인가 ?

A2: 무엇보다도 가격대 성능비가 우수하기 때문이다. 클러스터는 대량생산되는(COTS:commercially off-the-shelf) 부품, 즉 성능은 우수하나 가격이 저렴한 PC, 네트워크장비 및 무료 소프트웨어 등을 최대한 사용함으로써 가격을 기존의 등급 병렬컴퓨터에 비해서 10분의 일 이하로 낮출 수 있다. 실제로 미국 로스알라모스 국립연구소의 아발론 클러스터[3]의 경우에는 140개의 알파 시스템을 연결하여 10 Gflops 시스템을 구성하는데 15만불의 비용이 들었다고 보고하고 있다. 이 정도의 성능은 1998 11 현재 세계에서 113번째로 빠른 컴퓨터로서(Top 500 Supercomputers List[4]), 136-node의 Cray T3E 가 115 Gflops로서 약 10배의 성능을 보이지만 가격은 40배 정도라는 점을 감안한다면 클러스터의 우수한 가격대 성능비를 알 수 있다.

- Q3: 클러스터가 최근 관심이 집중되는 이유는 무엇인가 ?



그림 1 전형적인 클러스터

A3: 가격대비 성능이 우수한 부품이 대량생산되고 있으며 특히 고속 네트워크 장비의 개발과 양산으로 인해 고성능 클러스터의 제작이 가능하다고 보기 때문이다. 클러스터 시스템이 원하는 성능을 내기 위해서는 다수의 PC를 연결하는 연결망이 충분한 성능을 내주어야 한다. 기존의 병렬컴퓨터는 연결망에 있어서 독자적인 디자인으로 구성되어 프로세서간 통신이 매우 고속으로 이루어지는데 반하여, 일반적인 PC 간의 통신수단들은 10Mbps(Ethernet) 정도의 속도를 넘어설 수 없었다. 그러나 Fast Ethernet(100Mbps), Gigabit Ethernet(1Gbps), Myrinet(1.28Gbps)[5], ServerNet(1Gbps)[6] 등과 같은 고속 근거리 네트워크의 등장으로 속도문제가 해결되었고, 따라서 고성능의 클러스터 시스템

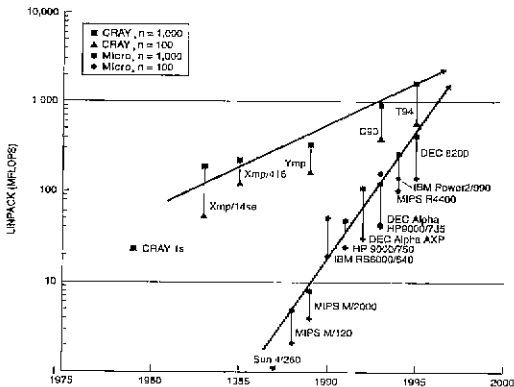


그림 2 고성능마이크로프로세서를 기반으로 한 시스템과 병렬시스템의 성능비교[6]

을 값싸게 구성할 수 있게 되었다. 즉, 이전에는 소위 Killer Micro의 등장으로 고성능 PC가 보편화되고 이에 따라 고성능 시스템 시장을 잠식하였으나(그림 2), 이들을 고속으로 연결할 수 있는 소위 Killer Switch(그림 3)의 등장으로 클러스터를 보편화시키며 대형시스템 시장의 판도를 바꿀 수 있는 큰 위협이 되고 있다고 할 수 있다.

네트워크 종류	데이터 전송속도	공급사 및 웹주소
Fast Ethernet	100Mbit/s	Digital 외 (http://www.compaq.com/products/networking/) 외
Gigabit Ethernet	1 Gbit/s	Digital (http://www.compaq.com/products/networking/) 외
Myrinet	1.28 Gbit/s	Myrinet (http://www.mvri.com)
ServerNet II	1.25 Mbyte/s	Tandem (http://www.servenet.com/)
Memory Channel	100 Mbyte/s	Digital (http://www.digital.com/info/hpc/systems/symc.html)
Synfinity	1.6 Gbyte/s	Fujitsu (http://www.fsl.com/)
SCI	400 Mbyte/s	Dolphin (http://www.dolphinics.com/)
ATM (OC-12)	155 (622) Mbit/s	Fore Systems (http://www.fore.com/) 외
HiPPI	800 Mbit/s	ODS (http://www.ods.com/) 외

그림 3 고속네트워크 및 데이터전송속도[6]

Q4: 클러스터 붐의 또 다른 중요한 이유가 있다면 ?

A4: 클러스터를 가능한 고성능, 고가용성 솔루션으로 되게 하는 또 다른 중요한 원인은 표준화된 소프트웨어환경의 급속한 보급이라고 말할 수 있다. 첫째는 프리웨어인 리눅스가 안정적으로 제공된다는 점이며, 둘째는 표준화된 병렬 프로그래밍 환경이라고 말할 수 있다. MPI(Message Passing Interface)[7]의 경우, 업계 표준으로 자리잡고 폭넓게 사용되는데 쉬운 인터페이스와 적절한 성능을 보여줌으로써 병렬프로그램도 쉽게 개발되며 또한 다른 컴퓨터에서의 호환성이 보장되어 쉽게 이식될 수 있다는 장점이 있다. 1992년부터 시작된 MPI에 대한

연구는 1994년 5월 MPI 1.0이 완성된 이후에도 지속적으로 발전하고 있으며, MPI 1.2의 경우 전체 125개의 API 중에서 기본적인 6개의 API 만을(그림 4) 사용하여서도 충분히 병렬프로그램을 작성할 수 있어 개발이 용이해졌다.

MPI API	프로토타입	기능
MPI_Init	(int argc, char **argv)	MPI 초기화
MPI_Comm_size	(MPI_Comm comm, int *size)	통신하는 프로세스 그룹의 크기
MPI_Comm_rank	(MPI_Comm comm, int *rank)	프로세스그룹 내에서의 해당 프로세스의 고유번호
MPI_Send	(void *buf, int count, MPI_Datatype datatype, int source, int tag, MPI_Comm comm)	프로세스 그룹 내에서 데이터 송신
MPI_Recv	(void *buf, int count, MPI_Datatype datatype, int source, int tag, MPI_Comm comm, MPI_Status status)	프로세스 그룹 내에서 데이터 받기
MPI_Finalize	()	MPI 끝내기

그림 4 MPI(Message Passing Interface)에서의 6개 기본 API[8]

Q5: 리눅스를 클러스터에서 사용하는 이유는 무엇인가 ?

A5: 클러스터는 가격대비 성능을 높이며 빠른 제작을 위하여 될 수 있는 대로 쉽게 구입할 수 있는 부품들을 조립하여 구성하고 있다. 그런 의미에서 프리웨어인 리눅스를 사용하는 것은 당연하며, 또한 클러스터를 목적에 맞게 사용하기 위하여는 운영체제를 비롯한 시스템소프트웨어의 일부분을 수정할 필요가 있는데, 리눅스를 사용함으로써 이들을 가능토록 하고 있다. 현재 리눅스는 안정성이 높게 향상되어 있고 GNU 프로젝트를 통한 다양한 POSIX(IEEE Portable Operating System Interface) 표준을 따르는 많은 툴과 유틸리티가 제공되며 X 윈도우시스템, Tcl/Tk, Perl 등은 물론이고 Java Kit, PDF viewer, Netscape 등과 같은 다양한 어플리케이션과 개발 툴이 무료로 제공되고 있어 리눅스를 이용한 클러스터의 개발과 운용에 많은

도움을 주고 있다.

Q6: 클러스터 컴퓨터가 유용한 분야는 ?

A6: 고성능분야, 상용 고가용성서버분야 등으로 크게 구분할 수 있다[9]. 고성능분야는 Global Climate Modeling, Ocean circulation, Earthquake modeling 등 전통적으로 대량계산을 요하는 분야로서 최근에는 Car crash simulation, Heart simulation, Drug design, Web search, Transaction processing, Semiconductor modeling, Nuclear weapons, Computer chess, 유전공학 등에서부터 주가 및 고객분석과 같은 데이터마이닝 분야 등에서 많은 양의 데이터를 단시간에 처리할 수 있는 시스템을 요구하고 있다. 고가용성 서버분야는 웹서버나 이메일 서버에서와 같이 서비스가 중단되어서는 곤란한 분야를 말하는데, 클러스터 기반의 서버는 고가용성 외에도 다수의 개별적인 서비스를 처리하고 사용자의 증가와 함께 확장이 용이한 구조를 갖추고 있으므로 이와 같은 상용서버에 적합하다고 할 수 있다.

Q7. 가장 널리 알려진 클러스터에는 어떠한 것이 있는가 ?

A7: 미국 버클리대학의 NOW(Network of Workstations) 클러스터는 스위크스테이션을 Myrinet 스위치로 연결한 형태이며 여기서의 연구결과는 Inktome[10]의 웹검색엔진에 활용되었다. 미국 항공우주국(NASA)의 Beowulf 클러스터[11]는 PC를 Ethernet으로 연결한 것으로서 이와 유사한 형태의 클러스터는 미국의 여러 국립연구소를 비롯한 많은 기관에서 연구, 개발되고 있으며, Beowulf 클러스터를 지원하는 리눅스관련 소프트웨어를 특별히 ExtremeLinux [12]로 부르고 있다. 그림 5에 클러스터의 예를 주요한 특징과 함께 나타내었다.

Q8: 산업체에서의 개발상황은 어떠한가 ?

A8: 앞서 언급한 바와 같이 여러 회사에서 클러스터 기술에 바탕한 솔루션을 제공하는데, 일반적인 Intel PC 또는 Alpha PC를 이용하여 클러스터를 구성하는 공급업체로는 Alta Tech, VA Linux, DCG, Paralogic, Microway, Hitech

시스템	하드웨어
ASCI Red Supercomputer Sandia National Laboratory (http://www.sandia.gov/ASCI/Red/)	4536 processors (1.8Tflops) Ethernet/ATM
Computational Plant Sandia National Laboratory (http://www.cs.sandia.gov/cplant/)	39x Alpha 21264 (500MHz) Myrinet Linux RedHat 6.0, 2.2.10 kernel
Avalon Los Alamos National Lab. (http://cnls.lanl.gov/avalon/)	140x Alpha 21164A (533MHz) Fast Ethernet Linux RedHat 5.0
Lok Los Alamos National Lab (http://loki-www.lanl.gov/)	16x Pentium Pro, Fast Ethernet, 12 Gflops
RWCP PC Cluster II Real World Computing Partners (http://pdswww.rwcp.or.jp/)	64x Pentium Pro (200MHz) Myrinet Linux 2.1.119
COMPaS Real World Computing Partners (http://pdswww.rwcp.or.jp/)	8x 4-way SMP (200MHz Pentium) Myrinet/100MHz Ethernet Solaris x86
theHIVE Goddard Space Flight Center, NASA (http://newton.gsfc.nasa.gov/thehive/)	4 subclusters (E, B: 66, 64x 2-way Pentium Pro C: 16x 2-way Pentium III Xeon DL: 10x 4-way Pentium III Xeon) Fast Ethernet/Gigabit Ethernet/Myrinet

그림 5 클러스터 예

USA, Penguin Computing, Carrera 그리고 PC 업체인 Compaq, Dell 등이 있다. VA Linux는 Debian GNU/Linux의 개발, 리눅스의 IA64 (Merced: 인텔과 HP의 차세대 프로세서) 포팅 등 활발한 리눅스 활동과 함께 ClusterCity라는 Beowulf 클러스터를 개발하는 한편 Argonne 국립연구소에 256개의 2-way SMP를 연결하는 초대형 클러스터 프로젝트인 Chiba City를 구성하고 있다(각 회사의 홈페이지 주소는 문헌란 [13]을 참고바람).

Q9: 이전에는 이러한 클러스터의 개념이 없었는가?

A9: 클러스터의 기본적인 개념은 오래 전부터 있어왔고, 이를 적용한 시스템도 많이 있었으나 현재 논의되고 있는 클러스터와의 차이점은 과

거의 클러스터는 독립된 시스템간의 연결이 운영체제와 시스템 하드웨어 등 전체 디자인에 영향을 미쳤다는 점이다.

예를 들면 디지털의 OpenVMS 클러스터[1]의 경우에는 운영체제인 OpenVMS OS에 클러스터링을 위하여 많은 부분이 고려되어 있으며 Star Coupler나 CI interconnect과 같은 특별한 하드웨어를 통하여 상호 연결망을 구축하고 있다. 텐덤은 전통적으로 고장을 허용하는 (Fault tolerant) 시스템으로 유명한데 히말라야 시스템의 경우 여러개의 노드를 DynaBus와 Torus-Net로 연결하는 클러스터 구조를 가지고 있는데, 하나의 노드는 두 개의 프로세서로 구성되어 고장을 허용하고 있고 NonStop 운영체제를 클러스터용으로 특별히 개발 사용하고 있다. 또한 IBM S/390 Parallel Sysplex 역시 90년대 중반

기에 개발된 시스템으로서 ESCON (Enterprise System Connection) 연결결과 클러스터링 소프트웨어를 별도로 개발하였다.

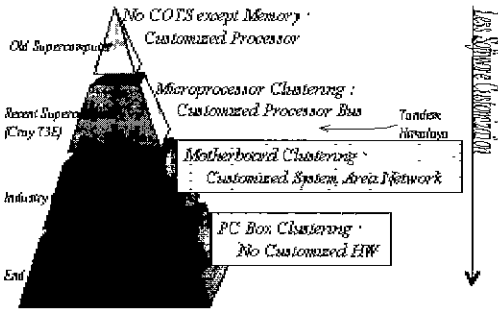


그림 6 클러스터링 수준

그림 6은 클러스터링하는 수준을 도식적으로 보여주는 그림으로 최근의 클러스터는 가장 하단의 PC Box Clustering을 추구하여 쉽고 빠르게 값싼 시스템 구성을 지원하고 있으며, 가장 상단의 클러스터는 프로세서에서부터 새롭게 디자인하여 병렬시스템을 구성하는 기존의 슈퍼컴퓨터를 의미하고 있다.

Q10: 클러스터의 연구과제 또는 문제점은 무엇인가 ?

A10: 클러스터를 사용하는 목적에 따라 여러 가지 다양한 요구조건들이 있겠으나, 일반적으로 가지는 문제점은 다수의 독립적인 PC들을 마치 하나의 시스템인 것처럼 보여주어야 하는 부분인데, 소위 단일이미지(Single System Image-SSI)의 지원이라고 불리운다[1,14]. 파일시스템의 경우, NFS와 같은 분산파일시스템을 사용하면 마치 하나의 파일시스템, 하나의 시스템인 것처럼 보여질 수 있는 것이 좋은 예이다.

클러스터에서 또 다른 연구과제는 효율적인 통신소프트웨어의 개발이다. 그림 7에서와 같이 프로세서와 네트워크 스위치의 성능은 매우 향상되었으나 통신 소프트웨어의 비효율성은 전체 통신통로에서의 병목지점을 형성할 수가 있다. 전통적인 TCP/IP 프로토콜은 업계 표준으로 광범위하게 사용되고 있으나 클러스터 시스템처럼 LAN보다도 가까운 거리에서는(SAN: System Area network), 예러가 많은 통신로의 가정하

에 구성된 프로토콜은 오버헤드만 과도하게 초래하는 결과를 낳게 된다. 일리노이대학의 FM (Fast Message), 버클리의 AM(Active Message), RWCP의 PM 그리고 컴팩/인텔/마이크로소프트의 VIA(Virtual Interface Architecture) 구조 역시 이러한 문제점을 해결하기 위한 방안이라고 할 수 있다[6].

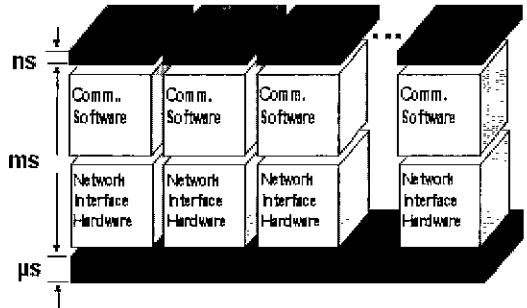


그림 7 통신소프트웨어에서의 병목현상[15]

클러스터 시스템에서 다수의 작업을 동시에 진행시키는 멀티태스킹 환경으로 사용하는 것이 앞으로의 발전방향이라고 할 때에 이를 지원하기 위한 작업스케줄링 및 부하균등화 등도 활발히 연구되는 과제이다. Genias사의 Codine (Computing in Distributed Network Environment), PBS(Portable batch Systems), Platform Computing 사의 LSF (Load Sharing Facility) 등이 있으며, 일리노이대학의 HPVM (High Performance Virtual Machine), 위스콘신대학의 Condor, 플로리다주립대의 DQS (Distributed Queueing System) 역시 좋은 예이다.

3. 결 론

1960년대 일리노이대학의 ILLIAC 프로젝트로부터 시작된 병렬컴퓨터에 대한 연구와 개발은 대규모 계산을 비롯한 여러분야에서의 활용을 기대하며 활발한 진행을 보였으나 고성능 프로세서가 출현하기까지는 매우 부진한 상황이었다. '90년대 중반 PC 및 마이크로프로세서의 등장으로 이를 이용한 병렬컴퓨터의 개발도 활성화되어 NCR, IBM, Fujitsu, Intel, Cray, Convex, Thinking Machines 등에서 많은 개발이 있었

다. 그러나 PC의 대대적인 성공으로 범용성과 상업성이 부족한 병렬컴퓨터에게는 오히려 그 관심이 위축되는 결과를 낳았다. 인터넷의 폭발적인 사용증가와 이를 활용하는 웹서버, 이메일 서버, 인터넷서버, 전자상거래용 서버 등 확장성 있는 대형시스템에 대한 요구가 많아지면서 다시 병렬컴퓨터, 특히 가격대 성능비가 우수한 클러스터에 대한 개발이 빠른 속도로 진행되고 있다. 프리웨어인 리눅스의 보급과 관련 소프트웨어의 개발 보급이 이러한 클러스터 붐을 가능하게 한다는 점에서 리눅스의 발전은 곧 클러스터의 발전으로 이어질 것으로 생각된다.

4. 추천문헌

(1) Gregory F. Pfister, In Search of Clusters, 2nd Edition, Prentice Hall, 1998 [1]: 이 책은 IBM의 저명한 학자인 Pfister의 최근 저서로서 가장 추천하고 싶은 책이다. 클러스터 컴퓨터에 대한 논의와 개발은 많으나 벤치마크나 개념적인 내용에서 제대로 정리되지 않은 현실에서 이를 나름대로 규정하고 클러스터와 관련된 대부분의 내용을 일관된 시각으로 정리하고 있다. 특히, 병렬컴퓨터 구조 및 병렬프로그래밍과 관련된 내용은 클러스터의 직접적인 내용은 아니지만 매우 흥미있는 내용이다. 그러나 초기 클러스터의 예로 든 IBM S/390 Parallel Sysplex, DEC OpneVMS Cluster, Tandem Himalaya 등의 시스템과 저자 자신이 현재 IBM에 몸담고 있다는 점에서인지, 클러스터를 일반인이 쉽게 조립할 수 있는 수준의 것보다는 IBM 정도의 회사가 구성할 수 있는 수준의 클러스터를 염두에 둔 듯하다. 실제로 IEEE TFCC의 게시판에서 논의된 저자의 논조에서도 이를 분명히 하고 있다("...if confined to clusters buildable by academics on a shoestring budget, I m out of here..."). 저자는 1985년 병렬컴퓨터의 연결망에서의 Hot Spot Contention 관련된 매우 유명한 논문의 저자이기도 하다. 이 책의 모든 자료에 대한 세미나 자료는 <http://vega.icu.ac.kr/~ppl/cluster/index.html>에서 구할 수 있다.

(2) Rajkumar Buyya, High Performance Cluster Computing: Architectures and Sys-

tems, Vol I. Prentice Hall 1999 [6]: 클러스터의 거의 모든 내용을 두루 망라하여 중요한 내용들을 엮은 여러 저자에 의해 씌여진 1999년판 최신 책이다. 따라서 장마다 내용이 질에 많은 차이가 있어, 가려서 읽을 필요가 있다. Part I. Requirements and General Issues, Part II. Networking, Protocols and I/O, Part III. Process Scheduling, Load Sharing, and balancing, Part IV. Representative Cluster Systems의 순서로 되어 있고 관심사에 따라 필요한 부분을 보면 되지만 High Speed Network 부분 및 Leight weight messaging 과 Part IV의 Beowulf, RWCP, COMPaS, ParPar, Pitt, RS/6000 SP, Web Server 등의 실제 클러스터 시스템에 대한 소개부분은 꼭 한번 볼 만하다. 이 책의 모든 자료에 대한 세미나 자료는 <http://vega.icu.ac.kr/~ppl/cluster/index.html>에서 구할 수 있다.

(3) B.Saphir, et.al, "Production Linux Clusters." Supercomputing '99 Tutorial, Nov. 1999(http://www.nersc.gov/research/tribble/production_linux_clusters_v1.pdf)[9]. 1999년말 Supercomputing 컨퍼런스에서 튜토리얼로 진행된 내용으로 미국 국립연구소에서의 클러스터 구성경험을 바탕으로 한 실제적인 내용으로 어느정도 규모있는 클러스터를 꾸미려고 할 때 사용해야 하는 부품과 설치와 운용시에 부딪치는 문제들을 많이 제시하고 이에 대한 해결책을 보여주고 있다. 위의 자료는 발표자료로서 다소 이해에 어려움이 있으나 실제적인 데이터를 알 수 있다는 점에서 의미가 크다.

(4) IEEE Task Force on Cluster Computing(TFCC). <http://www.dgs.monash.edu.au/~rajkumar/tfcc>[2]: 1999년도 초에 구성된 IEEE TFCC의 홈페이지

(5) 클러스터링 기술특집, 한국정보과학회 병렬처리시스템 연구회지 제10권 제2호

(6) JPC4 Beowulf and Beyond, The Fifth Joint DOE/NASA PC Clustered Computing Conference, <http://www.csm.ornl.gov/JPC4/JPCagenda.html>: 1999년 10월에 열린 클러스터 관련 컨퍼런스로서 관련 자료가 파워포인트 형식으로 많이 있다.

(7) Extreme Linux Workshop #2. <http://www.extremelinux.org/activities/usenix99/docs/>: 1999년 6월에 열린 Beowulf 클러스터를 위한 Extreme Linux 컨퍼런스로서 관련 자료가 파워포인트 형식으로 많이 있다.

참고문헌

[1] Gregory F. Pfister, In Search of Clusters, 2nd Edition, Prentice Hall, 1998.

[2] IEEE Task Force on Cluster Computing (TFCC), <http://www.dgs.monash.edu.au/~rajkumar/tfcc/>

[3] <http://cnls.lanl.gov/avalon/>

[4] <http://www.top500.org>

[5] N.Boden. et.al., "Myrinet: A Gigabit-per-Second Local Area Network," IEEE Micro. Vol.15 No.1, pp.29--36, February 1995.

[6] Rajkumar Buyya, High Performance Cluster Computing: Architectures and Systems, Vol I, Prentice Hall 1999.

[7] MPI Forum, <http://www-unix.mcs.anl.gov/mpl/>

[8] Parallel Programming Course, <http://vega.icu.ac.kr/~cyu/courses/prog99/99prog.html>

[9] B.Saphir, et.al, "Production Linux Clusters," Supercomputing '99 Tutorial, Nov. 1999(http://www.nersc.gov/research/tribble/production_linux_clusters_v1.pdf).

[10] <http://www.inktome.com>

[11] <http://newton.gsfc.nasa.gov/thehive/>, <http://www.beowulf.org>

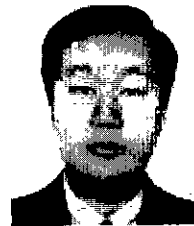
[12] <http://www.exetremelinux.org>

[13] Alta Tech : <http://www.altatech.com>
VA Linux: <http://www.valinux.com>
DCG: <http://www.dcginc.com>
Paralogic: <http://www.plogic.com>
Microway: <http://www.microway.com>
Hitech USA: <http://www.hitech-usa.com>
Penguin Computing: <http://www.penguincomputing.com>
Carrera: <http://www.carrera.com>
Compaq: <http://www.compaq.com>
Dell: <http://www.dell.com>

[14] Kai Hwang, et.al., "Designing SSI Clusters with Hierarchical Checkpointing and Single I/O Space," IEEE Concurrency, Jan.-Mar., 1999.

[15] D.E.Culler. High-Performance Clusters, Tutorial, PODC/SPAA, 1998.

유 찬 수



1982 서울대 전기공학과(학사)
1984 서울대 전기공학과(석사)
1991 펜실베이니아주립대 전산공학(박사, 병렬처리)
1984~1997 엘지전자 멀티미디어연구소 책임연구원
1998~현재 한국정보통신대학원대학교 공학부 조교수
E-mail cyu@icu.ac.kr
