

클러스터 시스템 기술 동향

부산대학교 오수철 · 정상화*

1. 서 론

최근에 사회의 모든 분야에서 정보화가 이루어지면서 컴퓨터로 처리해야 하는 정보가 증가하고 있다. 특히 인터넷의 급속한 보급으로 인터넷을 활용한 사회 활동이 급증하면서 웹서버, 전자상거래서버, VOD서버, 게임서버 등을 포함한 여러 분야에서 고성능 서버에 대한 수요가 증가하고 있다. 그러나, 고성능 서버는 각 서버 회사의 고유한 아키텍처에 기반하여 캐비닛 형태로 생산됨으로 가격이 수억에서 수십억원에 이르고 있어 구입 및 활용에 상당한 어려움을 가지고 있다.

이런 문제를 해결하기 위한 방안으로서 클러스터 시스템이 등장하였다. 클러스터 시스템은 그림 1처럼 저가의 고성능 PC 및 workstation을 고속 네트워크를 사용하여 연결함으로써 고성능 서버의 성능을 발휘하는 시스템을 제작하는 것으로 NOW(Network Of Workstations) 혹은 COW(Cluster Of Workstations)로도 불리운다.

클러스터 시스템의 장점은 저가의 상용제품을 사용함으로써 기존 고성능 서버보다 수배에서 수십배 작은 비용으로 동일한 성능의 시스템 구현이 가능하여 가격 대 성능비가 높다는 것이다. 그리고 사용자가 직접 상용부품을 사용하여 즉각적인 업그레이드가 가능함으로 시스템 유지 비용이 감소하고, 사용이 편리한 PC 및 workstation의 개발 환경을 그대로 사용할 수 있기 때문에 프로그램 개발이 용이하다.

클러스터 시스템이 고성능 서버에 필적하는 단

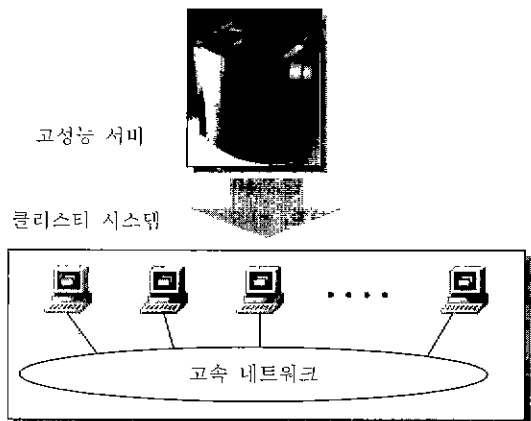


그림 1 클러스터 시스템 구성

일 컴퓨팅 시스템으로서의 성공 여부는 시스템 노드를 연결하는 네트워크의 성능과 지원 소프트웨어에 달려있다. 클러스터 시스템의 네트워크는 고성능 서버의 시스템 버스 역할을 담당하지만, 현재 클러스터 시스템에 사용중인 네트워크는 고성능 서버의 시스템 버스 성능과는 많은 차이가 있다. 이러한 문제를 해결하기 위해서 gigabit급의 네트워크들이 개발되고 있으며 통신에 필요한 소프트웨어 부하(overhead)를 최소화시킨 통신 라이브러리들이 개발되고 있다. 그리고 클러스터 시스템의 성능향상을 위해서는 각 노드에 분산된 자원을 효율적으로 관리하는 소프트웨어가 필요하다. 이를 통하여 다수의 컴퓨터로 구성된 클러스터 시스템을 사용자에게 하나의 시스템인 것처럼 보이게 하는 SSI(Single System Image)를 지원할 필요가 있으며 현재 이를 지원하는 소프트웨어들이 개발되고 있다.

* 중신회원

본 논문에서는 클러스터 시스템의 핵심적인 구성요소인 네트워크, 노드간 통신 라이브러리, SSI 지원 소프트웨어 등에 대하여 설명하고 최근 발전 동향을 소개한다.

2. 네트워크

2장에서는 현재 클러스터 시스템에 많이 사용되는 네트워크인 SCI(Scalable Coherent Interface), Myrinet, Fast Ethernet 및 Gigabit Ethernet의 특징과 성능에 대해서 설명한다.

2.1 SCI

SCI[1]는 SAN(System Area Network)을 지원하는 IEEE 1596-1992 표준으로 최대 64K개의 노드 연결이 가능하여 확장성이 매우 좋고 고속의 클러스터 시스템 구축을 가능하게 한다. 현재 SCI 표준을 만족하는 제품은 Dolphin Interconnect Solutions[2]사의 PC용인 PCI-SCI 카드와 Sparc station용인 SBUS-SCI 카드가 있다. SCI 카드는 링크 수준에서 400MByte/s의 대역폭을 가지며 2.3 μ s의 전송지연시간을 가진다. 애플리케이션 수준의 성능은 PCI-SCI 카드를 PC에 장착하고 공유메모리 방식을 사용했을 경우, 전송지연시간은 8 μ s이고 최대 대역폭은 메시지 크기가 16KByte 이상일 때 76MByte/s이다. SCI 카드는 point-to-point 연결을 사용하여 switch없이 링 형태로 연결하거나 전용 switch를 사용하여 연결할 수 있으며 switch는 cascading이 가능하다.

SCI 카드는 소프트웨어로 Linux와 NT용 드라이버를 제공한다. 그리고 다른 노드와의 데이터 교환을 위해 공유메모리 방식과 메시지 패싱 방식을 지원한다. 공유메모리 방식의 경우, 그림 2와 같이 노드 B의 로컬 메모리 영역을 SCI의 physical address space와 PCI address space를 사용하여 노드 A의 virtual address space로 매핑(mapping)함으로써 원격 노드의 메모리 지역 노드내의 메모리처럼 접근할 수 있다. 이러한 매핑은 하드웨어 수준에서 처리됨으로 소프트웨어 부하(overhead)가 없어 고속의 데이터 전송이 가능하다. 메시지 패싱 방식의 경우, 표준 메시지 패싱 라이브러리인 MPI(Message

Passing Interface)를 SCI위에 구현한 SCIMPI를 지원한다.

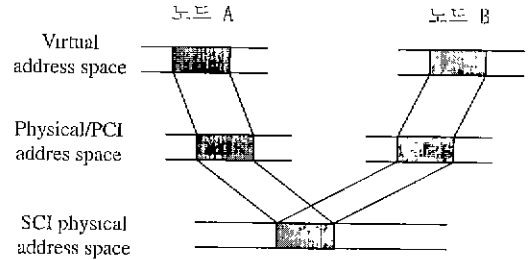


그림 2 SCI 카드의 메모리 매핑

현재 SCI를 사용한 클러스터 시스템은 독일의 Paderborn 대학[3]과 California 대학[24] 등에서 운영중이다. Paderborn 대학의 클러스터 시스템은 96개의 Dual Pentium II를 사용한 PC를 PCI-SCI 카드를 사용하여 연결한 것이다.

2.2 Myrinet

Myrinet은 Myricom[4]사에서 고속의 클러스터 시스템을 위해 만든 네트워크 카드로 PCI 버스에 장착된다. Myrinet은 링크 수준에서 1.28Gbps의 대역폭과 5 μ s의 전송지연시간을 가진다. 애플리케이션 수준에서는 PC에 메시지 패싱 라이브러인 Fast Message를 사용했을 경우, 전송지연시간은 10 μ s이다. 최대 대역폭은 메시지 크기가 3KByte 이상일 때 100MByte/s이다[16]. Myrinet 카드는 point-to-point 연결과 전용 switch를 사용하여 연결되며 switch는 cascading이 가능하다. Myrinet 카드에는 다른 네트워크 카드와 달리 최대 66MHz로 동작하는 LANai라는 RISC 프로세서가 장착되어 있으며, 사용자가 직접 프로그램 가능하여 원하는 작업 처리에 최적화시킬 수 있다.

Myrinet은 Linux와 NT를 위한 드라이버를 제공한다. 또한 다른 노드와의 데이터 교환을 위해 user-level 통신 라이브러리인 GM을 제공하고, GM위에 구현된 MPI 라이브러리를 지원한다.

Myrinet을 사용한 클러스터 시스템으로 Berkeley 대학의 NOW[5]와 Princeton 대학의 Shrimp[6] 등이 있다. NOW는 60대의

UltraSparc을 Myrinet을 사용하여 연결한 시스템이다.

2.3 Fast Ethernet, Gigabit Ethernet

Fast Ethernet은 LAN 용도로 개발되었으며 링크 수준에서 최대 100Mbps의 대역폭을 가진다. Fast Ethernet은 shared Fast Ethernet과 switched Fast Ethernet으로 구분할 수 있으며 최근에는 높은 성능을 가진 switched Fast Ethernet의 사용이 증가하고 있다. 현재 Fast Ethernet은 클러스터 시스템에 필요한 충분한 성능을 제공하지는 못하지만 SCI나 Myrinet에 비해 훨씬 저렴한 시스템 구축 비용 때문에 널리 사용되고 있다. 최근에는 1Gbps의 대역폭을 가지는 Gigabit Ethernet에 등장하였으나 높은 가격으로 인하여 많이 사용되고 있지 않다. Gigabit Ethernet의 애플리케이션 수준에서의 성능은 Alpha PC에서 TCP/IP를 사용했을 경우, 메시지 크기가 300KByte 이상일 때 30MByte/s의 최대 대역폭을 가진다[22]

Ethernet은 다른 노드와의 메시지 패싱을 위해 TCP/IP와 TCP/IP위에 구현된 메시지 패싱 라이브러리인 MPI와 PVM을 주로 사용한다. 그러나, TCP/IP는 통신에 소요되는 소프트웨어 부하가 많기 때문에 클러스터 시스템 전체의 성능을 저하시키고 있다.

Ethernet으로 연결된 클러스터 시스템을 보통 Beowulf[7]로 부르고 있으며, Fast Ethernet을 사용한 시스템으로 Los Alamos National Lab의 Avalon[8]이 있다. Avalon은 Alpha 프로세서를 사용한 PC 140개를 Fast Ethernet으로 연결한 시스템이다. 그리고 Gigabit Ethernet을 사용한 클러스터 시스템으로 Ames Lab의 Alpha-PC 클러스터가 있다[22]. AlphaPC 클러스터 시스템은 Alpha PC 24개를 Gigabit Ethernet을 사용하여 연결한 시스템이다.

3. 데이터 전송 방식

3장에서는 노드 사이의 데이터 교환을 위해 사용하는 메시지 패싱 라이브러리와 공유메모리 라이브러리에 대해서 설명한다.

3.1 메시지 패싱(Message-passing)

3.1.1 MPI(Message Passing Interface)와 PVM(Parallel Virtual Machine)

기존의 고성능 서버들은 독자적인 메시징 패싱 라이브러리를 가지고 있었기 때문에 이기종의 시스템 사이에 프로그램의 이식성이 좋지 않았다. 이러한 문제를 해결하기 위해서 표준화된 인터페이스를 가진 PVM[9]과 MPI[10]가 등장하였다. PVM은 1989년에 Oak Ridge National Lab에서 개발되었으며, MPI는 MPI Forum에 의해서 1992에 표준화가 되었다.

MPI와 PVM은 이식성과 기능성(functionality)을 고려하여 설계되었으며, 메시지 패싱의 초기화 및 각종 메시지 패싱 함수의 형식과 의미를 정의하고 있다. MPI와 PVM은 이식성을 위해 시스템의 네트워크 구조에 종속적이지 않은 인터페이스를 개발하였으며, 프로그램 작성에 필요한 다양한 형태의 메시지 패싱 함수를 지원함으로써 사용자의 프로그램 용이성을 돕고 있다.

MPI와 PVM은 이식성을 바탕으로 Ethernet, SCI, Myrinet 등을 포함한 대부분의 네트워크 카드에 구현이 되어 있으며, 고성능 서버에서 클러스터 시스템에 이르는 많은 서버 시스템에서 사용되고 있다. MPI의 성능은 Myrinet상에 구현된 Fast Message위에 구현했을 경우, 20 μ s의 전송지연시간과 메시지 크기가 200KByte 이상일 때 100MByte/s의 최대 대역폭을 가진다[16].

현재는 MPI와 PVM중에서 MPI가 보다 팔리 de facto 표준으로 자리 잡아 많은 사용자들이 주로 사용하고 있다.

3.1.2 U-Net

기존의 TCP/IP와 같은 메시지 패싱 시스템은 그림 3처럼 커널을 사용하여 통신을 수행한다. 이러한 방식은 시간이 많이 소비되는 커널 내부의 시스템 call을 사용하며, 사용자 프로그램에서 커널 및 네트워크 어댑터로 데이터를 여러 번 복사해야 하기 때문에 소프트웨어 부하가 크다. 이러한 문제 때문에 사용자 프로그램은 네트워크의 물리적 대역폭을 충분히 활용하지 못하였고, 고속의 네트워크가 등장해도 사용자 프로그램에서 얻을 수 있는 성능향상은 크지 않았다. 이러한 문제를 해결하기 위해서 그림 3처럼 메시지 패싱 과정에서 커널의 개입을 제거한 user-level

통신 라이브러리가 개발되었으며 대표적으로 U-Net이 있다.

U-Net[11]은 Cornell 대학에서 개발하였다. 커널의 개입을 없애기 위해 커널 내부에서 수행되던 프로토콜 처리 및 메시지를 저장하는 end-point(그림 4)를 user-level에 위치시키며, user-level에서 직접 네트워크 어댑터에 접근 가능하도록 네트워크 어댑터에 대한 virtual view를 제공한다. 이 방식을 통해 초기화를 제외한 모든 통신 과정이 커널 개입없이 user-level에서 이루어진다.

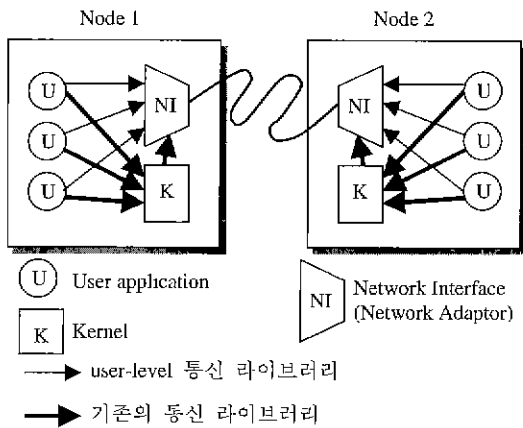


그림 3 통신 시스템의 구조

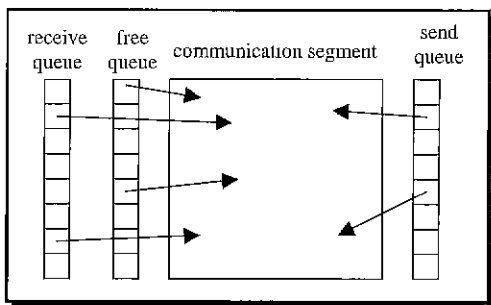


그림 4 U-Net endpoint 구조

U-Net에서 메시지 패싱은 end-point를 사용하여 진행된다. End-point는 메시지 데이터를 저장하는 communication segment, communication segment에 저장된 메시지에 대한 descriptor를 저장하는 send, receive, free queue

로 구성된다. End-point 생성시 커널의 도움을 받아 필요한 메모리 영역을 사용자 영역으로 매핑하며, end-point가 생성된 후에는 커널의 개입 없이 메시지 전송이 이루어진다. 메시지 전송시, communication segment에 메시지를 저장하고 send queue에 메시지에 대한 descriptor를 저장하면 네트워크 어댑터가 메시지를 읽어서 다른 노드로 전송을 한다. 메시지 수신시, 다른 노드에서 전송된 메시지는 communication segment에 저장되고 receive queue에 필요한 descriptor가 저장된다. 사용자 프로그램은 receive queue를 polling하고 있다가 메시지를 읽는다.

U-Net은 Berkeley 대학의 NOW팀에서 ATM, Myrinet, DEC의 21140 Tulip chip을 사용한 Fast Ethernet에 구현하였다. 성능은 U-Net을 Fast Ethernet상에 구현했을 경우 메시지 크기가 1KByte 이상일 때 최대 대역폭이 96Mbps이고, 120Mbps 링크를 사용한 ATM상에 구현했을 경우 메시지 크기가 1.5KByte 이상일 때 최대 대역폭이 118Mbps이다[12].

3.1.3 VIA(Virtual Interface Architecture)

User-level 통신 라이브러리의 우수한 성능이 입증되면서 Compaq, Intel, Microsoft는 VIA [13]라는 표준화된 user-level 통신 라이브러리를 개발하였다. VIA는 U-Net을 기반으로 개발되었으며 구조가 U-Net과 상당히 유사하다. VIA와 U-Net의 주된 차이점은 U-Net이 기존의 네트워크 어댑터위에 소프트웨어 형태로 구현된 반면, VIA는 spec을 하드웨어로 지원할 수 있는 네트워크 어댑터 형태의 제작을 권장하고 있다는 점이다.

VIA는 VI Provider, VI Consumer, Completion Queue, VI로 구성된다. VI Consumer는 사용자 프로그램과 OS vendor API로 구성되며, VI Provider는 네트워크 카드와 VI kernel agent로 구성된다. VI는 send & receive queue로 구성되며 Completion Queue는 종료된 작업에 관한 descriptor를 저장한다. 메시지 패싱 초기화시에 VI kernel agent를 사용하여 연결을 설정하며, 연결 설정후에는 커널 개입없이 통신이 진행된다. 전송되는 메시지는 send & receive queue에 저장되며, doorbell 메커니즘을

사용하여 네트워크 어댑터에 send & receive queue에 메시지가 있음을 알려준다.

VIA를 하드웨어로 구현한 것으로 GigaNet [14]사의 cLAN이 있으며 1.25Gbps의 대역폭을 가지고 PCI에 장착된다. 그리고 NOW 프로젝트에서는 VIA를 Myrinet상에 소프트웨어로 구현하였다[15,16]. Myrinet상에 구현된 VIA는 메시지 크기가 4KByte 이상일 때 최대 대역폭이 54MByte/s이다.

3.1.4 AM(Active Message)

AM[17]은 Berkeley 대학의 NOW팀에서 개발한 메시지 패싱 라이브러리이다. AM은 통신과 계산(computation)이 동시에 수행되도록 하여 통신에 소요되는 부하를 최소화시키고 있다.

AM은 메시지 송신시, 메시지 핸들러(message handler)라 불리는 user-level subroutine에 대한 포인터를 메시지의 헤더에 포함시켜 보낸다. 기존의 메시지 패싱 시스템은 메시지 수신측에서 메시지를 사용할 프로세스가 기존의 작업을 중지하고 도착한 메시지를 처리한다. 그러나, AM은 수신측에서 헤더 정보에 따라 메시지 핸들러를 새로운 thread로 실행시킴으로써 통신과 기존의 프로세스가 동시에 진행되도록 한다.

3.2 DSM(Distributed Shared Memory)

메시지 패싱은 원격 노드 메모리 접근시 노드 위치를 직접 지정해야 하는 단점을 가지고 있다. 이에 반해 DSM은 각 노드에 존재하는 메모리를 하나의 address space로 매핑하여 관리함으로써 변수가 위치하는 노드와 상관없이 변수 이름만으로 접근이 가능하여 프로그램이 용이하다는 장점이 있다. 따라서 많은 클러스터 시스템이 메시지 패싱 라이브러리와 함께 하드웨어 혹은 소프트웨어로 구현된 DSM을 지원한다.

하드웨어 DSM은 2.1장에서 설명했듯이 SCI 카드에서 지원되며, SCI를 제외한 네트워크는 소프트웨어로 DSM을 구현해야 한다. 소프트웨어 DSM은 공유메모리 지원 함수를 기존의 OS위에 라이브러리 형태로 제공한다. 소프트웨어 DSM은 각 노드를 하나의 address space에 매핑하여 관리하며 동기화를 포함한 공유메모리에 대한 접근을 처리한다. 또한, 원격 메모리의 접근 속도

향상을 위해 캐쉬기능을 지원하기도 한다.

소프트웨어 DSM으로 Rice 대학의 TreadMark[21]과 Munchen 대학의 SCI-VM[23] 등이 있다. TreadMark은 OS의 수정없이 Unix위에 run time 라이브러리 형태로 구현되어 있으며 캐쉬 기능을 지원한다. 현재 IBM, DEC, SUN, HP, SGI 등의 Unix 머신에서 수행 가능하다.

4. SSI(Single System Image)

여러 대의 PC 혹은 workstation으로 연결된 클러스터 시스템을 사용자에게 하나의 시스템인 것처럼 보이게 하는 것은 매우 중요하며 이것을 SSI라 부른다. SSI 지원 소프트웨어는 필요한 기능을 기존의 OS위에 미들웨어(middleware)형태로 구현한다. SSI 지원을 위해서는 각 노드에 분산되어 있는 자원들을 효율적으로 관리하여 transparent하게 접근 가능해야 하며 사용자에게 시스템에 대한 일관된 인터페이스를 제공해야 한다. SSI 지원을 위해 필요한 기능은 다음과 같다[18].

- Single entry point : 클러스터 시스템에 접속하기 위한 1개의 entry point를 제공함
- Single file hierarchy : 각 노드에 존재하는 파일 시스템을 하나의 파일 시스템으로 관리함(예 : NFS, AFS)
- Single control point : 한 장소에서 모든 노드를 관리함
- Single virtual networking : 각 노드에 존재하는 모든 네트워크 자원을 사용함
- Single memory space : 전체 메모리 주소를 하나의 메모리 공간으로 관리함(예 : DSM)
- Single job management system : 일관된 global job scheduler를 가짐
- Single user interface : 일관된 사용자 interface를 제공함(예 : X-Window)

현재 SSI를 지원하는 소프트웨어로 GLUnix[5], LSF[19], CODINE[20], TREAD-MARK[21] 등이 있으며, SSI 구현에 필요한 기능들이 각 소프트웨어에서 구현되었는지의 여부를 표 1에 나타내고 있다.

표 1 SSI를 지원하는 4가지 미들웨어

	GLUnix	Tread Mark	CODINE	LSF
Single entry point	Y	N	N	N
Single file hierarchy	Y	Y	Y	Y
Single control point	Y	N	Y	Y
Single virtual networking	N	N	N	N
Single memory space	N	Y	N	N
Single job management	Y	N	Y	Y
Single user interface	cmd-line	cmd-line	GUI/cmd-line	GUI

5. 결 론

클러스터 시스템은 저가로 고성능 서버의 성능을 발휘할 수 있는 시스템 제작이 가능하다는 장점 때문에 많이 사용되고 있다. 클러스터 시스템이 단일 컴퓨팅 시스템으로서의 성공여부는 네트워크의 성능과 지원 소프트웨어에 달려 있다. 네트워크의 성능은 SCI, Myrinet, Gigabit Ethernet 등과 같은 gigabit급 네트워크의 등장으로 많이 향상되고 있다. 지원 소프트웨어로는 메시지 패싱 시스템의 과도한 부하를 줄일 수 있는 U-Net과 같은 user-level 통신 라이브러리와 VIA가 등장하였다. 그리고 메시지 패싱 방식의 단점을 해결하여 프로그램이 용이한 DSM 방식이 지원되고 있으며 성능 향상을 위해 캐쉬기능을 포함한 소프트웨어 DSM도 개발되었다. 또한 클러스터 시스템 관리를 위해 SSI를 지원하는 소프트웨어도 개발되고 있다. 클러스터 시스템은 많은 장점 때문에 여러 분야에서 활용 및 연구되고 있으며 점차적으로 기존의 고성능 서버를 대체할 것으로 예상된다.

참고문헌

[1] IEEE Standard for Scalable Coherent Interface(SCI), IEEE Computer Society,

August 1993.

- [2] <http://www.dolphinics.com>
 [3] <http://www.uni-paderborn.de/pc2>
 [4] <http://www.myricom.com>
 [5] <http://now.cs.berkeley.edu>
 [6] <http://www.cs.princeton.edu/shrimp>
 [7] <http://www.beowulf.org>
 [8] <http://cnls.lanl.gov/avalon/>
 [9] <http://www.epm.ornl.gov/pvm>
 [10] <http://www.mpi-forum.org>
 [11] Thorsten von Eicken, Anindya Basu, Vineet Buch and Werner Vogels. "U-Net: A User-Level Network Interface for Parallel and Distributed Computing", Proceeding of the 15th ACM Symposium on Operating Systems Principles, 1995.
 [12] Matt Welsh, Anindya Basu and Thorsten von Eicken, "ATM and Fast Ethernet Network Interfaces for User-level Communication". Proceedings of the Third International Symposium on High Performance Computer Architecture (HPCA), 1997.
 [13] <http://www.viarch.org>
 [14] <http://www.giganet.com>
 [15] Philip Buonadonna, Andrew Geweke and David Culler, "An Implementation and Analysis of the Virtual Interface Architecture", Proceedings of SC98, Orlando, 1998.
 [16] Xin Liu. "Performance Evaluation of a Hardware Implementation of VIA", Technical Report, 1999.
 [17] Alan Mainwaring and David Culler, "Active Message Applications Programming Interface and Communication Subsystem Organization", Technical Document, 1995.
 [18] Kai Hwang, Hai Jin, Edward Chow, Cho-Li Wang and Zhiwei Xu, "Designing SSI Clusters with Hierarchical Checkpointing and Single I/O Space", IEEE Concurrency, Vol 7, No 1, pp60-69, 1999.

[19] <http://hpmag.cern.ch/computing/dict/l/lsf/index.html>

[20] <http://www.genias.de/products/codine/Welcome.html>

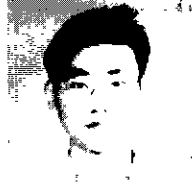
[21] <http://www.cs.rice.edu/~willy/TreadMarks/overview.html>

[22] <http://cmp.ameslab.gov/cmp/clusters/apc.html>

[23] Martin Schulz, "SCI-VM: A Flexible Base for Transparent Shared Memory Programming Models on Clusters of PCs", Fourth International Workshop on High-Level Parallel Programming Models and Supportive Environments, pp19-33, 1999.

[24] <http://www.cs.ucsb.edu/research/sci/index.html>

오 수 철



1995 부산대학교 컴퓨터공학과 학사
 1997 부산대학교 컴퓨터공학과 석사
 1997~1998 LG전자 멀티미디어 연구소 연구원
 1998~현재 부산대학교 컴퓨터공학과 박사과정
 관심분야 클러스터 시스템, 병렬처리, SCI, CC-NUMA
 E-mail:osc@hwowon.pusan.ac.kr

정 상 화



1985 서울대학교 전기공학과 학사
 1988 Iowa State University 전기 및 컴퓨터공학과 석사
 1993 University of Southern California 전기 및 컴퓨터공학과 박사
 1993~1994 University of Central Florida 전기 및 컴퓨터공학과 조교수
 1994~현재 부산대학교 컴퓨터공학과 조교수
 관심분야 클러스터 시스템, 병렬처리, 정보검색
 E-mail:shchung@hyowon.pusan.ac.kr

• 제27회 임시총회 및 춘계학술발표회 •

- 일 자 : 2000년 4월 28일(금) ~ 29일(토)
- 장 소 : 대구효성가톨릭대학교
- 논문모집 및 발표일정
 - 1) 접수기간 : 2000년 2월 10일(수) ~ 3월 4일(토)
 - 2) 심사결과 통보 : 2000년 3월 18일(토)
 - 3) 수정논문 접수마감 : 2000년 4월 1일(토)
 - 4) 논문발표 : 2000년 4월 28일(금), 4월 29일(토)
- 문 의 처 : 한국정보과학회 사무국

Tel. 02-588-9246/7, 4001/2

<http://www.kiss.or.kr>, E-mail:kiss@kiss.or.kr