SCI 네트워크 상의 소프트웨어 VIA 기반 PC클러스터 시스템

(A Software VIA based PC Cluster System on SCI Network)

신 정 희 † 정 상 화 †† 박 세 진 †††

(Jeonghee Shin) (Sang-Hwa Chung) (Sejin Park)

요 약 PC 클러스터 시스템에서 노드 사이의 데이타 교환을 위해 사용되는 TCP/IP 기반 통신 방식은 소프트웨어 부하가 크기 때문에 전체 시스템의 성능을 저하시키는 요인이 된다. 이러한 문제점을 해결하기 위해 사용자 수준 통신(user-level communication) 구조가 제안되었다. 사용자 수준 통신은 성능에 치명적인 영향을 미치는 커널을 통신 단계에서 제거함으로써 적은 지연 시간과 높은 대역폭을 제공하며, 이러한 우수한 성능은 업계 표준인 VIA(Virtual Interface Architecture)를 만들었다. 본 논문에서는 공유메모리 기반 Interconnect의 IEEE 표준인 SCI(Scalable Coherent Interface) 네트워크에 기반하여 VIA 클러스터 시스템을 구현하였다. 본 논문의 클러스터 시스템은 메시지 패싱 및 공유메모리 프로그래밍 환경을 동시에 제공하며, 최대 84MB/s의 대역폭과 8㎲의 지연 시간을 가진다. 또한, 본 시스템이 병렬 벤치마크 프로그램의 수행시 비교 대상 시스템들에 비해 성능이 우수함을 입증하였다.

키워드 : VIA, SCI, PC 클러스터, 병렬처리

Abstract The performance of a PC cluster system is limited by the use of traditional communication protocols, such as TCP/IP because these protocols are accompanied with significant software overheads. To overcome the problem, systems based on user-level interface for message passing without intervention of kernel have been developed. The VIA(Virtual Interface Architecture) is one of the representative user-level interfaces which provide low latency and high bandwidth. In this paper, a VIA system is implemented on an SCI(Scalable Coherent Interface) network based PC cluster. The system provides both message-passing and shared-memory programming environments and shows the maximum bandwidth of 84MB/s and the latency of 8㎲. The system also shows better performance in comparison with other comparable computer systems in carrying out parallel benchmark programs.

Key words : VIA, SCI, PC cluster, parallel processing

1. 서 론

최근 인터넷의 급속한 보급으로 웹 서버, 전자상거래 서버, VOD 서버 등을 포함하는 다양한 응용 분야에서 고성능 서버에 대한 수요가 증가하고 있으나, 고성능 서버의 높은 가격으로 인해 구입과 활용에 어려움이 따른다. 이러한 문제를 해결하기 위한 방안으로서, 고속 마이크로프로세서를 장착한 저가의 PC들을 고속 네트워크로 연결하여 하나의 컴퓨팅 시스템으로 사용하려는 클러스터링 기술이 등장하였다. 이러한 PC 기반 클러스터가 중대형 컴퓨터에 필적하는 단일 컴퓨팅 시스템으로서의 성공 여부는 PC를 연결하는 네트워크 성능과 지원소프트웨어에 달려 있다. 클러스터 시스템에 사용되는 네트워크는 Fast Ethernet, Myrinet, SCI 등이 있으며, 이중에서도 SCI(Scalable Coherent Interface)[1]는 ANSI/IEEE standard로서 최대 1GB/s의 대역폭을 지원하며, point-to-point 및 switch topology를 사용하므로 확장성이 우수하고, 고속의 클러스터링 시스템 구축

· 본 논문은 한국과학재단 목적기초연구(2000-2-30300-002-3)지원으로 수행되었음

† 비 회 원 : university of southern california 컴퓨터공학과
jeonghes@usc.edu
†† 종신회원 : 부산대학교 컴퓨터공학과 교수
shchung@pusan.ac.kr
††† 비 회 원 : 부산대학교 컴퓨터공학과
sejnpark@pusan.ac.kr

논문접수 : 2001년 2월 5일
심사완료 : 2002년 1월 28일

을 가능하게 한다. 또한 SCI는 이미 IBM의 NUMAQ 2000[2], Data General의 Aviion[3] 등의 중대형 시스템에 채택되어 그 성능이 입증되었다.

한편, 통신망 자체가 Gigabit급의 등장으로 급격히 발전하더라도 통신에 사용되는 TCP/IP와 같은 소프트웨어가 많은 부하를 가지면 실제 사용자 프로그램에서 사용 가능한 성능은 통신망의 물리적 성능에 훨씬 미치지 못하게 된다. 이러한 현상의 주요한 원인은 데이타 전송시 발생하는 커널로의 문맥 전환(context switch), 빈번한 데이타 복사, 그리고 데이타 수신시 발생하는 인터럽트(interrupt)이다[4][5]. 이와 같은 원인 해결을 위한 연구가 최근 활발하게 진행되었으며, 그 결과로 사용자 수준 통신(user-level communication) 모델들이 제시되었다[6][7][8][9][10]. 사용자 수준 통신 모델은 통신을 커널이 아닌 사용자 수준에서 처리하게 하여 커널 내부에서 소요되는 시간을 제거하였으며, 통신 계층을 단순화시킴으로써 통신 중에 발생하는 데이타 복사 회수를 최소화시켰다. [그림 1]은 TCP/IP와 같은 기존의 프로토콜과 사용자 수준 통신 모델을 비교하여 나타낸다. TCP/IP는 통신을 위해 커널의 개입이 발생하며, 이로 인한 문맥 전환과 데이타 복사가 일어난다. 반면, 사용자 수준 통신은 통신을 위한 초기화는 커널의 도움을 받아 이루어지지만 이후 통신 과정에서는 커널의 개입 없이 사용자 수준에서 통신이 직접 진행된다.

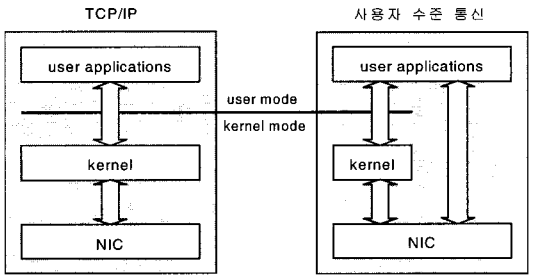

그림 1 TCP/IP와 사용자 수준 통신 비교

사용자 수준 통신 모델의 대표적인 연구로는 Cornell 대학의 U-Net[6], Princeton 대학의 SHRIMP[7], Berkeley NOW 시스템의 Active Message[8]와 Illinois 대학의 Fast Message[9]등이 있다. 이러한 사용자 수준 통신 모델의 우수한 성능이 입증되자 Compaq, Intel, Microsoft사는 U-Net을 기반으로 하여 사용자 수준 통신의 업계 표준인 VIA(Virtual Interface Architecture)[10]를 제안하였다. VIA는 통신하고자 하는 프로세스에게 통신망에 대한 가상의 인터페이스를 제공하여 통신망에 접근하도록 하며, 통신망 기기에 독립적인 API를 제공하여 다양한 플랫폼에서 사용 가능하다.

현재까지 Fast Ethernet이나 Myrinet 등의 네트워크 상에 VIA를 구현하여 우수한 성능을 얻은 많은 연구 사례가 발표되었다[11][12][13]. 본 논문에서는 Fast Ethernet이나 Myrinet보다 지연 시간 특성이 우수한 SCI네트워크 상에 VIA를 구현하였다. SCI는 NUMA (Non Uniform Memory Access) 특성인 RMA(Remote Memory Access)를 사용함으로써 시스템 호출 없이 다른 노드의 사용자 메모리 영역에 데이타 전송을 가능하게 하여, Fast Trap등을 발생시키는 다른 통신망 인터페이스에 비해 사용자 수준 통신인 VIA 구현시 최적의 환경을 제공한다. 본 논문에서 개발된 SCI 기반 VIA 시스템은 가장 최근 spec인 release 1.0에 준하여 VIA를 개발하였으며, 대표적인 소프트웨어 기반 VIA 시스템인 M-VIA[12]와 SCI 상에 표준 메시지 패싱 인터페이스를 제공하는 SCI-MPI[14]등과의 비교를 통하여 성능의 우수성을 입증하였다. 또한, SCI 기반 VIA 시스템은 기존의 SCI 공유메모리 시스템과 동시 사용이 가능하며, 커널의 재컴파일이나 디바이스 드라이버의 수정이 없어 시스템의 안정성을 보장한다.

SCI상의 VIA 구현은 이러한 시스템 기능 및 성능상의 장점 뿐 아니라 사용자 프로그래밍 편이성 측면에서도 유리하다. 클러스터 시스템에서 병렬 프로그램을 작성하는 방식에는 크게 나누어 메시지 패싱 방식과 공유메모리 방식이 있다. 메시지 패싱 방식은 프로그래머가 MPI[15]나 PVM[16]과 같은 병렬 라이브러리를 사용하여 프로그래밍하는 반면, 공유메모리 방식은 프로그래머가 단일 CPU를 보유한 시스템 상에서와 동일한 방법으로 프로그래밍한다. 본 논문에서 개발된 SCI 기반 VIA 시스템은 프로그래머가 원하는 방식을 사용하여 프로그래밍할 수 있도록 두 가지 방식의 동시 지원이 가능하다. 즉, SCI 공유메모리 구조 상의 공유메모리 방식 뿐 아니라 메시지 패싱 방식 환경에서 작업할 수 있도록 SCI 기반 VIA와 표준 병렬 라이브러리인 MPI를 제공한다.

본 논문의 구성은 다음과 같다. 2장에서 VIA의 연구 사례를 살펴보고, 3장에서는 SCI기반 VIA 시스템의 구조를 설명한다. 4장에서는 SCI기반 VIA 시스템의 성능을 분석하고, 마지막 5장에서는 결론 및 향후 연구 과제에 대하여 설명한다.

2. 관련 연구

현재까지 VIA의 소프트웨어 구현과 하드웨어 구현이 활발히 진행되고 있으며, 대표적인 VIA 관련 연구 및

구현은 다음과 같다.

하드웨어 VIA 구현의 대표적인 예로 GigaNet사의 cLAN[17]과 Finisar사의 Fibre Channel VI Host Bus Adapter[18]가 있다. 첫 번째 하드웨어 VIA 구현인 cLAN은 1.25Gbps의 성능을 내며, Fibre Channel VI Host Bus Adapter는 1.062Gbps의 최대 대역폭을 가진다. Gigabit Ethernet 기반 하드웨어 VIA인 Compaq/Tandem사의 ServerNet II[19]는 현재 베타 버전이 나와 있으며 180MB/s의 대역폭을 가진다. 또한, 독일의 Chemnitz 대학에서는 VIA와 SCI를 결합한 새로운 PCI-SCI bridge 하드웨어를 구현 중에 있으며[20], 아직 연구가 완료되지 않아 성능에 관해서는 보고된 바 없다.

이에 반해, 기존의 물리적 미디어를 그대로 사용하는 소프트웨어 VIA에 관한 연구도 진행되었다. 소프트웨어 VIA 구현의 대표적인 예로 Fujitsu사의 Synfinity CLUSTER[21] 그리고 NEC사의 V1000 네트워크 어댑터[22]가 있다. Synfinity CLUSTER는 최대 대역폭 108MB/s를 가지며, V1000은 1.25Gbps의 대역폭을 가진다. 또한, Intel 사는 VIA spec 검증 및 성능의 우수성을 증명하기 위해 Fast Ethernet과 Myrinet 상에 VIA를 소프트웨어로 구현하였으며, 각각 12MB/s와 87MB/s의 성능을 얻었다[11]. 이러한 VIA 개발은 업체 뿐 아니라 연구소에서도 진행되고 있는데 그 예로 Lawrence Berkeley Lab은 1998년 M-VIA[12]를 개발하였다. M-VIA의 특징은 소프트웨어 VIA의 모듈화로써 Fast Ethernet 및 Gigabit Ethernet 상에 구현되었으며, Myrinet을 지원할 예정이다. Fast Ethernet의 경우 11.9MB/s, Gigabit Ethernet의 경우 60MB/s의 대역폭을 가진다. Berkeley VIA project[13]는 VIA의 문제점을 분석하고 개선된 VIA를 제안하는데 그 목적을 두고 Myrinet 상에 구현되었으며, 최대 대역폭 68MB/s를 가진다.

이와 같이, 다양한 고성능 네트워크 상에서 VIA 연구가 활발히 진행되었으나 우수한 지연 시간과 대역폭을 가지는 SCI 상의 소프트웨어 VIA 연구는 아직 보고된 바 없다. 이와 유사한 SCI-MPI[14]가 있으나, 이는 SMI 라이브러리[23]를 기반으로 개발된 것으로 업계 표준인 VIA를 따른 것은 아니다.

3. SCI 네트워크 상의 VIA 기반 PC 클러스터 시스템

[그림 2]는 본 논문에서 개발된 SCI 네트워크 상의 VIA 기반 PC 클러스터 시스템(VIA/SCI)의 구조를 나타낸다. 이를 위하여 Dolphin사의 PCI-SCI Adapter Card[24]를 각 PC 노드의 PCI 슬롯에 장착하여 SCI 기반 상호 연결망을 구축하였다.

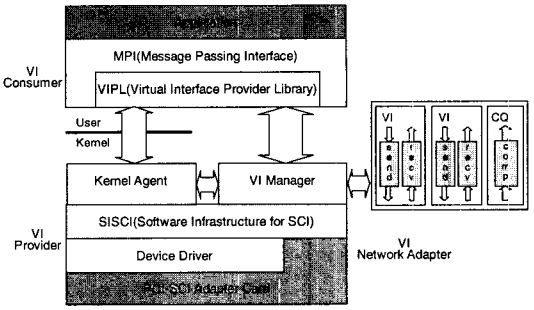

그림 2 SCI 네트워크상의 소프트웨어 VIA 기반 시스템 구조

3.1. VIA/SCI 구현

VIA[10]는 [그림 2]에서 보는 바와 같이 Virtual Interface(VI), Completion Queue(CQ), VI Provider 그리고 VI Consumer로 구성된다. VI manager에 의해 관리되는 VI는 Send Queue와 Receive Queue의 쌍으로 이루어진 Work Queue로 구성되며, VI Provider는 Kernel Agent와 VI 네트워크 어댑터로 구성된다. Kernel Agent는 초기화 및 자원 관리를 수행하며, VI 네트워크 어댑터는 데이타의 전송을 담당한다. VI Consumer는 VIPL(Virtual Interface Provider Library), MPI등의 Communication Interface와 사용자 프로그램으로 이루어진다. VI Consumer가 처음 Kernel Agent에 시스템 호출을 통해 VI를 생성하면, 이후의 Send 및 Receive 요청은 시스템 호출 없이 바로 VI를 통해서 이루어진다. 이러한 VI Consumer의 요청이 완료되었음은 해당 Work Queue와 짝을 이룬 CQ에 기록된다.

본 논문에서 구현된 VIA/SCI는 이러한 VIA 모듈을 Dolphin 사의 PCI-SCI Adapter Card를 위한 SISCI API[25]를 사용하여 구현하였다. Dolphin사의 PCI-SCI Adapter Card는 지역(local)노드의 프로세스가 자신의 메모리 주소 영역을 원격(remote)노드의 메모리 주소 영역에 매핑함으로써 원격노드의 사용자 메모리 영역에 직접 접근 가능하게 한다. 따라서, 이러한 방식의 시스템에서는 메시지 패싱 대신 Remote Read 및 Remote Write를 통해 노드간 통신이 이루어지게 되는데, [그림 3]은 노드 B가 노드 A에 Remote Write를 수행하는 모습을 보여준다. 노드 A가 자신의 메모리 영역에 메모

리를 할당하고(①), 노드 A와 노드 B 사이의 매핑이 이루어지면(②) 두 노드 사이의 데이타 전송이 가능해진다(③). 이 때, 매핑된 노드 B의 메모리 주소 영역은 Remote Segment, 노드 A의 메모리 주소 영역은 Local Segment라 부르기로 한다.

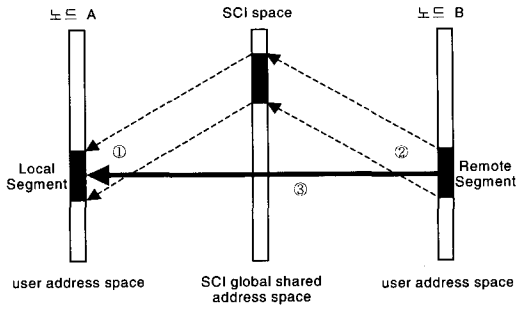

그림 3 PCI-SCI mapping

다음에서는 VIA/SCI 구현의 주요 메커니즘인 초기화와 데이타 전송에 대해 설명한다.

3.1.1 Kernel Agent에 의한 초기화 매커니즘

Kernel Agent는 초기화 메커니즘에서 VI manager에게 VI의 생성을 요청하고, 통신을 원하는 노드간의 VI connection을 설정한다. [그림 4]는 노드 A와 노드 B 사이의 VI connection을 위한 flow chart를 보여준다.

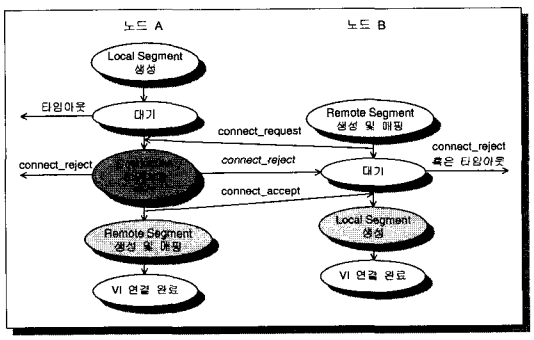

그림 4 VIA/SCI의 VI connection시 flow chart

노드 A는 SCI 공유 메모리 영역에 Local Segment를 생성한 후 노드 B로부터의 connect_request를 기다린다. 노드 A와 통신을 원하는 노드 B는 SCI 공유 메모리 영역에 Remote Segment를 생성해서 노드 A의 Local Segment에 매핑한 후 자신의 주소 및 connection 설정 정보를 가진 connect_request를 노드 A에 보낸다. 즉, 노드 B는 노드A를 위해서 매핑된 특정 영역에 자신의 정보를 write한다. 노드 A는 노드 B로부터 받은 connection attribute를 조사하여 부합 여부에 따라connection의 reject 혹은 accept 여부를 결정하고, 노드 B에게 결과를 알려준다. Connection이 accept되었을 경우, 노드 B의 Local Segement 생성과 노드 A의 Remote Segment 생성 및 매핑이 필요하게 되는데, 이는 송신과 수신시 각각 다른 Segment가 사용되기 때문이다. 즉, Local Segment는 데이타 수신을 위해서, Remote Segment는 데이타 송신을 위해서 사용된다. Dolphin사의 SISCI API에서 제공하는 Remote Write의 지연 시간은 Remote Read의 지연 시간에 비해 약 10배정도 적게 걸린다. 따라서, 본 논문에서는 위와 같이 두개의 영역을 따로 관리함으로써 지연 시간이 적은 Remote Write만을 사용하여 데이타 전송이 가능하게 하였다. 이러한 과정을 통해 두 노드 사이의 VI connection이 형성되면 데이타 전송이 가능해진다.

3.1.2 데이타 전송 메커니즘

3.1.1절의 초기화 메커니즘을 통해 VI connection이 형성된 두 노드 사이의 데이타 전송은 다음과 같이 이루어진다.

우선, 송신측에서는 보낼 데이타의 Descriptor를, 수신측에서는 받을 데이타의 Descriptor를 만들어 VI manager를 통해Work Queue에 넣게 되고, VI 네트워크 어댑터는 Send Queue로부터 Descriptor를 꺼내 보낼 데이타를 통신망에 전송한다. 데이타 전송을 위해서는 [그림 5]와 같은 handshake가 필요하며, 실제 데이타 전송은 SISCI API를 이용한 Remote Write로 개발된다. 즉, 송신할 데이타가 SCI 공유 메모리 상에 매핑된 Remote Segment에 write되고, 이는 초기화 메커니즘을 통해 형성된 매핑에 따라 수신측의 Local Segment로 보내지게 된다. 따라서 송신을 원하는 데이타를 매핑된 Remote Segment에 write하기만 하면 수신측은 Local Segment에서 바로 읽을 수 있다. 이러한 VIA/SCI의 데이타 전송 메커니즘은 다른 통신망에서 데이타 전송시 발생하는 Fast Trap등의 시스템 호출이 일어나지 않으므로 사용자 수준 통신인 VIA의 특성에 적합하며, 시스템 호출로 인한 부하를 제거한다.

통신망을 통해 수신된 데이타는 수신자의 해당 VI의 Receive Queue에 있는 Descriptor가 지정한 사용자 버퍼에 저장된다. 이러한 일련의 전송이 완료되면 네트워크 어댑터는 CQ에 전송이 완료되었음을 표시한다. 이러한 데이타 전송의 과정에서 불필요한 데이타 복사는 발생하지 않으며, 인터럽트의 발생을 방지하기 위해 폴링 방식을 사용하였다.

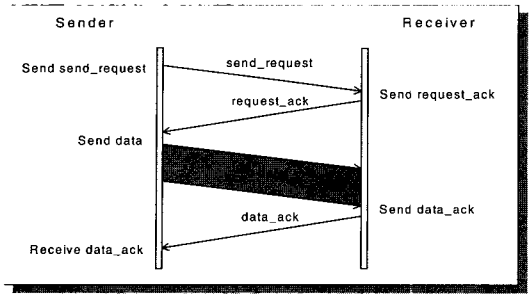

그림 5 데이타 전송시 handshake

이와 같이 본 논문에서는 VIA/SCI의 구현시 사용자 수준 통신의 특성에 충실하기 위해 소프트웨어 부하의 주요한 원인들을 제거하는데 중점을 두었다. 즉, SCI의 NUMA 특성인 RMA를 사용하여 시스템 호출 없이 다른 노드의 사용자 메모리 영역에 데이타를 전송함으로써 커널로의 문맥 전환을 제거하였다. 또한, 데이타 전송시 사용자 버퍼에서 통신망으로 데이타를 바로 전송함으로써 불필요한 데이타 복사를 완전히 제거하였다. 마지막으로 데이타 수신시, 커널로의 문맥 전환의 원인인 인터럽트 발생을 방지하기 위하여 폴링 방식을 사용하였다.

3.2. VIA/SCI 상에 MPI 모듈 구현

본 논문에서는 메시지 교환에 의한 노드간의 동기화, 응용 프로그램 작성의 용이성 및 시스템의 표준화를 위해 표준 메시지 패싱 라이브러리인 MPI2 Standard[15] 함수를 개발하였다. [그림 6]은 VIA/SCI 시스템을 위한 그림을 나타내며, MPI2 표준 중에서 실제로 VIPL을 이용하여 SCI 상에 개발된 함수는 초기화 관련 함수, 동기화 관련함수, 송수신 관련함수, Reduction 관련함수, 그리고 종료함수로 구분할 수 있고 다음과 같이 정리할 수 있다.

- 초기화 함수
  - MPI_Init
  - MPI_Comm_rank : 프로세서의 rank를 return하는 함수
  - MPI_Comm_size : 총 실행되는 프로세서의 수를 알려주는 함수
- 동기화 함수
  - MPI_Barrier
- 송수신 관련 함수
  - MPI_Send : 송신 함수
  - MPI_Recv : 수신함수
- 프로그래머의 편의를 위한 Reduction 관련 함수
  - MPI_Reduce : 연산자에 따라 모든 프로세서가 가지는 송신 버퍼 값들의 연산 결과를 루트에서 볼 수 있는 함수
  - MPI_Allreduce : 모든 프로세스들의 값들에 대해 요구한 연산을 수행한 뒤 수신 받기를 원하는 버퍼에 돌려주는 함수
- 종료함수
  - MPI_Finalize

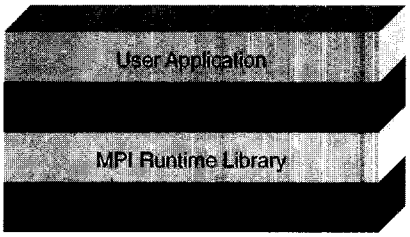

그림 6 VIA/SCI 시스템을 위한 MPI 구조

4. 실험

4.1 실험 환경

본 논문에서는 성능 측정 실험을 위하여 8개 노드의 PC 클러스터 시스템을 사용하였다. 각 노드는 단일 350MHz Intel Pentium II 프로세서, 128MB 주 메모리, 4.3GB 용량의 SCSI 하드 디스크를 탑재하고 있으며, 운영체제는 Linux를 사용한다. 이러한 8개의 노드는 [그림 7]과 [그림 8]에서 보는 바와 같이 두 가지 형태로 연결되어 있다. [그림 7]은 각 노드에 Dophin사의 PCI-SCI Adapter Card를 장착한 후, 각각 2개의 노드를 연결한 4개의 링이 4×4 SCI Switch에 연결된 PC 클러스터 시스템의 구조를 보여주며, [그림 8]은 각 노드에 DC21140 Fast Ethernet Card를 장착하고, 3COM SuperStack II Switch 3300[26]으로 연결된 PC 클러스터 시스템의 구조를 보여준다.

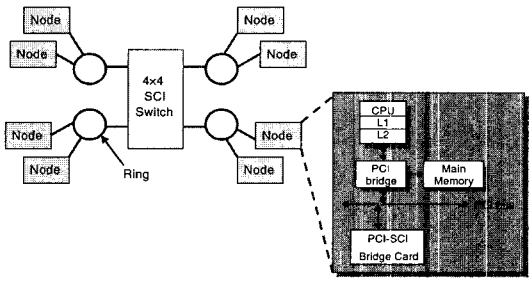

그림 7 SCI 기반 8-노드 PC 클러스터 시스템의 구조

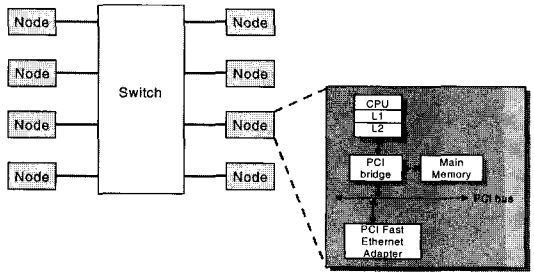

그림 8 Fast Ethernet 기반 8-노드 PC 클러스터 시스템의 구조

4.2 실험 결과 및 분석

본 논문에서 구현된 SCI 기반 VIA 시스템은 다른 시스템들과의 비교를 통해 성능을 평가하였으며, Fast Ethernet 상의 M-VIA, TCP/IP, 그리고 SCI-MPI를 비교 대상으로 삼았다. Lawrence Berkeley Lab에서 개발된 M-VIA[12]는 대표적인 소프트웨어 VIA로써 Linux 상에 VIA를 모듈로 구현하여 구축이 용이하고 다양한 네트워크 디바이스를 지원하며 우수한 성능을 가지는 것으로 알려져 있다. 또한, Aachen 대학의 SCI-MPI[14]는SMI 라이브러리[23]를 이용하여 개발되었으며, 공유 메모리를 제공하는 SCI 기반 클러스터 시스템에서 MPI를 제공함으로써 SCI 사용을 다양화하는데 기여하였다.

본 실험의 성능 비교는 대역폭 및 지연 시간 측정과 병렬 벤치마크 프로그램의 수행 시간 측정을 통하여 이루어졌다.

4.2.1 지연 시간 및 대역폭을 통한 성능 측정

[그림 9]는 네 가지 통신망 인터페이스에서 메시지 크기 증가에 따른 지연 시간의 변화를 보여준다. VIA/SCI는 최소 8μs의 지연 시간을 가지는데, 이는 동일 SCI 네트워크 상에 구현된 SCI-MPI의 지연 시간인 12μs와 비교해 볼 때 4μs가 단축되었으며, 이러한 지연 시간의 차이는 메시지 크기가 증가함에 따라 더욱 커짐을 알 수 있다. M-VIA(VIA/FE)는 25μs의 지연 시간을 가지며, TCP/IP 프로토콜을 사용했을 경우 357μs의 지연 시간을 나타낸다. 이와 같이 사용자 수준 통신은 TCP/IP 프로토콜의 과도한 소프트웨어 부하를 제거한 효과를 보여주는데, 특히 소프트웨어 부하가 치명적인 영향을 미치는 적은 크기 메시지의 경우 효과가 두드러진다.

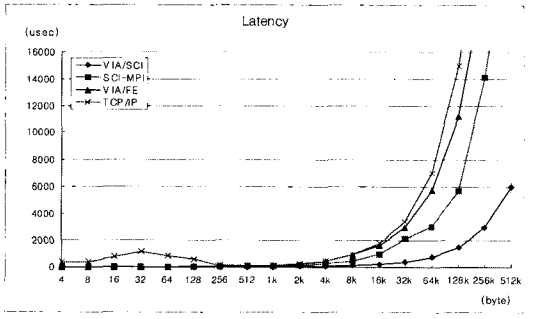

그림 9 지연 시간 비교

[그림 10]은 메시지 크기 증가에 따른 대역폭의 변화를 보여준다. VIA/SCI의 경우 최대 84MB/s의 대역폭을 가지며, SISCI를 사용한 NUMA 방식의 최대 대역폭에 가깝게 나타남을 알 수 있다. 반면, SCI-MPI는 최대 대역폭인 18MB/s를 나타내는데, 이는 SCI-MPI 구현시 사용된 C 함수인 *MEMCPY*가 Linux상에서 나쁜 성능을 가지기 때문이다. M-VIA의 경우도 11.4MB/s의 대역폭을 가지며 Fast Ethernet의 물리적 성능에 가까운 값을 나타낸다.

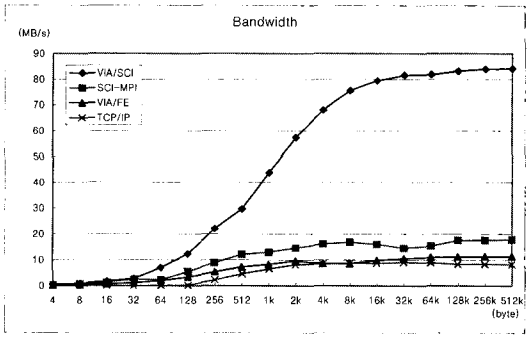

그림 10 대역폭 비교

VIA/SCI의 지연 시간과 대역폭은 Berkeley 대학에서 구현한 Myrinet 상의 VIA 시스템[13]의 지연 시간인 30μs 보다 약 4배 정도 적으며, 최대 대역폭인 68MB/s 보다 16MB/s 더 크다. 또한, 이러한 VIA/SCI의 성능은 Intel 사에서 구현한 Myrinet 상의 VIA 시스템[11]의 지연 시간인 53μs보다 7배 정도 적으며, 최대 대역폭인 87MB/s와 유사한 성능을 보여줄 뿐 아니라 대표적인 하드웨어 VIA인 GigaNet사의 cLAN[17]이 응용프로그램 수준에서 측정된 지연 시간인 7μs와 대역폭인 130MB/s에도 크게 뒤지지 않는다. 추후 Dolphin사의 PCI-64 PCI-SCI Adapter Card(지연 시간 1.46μs, 대역폭 304MB/s)를 사용하여 본 실험 환경을 개선한다면 더 나은 성능을 제공하게 될 것이다.

4.2.2 병렬 벤치마크 프로그램의 수행 시간을 통한 성능 측정

현재까지 발표된 VIA 시스템의 성능은 단순한 지연 시간과 대역폭이 대부분으로 응용 프로그램 상에서의 성능이라 보기에는 무리가 있었다. 따라서, 본 논문은 NASA에서 개발된 병렬 벤치마크 프로그램인 NAS (Numerical Aerodynamics Simulation) Parallel Benchmark(NPB) 버전 2.3[27]을 VIA/SCI 시스템 및 각종 비교 대상 시스템 상에 구현하여 응용 프로그램 수준에서의 성능을 측정하였으며, 수행 노드수가 1인 경우 NPB의 serial 버전을 사용하였다. 본 논문에서 사용된 벤치마크와 해당 problem size는 [표 1]에 나타나있다.

표 1 NPB problem size

Benchmark Program	Problem Size (byte)	Iteration 수
Embarrassingly Parallel (EP)	33554432	0
Conjugate Gradient (CG)	14000	15
Multigrid (MG)	64x64x64	40
LU solver (LU)	33x33x33	300

[그림 11]부터 [그림 14]까지는 노드수 증가에 따른 각 벤치마크 프로그램의 수행 시간과 통신 시간을 보여준다.

LU solver(LU)는 수행 중 노드간 40-byte 이하의 크기가 적은 메시지의 통신이 발생하는 프로그램으로 speedup이 잘 나타나며, VIA/SCI와 TCP/IP의 경우 수행 시간 차는 8초 이상이다. 그러나 VIA/SCI와 SCI-MPI의 경우 크기가 적은 메시지에서 지연 시간의 차가 10μs 정도이므로 전체 수행 시간의 차가 적다. 또한, VIA/SCI와 VIA/FE의 수행 시간 차는 4초 정도이다.

Maltigrid(MG)와 Conjugate Gradient(CG)의 경우 수행 중 노드간 전송되는 데이타의 크기가 커서 통신 인터페이스 종류에 따른 성능 차이가 크다. 특히, Fast Ethernet을 사용한 경우 통신 시간으로 인한 병렬화 오버헤드가 커서 speedup을 나타내지 못한다. 그러나, SCI 네트워크 사용시에는 뚜렷한 speedup을 보여주고 있으며, VIA/SCI의 경우 SCI-MPI보다 나은 성능을 나타낸다.

Embarrassingly Parallel(EP)의 경우는 노드간의 낮은 통신 빈도수로 인해 병렬화에 따른 오버헤드가 적어 뚜렷한 speedup을 보여줄 뿐 아니라 통신망 인터페이스 종류에 관계 없이 유사한 성능을 나타낸다.

[그림 11]부터 [그림 14]까지 나타내는 바와 같이, 동일한 SCI 네트워크를 사용한 VIA/SCI와 SCI-MPI의 경우 통신 시간에 차이를 보여준다. 이와 같은 현상은 VIA/FE와 TCP/IP에서도 나타난다. 또한, VIA/SCI의 경우 평균 6.1의 speedup을 보여준다(단, speedup 측정

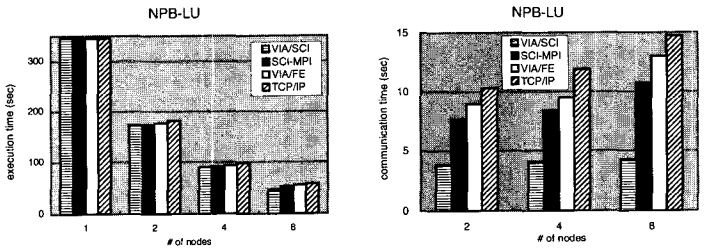

그림 11 NPB benchmark : LU

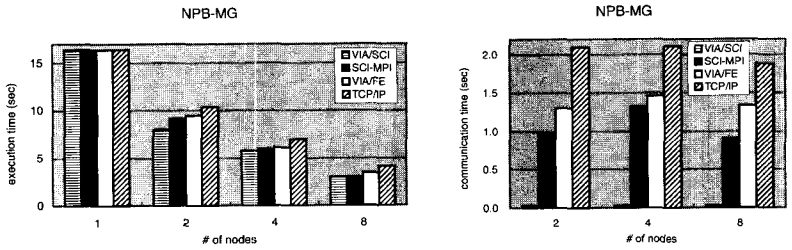

그림 12 NPB benchmark : MG

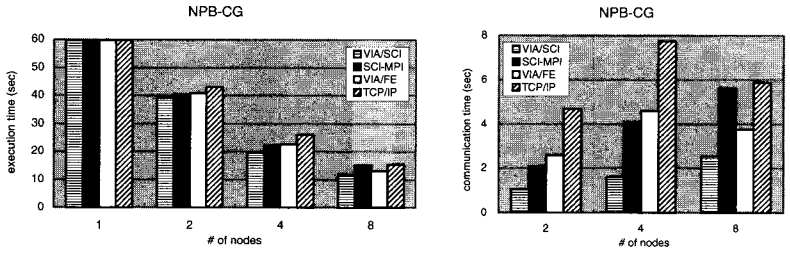

그림 13 NPB benchmark : CG

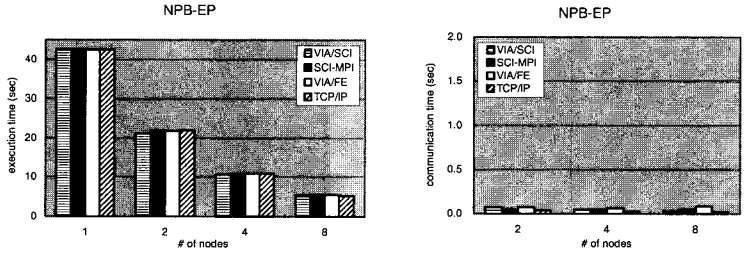

그림 14 NPB benchmark : EP

시 통신 시간이 거의 없는 EP는 제외하였다).

5. 결론 및 향후 과제

본 논문에서는 SCI 네트워크를 기반으로 한 PC 클러스터 시스템 상에 사용자 수준 통신의 업계 표준인 VIA를 개발하였다. VIA/SCI 시스템은 SCI의 NUMA 특성을 이용하여 통신 과정에서 커널의 개입을 완전히 제거함으로써 우수한 성능을 제공하고, 기존의 SCI 공유메모리 시스템과 동시 사용이 가능하며, 커널의 재컴파일이나 디바이스 드라이버의 수정이 없어 시스템의 안정성을 보장한다. 또한, 병렬 프로그래머에게는 공유메모리 방식을 제공함과 동시에 표준 병렬 라이브러리인 MPI를 제공함으로써 메시지 패싱 방식 환경을 제공한다. VIA/SCI는 최대 84MB/s의 대역폭과 8μs의 지연시간을 가지며 우수한 성능을 나타내고, 병렬 벤치마크 프로그램의 수행시에도 비교 대상의 타 시스템 보다 나은 성능을 보여 응용 프로그램 수준에서도 우수함을 입증하였다. 본 논문에서 개발된 VIA/SCI가 향후 시스템의 최적화를 거친다면 PC 클러스터 시스템 상에 우수한 통신 인터페이스를 제공하게 될 것이다.

참고 문헌

[1] IEEE Standard for Scalable Coherent Interface, IEEE Std 1596-1992, IEEE Computer Society, Aug. 1993.

[2] http://www.sequent.com

[3] http://www.dg.com/about/html/sci_interconnection_and_adapter.html

[4] J. Kay and J. Pasquale, "Profiling and Reducing Processing Overheads in TCP/IP", IEEE/ACM Transactions on Networking, Vol. 4, No. 6, pp. 817-828, Dec. 1996.

[5] R. A.F. Bhoedjang, T. Ruhl, and H. E. Bal, "User-Level Network Interface Protocols", IEEE Computer, Vol. 31, No. 11, pp. 53-60, Nov. 1998.

[6] T. von Eicken, A. Basu, V. Buch and W. Vogels, "U-Net:A User-Level Network Interface for Parallel and Distributed Computing", Proceeding of the 15th ACM Symposium on Operating Systems Principles, 1995.

[7] M. Blumrich, C. Dubnichi, E. W. Felten and K. Li, "Virtual Memory-Mapped Network Interfaces", IEEE Micro, pp. 21-28, Feb. 1995.

[8] A. Mainwaring and D. Culler, "Active Message Applications Programming Interface and Communication Subsystem Organization", Technical Document, 1995.

[9] S. Pakin, M. Lauria, and A. Chien, "High Performance Messaging on Workstations: Illinois Fast Messages(FM) for Myrinet", Proceeding of Supercomputing 95, San Diego, California.

[10] http://www.viarch.org

[11] F. Berry, E. Deleganes, and A. M. Merritt, "The Virtual Interface Architecture Proof-of-Concept Performance Results", Server System Technology, Intel Corporation, Feb. 1998.
[12] http://www.nersc.gov/research/FTG/via
[13] http://www.cs.berkeley.edu/~philipb/research.html, P. Buonadonna, A. Begel, D. Gay, and D. Culler, "An Analysis of VI Architecture Primitives in Support of Parallel and Distributed Communication", Apr. 2000.
[14] J. Worringen and T. Bemmerl, "MPICH for SCI-connected Clusters", Proceeding of SCI Europe '99, pp. 3-11, Sep. 1999.
[15] http://www-unix.mcs.anl.gov/mpi/
[16] "PVM: A Framework for Parallel Distributed Computing", V. S. Sunderam, Concurrency: Practice and Experience, 2, 4, pp. 315-339, Dec. 1990.
[17] http://www.giganet.com/products/indexlinux.htm
[18] http://www.finisar.com
[19] http://servernet.himalaya.compaq.com/
[20] M. Trams, and W. Rehm, "A new generic and reconfigurable PCI-SCI bridge", Preceedings of SCI-Europe'99, Sep. 1999.
[21] http://www.fjst.com/products/synfinitycluster
[22] http://necsvl.com/wpapers/oracle.html
[23] http://www.lfbs.rwth-aachen.de/users/joachim/SMI/
[24] http://www.dolphinics.no/pdf_filer/PCI_SCI_Overview.pdf
[25] http://www.dolphinics.no/customer/software/_linux/index.html
[26] http://www.3com.com/products/dsheets/800903.html
[27] http://www.nas.nasa.gov/Software/NPB/

신 정 희

1999년 부산대학교 컴퓨터공학과 학사. 2001년 부산대학교 컴퓨터공학과 석사. 2001년 ~ 현재 University of Southern California 컴퓨터공학과 박사과정. 관심분야는 클러스터 시스템, 병렬처리, VIA, SCI

정 상 화

1985년 서울대학교 전기공학과 학사. 1988년 Iowa State University 전기 및 컴퓨터공학과 석사. 1993년 University of Southern California 전기 및 컴퓨터공학과 박사. 1993년 ~ 1994년 University of Central Florida 전기 및 컴퓨터공학과 조교수. 1994년 ~ 현재 부산대학교 컴퓨터공학과 부교수 및 컴퓨터및정보통신연구소 연구원. 관심분야는 클러스터 시스템, 병렬처리, 정보검색, VOD, Infiniband

박 세 진

1998년 부산대학교 컴퓨터공학과 학사. 2000년 부산대학교 컴퓨터공학과 석사. 2000년~현재 부산대학교 컴퓨터공학과 박사과정. 관심분야 : 클러스터 시스템, 병렬처리, VIA, Infiniband