효율적인 정보 검색을 위한 VIA 기반 PC 클러스터 시스템

(VIA-Based PC Cluster System for Efficient Information Retrieval)

강 나 영 † 정 상 화 †† 장 한 국 †††
(Nayoung Kang) (Sang-Hwa Chung) (Hankook Jang)

요 약 PC클러스터 기반 정보 검색 시스템은 질의를 클러스터 상의 노드에 분산시켜 병렬로 처리함으로써 전체 시스템의 성능을 향상시킬 수 있다. 그러나, 노드 사이의 데이터 교환을 위하여 TCP/IP 기반 통신을 사용하는 것은 전체 시스템 성능 저하의 원인이 된다. 이를 해결하기 위해 개발된 것이 사용자 수준 통신(user-level communication)이다. 이것은 성능에 치명적인 영향을 미치는 커널 접근을 통신 단계에서 제거함으로써 적은 지연시간과 높은 대역폭을 제공한다. 본 논문에서는 사용자 수준 통신 방법의 업계 표준인 VIA(Virtual Interface Architecture)를 기반으로 한 효율적인 병렬 정보 검색 시스템을 제안한다. 본 논문의 정보 검색 시스템은 SCI(Scalable Coherent Interface) 기반의 VIA 방식, SCI 기반의 VIA/MPI 방식 그리고 Fast Ethernet 기반의 VIA/MPI 방식으로 구현되었으며 실험을 통하여 세 방식의 성능을 비교 분석하였다.

키워드 : 정보검색, VIA, SCI, MPI, PC 클러스터, 병렬처리

Abstract PC cluster-based Information Retrieval (IR) systems improve their performances by parallel processing of query terms using cluster nodes. However TCP/IP based communication used to exchange data between cluster nodes prevents the performance from being improved further. The user-level communication mechanisms solve the problem by eliminating the time-consuming kernel access in exchanging data between cluster nodes. The Virtual Interface Architecture (VIA) is one of the representative user-level communication mechanisms which provide low latency and high bandwidth. In this paper, we propose a VIA-based parallel IR system on a PC cluster. The IR system is implemented using the following three communication methods: Scalable Coherent Interface (SCI) based VIA, MPI on SCI based VIA, MPI on Fast Ethernet based VIA. Through experiments, the performances of the three methods are analyzed in various aspects.

Key words : IR, VIA, SCI, MPI, PC Cluster, Parallel Processing

1. 서 론

최근 인터넷의 보급이 일반화됨에 따라 인터넷 상에는 막대한 양의 정보가 넘쳐나고 있다. 그러나, 이렇게 방대한 양의 정보 중에서 사용자가 필요로 하는 정보를 신속 정확히 제공하기란 쉽지 않다. 따라서 고품질의 정보를 신속하게 제공하는 검색 시스템에 대한 요구가 더욱 커지고 있다. 이를 위해 많은 정보 검색 서비스 제공 업체들은 값비싼 중대형 서버 또는 슈퍼컴퓨터를 사용하여 서비스를 제공하고 있다. 그러나, 중대형 컴퓨터는 고가의 비용 때문에 구입 및 활용에 어려움이 있다. 이러한 문제를 해결하기 위해서 고속 마이크로프로세서를 장착한 저가의 PC들을 Fast Ethernet, Myrinet, SCI (Scalable Coherent Interface)[1] 등의 고속 네트워크로 연결하여 하나의 컴퓨팅 시스템으로 사용하려는 클러스터링 기술이 등장하였다.

클러스터링 기술을 활용하여 검색 시스템을 구성하면,

· 본 연구는 한국과학재단 목적기초연구(2000-2-30300-002-3) 지원으로 수행되었음.
† 비 회 원 : 삼성전자 무선사업부 연구원
kangny@pusan.ac.kr
†† 종신회원 : 부산대학교 컴퓨터공학과 교수
shchung@pusan.ac.kr
††† 비 회 원 : 부산대학교 컴퓨터공학과
hkjang@pusan.ac.kr
논문접수 : 2002년 2월 26일
심사완료 : 2002년 8월 19일

노드들이 질의를 나누어 처리함으로써 디스크 I/O를 비롯한 연산의 일부를 분산하여 처리할 수 있으므로 검색 시스템의 효율을 높일 수 있다. 이러한 PC클러스터를 사용하는 검색 시스템의 성능은 각각의 질의에 대해 부하 균등화가 얼마나 적절히 이루어졌는지 여부와 데이터를 주고받기 위한 클러스터간의 통신성능에 달려있다.

위에서 언급한 사항 중 통신성능에 영향을 끼치는 요소로는 통신망 자체의 물리적 특성이 문제가 될 수 있다. 그러나, 통신망의 물리적 성능이 발전하여 Gigabit급 또는 그 이상으로 되더라도 TCP/IP와 같은 기존의 통신 프로토콜을 사용할 경우 사용자 프로그램에서는 실제 물리적 성능을 충분히 활용할 수 없다. 이것은 통신에 있어서 기존의 통신 프로토콜 자체가 차지하는 부하가 막대하기 때문이다. 이와 같은 문제를 해결하기 위해 사용자 수준 통신(user-level communication) [2][3][4][5]모델이 제시되고 있다. 사용자 수준 통신이란 통신을 사용자 수준에서 처리하게 해서 커널수준에서 통신할 때와 달리 커널 내부에서 소요되는 시간을 제거하는 것이다. 또한, 실제 데이터들의 통신시에는 커널의 개입이 배제되는 단순화된 계층구조로 이루어지므로 데이터의 송·수신시에 발생하는 데이터 복사 회수를 최소화하였다.

사용자 수준 통신 모델의 성능의 우수함이 여러 연구를 통해 입증되자 Intel, Compaq, Microsoft사는 이에 관한 대표적인 연구중의 하나인 Cornel 대학의 U-Net을 기반으로 하여 사용자 수준 통신의 업계 표준인 VIA (Virtual Interface Architecture)를 제안하였다. VIA는 통신망에 대한 가상의 인터페이스를 통신하고자 하는 프로세스에게 제공하며, 통신기기에 독립적인 API를 제공하여 플랫폼에 제한을 받지 않는 특성을 가진다.

본 논문에서는 부가적인 계층이 추가되지 않은 SCI(Scalable Coherent Interface) 공유메모리 방식 및 SCI 상에서 구현된 VIA, 그리고 대표적인 소프트웨어 VIA인 M-VIA[6]의 세 가지 통신망 인터페이스를 각각 병렬 정보 검색 시스템에 적용하고 그 성능을 비교 분석한다.

본 연구에서 사용되는 고성능 네트워크인 SCI는 ANSI/IEEE standard로서 최대 1GB/s의 대역폭을 지원하며, point-to-point 및 switch topology를 사용하여 확장성이 우수하고, 고속의 클러스터링 시스템 구축을 가능하게 한다. 그리고, SCI는 RMA(Remote Memory Access)를 사용하여 다른 노드의 사용자 메모리에 접근할 수 있는 NUMA(Non Uniform Memory Access) 특성을 가진다. 즉, 노드간의 데이터 전송을 위해 부가적인 계층이 추가되지 않아 근본적으로 커널의 간섭이 없다. 따라서 커널의 간섭을 배제하는 것이 목적인 VIA의 구현에 사용되기에 적합하다. 그리고, SCI 상에 개발된 VIA시스템의 활용도를 높이기 위해 많은 병렬 프로그래머에게 익숙한 통신 라이브러리인 MPI를 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 관해 소개하고, 3장에서는 병렬 정보 검색 시스템을 살펴본다. 다음 4장에서는 VIA를 기반으로 하는 병렬 정보 검색 시스템의 구조에 관해 설명한다. 그리고, 5장에서는 실험을 통하여 다른 시스템들과 성능을 비교 분석한다. 마지막으로 6장에서는 결론 및 향후 연구 과제를 제시한다.

2. 관련 연구

대용량의 자료를 효율적으로 검색하기 위한 병렬 정보 검색 시스템에 대한 연구가 과거에는 전용 네트워크를 채택한 중형급 이상의 컴퓨터 시스템 상에서 이루어졌다 [7][8][9]. 그러나, 최근에는 표준화된 네트워크의 성능이 향상됨에 따라 PC 기반 클러스터 시스템 상에서의 구현이 중요하게 연구되고 있으며, 이러한 PC 기반 클러스터 시스템 상에서의 병렬 정보 검색에는 특히 네트워크의 성능이 전체 정보검색 시스템의 성능에 가장 큰 영향을 끼치는 것이 입증되었다 [10][11].

그리고, 현재까지 네트워크의 성능을 최대한 활용하기 위한 방안에 대한 연구가 다방면으로 이루어지고 있다. 특히 VIA 프로토콜을 소프트웨어 및 하드웨어로 구현하려는 노력이 계속되고 있으며, 대표적인 VIA 관련 연구 및 구현 사례는 아래와 같다.

먼저 하드웨어 구현에 관해 살펴보면 EMULEX사에서 개발한 CL1000[12]과 Qlogic사에서 개발한 SANblade 2300 Fibre Channel Host Adapter[13] 등이 있다. 그리고, 연구 수준에서 이루어지고 있는 독일 Chemnitz 대학의 PCI-SCI bridge[14]가 있다.

한편, 소프트웨어 VIA 구현도 계속되고 있다. Intel사는 VIA spec의 검증 및 우수성을 보이기 위해 Fast Ethernet과 Myrinet 상에서 VIA를 개발하였다. 또한 업체뿐만 아니라 연구소 차원의 VIA 개발연구도 진행되고 있다. 대표적인 예는 Lawrence Berkeley Lab의 M-VIA가 있다. 이것은 1998년에 개발되었으며, 소프트웨어 VIA를 모듈로 구현한 것으로서 Fast Ethernet 및 Gigabit Ethernet 상에 구현되었다. 또한 VIA의 문제점을 분석하고 보다 향상된 VIA를 제안하기 위한 Berkeley VIA 프로젝트[15]가 Myrinet 상에서 구현되었다.

이와 같이 VIA는 현재 여러 종류의 고성능 네트워크

상에서 연구되고 있다. 그러나, 벤치마크 수준의 응용프로그램에 대한 연구 결과가 나와있을 뿐 실제 응용프로그램에 적용된 것은 알려진 바가 없다. 따라서 본 논문에서는 실제 병렬 정보 검색 시스템에 VIA를 적용하기 위하여 대역폭 및 지연시간 모두 우수한 성능을 가지는 SCI 네트워크 상에 개발된 VIA를 제안하고, 그 활용도를 높이기 위해 MPI를 제공한다.

3. 병렬 정보 검색 시스템

PC 클러스터 기반 병렬 정보 검색 시스템의 일반적인 구조는 <그림 1>에서 보는 바와 같이 사용자와 인터페이스 및 질의 결과를 연산하는 주노드(master node)와 주노드로부터 색인어를 받아서 색인어 역파일을 읽어와 처리하는 종속노드(slave node)로 구성된다.

주노드와 종속노드 각각의 역할에 대한 설명은 <그림 2>와 같다. <그림 2>에서 보는 바와 같이, 주노드는 먼저 사용자로부터 여러 개의 색인어로 이루어진 질의를 받아들인다. 그리고, 질의를 구성하는 색인어들에 대하여 각각의 역파일이 클러스터링된 노드 정보를 기준으로 색인어를 처리할 종속노드를 결정한다. 주노드를 비롯한 종속노드는 자신의 데이터베이스를 따로 가지고 있으며, 주노드와 종속노드 모두는 자신의 노드로 할당된 색인어 역파일을 하드디스크로부터 읽어와서 유사도에 관한 연산을 수행한 후 그 결과를 주노드로 보낸다. 그러면 주노드에서는 각각의 노드에서 보내온 결과를 병합한 뒤, 우선순위를 결정하여 사용자에게 보여준다.

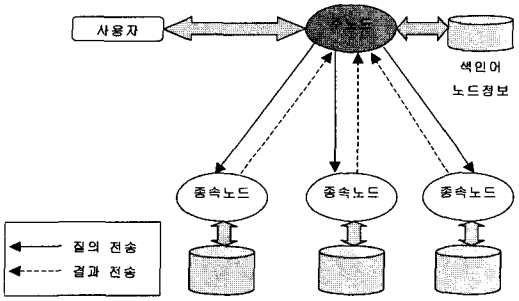

그림 1 병렬 정보 검색 시스템 구조

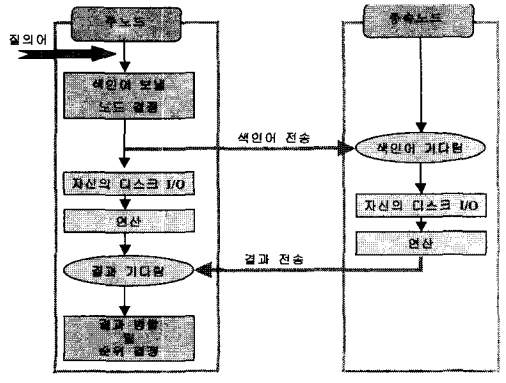

그림 2 병렬 정보 검색 시스템에서 주노드와 종속노드의 역할

4. VIA기반 병렬 정보 검색 시스템

4.1. VIA

먼저 TCP/IP/MPI를 기반으로 하는 병렬 정보 검색 시스템의 통신 구조는 <그림 3>과 같다.

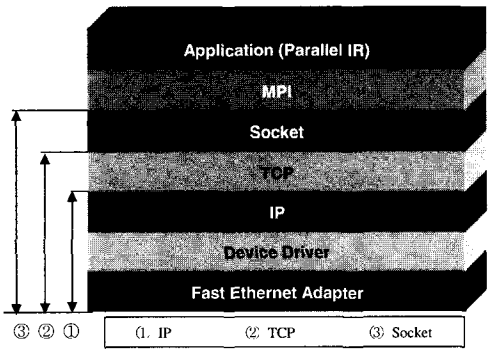

그림 3 TCP/IP/MPI 기반 병렬 정보 검색 시스템의 통신 구조

기존의 TCP/IP 기반 통신 모델에서는 데이터의 전송이 요구될 때 데이터가 커널 내부의 프로토콜 스택을 거치게 된다. 이 데이터가 프로토콜 스택의 각 계층(layer) 사이를 통과해 갈 때마다 버퍼로의 복사가 발생하게 되는데, 이로 인해 각 계층을 통과해가면서 계속 지연시간이 누적되고 결과적으로 대역폭에서도 손해를 보게 된다.

이러한 단점을 극복하기 위해 커널 내부에서의 복사를 줄이고자 하는 사용자 수준 통신이 연구되어 왔으며, 최근에 그 표준으로 VIA(Virtual Interface Architecture)가 제안되었다. 본 논문에서는 VIA 프로토콜을 SCI 상에 구현하였고, Lawrence Berkeley Lab에서 Fast Ethernet 상에 구현한 M-VIA와 본 연구의 SCI/VIA를 함께 사용하여 TCP/IP 기반 네트워크를 사용한 시스템과 성능을 비교하였다.

먼저 VIA의 구성요소를 살펴보면, VIA는 가상 인터

페이스(VI), 컴플리션큐(CQ), VI 제공자(VI Provider) 그리고 VI 소비자(VI Consumer)로 이루어진다. VI에는 작업큐(Work Queue)가 있는데 이것은 송신큐(Send Queue)와 수신큐(Receive Queue)가 한 쌍이 되어 구성된다. 그리고, VI 제공자는 초기화 및 자원 관리를 수행하는 커널 관리자(Kernel Agent)와 실제 데이터 전송을 담당하는 VI 네트워크 어댑터로 구성된다. 마지막으로 VI 소비자는 VIPL(*Virtual Interface Provider Library*), MPI 등의 통신 인터페이스와 사용자 프로그램으로 이루어진다.

다음으로 VIA의 통신방법을 보면, 가장 처음 수행되는 과정이 VI 소비자가 통신을 위한 VI를 생성하는 것이다. 이것은 커널관리자에게 시스템 호출을 함으로써 수행된다. 일단 VI가 생성되고 나면 그 후의 송신 및 수신 요청은 시스템 호출 없이 바로 VI를 통해서 이루어진다. 통신이 끝난 후 VI 소비자의 요청이 끝났음이 그 작업큐에 짝지워진 CQ에 기록된다. 다음 <그림 4>는 위에서 설명한 VIA의 구조를 나타낸 것이다.

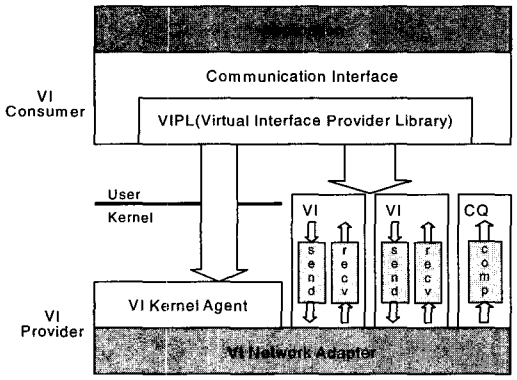

그림 4 VIA의 구조

4.2. SCI

공유메모리 프로그램 방식을 사용하게 되면 지역(local) 노드에서 원격(remote) 노드의 메모리에 직접 접근할 수 있다. 그러나, SCI기반 클러스터 시스템에서 임의의 지역메모리를 전역메모리로서 사용하기 위해서는 PCI-SCI 어댑터의 디바이스 드라이버에서 매핑(mapping) 과정을 거쳐야 한다. 이러한 매핑이 완료되어야만 원격 노드의 메모리 영역에 지역 노드의 프로세스가 직접 접근할 수 있게 된다.

원격 노드에서 지역 노드 메모리에 접근이 가능하게 하려면 우선 지역노드에서 일정 영역을 할당하고 원격 노드에서 접근 가능하게 허용해 주어야 한다. 그런 다음 통신을 원하는 원격 노드는 해당 지역 노드와 매핑을 수행한다. 그러면 원격 노드는 매핑한 영역에 대해서는 자신의 지역 메모리처럼 읽고 쓰기를 할 수 있게 된다.

앞으로의 설명의 편의를 위해 지역 메모리에 접근을 허가해 주는 노드의 메모리 주소 영역을 Local Segment라 하고, 접근을 시도하려는 원격노드의 주소 영역을 Remote Segment라 한다.

본 연구에서 VIA기반 시스템과 비교를 위해 참조되는 SCI 공유메모리 기반 병렬 정보 검색 시스템에 쓰이는 통신 라이브러리는 SISCI(Software Infrastructure for SCI) API[16]를 사용하여 개발되었다. 본 구현에서 모든 메모리 트랜잭션은 원격 쓰기로 구현되었다. 이것은 Dolphin사의 PCI-SCI 어댑터의 경우 원격 노드에 위치한 공유메모리에 대해서 원격 쓰기가 원격 읽기에 비해 10배 이상 빠른 동작 특성을 보이고 있기 때문이다.

4.3 SCI/VIA 및 SCI/VIA/MPI

SCI 네트워크 상에서 VIA를 구현하기 위한 과정으로는 먼저 Kernel Agent가 초기화 과정에서 커널의 도움을 받아 VI를 생성하고, 다음으로 생성된 VI간의 연결을 설정하는 초기화 과정이 필요하며, 이후에 연결 설정을 바탕으로 한 데이터 전송 과정이 필요하다.

우선, VI간의 연결 설정 과정을 살펴보면 자신의 지역메모리에 접근을 허용하는 Server역할의 노드는 공유메모리 영역에 Local Segment를 만들고 통신을 원하는 노드의 요청이 생길 때까지 기다린다. 이 노드와 통신을 원하는 Client노드는 Remote Segment를 만들고 Server의 Local Segment에 매핑을 수행한 뒤 Server에게 자신의 주소를 비롯한 연결 설정 정보를 포함하는 연결요청 데이터를 보낸다. Server는 이 요청 데이터를 보고 연결 수락 및 거절을 판단하여 알려준다. 연결이 수락되면 Client 노드는 Local Segment를 생성하고 Server 역시 Remote Segment를 생성해서 Client노드의 Local Segment에 매핑한다. 이렇게 하면 송신에는 Remote Segment를 사용하고 수신에는 Local Segment를 이용할 수 있으며, 실제 데이터 전송시에는 Dolphin사의 원격 쓰기를 사용한 구현이 가능해진다.

다음으로 데이터 전송과정을 살펴보면, VIA에서 데이터 전송은 Descriptor를 통해 이루어진다. 즉, 송신측은 송신할 데이터의 정보가 담긴 *Descriptor*를 송신큐에 넣고, 수신측은 수신할 데이터의 Descriptor를 만든 후 수신큐에 넣는다. 그러면 VI 네트워크 어댑터는 송신큐의 Descriptor의 내용을 보고 그 정보에 따라 데이터를 전송한다. 송신 데이터는 그것이 매핑된 Remote Segment에 원격 쓰기로 수행되어지고, 그렇게 되면 초기화 과정

에서 매핑되었던 Local Segment에서 수신하게 된다. 이렇게 수신된 데이터는 수신자의 VI의 수신큐에 있는 Descriptor의 정보를 참고해서 해당 버퍼에 저장된다. 이러한 과정이 모두 완료되면 네트워크 어댑터는 CQ에 전송이 완료되었음을 표시한다.

MPI는 이식성(portability)과 기능성(functionality)을 고려하여 설계되었으며, 메시지 패싱의 초기화 및 각종 메시지 패싱 함수의 형식과 의미를 정의하고 있다. MPI는 이식성을 위해 시스템의 네트워크 구조에 종속적이지 않은 인터페이스를 개발하였으며, 프로그램 작성에 필요한 다양한 형태의 메시지 패싱 함수를 지원함으로써 사용자에게 프로그램 개발의 용이성을 제공한다. 또한 기존의 병렬 프로그램들이 MPI를 사용하여 개발된 경우가 많으므로, 새로운 시스템에서의 호환성과 이식성을 유지하기 위해 MPI를 지원하는 것이 필요하다.

본 연구에서는 SCI 네트워크 기반으로 구현된 SCI/VIA상에 MPI를 제공하여 프로그래머에게 편리한 프로그래밍 환경을 제공하며, MPI로 작성된 기존의 응용 프로그램이 SCI/VIA 위에서 별다른 수정없이 사용될 수 있도록 한다.

일반적으로 MPI는 초기화 모듈, 송·수신 모듈 및 동기화 모듈 등이 필요하다. 여기서 VIA 초기화 및 연결에 관련된 작업은 모두 MPI 초기화 모듈에서 수행하며, 일단 MPI 초기화가 완료되면 SCI 공유메모리 상에 통신에 사용될 VI가 생성되고 통신을 원하는 다른 노드들과 연결이 설정된 상태가 된다. 따라서 그 후에 발생하는 MPI 데이터 전송은 모두 공유메모리에 위치한 VI를 통해서 이루어진다.

SCI/VIA/MPI를 기반으로 하는 병렬 정보 검색 시스템을 구현하기 위해서는 다음 <그림 5>와 같은 통신 구조가 필요하다. <그림 5>에서 보는 바와 같이, SCI를 네트워크로 사용하고 있으므로 SCI계층이 최하위에 있게 된다. 그리고, SCI의 API를 사용하여 VIA프로토콜을

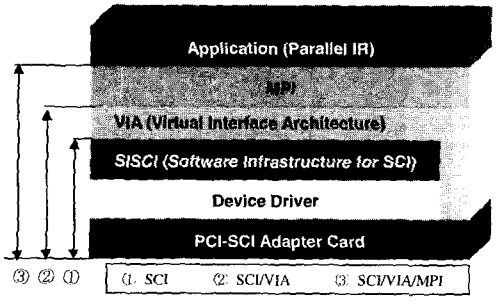

그림 5 SCI/VIA/MPI 기반 병렬 정보 검색 시스템의 통신 구조

구현한다(SCI/VIA). 그리고, 시스템의 이식성과 활용성을 위하여 병렬 프로그래밍의 표준 통신 라이브러리인 MPI(Message Passing Interface)계층을 추가한다(SCI/VIA/MPI). 마지막으로 최상위 계층인 응용프로그램으로 정보 검색 소프트웨어를 사용하게 된다.

4.4 FE/VIA 및 FE/VIA/MPI

앞에서 언급한 바와 같이, 대표적인 소프트웨어 VIA로는 Lawrence Berkeley Lab의 M-VIA가 있다. M-VIA는 먼저 Fast Ethernet 어댑터의 디바이스 드라이버를 VIA의 동작 구조에 맞추어 수정하고, 그 위에 VIPL을 구현하였다. 또한 M-VIA와 MPI의 효율적인 결합을 위해 MVICH[17]가 개발되었다. 본 연구에서는 Fast Ethernet 상에 M-VIA와 MVICH를 이식하여 구현한 FE/VIA/MPI 기반 병렬 정보 검색 시스템을 SCI/VIA/MPI 기반 시스템과 비교하였다.

FE/VIA/MPI 기반 병렬 정보 검색 시스템의 통신 구조는 <그림 6>과 같다. FE/VIA는 SCI/VIA와 비슷한 구조이지만, API를 거치지 않고 직접 네트워크 어댑터에 접근하게 된다.

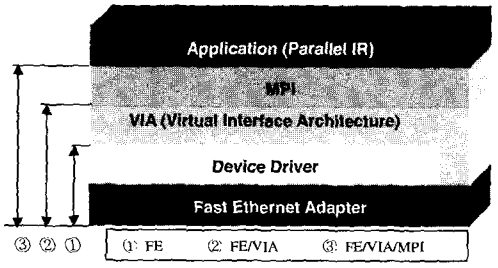

그림 6 FE/VIA/MPI 기반 병렬 정보 검색 시스템의 통신 구조

5. 실 험

5.1 실험 환경

본 연구에서는 8대의 Pentium-II PC를 연결한 실험용 PC 클러스터 시스템을 사용하였으며, 각 노드는 각각 한 개의 350MHz Pentium-II 프로세서, 128MB의 주메모리, 4.3GB 용량의 SCSI 하드디스크를 탑재하고 있으며, 운영체제로는 Linux 커널 2.2.9를 사용하였다. <그림 7>는 Fast Ethernet 어댑터와 Switching HUB로 연결한 8-노드 PC 클러스터 시스템의 구조를 나타내고 있으며, <그림 8>은 PCI-SCI 어댑터와 Dolphin사의 CluStar SCI switch를 사용하여 연결한 8-노드

PC 클러스터 시스템의 구조를 나타내고 있다.

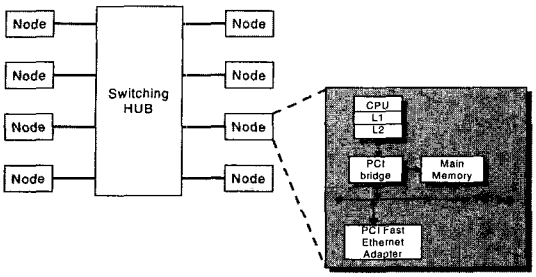

그림 7 Fast Ethernet 기반 8-노드 PC클러스터 시스템의 하드웨어 구조

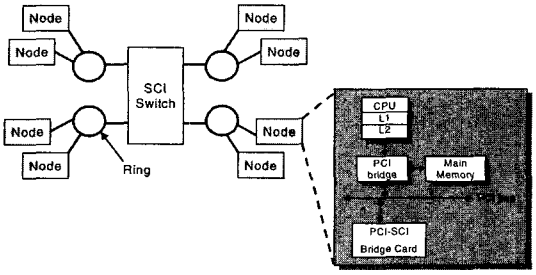

그림 8 SCI 기반 8-노드 PC 클러스터 시스템의 하드웨어 구조

본 연구에서는 relevance feedback을 이용한 병렬 정보 검색 시스템을 구현하였다. Relevance feedback 방법을 이용한 검색에서는 사용자가 입력한 질의와 관련된 문서를 검색 시스템이 제시한 후, 다시 사용자에 의해 선택된 문서들 내에서 질의와 관련되면서 동시에 문서를 대표할 만한 중요한 색인어들을 검색 시스템이 선택하여 새로운 확장 질의를 구성하게 된다.

또한 본 연구에서는 약 50만 건의 신문기사 문서와 여기에서 추출된 색인어들을 실험의 대상으로 사용하였다. 최근 3년 간의 일간지 기사를 대상으로 하여 총 489,772개의 문서를 추출하고, 이로부터 4,701,388개의 색인어 집합을 구성한 후 이 색인어 집합에서 2개 이하의 문서에 나타나는 색인어를 제거하였다. Relevance feedback 검색을 위해 각 질의의 색인어 수는 24개로 확장되었으며, 총 500개의 질의를 사용하였다.

본 실험에서는 색인어의 문서내 중요도값의 척도로 *tf* * *idf* 값을 사용한다. *tf* 와 *idf* 에 대한 설명은 아래와 같다. 여기에서 *tf* 는 한 문서가 선택될 때마다 문서 내의 색인어별로 새로 계산되고, *idf* 는 모든 색인어와 모든 문서에 대해 미리 계산되어 있어야 한다.

tf = term frequency (한 문서 내에서 색인어 t_i 가 나타나는 빈도 수)

$idf = \log_2(N/n) + 1$

N : 총 문서수

n : *df* (document frequency : 임의의 색인어 t_i 가 나타나는 문서의 수)

여기에서 한 색인어가 어떤 문서 내에서 등장 횟수가 많으면 그 문서의 중요한 색인어일 가능성이 높고, *tf* 값이 커지게 된다. 그러나, 문서 라이브러리 전체에서 공통적으로 많이 나타나는 색인어는 일반적인 색인어일 가능성이 높아지므로 문서를 대표할 수 있는 중요 색인어일 가능성이 낮고, 이런 색인어는 n 값이 커지므로 *idf* 값이 작아져서 그 중요도가 낮아지게 된다.

그리고, 본 실험에서는 색인어 역파일 분산 방법으로 무작위 방안을 사용하였다. 무작위 방안은 색인어 역파일을 임의의 노드에 분산시키는 방법으로 노드간의 부하편중 여부를 고려하지 않는 특성이 있다.

5.2 실험 결과 및 분석

5.2.1. 대역폭 및 지연시간 측정을 통한 성능 비교

먼저 통신 성능 평가의 가장 기본적인 방법인 대역폭 및 지연시간을 각 시스템 별로 살펴본다. <그림 9>는 SCI/VIA, SCI/VIA/MPI, FE/VIA, FE/VIA/MPI, TCP/IP/MPI의 대역폭을 비교하고 있고, 다음 <그림 10>은 이 5 종류 시스템의 지연시간을 비교하고 있다.

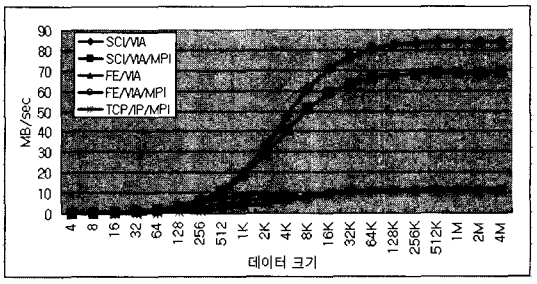

그림 9 시스템 대역폭 비교

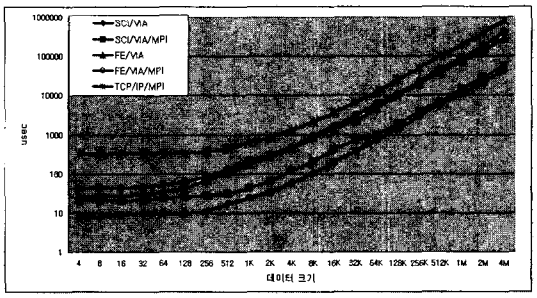

그림 10 시스템 지연시간 비교

최대대역폭의 경우를 살펴보면 SCI/VIA는 약 84MB/s에 이르러 SCI 공유메모리 라이브러리만을 사용하는 경우에 근접하는 우수한 성능을 보인다[18]. SCI/VIA/MPI의 경우 SCI/VIA 상에 MPI를 구현하는데 약간의 오버헤드가 있어 약 69MB/s의 최대대역폭을 보인다. 한편 FE/VIA는 약 12MB/s, FE/VIA/MPI는 약 11.4MB/s의 대역폭에 이르러 Fast Ethernet의 물리적 성능의 한계에 근접하는 성능을 보여준다. 그리고, TCP/IP/MPI의 경우에도 최대대역폭은 약 10.7MB/s에 달함으로써 FE/VIA에 비해 큰 차이가 없다.

그러나, 최소지연시간의 측면에서 비교하면 FE/VIA가 약 26μsec의 지연시간을 보이는데 반해 TCP/IP/MPI는 최소지연시간이 수백 μsec에 이르러 큰 차이를 보이고 있어 비교적 작은 크기의 데이터를 전송할 경우에는 FE/VIA가 압도적으로 유리한 성능을 발휘할 수 있다. 그리고, FE/VIA/MPI는 FE/VIA를 사용한 효과로서 42μsec의 비교적 적은 지연시간을 보이고 있고, SCI/VIA와 SCI/VIA/MPI는 각각 8μsec와 20μsec의 최소지연시간을 보인다.

병렬 정보 검색을 위한 실제 실험은 앞에서 말한 시스템 중 VIA 상에 MPI 계층이 추가된 시스템을 기반으로 한다. 이것은 대역폭 및 지연시간에서 MPI 계층의 추가로 인한 성능 저하가 크지 않음을 실험을 통해 확인하였고, 실제 응용프로그램 개발시 MPI를 기반으로 하는 것이 현실적이기 때문이다.

5.2.2 병렬 정보 검색 시스템을 적용한 결과 및 성능 분석

본 논문에서 앞으로의 실험에 사용된 병렬 검색을 위한 기반이 되는 시스템은 다음과 같다.

- SCI/VIA/MPI : SCI 상의 VIA 프로토콜 기반 MPI
- SCI/VIA : SCI 기반 VIA 프로토콜
- FE/VIA/MPI : Fast Ethernet 상의 VIA 프로토콜 기반 MPI
- TCP/IP/MPI : Fast Ethernet 상의 TCP/IP 프로토콜 기반 MPI

실험은 클러스터 상에서 노드 수를 증가시키면서, 위에서 말한 네 시스템을 기반으로 하는 병렬 정보 검색 시스템의 성능을 비교분석 한다. 실험 결과의 수치는 질의 집합 500개를 반복 실험 한 후 주노드에서 나타난 결과를 평균한 것이다.

아래 [표 1]은 앞으로의 그림에 나타날 용어를 설명한다.

다음 [표 2]와 <그림 11>은 SCI/VIA/MPI를 사용한 경우의 질의시간 비교이다.

표 1 통신 시간 측정에 사용된 용어

용어	설명
질의전송	각 종속노드로 색인어를 모두 보내는데 걸린 시간
디스크 I/O	분배된 색인어에 따른 실제적인 디스크 접근에 걸리는 시간
병렬화연산	병렬화하여 수행하는 색인어에 대한 연산 (AND/OR 연산 및 유사도 계산)
결과전송	병렬화 연산이 모두 종료된 후 그 결과 데이터를 전송하는데 소요되는 시간
결과지연	주노드가 모든 종속노드로부터 병렬화 연산의 결과를 다 받아들일 때까지 기다리는 시간
주노드연산	주노드가 모든 노드로부터 도착한 결과를 병합하고 순위를 정해서 정렬하는 데 걸리는 시간
총소요시간	하나의 질의 처리에 걸린 시간의 총합 (질의전송 + 디스크 I/O + 병렬화연산 + 결과전송 + 결과지연 + 주노드연산)

표 2 SCI/VIA/MPI를 사용한 실험시 요소별 소요된 시간 (단위:초)

	1노드	2노드	4노드	8노드
질의전송	0	0.0001	0.0002	0.0007
디스크 I/O	0.5541	0.3233	0.1878	0.1029
병렬화연산	0	0.5955	0.1870	0.0486
결과전송	0	0.0184	0.0212	0.0273
결과지연	0	0.3911	0.3941	0.3435
주노드연산	2.0368	0.0559	0.0741	0.0942
총소요시간	2.5909	1.3843	0.8644	0.6172

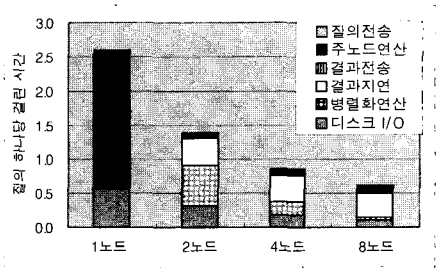

그림 11 SCI/VIA/MPI를 사용한 실험시 성능 향상 정도 (단위:초)

표 3 SCI/VIA를 사용한 실험시 요소별 소요된 시간 (단위:초)

	1노드	2노드	4노드	8노드
질의전송	0	0.0001	0.0002	0.0007
디스크 I/O	0.5541	0.3250	0.1869	0.1019
병렬화연산	0	0.5963	0.1869	0.0486
결과전송	0	0.0162	0.0205	0.0272
결과지연	0	0.3785	0.3878	0.3347
주노드연산	2.0368	0.0560	0.0738	0.0935
총소요시간	2.5909	1.3721	0.8561	0.6066

다음 [표 3]과 <그림 12>은 SCI/VIA를 사용한 경우의 질의시간 비교이다.

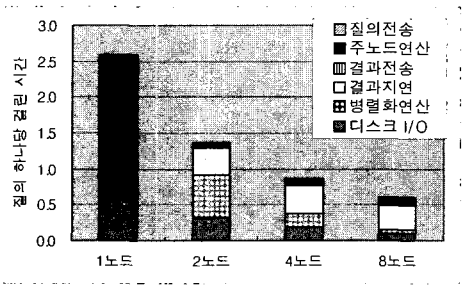

그림 12 SCI/VIA를 사용한 실험시 성능 향상 정도 (단위:초)

다음 [표 4]와 <그림 13>은 FE/VIA/MPI를 사용한 경우의 질의시간 비교이다.

표 4 FE/VIA/MPI를 사용한 실험시 요소별 소요된 시간 (단위:초)

	1노드	2노드	4노드	8노드
질의전송	0	0.0003	0.0015	0.0029
디스크 I/O	0.5541	0.3188	0.1785	0.0968
병렬화연산	0	0.5981	0.1834	0.0476
결과전송	0	0.0325	0.0481	0.0584
결과지연	0	0.3973	0.4059	0.3446
주노드연산	2.0368	0.0513	0.0679	0.0864
총소요시간	2.5909	1.3983	0.8853	0.6367

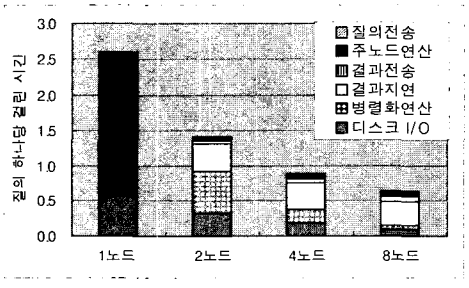

그림 13 FE/VIA/MPI를 사용한 실험시 성능 향상 정도 (단위:초)

다음 [표 5]와 <그림 14>는 TCP/IP/MPI를 사용한 경우의 질의시간 비교이다.

표 5 TCP/IP/MPI 를 사용한 실험시 요소별 소요된 시간 (단위:초)

	1노드	2노드	4노드	8노드
질의전송	0	0.0005	0.0019	0.0041
디스크 I/O	0.5541	0.3243	0.1869	0.1039
병렬화연산	0	0.6002	0.1912	0.0492
결과전송	0	0.0531	0.0821	0.0959
결과지연	0	0.4442	0.3904	0.3671
주노드연산	2.0368	0.0602	0.0804	0.0997
총소요시간	2.5909	1.4825	0.9329	0.7199

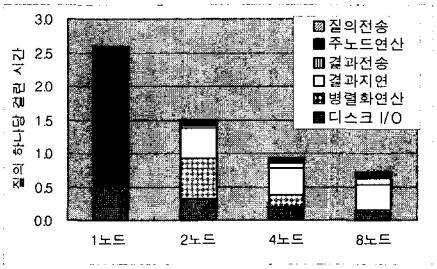

그림 14 TCP/IP/MPI를 사용한 실험시 성능 향상 정도 (단위:초)

위에 나타난 표와 그림에서 보는 바와 같이 사용된 시스템의 종류에 상관없이 노드를 증가시킬수록 병렬화 효과로 인해 질의 하나당 총 수행시간은 감소한다. SCI/VIA/MPI의 경우 노드 증가에 따라 최대 4.198배의 속도 향상을 보이고 있고, SCI/VIA는 최대 4.271배의 속도 향상을 보인다. FE/VIA/MPI의 경우 4.069배의 속도 향상이 있으며, TCP/IP/MPI의 경우 속도 향상은 3.599배를 보인다. SCI/VIA/MPI 기반의 병렬 정보 검색 시스템의 성능은 SCI/VIA 기반 병렬 정보 검색 시스템의 성능에 비해 크게 뒤지지 않는다. 그리고, FE/VIA/MPI도 TCP/IP/MPI에 비해서 약 12% 정도 더 우수한 성능을 보이고 있다.

본 실험의 결과 TCP/IP/MPI의 성능이 가장 나쁘게 나타나는데, 이는 <그림 10>의 시스템 지연시간에서 나타난 바와 같이 물리 네트워크간의 성능 차이와 더불어 커널 수준에서의 간섭으로 인해 지연시간이 나쁜 것이 원인이 된다. 다음으로 Fast Ethernet 기반의 FE/VIA/MPI의 속도 향상이 낮은데, 이는 물리 네트워크간의 성능 차이로 인해 FE/VIA/MPI의 지연 시간이 SCI/VIA/MPI의 지연시간보다 더 큰 것이 원인이 된다.

각 요소별로 실행시간을 분석해보면, 노드가 증가함에

따라 병렬화 효과로 인해 실제 디스크 접근 시간(디스크 I/O)과 병렬화 연산의 실행 시간 비율은 줄어들고 있다.

한편, 성능 향상의 가장 큰 장애 요인은 노드가 증가할수록 더 큰 비율을 차지하는 결과 지연 시간이다. 이것은 색인어 역파일이 적절하게 분산되어 있지 않아 질의 처리시에 특정 노드에 색인어가 편중되었기 때문이다. 즉, 특정 노드에 디스크 I/O 및 정보 검색에 관련된 연산의 편중을 초래하였다. 이러한 특정 노드가 종속노드 중 하나인 경우 주노드 자신의 연산이 모두 끝나더라도 종속노드의 처리가 끝나기를 기다려야 한다. 노드가 증가할수록 주노드보다 처리시간이 긴 종속노드가 발생할 가능성은 더욱 커진다. 따라서 결과 지연 시간의 비율을 줄이는 것이 성능 향상을 위해 해결해야 할 가장 큰 문제이다. 이를 위해서 색인어 역파일을 노드들 간의 처리 부하가 균등하게 분산 저장하는 방안의 개발이 요구된다. 또, 색인어 역파일을 복제하여 2개 이상의 노드에 중복 저장해 두고 작업 부하가 적은 곳으로 색인어를 동적으로 할당하여 처리하게 하는 방안을 고려해 볼 수도 있다.

증가하는 비율을 보이는 다른 요소는 주노드연산 시간이다. 이것은 주노드가 모든 종속노드의 처리 완료된 질의를 받은 후, 이것을 병합하는데 시간이 많이 걸리기 때문이다.

그리고, 종속노드의 결과 데이터 전송이 완료되는데 소요된 시간인 결과전송 시간 역시 노드 수가 증가할수록 늘어난다. 이것은 연산시간이 포함되지 않는 순수 통신시간만을 나타내는 것으로 노드가 증가할수록 주노드로 전송되는 데이터의 양이 증가하기 때문이다.

현재 실험에 사용된 정보 검색의 결과는 SCI/VIA/MPI 기반의 시스템과 Fast Ethernet 기반의 FE/VIA/MPI를 사용하는 시스템간의 성능 차이가 크지 않다. 그 원인은 검색 결과 데이터의 대역폭 요구량이 그다지 크지 않기 때문으로 분석된다. 노드 증가에 따른 결과 데이터 크기의 분포는 아래 <그림 15>와 같이 나타난다.

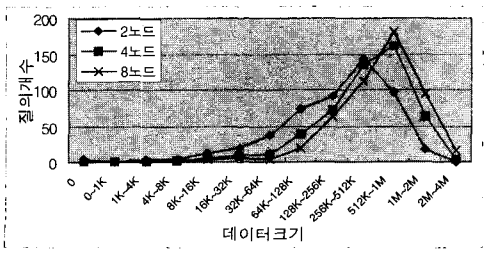

그림 15 노드 수 증가에 따른 검색 결과 데이터의 크기

즉, 검색에 참여하는 노드 수가 2개인 경우 출현빈도가 가장 높은 결과 데이터 크기는 256KB ~ 512KB 사이이고, 4개인 경우는 256KB ~ 1MB 사이이다. 그리고, 노드 수가 8개인 경우 결과 데이터의 크기는 512KB ~ 1MB 사이가 가장 많이 나타난다. 전체적으로 노드 수가 늘어날수록 검색결과 데이터의 크기가 커진다. 이것은 색인어 역파일이 분산 저장되어 있기 때문에 노드 수가 증가할수록 질의내의 색인어들 중 종속노드가 처리하는 색인어 개수가 많아지기 때문이다. 그러므로 검색에 참여하는 노드 수에 따라 종속노드들이 주노드로 보내는 결과데이터의 총 합의 크기도 커진다.

실제로 질의 검색 500회를 8노드로 검색하는 경우, 모든 횟수를 수행하는 동안 종속노드들이 주노드로 전송하는 결과 데이터 크기의 총합은 350.79MB가 된다. 즉, 질의 하나당 평균 데이터의 크기는 718KB정도이다. 여기서 SCI/VIA/MPI를 사용하는 경우, 요구되는 시간당 평균 데이터량은 약 2MB/s 정도이다. 이는 Fast Ethernet의 대역폭만으로도 충분히 만족하는 값이며, 따라서 SCI/VIA/MPI 기반 시스템과 FE/VIA/MPI 기반 시스템의 성능 차이가 두드러지지 않는다.

그러나, 사용자에게 보다 더 신뢰받을 수 있는 높은 품질의 정보를 제공하기 위해서는 검색의 결과로 문서 ID와 유사도 정보를 보내는데 그치지 않는다. 실제로 현재 정보 검색제공 업체들은 검색 결과로 요약정보, 미리 보기, 비슷한 페이지, 연결 페이지 검색 등을 제공하는데, 이런 서비스를 제공하기 위해서는 보다 많은 양의 데이터를 전송해야 한다. 따라서 Fast Ethernet의 대역폭으로는 검색할 기반의 데이터 양이 많아지고 검색결과가 많아질수록 효율이 떨어진다. 이러한 경우 대역폭 크기가 훨씬 큰 SCI/VIA 기반 시스템이 훨씬 좋은 효율을 발휘할 것이다.

6. 결론 및 향후 과제

본 연구에서는 효율적인 병렬 정보 검색 시스템을 위해 SCI 네트워크 상에 개발된 VIA인 SCI/VIA를 기반으로 시스템을 구성하였다. 또한 이 시스템의 활용도를 높이기 위해 SCI/VIA 상에 병렬 프로그래밍의 표준 인터페이스인 MPI를 구현하였다. SCI/VIA 기반 시스템은 데이터 전송시에 커널의 간섭을 배제함으로써 SCI 공유메모리 라이브러리만을 사용하는 시스템에 근접하는 우수한 성능을 지녔음을 보였다. 그리고, 병렬 정보 검색 시스템에 SCI/VIA/MPI와 SCI/VIA를 적용한 실험을 통해 SCI/VIA 상에 MPI 계층이 추가되어도 검색 시스템의 성능에 큰 차이가 없음을 보여 SCI/VIA가

MPI를 위한 기반으로서도 우수함을 증명하였다.

그리고, Fast Ethernet 상에서 M-VIA 기반의 MVICH를 사용한 FE/VIA/MPI 시스템과 TCP/IP 기반 MPI 시스템의 성능을 병렬 정보 검색 실험을 통해 비교하였다. FE/VIA/MPI 시스템이 TCP/IP/MPI 시스템보다 우수한 성능을 보였고, Fast Ethernet 상에서도 VIA를 사용함으로써 하드웨어의 성능을 최대한 활용할 수 있음을 입증하였다.

실험에 사용된 문서기반의 병렬 정보 검색 시스템에서는 대역폭의 요구량이 Fast Ethernet 상에서도 많은 부하를 발생시키지 않아서 SCI 기반의 SCI/VIA를 활용하는 것과 Fast Ethernet 기반의 M-VIA를 사용하는 것에서 뚜렷한 성능의 차이를 보이지는 않았다. 그러나, 앞으로 문서 ID와 유사도 뿐만 아니라 link정보, 요약정보 등의 보다 많은 데이터를 통신하게 될 경우나, 또 실제 사용자에게 적합한 서비스를 제공하기 위해 검색 결과 데이터의 대역폭의 요구량이 많아지면 데이터를 전송하는데 소요되는 통신 시간을 최대한 줄이기 위해 SCI/VIA를 사용하는 것이 보다 효율적인 서비스를 제공하게 될 것이다. 그리고, 검색에 참여하는 노드 수가 증가할 경우 색인어 역파일을 노드들 간의 처리 부하가 균등하게 분산 저장하는 방안 및 색인어 역파일을 복제하여 중복 저장해 두고 작업 부하가 작은 곳으로 색인어를 동적으로 할당하여 처리하는 방안 등을 활용하여 주노드의 결과지연 시간을 대폭 줄일 수 있다면 성능 향상 정도가 더욱 높아질 것이다.

향후 과제로는 SCI와 견줄만한 물리적 성능을 보이는 Gigabit Ethernet, Myrinet 등의 고속의 네트워크로 구성된 클러스터 상에서 VIA프로토콜을 이용하는 병렬 정보 검색 시스템을 구성하여 그 성능을 비교 분석해볼 수 있을 것이다. 그리고, 결과지연 시간을 줄이기 위한 방안에 관련한 연구 또한 중요한 부분이 될 것이다.

참 고 문 헌

[1] Microprocessor and Microcomputer Standards Subcommittee, *"IEEE Standard for Scalable Coherent Interface," IEEE Std 1596-1992,* IEEE Computer Society, August 1993.

[2] Basu, A., Buch, V., Vogels, W. and von Eiken, T., *"U-Net: A User-Level Network Interface for Parallel and Distributed Computing," Proceeding of the 15th ACM Symposium on Operating Systems Principles*, pp. 40-53, Copper Mountain, Colorado, United States, December 1995.

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

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

[5] Pakin, S., Karamcheti, V. and Chien, A. A., *"Fast Messages (FM): Efficient, Portable Communication for Workstation Clusters and Massively-Parallel Processors," IEEE Concurrency*, Vol. 5, No. 2, pp. 60-73, 1997.

[6] http://www.nersc.gov/research/FTG/via

[7] Sharma, R., *"A Generic Machine for Parallel Information Retrieval," Information Processing and Management*, Vol. 25, No. 3, pp. 223-235, 1989.

[8] Cringean, J. K., England, R., Manson, G. A. and Willett, P., *"Network Design for the Implementation of Text Searching Using a Multicomputer," Information Processing & Management*, Vol. 27, No. 4, pp. 265-283, 1991.

[9] Stanfill, C. and Thau, R., *"Information Retrieval on the Connection Machine: 1 to 8192 Gigabytes," Information processing & Management*, Vol. 27, No. 4, pp. 285-310, 1991.

[10] Chung, S., Kwon, H., Ryu, K., Jang, H., Kim, J., and Choi, C., *"Parallel Information Retrieval on an SCI-Based PC-NOW," Lecture Notes in Computer Science*, 1800, pp. 81-90, May 2000.

[11] Chung, S., Kwon, H., Ryu, K., Chung, Y., Jang, H. and Choi, C., *"Information Retrieval on an SCI-Based PC Cluster," Journal of Supercomputing*, Vol. 19, Issue 3, pp. 251-265, July 2001.

[12] http://www.emulex.com/products/vi/clan1000.html

[13] http://www.qlogic.com/products/sanblade/sanblade_2300.asp

[14] Trams, M., Schlosser, R. and Rehm, W., *"Design Choices and First Results of Our VIA-Capable PCI-SCI Bridge," Proceedings of CLUSTER 2000*, pp. 349-350, Chemnitz, Germany, November 2000.

[15] Buonadonna, P., Geweke, A. and Culler, A., *"An Implementation and Analysis of the Virtual Interface Architecture," Proceedings of SC98*, Orlando, Florida, United States, November 1998.

[16] Giacomini, F., Amundsen, T., Bogaerts, A., Hauser, R., Johnsen, B. D., Kohmann, H., Nordstrøm, R. and Werner, P., *"Esprit Project 23174 - Software Infrastructure for SCI (SISCI), Version 2.1.1," White Paper*, Dolphin Interconnect Solutions, 1999.

[17] http://www.nersc.gov/research/FTG/mvich/index.html

[18] Shin, J., Chung, S. and Hahn, W., *"An SCI-based Software VIA System for PC Clustering," 2001*

IEEE International Conference on Cluster Computing, Newport Beach, United States, October 2001.

강 나 영
2000년 부산대학교 컴퓨터공학과 학사. 2002년 부산대학교 컴퓨터공학과 석사. 2002년 ~ 현재 삼선전자 무선사업부 연구원. 관심분야는 컴퓨터구조, 클러스터 시스템, 병렬처리, 정보검색

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 ~ 2000년 부산대학교 컴퓨터공학과 조교수 및 컴퓨터 및 정보통신연구소 연구원. 2000년 ~ 현재 부산대학교 컴퓨터공학과 부교수 및 컴퓨터및 정보통신연구소 연구원. 관심분야는 클러스터 시스템, 병렬처리, 정보검색, VOD, Infiniband

장 한 국
1999년 부산대학교 컴퓨터공학과 학사. 2001년 부산대학교 컴퓨터공학과 석사 수료. 2001년 ~ 현재 부산대학교 컴퓨터공학과 석박사 통합과정. 관심분야는 컴퓨터구조, 클러스터시스템, 병렬처리, 정보검색, InfiniBand