링 구조 NUMA 시스템에서 디스크 입출력의 성능 향상을 위한 효율적인 방안

(Efficient Schemes for Enhancing Performance of Disk I/O in Ring based NUMA Systems)

김 철 홍 † 김 명 주 †† 장 성 태 ††† 엄 성 용 †† 전 주 식 ††††

(Cheol Hong Kim)(Myuhng-Joo Kim)(Seong Tae Jhang)(Seong Yong Ohm)(Chu Shik Jhon)

요 약 NUMA 구조 다중 프로세서 시스템에서는 상호 연결망으로서의 버스의 제약을 극복하기 위해 지점간 링크를 이용한 링 구조가 제안되었다. 링 구조 NUMA 시스템에서, 전송이 페이지 단위(2K 바이트 이상)로 이루어지는 디스크 입출력 요구는 지점간 링크에 대한 오랜 접근을 요구하여 지점간 링크의 트래픽을 증가시키는 원인으로 작용한다. 본 논문에서는 지점간 링크의 트래픽을 줄이기 위한 방안으로 입출력 전용 채널을 사용하는 세 가지 디스크 입출력 구조를 제시한다. 제시되는 디스크 입출력 구조를 사용하는 시스템은 디스크 입출력 요구로 인한 지점간 링크의 접근을 없앰으로써 기존 시스템에 비해 트랜잭션의 수행시간을 줄일 수 있다는 장점이 있다. 중앙집중형, 병렬형, 분산형의 세 가지 디스크 입출력 구조를 설계하기 위한 고려 사항과 구현 비용 등을 생각해 본 후, 노드 수, 원격 디스크 접근 확률, 디스크 입출력 전송 데이타 크기 등을 달리한 여러 시스템 환경에서의 각 구조의 성능을 비교, 분석해 본다.

Abstract In the past several years, NUMA systems adopted ring topology with high speed unidirectional point-to-point links to overcome the limit of bus for interconnection network. In the ring based NUMA system, the disk I/O transaction increases the traffic of the ring because it has the data whose size is more than 2K bytes. In this paper, we suggests various kinds of disk I/O architectures using I/O channel which transfers disk I/O data to decrease the traffic of the ring. The system using these disk I/O architectures can reduce the transaction execution time because the disk I/O data are not transferred through the ring. We thought about the problems and cost to design Centralized, Parallel, Distributed disk I/O architecture. We analyzed the performance of each architecture varying node counts, remote disk access probability and disk I/O transfer size.

1. 서 론

공유 메모리 다중 프로세서 시스템은 메모리의 공유 방법 또는 분산에 따르는 응답지연시간의 특성에 따라 크게 UMA(Uniform Memory Access), NUMA(Non-Uniform Memory Access) 구조로 분류된다. NUMA 구조는 캐쉬 일관성을 유지하는 유형에 따라 CC-NUMA(cache coherent non-uniform memory access)와 COMA(Cache Only Memory Architecture) 등으로 세분화된다[1].

UMA 구조 다중 프로세서 시스템은 공유 메모리가 모든 프로세서에 의해 동일하게 접근되므로 프로그래밍이 쉽다는 장점을 가지지만, 메모리를 접근하는데 항상 일정한 시간이 소요되고 여러 개의 메모리 접근 요구가 동시에 발생할 경우 메모리의 병목현상으로 인해 전체 시스템의 성능이 급격히 저하되는 단점이 있다. CC-NUMA 구조 다중 프로세서 시스템은 UMA 구조에서 나타나는 메모리의 병목 현상을 완화하기 위해 메모리를 지역적으로 분산시켜 놓고 이러한 여러 지역 메

· 이 논문은 2000년도 두뇌한국21 사업에 의하여 지원되었음

† 비 회 원 : 서울대학교 컴퓨터공학부
kimch@panda.snu.ac.kr
†† 종신회원 : 서울여자대학교 정보통신공학부 교수
mjkim@cs.swu.ac.kr
osy@cs.swu.ac.kr
††† 종신회원 : 수원대학교 전자계산학과 교수
stjhang@mail.suwon.ac.kr
†††† 종신회원 : 서울대학교 컴퓨터공학부 교수
csjhon@riact.snu.ac.kr

논문접수 : 2000년 2월 10일
심사완료 : 2001년 1월 18일

모리들이 모여서 하나의 전역 주소 공간을 이루게 된다. 공유 메모리를 분산시킴으로 인해 시스템의 확장성에 제약이 덜하다는 장점을 가지는 반면, 지역 메모리 접근에 비해 상대적으로 큰 원격 메모리에 대한 데이타 전송 시간이 시스템의 성능을 저하시킨다는 단점이 있다. COMA 구조 다중 프로세서 시스템은 메모리가 특정 노드에 고정되어 있지 않고 캐쉬와 같이 필요에 따라 동적으로 이동한다는 특성으로 인해 CC-NUMA 구조의 단점인 원격 메모리 접근 시간을 줄일 수 있다는 장점을 가지지만, 캐쉬 일관성을 유지하기 어렵고 구현이 복잡하다는 단점이 있다[2]. CC-NUMA 구조 다중 프로세서 시스템의 성능 향상을 위한 방안으로, 원격 메모리에 대한 응답 지연을 줄이고 전역 연결망의 트래픽을 감소시키기 위해 지역 메모리 영역이 아닌 원격 메모리 영역만을 캐싱하는 원격 접근 캐쉬(RAC : Remote Access Cache)가 제안되었다[3]. CC-NUMA 구조 다중 프로세서 시스템은 원격 접근 캐쉬를 첨가함으로 인해 성능에 있어서 COMA 구조 다중 프로세서 시스템을 앞지를 수 있다[4].

상기한 다중 프로세서 시스템 구조를 설계하는 데 있어서, 입출력 시스템에 관한 연구는 지금까지 많은 비중을 차지하기 못해왔다. 컴퓨터의 성능을 측정할 때에도 대부분이 CPU 시간에 관한 것이다. CPU는 매년 55 % 정도의 성능 향상을 가져오고 있지만, 입출력 시스템 쪽은 그러하지 못하다[5]. 그러므로, 다른 부분들에서 성능 향상을 도모한다고 하더라도 결국은 입출력 시스템 때문에 컴퓨터 시스템 전체의 성능 향상에는 제한이 따르게 된다. 더욱이 멀티미디어 시대가 도래함에 따라 이미지, 동영상 등 전송해야할 데이타의 양이 점점 많아지고 있는 실정에서 전체 시스템의 성능은 입출력 시스템의 성능에 좌우될 수밖에 없는 상황이 된 것이다. 입출력 시스템의 성능 향상을 위한 방안으로는 크게 두 가지를 살펴볼 수 있다. 하나는 입출력 장치의 성능 향상을 꼽을 수 있는데, 디스크의 성능 향상으로 RAID (Redundant Array of Inexpensive Disks) 같은 것을 예로 들 수 있다. 다른 것으로는 입출력 장치와 프로세서를 연결하는 입출력 구조의 성능 향상에 관한 것이다.

입출력 시스템을 접근하는 버스 트랜잭션은 일반적인 입출력 장치에 대한 접근, 인터럽트 제어기에 대한 접근, 디스크 접근의 세 가지로 나눌 수 있다. 일반적인 입출력 장치에 대한 접근과 인터럽트 제어기에 대한 접근은 전송 데이타의 양이 최대 64 비트인데, 대부분의 시스템 버스나 입출력 버스에서 32 바이트의 burst 전송을 지원하므로 이 부분에서 성능 향상을 기대하기는 어렵다. 반면, 디스크 접근의 데이타 양은 일반적으로 페이지 단위이고, 그것은 보통 2K 바이트 이상이다[5]. 디스크에 접근하기 위해 시스템 버스를 여러 번에 나누어 오래 점유한다면, 요구 자체가 느리게 처리되고 시스템 버스의 경쟁이 늘어나 전체 시스템의 성능이 저하되는 문제가 발생하므로 이 부분을 개선함으로써 전체 시스템의 성능을 크게 향상시킬 수 있다.

본 논문에서는 시스템 설계의 부수적인 부분으로 입출력 시스템을 고려한 것이 아니라, 입출력 시스템의 성능에 영향을 많이 받는 환경에서 사용되는 컴퓨터 시스템 - 예를 들어, 데이타베이스 서버 같은 시스템 - 을 설계하기 위하여 입출력 시스템의 성능을 높이기 위한 디스크 입출력 구조의 설계 방안을 제시한다. 기존의 지점간 링크 외에 디스크 접근 트랜잭션만을 처리하는 입출력 채널을 새로이 도입한 중앙집중형, 병렬형, 분산형의 세 가지 디스크 입출력 구조를 기반으로 하는 시스템의 구현 가능성 및 설계 비용 등을 고려하고, 시뮬레이션을 통해 각 구조의 성능을 비교, 분석해 본다.

본 논문에서 대상으로 하는 시스템은 서울대학교 PANDA 연구실에서 제안한 PANDA (Progressive Approach of NUMA model based on Distributed shared memory Architecture) 시스템[6]으로 이 시스템은 스누핑 방식의 캐쉬 일관성 유지방법을 사용하고 하나의 트랜잭션을 연속된 여러 개의 패킷으로 나누어 전송하는 레지스터 삽입 방식의 단방향 링크를 사용한 CC-NUMA 시스템이다. PANDA 시스템은 응답 지연 시간의 단축으로 디렉토리 캐쉬 일관성 유지 방법을 사용한 시스템에 비해 성능이 우수함이 발표되었고[7], 최근에는 기존의 시스템을 방향 분리 이중 링 형태로 확장하여 링 대역폭을 확장하는 것이 단순히 단일 링 형태로 링 대역폭을 두 배로 하는 방법보다 더 좋은 성능을 보인다는 것이 발표되었으며[8], 이 결과를 바탕으로 현재 PANDA II 시스템의 설계가 진행중이다.

2장에서는 논문의 연구 대상이 되는 이중 링을 사용한 PANDA 시스템의 전체 구조 및 입출력 시스템의 구조를 살펴본다. 3장에서는 세 가지 디스크 입출력 구조를 제안하고, 각 구조에 대해 정성적 분석을 한다. 4장에서는 3장에서 제시한 각 구조들의 특성 및 성능을 모의실험 결과를 통해 분석해 보고 그 결과를 기술한다.

2. PANDA 시스템의 구조

2.1 PANDA 시스템의 전체 구조

본 연구에서 대상으로 하는 PANDA 시스템의 모형은 그림 1과 같다. 시스템은 메모리 주소 영역을 공유하

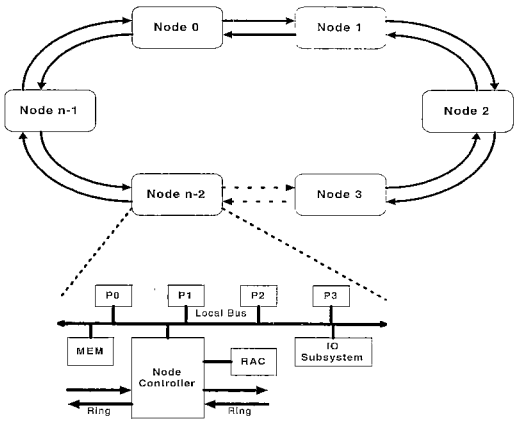

그림 1 PANDA 시스템의 전체 구조도

는 CC-NUMA 구조 다중 프로세서 시스템으로 각 연산 노드가 단방향 지점 간 링크 두 개를 사용하여 방향 분리 이중 링(point-to-point dual ring)을 구성하고 있다. 하나의 링은 시계방향 전송을 하고 다른 하나의 링은 반시계 방향의 전송을 한다. 단일 전송 패킷의 경우에는 두 가지 방향 중에서 현재 노드에서 목적지 노드로 가는데 가까운 방향을 결정하여 해당 링을 사용하고, 방송 패킷인 경우는 주소를 보고 odd/even으로 구분하여 해당 링을 사용하게 구성함으로써 패킷의 전송 경로를 분산시켜 평균 응답 시간을 줄인다. 지점간 링크로 사용하는 IEEE 표준 SCI[9] 링크는 16bit 데이타 폭으로 1GBytes/sec (500MHZ)의 전송률을 가진다. 이하 연산 노드들을 연결하는 링은 시스템 버스라 칭하기로 한다.

시스템 버스는 방송 트랜잭션을 지원하므로 논리적으로 버스와 동일한 동작을 수행하므로 스누핑 방식으로 캐쉬 일관성을 유지하게 된다. 링 구조에서 스누핑 방식의 캐쉬 프로토콜을 사용하는 또 다른 시스템인 Express Ring[10]에서는 슬롯 링(Slotted Ring) 방식으로 구성되어 있는 반면에, PANDA 시스템에서는 레지스터 삽입 방식을 사용한다. 슬롯 링 방식에서 데이타를 전송하고자 하는 노드는 링을 돌고 있는 각기 크기가 다른 여러 슬롯들 중에서 전송할 데이타와 크기가 맞는 빈 슬롯이 지나갈 때에만 전송을 수행할 수 있으므로 링의 이용률이 낮은 단점이 있다. 하지만, 레지스터 삽입 방식에서는 각 링크 사이에 존재하는 FIFO 큐를 사용하여 데이타를 전송하므로 슬롯 링에 비해 링의 이용률을 높일 수 있다는 장점을 가진다.

시스템 버스로 연결된 하나의 연산 노드는 지역 버스로 연결된 4개의 프로세서 모듈(1차 캐쉬와 2차 캐쉬를 내장), 지역 메모리, 링과 지역 버스를 연결하는 노드 제어기, 원격 캐쉬, 입출력 부시스템으로 구성된다. 지역 버스는 요청 트랜잭션과 응답 트랜잭션이 분리될 수 있는 버스, 즉 split 트랜잭션을 지원하는 버스이며 스누핑 방식에 의한 캐쉬 일관성 유지 프로토콜을 사용한다. 지역 메모리는 분산형 공유 메모리로 전체 시스템 메모리의 물리 주소 영역 중 일부를 구성하고 있다. 노드 제어기는 원격 노드로의 접근 지연 시간을 단축하기 위해 원격 접근 캐쉬를 유지하고 있으며, 지역 버스에서 요청이 발생하면 RAC나 전역 링크를 통한 원격 노드로부터의 데이타 제공을 담당한다. 또한, 전역 링크에서 발생한 요청에 대해서도 노드 제어기가 응답할 책임을 진다. 이밖에, 노드 제어기는 지역 버스 및 전역 링크에서 발생하는 모든 트랜잭션을 스누핑하여 메모리-캐쉬 일관성 유지를 위해 필요한 제어를 수행한다. 입출력 부시스템은 프로세서에서 요구한 입출력 디바이스에 대한 읽기 및 쓰기를 수행해 준다. 모든 입출력 관련 트랜잭션은 지역 버스에 연결된 PCI 버스를 통해 처리하게 된다.

2.2 입출력 구조

PANDA 시스템에서는 여러 가지 표준 입출력 버스 중에서 처리 능력 및 주변 장치들과의 연결성 측면 등을 고려하여 PCI 버스를 입출력 버스로 사용하였다. PCI를 입출력 버스로 사용하는 대부분의 시스템에서는 PCI를 시스템 내부에 도입하는데 있어서 시스템 버스 또는 지역 버스와의 인터페이스를 수행하는 카드를 통하여 시스템에 적용하기보다는 시스템 내부에 온보드화 하여 사용하고 있는 실정이다. PCI 입출력 버스의 온보드화에 따라 생각할 수 있는 문제점은 확장성이다. 그러나 PCI의 경우는 자체의 확장성이 매우 크고 단순히 PCI-to-PCI 브리지를 사용하여 매우 큰 확장성을 얻을 수 있으므로, 시스템에 온보드화 되어도 확장성에는 별 문제가 생기지 않는다. 이러한 PCI의 특성을 반영하여 PANDA 시스템에서도 PCI 버스를 도입하기 위한 입출력 보드를 따로 설계하지 않고 보드 내에 온보드화 하는 방법으로 입출력 버스를 도입하였다. 본 시스템에서는 OPB(Orion PCI Bridge)라는 P6 지역 버스와 PCI 버스 사이의 인터페이스를 제공하는 인터페이스 칩을 사용하여 PCI 입출력 버스를 시스템에 온보드화 하여 사용한다.

현재 PANDA 시스템의 입출력 구조는 그림 2의 원 내부와 같다. 입출력 버스인 PCI 버스는 PCI 입출력 모듈(IOC)을 통해 프로세서와 메모리를 연결하는 지역 버스와 연결된다. IOC는 지역 버스와 PCI 버스 사이의 인터페이스를 제공하기 위해 서로 다른 버스간의 전송 속도 차를 보상하며, 버스 상호간의 프로토콜을 변경시

켜주는 기능을 수행한다. PANDA 시스템에서는 적은 응답 지연으로 높은 처리속도를 가짐과 동시에 외부적인 추가 로직이 없이 P6 지역 버스에 직접 연결이 가능하고, 동기적 또는 비동기적인 PCI 인터페이스를 지원하며, 다중의 입출력 브리지 사이의 상호 통신을 통한 시스템 성능 향상 방안을 제공하는 특성을 가진 OPB를 이용하여 PCI 버스와 지역 P6 버스를 연결하는 방안을 제시하고, 이를 기반으로 IOC를 구현하였다. PCI 버스 하부에는 여러 개의 입출력 장치가 연결될 수 있으며, 특히 가장 중요한 저장 장치 시스템과는 SCSI를 통해 연결된다. 디스크 등의 저장 장치가 연결된 SCSI 버스는 디스크 입출력 모듈인 SCSI 인터페이스를 통해 PCI 버스와 연결된다.

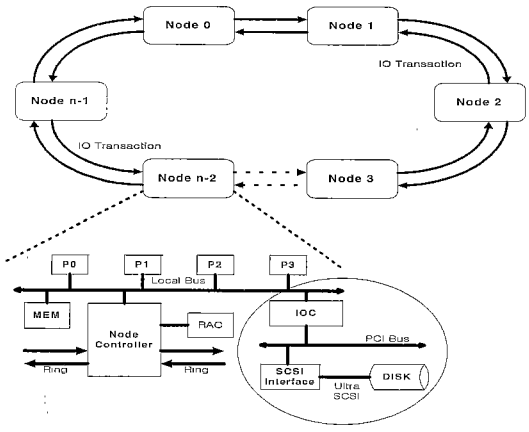

그림 2 PANDA 시스템의 입출력 구조

3. 디스크 입출력 구조의 구성 및 정성적 분석

이 장에서는 입출력 시스템의 구성 요소인 디스크 입출력 구조에 대해 기존의 구성 방식과 새로운 세 가지 방식을 설명하고 이들에 대한 정성적인 분석을 통해 시스템의 성능을 예측해 본다.

입출력 시스템을 접근하는 버스 트랜잭션 중에서 시스템의 성능을 좌우하는 것은 디스크에 대한 접근이므로, 디스크 입출력 구조의 성능 향상을 통해 입출력 시스템의 성능 향상을 꾀하고자 한다. 디스크 입출력 모듈은 앞서 설명한 PCI 입출력 모듈(IOC)과의 인터페이스를 유지하여 디스크에 발생하는 읽기 및 쓰기 트랜잭션을 제어하는 역할을 담당한다.

3.1 PANDA 시스템의 기존 디스크 입출력 구조

PANDA 시스템에서 사용한 기존의 디스크 입출력 모듈은 그림 2에서의 SCSI 인터페이스에 해당한다. SCSI 인터페이스는 PCI 입출력 모듈에 연결되어 시스템에 디스크 등을 연결할 수 있도록 SCSI 버스와의 중재자 역할을 수행하는 버스간의 인터페이스이다. PANDA 시스템에서는 SCSI-2의 확장 모델인 Ultra SCSI를 사용하였다.

기존의 디스크 입출력 구조로 설계된 PANDA 시스템에서의 디스크 입출력은 다음과 같이 처리된다. 디스크 읽기나 쓰기 트랜잭션이 P6 지역 버스 상에서 관찰되면, 지역 노드의 디스크에 대한 트랜잭션의 경우에는 IOC가 이를 받아 바로 디스크에 해당 요구를 처리하고, 원격 노드의 디스크에 대한 트랜잭션의 경우에는 노드 제어기가 시스템 버스를 통해 원격 노드로 요구를 내보낸다. 요구를 받은 원격 노드는 자신의 디스크에서 처리를 마친 후, 처리 결과를 시스템 버스를 통해 요구한 노드로 다시 보내준다. 이 경우에 디스크 입출력 트랜잭션이 노드 내 디스크에 대한 요구일 경우에는 빠른 처리가 가능하지만, 원격 노드로 요구를 보내야 할 경우에는 시스템 버스를 오랜 시간동안 잡고 있어야 한다. 디스크 입출력 트랜잭션은 처리해야 할 데이타의 크기가 보통 2K 바이트를 초과하기 때문에 기타 트랜잭션에 비해 시스템 버스의 트래픽을 상당히 높이는 결과를 유발하여 전체 시스템의 성능을 현저히 저하시킨다. 시스템이 확장되거나 디스크 입출력 전송 데이타의 크기가 커질수록 이러한 현상은 더욱 크게 나타난다. 최근에는 이러한 문제점을 해결하기 위해 기존의 시스템 버스 외에 디스크 입출력 트랜잭션만을 처리하는 입출력 채널을 시스템에 추가함으로써 성능을 향상시킬 수 있는 방식이 연구되고 있다.

3.2 중앙집중형 디스크 입출력 구조

이 모델의 구조는 그림 3과 같다. PANDA 시스템의 기존 디스크 입출력 구조와 비교할 때 SCSI 인터페이스가 입출력 채널 인터페이스(I/O Channel Interface)로 바뀌었다. 입출력 채널 인터페이스가 SCSI 인터페이스와 가장 다른 점은 SCSI 프로토콜 칩을 PCI-to-SCI 칩으로 바꿈으로써 바로 디스크와 연결시키지 않고 입출력 채널과의 인터페이스를 유지한다는 것이다. 디스크는 각 노드에 분산되어 있지 않고 입출력 채널에 연결되어 모든 노드가 동일하게 디스크 접근을 수행하도록 한다. 디스크 접근 트랜잭션으로 인한 디스크 접근은 모두 중앙 디스크 제어기를 통해 관리된다. P6 지역 버스 상에 입출력 트랜잭션이 관찰되면, IOC는 입출력 채널 인터페이스를 통해 입출력 채널로 트랜잭션을 내보내준다. 입출력 채널을 통해 중앙 디스크 제어기로 전달된

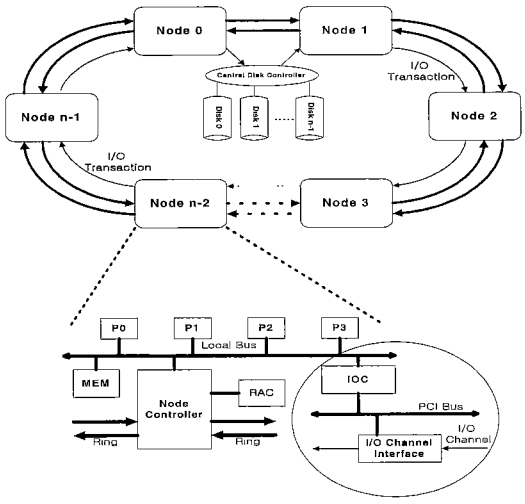

그림 3 중앙 집중형 디스크 입출력 구조

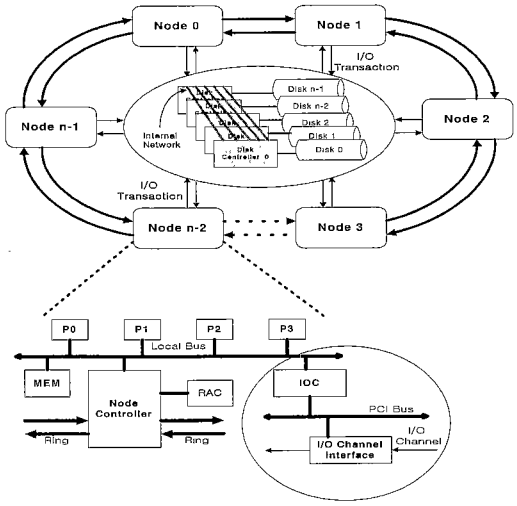

그림 4 병렬형 디스크 입출력 구조

트랜잭션은 해당 작업을 디스크에 수행하고, 처리 결과를 입출력 채널을 통해 요구 노드로 보내 준다.

이 구조는 시스템 버스에 생기는 디스크 입출력 데이타로 인한 트래픽을 없애 주므로 기존 구조에 비해 큰 성능향상 효과를 얻을 수 있다. 하지만, 모든 디스크 입출력 트랜잭션이 입출력 채널을 통해 외부의 중앙 디스크 제어기로 전달되므로 시스템이 확장되거나, 디스크 입출력 데이타의 전송 크기가 커지는 경우에는 입출력 채널의 트래픽이 증가하여 처리 속도가 느려지는 단점이 있다. 또한, 이러한 방식의 디스크 입출력 구조를 PANDA 시스템에 적용하기 위해서는 입출력 채널과의 인터페이스를 유지하고 디스크를 제어하는 중앙 디스크 제어기를 설계하여야 하며, 이에 따르는 새로운 디스크 관련 소프트웨어(ex. 디바이스 드라이버)를 제작해야 하는 오버헤드가 생기게 된다.

이 방식의 입출력 구조는 AV 25000 System[11]과 같은 최근의 다중 프로세서 시스템에서 사용되고 있다.

3.3 병렬형 디스크 입출력 구조

병렬형 디스크 입출력 구조는 그림 4와 같다. 3.2에서 설명한 중앙 집중형 방식의 디스크 입출력 구조에서는 하나의 집중된 디스크 제어기를 사용하기 때문에 이로 인한 병목 현상이 생길 수 있고, 하나의 입출력 명령을 수행하기 위해 입출력 채널을 한 번 순회해야 하므로 상당한 지연이 생기기 때문에 시스템의 확장성에 있어 문제점이 있다. 이러한 문제점을 해결하기 위해 제안하는 구조가 병렬형 디스크 입출력 구조이다.

하나의 집중된 제어기 대신 내부 연결망을 통해 연결되어 있는 여러 개의 디스크 제어기를 사용하여 하나의 디스크 노드를 구성하였다. 입출력 트랜잭션이 발생되어 IOC까지 전달되는 과정은 중앙집중형 입출력 구조와 동일하다. IOC로부터 입출력 트랜잭션이 입출력 채널 인터페이스로 전달되면, 입출력 채널 인터페이스는 디스크 노드를 구성하고 있는 하나의 디스크 제어기와 전용으로 통신을 한다. 디스크 제어기는 노드에서 요구한 디스크 입출력 트랜잭션이 자신이 담당하는 디스크 영역에 해당하면 바로 처리해서 응답을 보내주고, 다른 제어기가 담당하는 디스크 영역에 해당하는 요구인 경우에는 노드 내부 연결망을 통해 해당 제어기와의 통신을 하여 처리를 한 후 이에 대한 결과를 요구 노드로 보내준다. 각각의 디스크 제어기가 하나의 연산 노드에 전용으로 사용됨으로써 중앙집중형 방식에서 보이는 디스크 제어기의 부담을 여러 제어기로 분산시켜 병목 현상을 줄일 수 있다.

병렬형 디스크 입출력 구조는 입출력 채널의 병목 현상으로 인한 시스템의 성능 저하를 방지하고, 디스크 접근에 따르는 요구 채널의 수가 중앙집중형 디스크 입출력 구조에서의 '노드 개수 + 1'에서 2개(지역 디스크의 경우, 원격 디스크 접근의 경우에는 내부 연결망을 사용하는 오버헤드가 부가됨)로 줄어드는 장점이 있다. 그러므로, 노드 수가 증가되고 디스크 입출력 전송 데이타의 크기가 커지더라도 우수한 성능이 보장된다. 하지만, PANDA 시스템에 적용하기 위해서는 데이타 메쉬 구조[12]와 같은 디스크 노드의 설계에 따른 과도한 비용

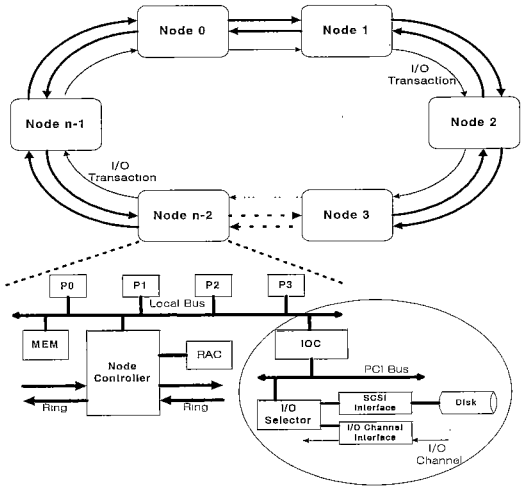

그림 5 분산형 디스크 입출력 구조

이 들게 되고, 디스크와 관련된 디바이스 드라이버 등의 소프트웨어를 새로 제작해야 하는 단점을 가지고 있다.

3.4 분산형 디스크 입출력 구조

분산형 디스크 입출력 구조는 그림 5에서 보는 바와 같이 기존의 PANDA 시스템의 입출력 구조와 동일하게 각각의 연산 노드가 디스크를 내부에 포함하고 있다.

P6 지역 버스에 입출력 트랜잭션이 관찰된 후부터 IOC까지 전달되는 과정은 중앙집중형 디스크 입출력 구조나 병렬형 디스크 입출력 구조와 동일하다. IOC에서 PCI 버스로 트랜잭션을 전달하면, 입출력 선택기(I/O Selector)는 트랜잭션을 보고 노드 내 디스크에 해당하는 영역인지, 원격 노드의 디스크에 해당하는 영역인지를 판단한다. 노드 내 디스크에 대한 입출력 요구의 경우에는 SCSI 인터페이스를 통해 곧바로 처리가 가능하고, 원격 노드 디스크에 대한 요구인 경우에는 입출력 채널 인터페이스를 통해 입출력 채널로 트랜잭션을 내보낸다. 이 트랜잭션은 입출력 채널을 통해 목적지 노드에 전달되어 디스크에 대한 처리가 이루어지며, 처리된 결과는 입출력 채널을 통해 요구한 노드로 되돌아온다.

이 모델에서는 입출력 요구가 노드 내 디스크에 해당하는 경우에는 노드 내에서 빠른 처리가 가능하므로 중앙집중형 구조나 병렬형 구조에 비해 우수한 성능을 보이는 장점이 있다. 또한, 중앙집중형 구조나 병렬형 구조에 비해 훨씬 적은 비용으로 기존의 PANDA 시스템에 적용할 수 있다. 새로운 디스크 제어기를 설계하지 않아도 되고, 별도의 디스크 관련 소프트웨어를 새롭게 제작하지 않아도 된다는 것이다. 하지만, 원격 노드로 요구를 보내야 하는 경우에는 입출력 채널을 한번 순회해야 하므로 중앙집중형 구조와 비슷한 성능을 보여 병렬형 구조에 비해서는 성능이 떨어지게 되는 단점이 있다.

4. 시스템 성능 모의실험 및 분석

4.1 모의실험 환경

본 논문에서는 위에서 제안한 중앙집중형, 병렬형, 분산형 디스크 입출력 구조를 적용한 다중 프로세서 시스템을 분석하기 위한 모의실험 도구로 확률 기반 시뮬레이터인 SES/Workbench를 사용한다. SES/Workbench는 큐잉 모델에 기반을 두는 모델링을 통해 시스템의 성능 및 무결성을 검사할 수 있는, 과학이나 공학 분야에서 많이 사용되고 있는 모의실험 도구이다[13].

이 실험에서 SES/Workbench의 입력 파라미터로는 TPC-A[14]를 사용한다. TPC-A 벤치마크는 집약적 갱신을 특징으로 하는 OLTP(On-Line Transaction Processing) 응용 데이타베이스 서비스 시스템의 환경에서 실제 궤적(trace)을 따라 추출한 통계자료이다. OLTP 환경은 다음과 같은 특징을 갖는다.

> 다중 온라인 터미널 세션
> 많은 디스크 입출력
> 적정한 시스템과 응용 프로그램의 실행 시간
> 트랜잭션의 무결성

위 특징에서 보듯이 많은 디스크 입출력을 특징으로 하기 때문에 입출력 시스템의 성능을 테스트하는데 있어서 비교적 정확한 결과를 보장하는 벤치마크이다.

4.2 모의실험 인자

모의실험에서의 각 노드는 최근 발표된 상용 제품의 성능에 근거하여 500 MHz의 클럭을 가지는 4개의 CPU가 100MHz의 지역 버스에 연결되어 있는 것으로 모델링하였다. 또한, 각 노드를 연결시켜 주는 시스템 버스는 IEEE 표준에 의거해 500 MHz로 동작하게 한다. 입출력의 경우 모든 입출력 관련 트랜잭션은 지역 버스에 연결된 PCI 버스를 통해 처리되도록 구현하였다. 디스크는 Ruemmler의 논문에 의거하여 모델링하였으며, 모델링한 디스크는 디스크 버퍼 캐쉬를 포함하여 read-ahead, write-behind를 제공한다[15]. 시스템 관련 인자의 내용은 표 1과 같다.

SES/Workbench는 트랜잭션의 종류를 결정하거나 캐쉬의 적중 여부, 프로세서간 라인의 공유 상황, 입출력에 관한 사항들을 입력된 확률을 통해 결정하므로, TPC-A에 의거하여 추출한 값들을 입력으로 주었다.

4.3 모의 실험 결과 및 분석

표 1 시스템 관련 공통 인자

프로세서 클럭	500 MHz
지역 버스 클럭	100 MHz
메모리 버스 클럭	100 MHz
캐쉬 라인 크기	32 Bytes
2차 캐쉬 크기	512 KB
3차 캐쉬 크기	32 MB
시스템 버스 클럭	500 MHz
PCI 버스 클럭	66 MHz
SCSI 버스 전송률	40 MB/s
입출력 채널 전송률	1 GB/s
디스크 컨트롤러 오버헤드	0.3 ms

디스크 입출력 시스템의 성능을 측정할 때, 비교 대상이 되는 수치로는 평균 응답 시간과 처리량(throughput)을 들 수 있다. 모의 실험을 통해 중앙집중형, 병렬형, 분산형의 3 가지 디스크 입출력 구조를 사용하는 각각의 시스템에 대해 평균 응답 시간과 처리량을 측정해 보았다. 분석 결과 평균 응답 시간과 처리량 사이에 큰 차이가 보이지 않으므로 이 절에서는 측정된 처리량을 통해 각 입출력 구조의 성능을 분석한다.

디스크 입출력 접근은 일반적으로 페이지 단위로 이루어지므로 전송 데이타 단위의 크기는 보통 4 KB에서 64 KB에 이른다[5]. 본 실험에서는 4 KB, 16 KB, 64 KB의 세 가지 전송 크기에 대해 성능 비교를 함으로써, 전송 데이타 크기에 따른 각각의 성능 특성을 알아본다.

4.3.1 4 KB 전송 크기를 사용할 때의 성능

4 KB 크기의 전송 단위를 사용하는 시스템에서의 입출력 구조의 성능을 비교하여 봄으로써 상대적으로 작은 전송 단위를 선택할 때의 각 구조의 성능을 알아본다. 그림 6에서 그림 9는 노드 수를 변화시킬 때 세 구조의 상대적 성능을 나타낸다.

각 그래프에서 세로축은 중앙집중형 구조의 처리량을 100으로 볼 때의 상대적인 처리량을 나타내고, 가로축은 원격 디스크에 대한 접근확률을 나타낸다. 원격 디스크는 분산형 구조의 경우 외부 노드가 포함하는 디스크를

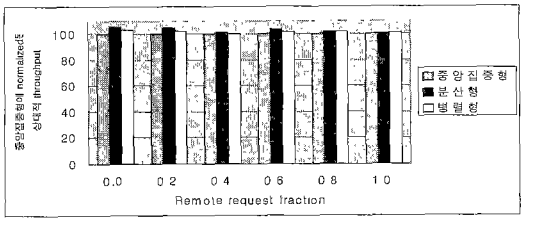

그림 6 노드 4개, 전송크기 4KB 경우 상대적 throughput

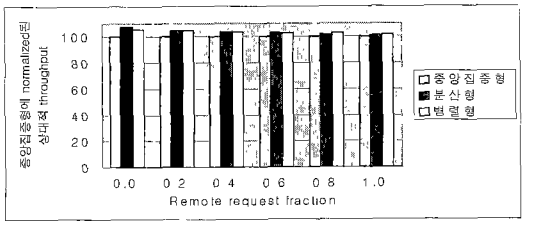

그림 7 노드 8개, 전송크기 4KB 경우 상대적 throughput

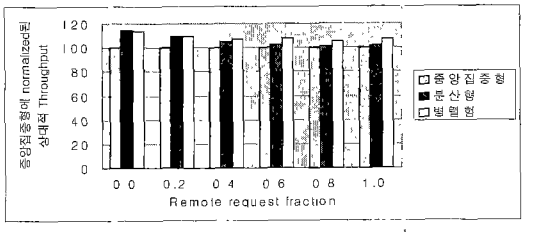

그림 8 노드 16개, 전송크기 4KB 경우 상대적 throughput

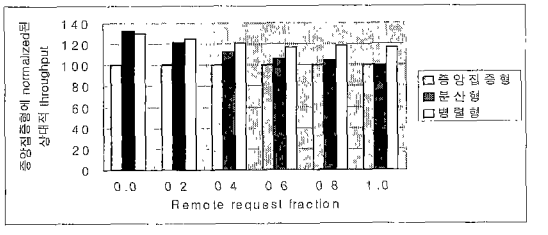

그림 9 노드 32개, 전송크기 4KB 경우 상대적 throughput

말하며, 병렬형 구조의 경우에는 노드와 직접적으로 통신하는 디스크 제어기 외의 다른 제어기가 관리하는 디스크를 말한다. 여러 원격 디스크 중에서 어느 것을 선택할 지는 같은 확률로 선택하였다.

각각의 구조의 성능은 노드 수와 원격 디스크 접근 확률에 따라 변한다. 그래프에서 보는 바와 같이 노드 수가 적거나 원격 디스크 접근 확률이 작을 경우에는 분산형 구조의 성능이 가장 우수하게 나타나며, 노드 수가 많아지거나 원격 디스크 접근 확률이 커지면 병렬형 구조의 성능이 우수하게 나타난다. 중앙집중형 구조의 경우에는 모든 경우에 있어서 분산형 구조나 병렬형 구조에 비해 좋지 않은 성능을 보이고 있다.

중앙집중형 구조에서는 지역 디스크 접근이나 원격 디스크 접근의 구별 없이 모든 입출력 트랜잭션은 입출력 채널 링을 한 번 순회해야 하므로 다른 두 구조에 비해 상대적으로 좋지 않은 성능을 보인다.

분산형 구조의 경우 지역 디스크를 접근할 시에는 노

드 내에서 처리가 가능하기 때문에 다른 두 구조에 비해 우수한 성능을 보이지만, 원격 디스크를 접근할 때에는 중앙집중형 구조와 마찬가지로 입출력 채널 링을 한 번 순회하여야 하므로 지역 디스크 접근에 비해 많은 시간을 필요로 한다. 노드의 수가 증가할수록 한 번 순회할 때 잡아야 하는 채널의 수가 많아지므로, 원격 디스크 접근에는 더 많은 오버헤드가 생긴다.

병렬형 구조의 경우에는 지역 디스크를 접근할 때에는 연결된 디스크 제어기와의 통신을 통해 처리되므로, 분산형 구조에 비해서는 많은 시간이 걸리지만, 중앙집중형 구조에 비해서는 우수한 성능을 보인다. 원격 디스크를 접근할 때에는 디스크 제어기에 요구를 하면, 디스크 제어기는 내부 연결망을 통해 해당 디스크를 제어하는 제어기와 통신을 하여 처리한다. 그러므로, 다른 두 구조에 비해 보다 빠르게 원격 디스크 접근을 처리할 수 있다. 이 구조에서는 노드의 수가 커지더라도 특별한 오버헤드의 증가 없이 원격 디스크 접근을 수행할 수 있으므로 노드 수가 많아지거나 원격 디스크 접근 확률이 커질수록 다른 두 구조에 비해 보다 우수한 성능을 보인다.

4.3.2 16 KB 전송 크기를 사용할 때의 성능

16 KB 크기의 전송 단위를 사용하는 시스템에서의 입출력 구조의 성능을 비교하여 봄으로써 중간 크기 정도의 전송 단위를 선택할 때의 각 구조의 성능을 실험한 결과가 그림 10에서 그림 13에 나타난다. 앞에서와 마찬가지로 노드 수를 변화시키면서 세 구조의 상대적 성능을 알아보았다.

4 KB 크기의 전송 단위를 사용하는 경우와 비슷하게 노드 수가 적고 원격 디스크 접근 확률이 낮은 경우에는 분산형 구조가 좋은 성능을 보이며, 노드 수가 많고 원격 접근 확률이 높은 경우에는 병렬형 구조가 우수한 성능을 보이고 있다.

16 KB 크기의 전송 단위를 사용하는 경우에는 4 KB 크기의 전송 단위를 사용하는 경우에 비교해서 각 환경에서의 구조상에 나타나는 성능의 차가 보다 커진다. 이는 전송 데이타의 크기가 커지면서 하나의 디스크 입출력 트랜잭션을 처리하는데 있어서 입출력 채널을 잡는 시간이 길어지므로 각 구조의 장단점이 보다 확실하게 나타나기 때문이다.

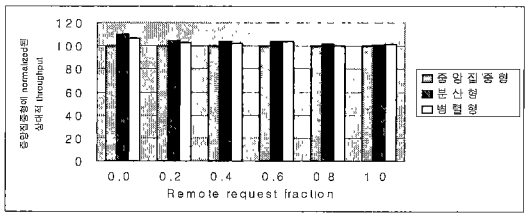

그림 10 노드 4개, 전송크기 16KB 경우 상대적 throughput

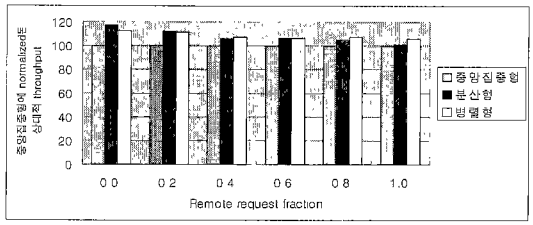

그림 11 노드 8개, 전송크기 16KB 경우 상대적 throughput

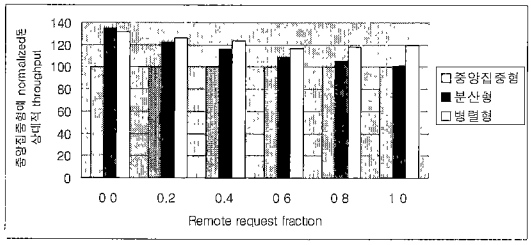

그림 12 노드 16개, 전송크기 16KB 경우 상대적 throughput

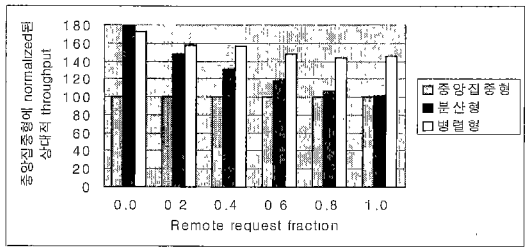

그림 13 노드 32개, 전송크기 16KB 경우 상대적 throughput

4.3.3 64KB 전송 크기를 사용할 때의 성능

그림 14에서 그림 17은 64KB 크기의 전송 단위를 사용하여 상대적으로 큰 전송 단위를 선택할 때의 각 구조의 성능 비교를 보여준다. 앞의 실험들과 마찬가지로 노드 수를 변화시키면서 세 구조의 상대적 성능을 비교해 보았다.

앞의 두 경우와 유사하게 노드 수가 적고 원격 디스크 접근 확률이 낮은 경우에는 분산형 구조가 좋은 성능을 보이며, 노드 수가 많고 원격 접근 확률이 높은 경우에는 병렬형 구조가 우수한 성능을 보인다.

64KB 크기의 전송 단위를 사용하는 경우에는 하나의 입출력 트랜잭션을 처리하는데 드는 비용이 앞의 두 경

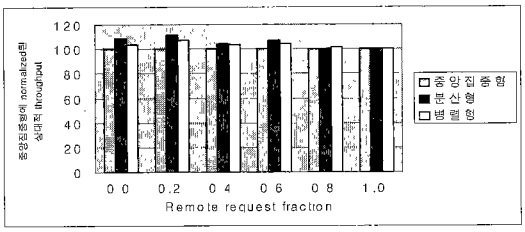

그림 14 노드 4개, 전송크기 64KB 경우 상대적 throughput

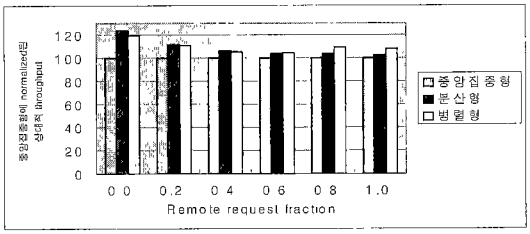

그림 15 노드 8개, 전송크기 64KB 경우 상대적 throughput

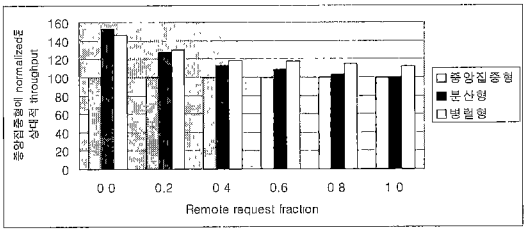

그림 16 노드 16개, 전송크기 64KB 경우 상대적 throughput

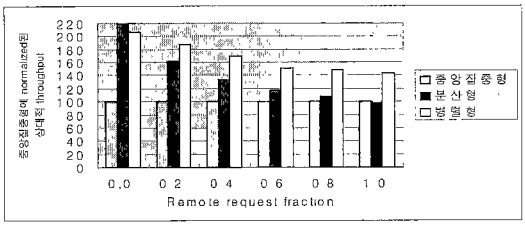

그림 17 노드 32개, 전송크기 64KB 경우 상대적 throughput

우에 비해 훨씬 크기 때문에, 시스템 환경에 따른 각 구조의 성능 차이가 가장 크게 나타난다.

위의 실험 결과들을 토대로 볼 때, 노드 수가 작거나 원격 디스크의 접근이 많이 일어나지 않는 경우에는 분산형 디스크 입출력 구조의 성능이 가장 뛰어나며, 노드 수가 많거나 원격 디스크의 접근이 자주 일어나는 경우에는 병렬형 디스크 입출력 구조의 성능이 가장 뛰어나다는 것을 알 수 있다. 중앙집중형 구조의 경우에는 각각의 시스템 환경에서 가장 좋지 못한 성능을 보인다. 이러한 특성은 데이타 전송 크기가 커질수록 보다 뚜렷하게 나타난다.

분산형 구조와 병렬형 구조는 각 환경에 따라 성능의 우위가 바뀌기 때문에 어느 구조가 확실하게 더 좋은 것이라고 말하기는 힘들다. 시스템 설계자는 시스템이 사용되는 환경을 잘 파악한 후 알맞은 구조를 선택하여야 할 것으로 보인다.

위의 각각의 디스크 입출력 구조를 시스템에 적용하는 데 있어서 반드시 생각되어야 하는 문제는 시스템의 설계 비용이다. 병렬형 구조의 경우에는 시스템이 커지고 원격 디스크 접근이 많아지면 가장 우수한 성능을 보이지만, 연산 노드를 설계하는 만큼의 비용이 디스크 제어기 설계에 필요하므로 다른 두 구조에 비해 큰 부담이 뒤따른다.

PANDA 시스템의 경우에는 기존의 구조에서 가장 적은 비용으로 확장할 수 있으며, 중앙집중형 구조에 비해서는 확실히 좋은 성능을 보이는 분산형 구조를 입출력 시스템에 적용하는 것이 좋을 것으로 판단된다.

5. 결론 및 향후 과제

버스 구조의 물리적 확장성 및 대역폭의 한계를 극복하기 위해 지점간 링크를 이용한 링 구조의 NUMA 시스템이 많이 보편화되었다. 하지만, 디스크 입출력 요구는 페이지 단위로 이루어지므로 지점간 링크를 오랜 시간 점유하여 지점간 링크의 트래픽을 더욱 높이는 요소로, 시스템 성능 향상에 있어서 걸림돌이 되고 있다.

본 논문에서는 디스크 입출력 요구로 인한 지점간 링크의 트래픽이 증가하는 것을 예방하기 위해 입출력 채널을 이용하여 디스크 입출력 요구를 처리하는 몇 가지 디스크 입출력 구조를 설계하여 시스템 성능을 향상시킬 수 있는 방법을 제시하였다.

중앙집중형, 병렬형, 분산형 디스크 입출력 구조는 노드의 수 및 원격 디스크 접근 확률, 디스크 입출력 전송 데이타 단위의 크기에 따라 각각 다른 성능 특성을 보였다. 분산형 디스크 입출력 구조는 노드의 수가 적고 원격 디스크 접근 확률이 낮은 경우 가장 우수한 성능을 보였으며, 병렬형 디스크 입출력 구조는 노드의 수가 많고 원격 디스크 접근 확률이 높은 경우 다른 두 구조에 비해 우수한 성능을 보였다.

전체 시스템의 특성 및 구조 설계에 드는 비용 등을 고려하여 비용 대비 성능이 가장 우수한 디스크 입출력 구조를 선택하는 것은 어려운 일이다. 본 논문에서 보인

성능 비교를 통해 알맞은 디스크 입출력 구조를 선택하고, 입출력 설계에 따르는 인터럽트 등의 기타 문제들을 처리하는 것은 앞으로의 과제가 될 것이다.

참 고 문 헌

[1] Kai Hwang and Zhiwei Xu, "Scalable parallel Computing : Technology, Architecture, Programming," McGraw-Hill, 1998.

[2] Per Stenstrom, Truman Joe and Anoop Gupta, "Comparative Performance Evaluation of Cache-Coherent NUMA and COMA Architectures," In the 19th Int'l Symp. on Computer Architecture, pp 80-91, 1992.

[3] Daniel Lenoski, Anoop Gupta et. "The Stanford Dash Multiprocessor," IEEE Computer, Mar 1992.

[4] Zhang, Z. and J. Torrellas, "Reducing Remote Conflict Misses: NUMA with Remote cache versus COMA," In Proc. of the 3rd IEEE Symp. on High Performance Computer Architecture (HPCA-3), pp 272-281, Feb 1997.

[5] J.L. Hennessy and D.A Patterson, "Computer Architecture: A Quantitative Approach," Second Edition, Morgan Kaufmann Publishers, 1996.

[6] 김형호, "지점간 링크를 이용한 스누핑 버스의 설계 및 성능 분석", 서울대학교 석사학위 논문, 1996.

[7] Sung Woo Chung, Seong Tae Jhang and Chu Shik Jhon, "PANDA: Ring-Based Multiprocessor System using New Snooping Protocol," In the Proceeding of ICPADS'98, pp 10-17, Dec 1998.

[8] 장병순, "PANDA 시스템에서 링 대역폭 확장을 위한 효율적인 방안", 서울대학교 석사학위 논문, 1999.

[9] N. M. Aboulenein, S. Gjessing, J. R. Goodman and P. J. Woest, "Hardware support for synchronization in the scalable coherent interface(SCI)," Technical Report CS-TR-92-1117, U of Wisconsin-Madison, Nov 1992.

[10] L. Barroso and M. Dubois, "The Performance of Cache-Coherent Ring-based Multiprocessors," In Proceedings of the 20th Int'l Symp. on Computer Architecture, pp.268-277, May 1993.

[11] Roy Clark, "SCI Interconnect Chipset and Adapter: Building Large Scale Enterprise Servers with Pentium II Xeon SHV Nodes," Jan 1999.

[12] Chia Chao, Robert English, David Jacobson, Bart Sears, Alexander Stepanov, and John Wilkes, "DataMesh architecture 1.0," HP Laboratories Technical Report, Jun 1992.

[13] "SES/Workbench Technical Reference, Scientific and Engineering Software," 1995.

[14] "Transaction Processing Performance Council, Overview of the TPC Benchmark A," In http://www.tpc.org/adetail.html.

[15] Chris Ruemmler, John Wilkes, "An Introduction to Disk Drive Modeling," IEEE Computer, Mar 1994.

김 철 홍

1998년 2월 서울대학교 컴퓨터공학과 공학사. 2000년 2월 서울대학교 대학원 컴퓨터공학부 석사. 2000년 3월 ~ 현재 서울대학교 대학원 컴퓨터공학부 박사과정. 관심분야는 컴퓨터 구조, 병렬 처리 시스템

김 명 주

1986년 서울대학교 컴퓨터공학과 공학사. 1988년 서울대학교 대학원 컴퓨터공학과 석사. 1993년 서울대학교 대학원 컴퓨터공학과 박사. 1993년 ~ 1995년 컴퓨터 신기술 공동연구소 자료실장. 1995년 ~ 현재 서울여자대학교 정보통신공학부 부교수, 전산교육원장. 관심분야는 병렬처리, 정보보안, 웹 기술

장 성 태

1986년 2월 서울대학교 전자계산기공학과 공학사. 1988년 2월 서울대학교 대학원 컴퓨터공학과 석사. 1994년 2월 서울대학교 대학원 컴퓨터공학과 박사. 1994년 3월 ~ 현재 수원대학교 전자계산학과 교수. 관심분야는 다중 프로세서컴퓨터구조, 병렬처리, 캐쉬 일관성 유지 프로토콜

엄 성 용

1985년 서울대학교 컴퓨터공학과 공학사. 1987년 서울대학교 대학원 컴퓨터공학과 석사. 1992년 서울대학교 대학원 컴퓨터공학과 박사. 1992년 ~ 1995년 컴퓨터 신기술 공동연구소 특별연구원. 1993년 ~ 1995년 Univ. of California, Irvine 한국과학재단 파견 Post Doc. 1996년 ~ 현재 서울여자대학교 정보통신공학부 부교수, 연구지원실장. 관심분야는 상위단계 합성, VLSI/CAD, 컴퓨터 그래픽스, IEEE 1394 응용 기술 개발 등

전 주 식

1975년 2월 서울대학교 응용수학과 학사. 1977년 2월 한국과학기술원 전산학과 석사. 1983년 2월 미국 Univ. of Utah 박사. 1983년 ~ 1985년 Univ. of Iowa 조교수. 1985년 ~ 현재 서울대학교 컴퓨터공학과 교수. 1994년 ~ 현재 컴퓨터신기술공동연구소 소장. 관심분야는 컴퓨터 구조, 병렬처리, VLSI/CAD