VIA기반의 멀티미디어 클러스트 시스템의 데이터를 동반한 메시지 전송 기법

*박시용[0] 박성호 정기동
부 산 대 학 교 *멀 티 미 디 어 협 동 과 정, 전 자 계 산 학 과
{sypark, shpark, kdchung}@cs.pusan.ac.kr

Message and Data Passing Method in VIA-based Multimedia Cluster System

*Si-Yong Park[0] Sung-Ho Park Ki-Dong Chung
Dept. of Computer Science, Pusan National University
*Dept. of Multimedia co-operation course, Pusan National University

요 약

본 논문에서는 클러스트의 프로토콜 스택 오버헤드를 줄인 VIA(Virtual Interface Architecture)를 이용한 멀티미디어 클러스트 시스템을 제안하고 제안된 VIA 기반의 멀티미디어 클러스트 시스템의 각 서버들간의 통신 및 데이터 전송을 위한 메시지들을 효율적으로 전송하기 위해서 이중 버퍼를 사용하는 방법을 고려하였다. 그리고 프로토콜 스택 오버헤드를 줄이기 위해서 간단한 프로토콜만을 제공하는 VIA의 명세에서는 제공하지 않는 재전송 메커니즘을 제안하여 시스템의 안정성을 높였다. 본 논문에서 제안한 시스템의 실험 결과 이중 버퍼를 이용한 데이터 전송의 경우 순차적인 전송 기법보다 데이터의 양이 많을수록 더 좋은 성능을 보였다. 그리고 적절한 데이터 전송 횟수와 버퍼량을 구하기 위한 실험 결과 총전송량을 10Mbyte로 고정시키고 버퍼의 양과 전송 횟수를 변화시킨 결과 1Mbyte를 10번 전송할 때 보다 0.1Mbyte를 100번 전송할 경우 네트웍 대역폭은 2배 이상 높게 나타났다.

1. 서론

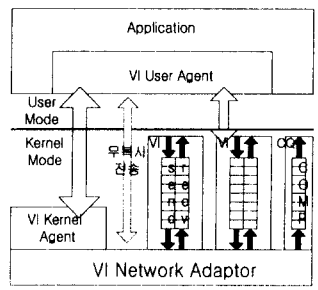
하드웨어의 급속한 발전으로 인하여 우리는 과거에 비하여 월등한 연산능력을 가진 컴퓨터를 가지게 되었다. 그리고 네트웍에 있어서는 100Mbps 내지는 1Gbps급 이상의 고성능 네트웍들이 계속 등장하였다. 성능이 향상된 PC나 워크스테이션을 고성능 네트웍을 통해서 하나의 연산 단위로 통합한 더 강력한 연산 능력을 가지게 하는 클러스트 시스템이 등장하게 되었다. 일반적으로 멀티미디어 시스템을 구성하기 위해서는 많은 데이터 저장 공간을 필요로 하며, 멀티미디어 데이터의 특성상 많은 양의 I/O가 발생한다. 본 논문에서는 I/O를 분산시키고 멀티미디어 데이터의 병렬성을 높이기 위해서 클러스트 구조의 병렬 멀티미디어 시스템을 제안하였다. 그러나 클러스트 시스템에서의 문제점으로는 다단계 프로토콜 스택으로 구성하는데 있다[1]. 그렇기 때문에 네트웍 하드웨어보다는 소프트웨어에서 오는 오버헤드가 네트웍에 더 큰 영향을 미친다. 이런 문제점들을 줄이기 위해서 클러스트 구조에 적합한 많은 통신 프로토콜들에 관련된 연구가 진행되었다. 연구의 대표적인 예로는 Active Message(AM), Cornell 대학의 U-Net, VMMC(Virtual Memory-Mapped Communication), Virtual Interface Architecture(VIA)등이 있다[2][3]. 특히 U-Net을 기반으로 하여 제안되어진 VIA의 경우에는 사용자수준의 무복사 전송을 제공하여 통신 프로토콜에서 소프트웨어 오버헤드를 줄여준다[3]. 본 논문에서는 VIA 기반의 멀티미디어 클러스트 시스템을 제안하고 VIA를 병렬 멀티미디어 시스템에 적용시키기 위하여 대용량의 데이터를 효율적으로 전송시키기 위한 방법을 고려하였다. 본 논문의 구성을 살펴보면 2장에서는 VIA기반의 병렬 멀티미디어 시스템을 구성하는데 필요한 관련 연구에 대해서 언급하고 3장에서는 본 논문에서 제안하는 시스템의 구조와 시스템에서 필요로 하는 메시지 전송 기법에 대해서 살펴보기로 한다. 4장에서는 제안한 메시지 전송기법에 따른 성능평가를 하고 5장에서 결론 및 향후과제를 말한다.

2. 관련연구

2.1 VIA(Virtual Interface Architwcture)

VIA는 커널 버퍼로의 복사를 생략하여 사용자 영역의 버퍼에서 직접 NIC(Network Interface Card)으로 전송하는.사용자 수준의 무복사 전송 기법을 제공한다. VIA는 Compaq, Intel, Microsoft등의 회사가 참여하여 제안하였고 소프트웨어와 하드웨어 모두에서 구현이 가능하다[3]. VIA의 소프트웨어로의 구현은 Berkely의 Linuc기반 M-VIA와 Intel의 Windows NT기반의 VIA가 있다.

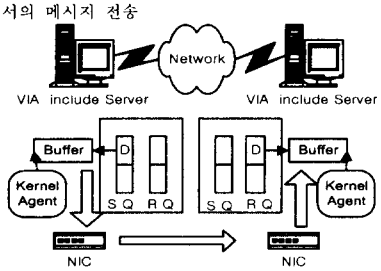
• VIA 구조



[그림 1] VIA 구조

VIA는 User Agent, VI(Virtual Interface), Kernel Agent, Complete Queue의 4부분으로 나눌 수 있다. 사용자가 메시지를 전송하기 위해서는 먼저 메시지의 디스크립트를 만든 후 그 메시지를 VI의 Send Queue에 삽입한다. 만약 Data가 있을 경우 디스크립트의 데이터 세그먼터에 데이터와 관련된 정보를 기록한다. Kernel Agent는 Send Queue의 Descriptor에 따라서 Message를 만들어 VI Network Adaptor로 전송한다. 그리고 디스크립트의 완료 여부를 Kernel Agent가 Complete Queue를 조사하여 알아낸다. 각각의 VI는 송신을 위한 Send Queue와 수신을 위한 Receive Queue로 구성된다. Kernel Agent는 실제 데이터 전송과 주소 변환 등을 수행한다[4].

• VIA에서의 메시지 전송



[그림 2] VIA 메시지 전송 구조

Send Queue와 Receive Queue에 들어가는 디스크립트의 내용은 실제 Descriptor의 포인터를 저장하고 있다. 전송측에서 Send 디스크립트를 Send Queue에 삽입하면 Kernel Agent가 메시지를 VI Network Adaptor를 통하여 전송한다. 수신측 Kernel Agent는 전송받은 메시지를 Receive Queue에 미리 삽입되어 있는 Receive 디스크립터가 지시하는 메모리 영역에 저장한다. 통신을 위해서는 VI들 간에 전송을 위한 Send 디스크립트와 수신을 위한 Receive 디스크립터가 서로 동기화가 되어야 한다. 이러한 메커니즘으로 인하여 VI들은 수신측 VI와 송신측 VI간의 1대1통신을 한다[4].

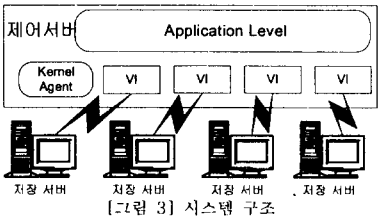
2.2 클러스트 미디어 서버

클러스트 미디어 서버는 저장 서버와 제어 서버의 위치에 따라서 저장 서버와 제어서버가 하나의 노드에 존재하는 수평 구조와 서로 다른 노드에 존재하는 2계층 구조로 나눌 수 있고 미디어 객체의 배치 정책에 따라서 독립 구조와 분산 구조로 나눌 수 있다[5].

3. 병렬 멀티미디어 시스템에서의 메시지 전송 기법

3.1. 시스템 구조

본 논문에서 제안하는 시스템의 구조는 제어 서버와 저장 서버가 서로 다른 노드에 존재하고 데이터를 분산하여 배치하는 2계층 분산 구조이다. 그리고 제어 서버는 저장 서버와의 통신을 위해서 저장 서버의 수만큼 VI를 가지고 있고 각각의 저장 서버는 제어 서버와의 통신을 위해서 하나의 VI를 가진다[그림 3].



[그림 3] 시스템 구조

• 제어 서버 구조

제어 서버는 Kernel Level에서 동작하는 VI와 VI Kernel Agent로 구성되어 있고 User Level에서는 Request를 수신하고 스케쥴링하는 Request Manager Group과 VI의 생성,제거 및 오류를 인식하여 사용자 프로세스에게 알려주는 역할을 하는 VI Manager로 구성되어 있다. User Agent를 기반으로 User Level이 구성된다.

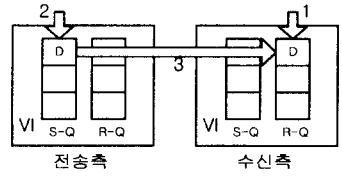
• 저장 서버 구조

저장 서버는 Kernel Level에 제어 서버와 동일하게 VI와 VI Kernel Agent로 구성되고 User Level에는 VI를 관리하는 VI Manager와 Request를 스케쥴하는 Request Reader, Request를 처리하는 Job Scheduler 그리고 데이터의 서비스를 위한 Transmission Scheduler로 구성되어 있다.

3.2. 데이터를 동반한 효율적인 메시지 전송

메시지 전송을 위해서는 모든 VI는 초기 메시지 수신을 위해서 Receive Queue에 적당량의 Receive 디스크립트를 삽입해놓는다. 그리고 모든 VI는 하나의 Send 디스크립트를 수신 후 항상 Receive 디스크립트를 자동으로 삽입한다.

• VIA에서의 일반적인 메시지 전송

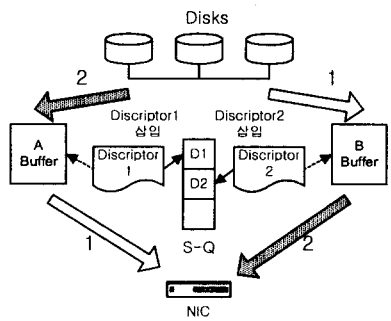


[그림 4] 데이터 없는 전송

전송측에서 제어 메세지를 포함하고 있는 Send 디스크립트를 Send Queue에 삽입하고 미리 준비된 수신측의 Receive Queue의 Receive 디스크립트를 이용하여 제어 메시지를 수신한다. 그리고 제어 메시지의 내용에 따라서 처리해준다. 데이터가 없는 전송의 경우에는 순차적인 메시지 전송을 사용한다[그림 4].

• 데이터가 동반한 메시지 전송 기법

병렬 멀티미디어 시스템에서는 다른 클러스트 시스템에 비하여 많은 양의 데이터 전송이 일어난다. 그러므로 본 논문에서는 데이터의 디스크 I/O 입출력으로 인한 전송 지연 시간을 줄이기 위해서 두개의 버퍼를 이용한 전송을 사용한다.



[그림 5] 데이터가 있는 메시지 전송

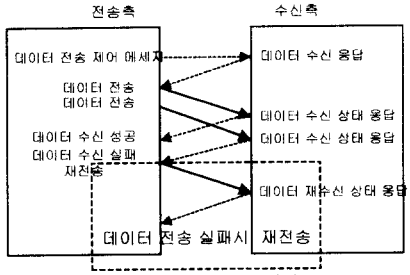
Send 디스크립트의 Data 세그먼트 영역을 설정한 후 전송측의 Send Queue에 디스크립트를 삽입하고 메시지를 전송. 수신측의 미리 삽입된 Receive 디스크립트를 통하여 메시지를 수신한다. 그리고 각각의 설정된 Buffer 영역을 통하여 데이터를 전송을 시작한다. 두개의 버퍼 영역을 이용하여 연속적인 Send 디스크립트 삽입을 통하여 데이터를 전송한다[그림 5].

◆ A 버퍼가 데이터를 NIC(Network Interface Card)으로 전송하는 사이에 B 버퍼는 디스크로부터 데이터를 전송 받고 디스크립터를 Send Queue에 삽입한다.

◆ B 버퍼가 데이터를 NIC(Network Interface Card)으로 전송하는 사이에 A 버퍼는 디스크로부터 데이터를 전송 받고 디스크립트를 Send Queue에 삽입한다.

3.3 재전송 메커니즘

VIA에서는 프로토콜 스택을 단순화하기 위해서 따로 재전송 메커니즘을 두지 않는다. 그래서 본 논문에서는 데이터의 전송이 많이 사용됨으로 전송 중 데이터의 손실이 일어났을 때 이를 복구해주기 방법으로 재전송 기법을 사용한다[그림 6].



[그림 6] 재전송 기법

◆ 전송측에서 데이터 전송을 위해서 Send 디스크립트를 삽입할 때 전송측의 Receive Queue에도 Ack 메시지 수신을 위한 Receive 디스크립트를 하나 삽입한다.

◆ 수신측에서 데이트를 수신한 후 그 데이터의 손실 및 변질 여부를 확인하고 Ack 메시지를 전송한다.

◆ 전송측에서 Ack 메시지를 수신한다.
  Success : 데이터 전송 성공
  Fail : 데이터 전송 실패, 데이터 재전송

◆ 전송측에서 정해진 시간 동안 Ack 메시지를 전송받지 못할 경우 Ack 메시지 수신을 위한 메시지를 수신측에 전송한다.

4. 실험 및 성능 평가

4.1. 실험 환경

본 논문에서 제안한 시스템은 멀티미디어 데이터의 특성상 빈번한 데이터의 전송이 발생한다. 이에 본 논문에서 제안한 이중 버퍼를 이용한 데이터 있는 전송 기법에 관하여 성능 평가를 한다. 그리고 데이터를 전송할 때 전체 시스템의 성능을 고려한 적절한 데이터 전송 사이즈와 전송 횟수를 선택하기 위한 실험을 실시하였다. 기본적으로 VIA와 다른 프로토콜간의 비교는 M-VIA에서의 결과를 가정하였다[표 1][6].

[표 1] VIA와 TCP 비교

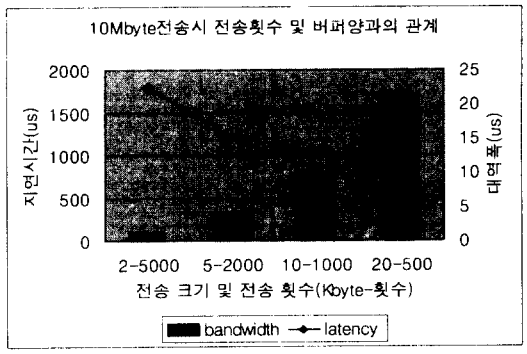
NIC종류	Protocol	Latency(us)	Bandwidth(MB/s)
Packet Engines GNIC II	TCP	59	31
Packet Engines GNIC II	M-VIA	19	60
Tulip East Ethernet	TCP	65	11.4
Tulip East Ethernet	M-VIA	23	11.9

[표 2] 실험 환경

제어서버 CPU	Intel Pentium II 300MHz Dual
저장서버 CPU	Intel Pentium 166MHz
Network Interface Card	Intel Ether Express Pro 100
Switching Hub	OmmiSTACK Hub
운영체제	Linux Kernel 2.2.12
VIA	M-VIA 1.0

하나의 저장 서버와 두개의 저장 서버로 실험을 위한 환경을 구축하였다.

4.2. 데이터를 동반한 메시지 전송 비교



[그림 7] 데이터 전송 시 버퍼량과 전송횟수 관계

전송 데이터의 양은 고정시키고 전송 횟수와 버퍼의 양만을 조절하였을 때는 버퍼의 양을 증가시키고 전송 횟수를 감소시킬수록 더 좋은 결과를 보이고 있다. 전송 횟수가 많을수록 더욱 더 빈번하게 디스크를 접근하므로 전송횟수가 작을수록 VIA기반의 멀티미디어 클러스트 시스템에 더 적합하다고 할 수 있다.[그림 7].

5. 결론 및 향후 과제

본 논문에서는 프로토콜 스택을 단순화하여 네트웍상에서 소프트웨어 과부하를 줄인 VIA를 기반으로 한 병렬 멀티미디어 시스템을 제안하였고 VIA의 명세에서는 제공하지 않는 메시지 전송 기법을 제안하였다. 특히 멀티미디어를 지원하기 위한 시스템에서는 많은 양의 데이터 전송이 발생하기 때문에 이중 버퍼를 이용한 메시지 전송 기법을 제안하였다. 이중 버퍼를 사용하였을 때 데이터 전송 성능은 단일 버퍼를 이용한 전송의 경우보다 월등히 좋은 것으로 나타났다. 빈번한 데이터 전송으로 인한 시스템의 성능을 높이기 위해서 전송 버퍼 크기와 전송 횟수와의 관계를 고려한 실험의 결과 버퍼의 크기를 증가시키고 전송횟수를 줄일 경우 디스크 접근 횟수를 감소시키므로 더 좋은 성능을 보이고 있다. 그리고 VIA에서는 제공하지 않는 재전송 메커니즘을 제안하여 VIA기반의 멀티미디어 시스템의 안정성을 높여주었다. 앞으로 연구가 진행되어야 할 사항으로는 동시 사용자 수의 증가로 인한 VI 의 제어에 대한 연구와 좀 더 효율적인 재전송 기법에 관한 연구가 진행되어야 할 것이다.

6. 참고 문헌

[1] Jonathan Kay and Joseph Pasquale, "Profiling and Reducing Processing Overheads in TCP/IP," IEEE Transactions on Networking, Vol.4, 1996

[2] Anindya Basu, Vineet Buch, Werner Vogels, Thorsten von Eicken, "U-Net: A User-Level Network Interface for Parallel and Distributed Computing," SOSP, Dec. 1995

[3] Dave Dunning, Greg Regnier, Gary McAlpine, Don Cameron, Bill Shubert, Frank Berry, Anne Marie Merritt, Ed Gronke, Chris Dodd, "The Virtual Interface Architecture," IEEE Micro, Mar.-Apr. 1998

[4] Compaq, Intel, Microsoft, "Virtual Interface Architecture Specification Version 1.0," 1997

[5] 박시용, 석창규, 박성호, 김영주, 정기동, "Multimedia Data를 위한 병렬 파일 시스템", 정보과학회 봄 학술발표 논문집(A), 2000

[6]M-VIA Document, http://www.nersc.gov/research/FTP/via