2000년 한국정보처리학회 추계 학술발표논문집 제 7 권 제 2 호

공간 데이터의 병렬성을 고려한 VIA 기반의 클러스트 시스템 설계 및 구현

박시용*, 박성호**, 정기동**
*부산대학교 멀티미디어 협동과정
**부산대학교 전자계산학과
e-mail : {sypark, shpark, kdchung}@melon.cs.pusan.ac.kr

The Design and Implementation of VIA-based Cluster System for spatial data's parallelism

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

요 약

본 논문에서는 공간데이터의 병렬성을 고려한 클러스트 시스템을 제안하였다. 클러스트 시스템의 큰 단점인 다단계 프로토콜 스택에서 오는 메시지 전송 부하를 줄인 VIA(Virtual Interface Architecture)를 기반으로 클러스트 시스템을 구성하고 저장 서버들간에는 공간데이터의 지역성에 기반하여 데이터를 배치하며 저장 서버들 내에서는 공간 데이터의 병렬성을 고려하여 EPR(Enhanced Parallel R-tree)로 데이터를 배치하였다. 위의 클러스트 시스템을 기반으로 적절한 전송 데이터 크기와 전송 횟수를 구하기 위한 실험을 실시하였다.

1. 서론

지리 정보 시스템(Geographical Information system)에서 사용되는 공간 데이터는 공간 객체의 위치, 속성 정보를 표현하는 방대한 데이터로 구성된다. 이러한 이와 같은 대용량 데이터는 디스크의 입출력 과부하를 발생시켜 전체 시스템의 성능을 저하시키는 가장 큰 요인들 중의 하나이다. 이러한 지리 정보 시스템의 입출력 과부하를 해결하기 위하여 병렬 입출력 시스템에 대한 연구가 활발히 진행되고 있다 [1,2]. 입출력을 병렬화 시키기 위해서 본 논문에서는 입출력의 부하를 분산시킬 수 있는 병렬 입출력 시스템을 제안한다. 우리가 제안하는 병렬 입출력 시스템의 기본 구조는 고성능의 네트웍을 기반으로 여러 대의 컴퓨터가 단일 연산 및 입출력을 담당하는 클러스트 시스템을 채택하였다. 본 논문에서는 이러한 클러스트 시스템은 사용자 요구를 분산하여 처리할 수 있는 장점을 가진는 반면 클러스트 노드들간의 전송 메시지처리 과부하를 발생시키는 단점을 가지고 있다[3]. 다단계 프로토콜 스택의 처리과정에서 오는 전송 메시지 처리하는 과정에서 발생하는 시스템의 전체적인 성능 저하를 해결하기 위해서 SAN(System Area Network)기반의 프로토콜 중의 하나인 VIA(Virtual Interface Architecture)로 클러스트 시스템 을 구성 하였다.

*본 연구는 학술 진흥 재단 중점 연구소 지원사업 연구비에 의해 수행되었음.

본 논문의 2 장에서는 우리가 제안한 시스템 을 구성하기 위한 관련 연구를 소개하고 3 장에서는 제안하는 시스템의 구조 및 클러스트 시스템의 특성을 고려한 공 간 데이터의 배치 방법에 대해서 소개한다. 4 장에서는 제안한 시스템의 성능 평가를 보여주고 끝으로 5 장에서 결론 및 향후 과제를 말한다.

2. 관련 연구

2.1. 공간 데이터 특성

공간 데이터의 기본적인 특징은 아래와 같다[4].

- 대용량의 데이터이다.
- 영역 질의는 특성상 인접한 영역의 공간 데이트를 동시에 요구한다.
- 인접한 영역의 데이터는 디스크상에 물리적으로 인접하게 저장하여 디스크의 입출력 횟수를 줄일 수 있다.

2.2. 클러스트 시스템

최근 가격대비 마이크로프로세서의 고성능화, 메모리의 대용량화, 그리고 네트웍 기술의 발전 등으로 저가형 컴퓨터 시스템을 이용하여 고성능 컴퓨터를 구축할 수 있는 클러스트링 기술이 크게 발전하고 있다.

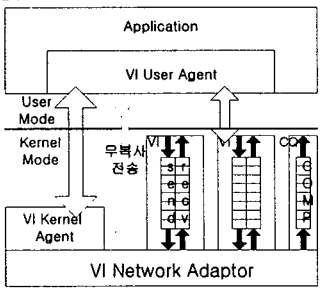
일반적으로 클러스트를 구성하는 하드웨어 요소로는 다수의 컴퓨터와 네트웍 인터페이스 카드, 네트웍 스위치이고 소프트웨어 요소로는 통신 프로토콜, 클러스트 미들웨어, 운

영체제 등으로 구성된다. 클러스트 시스템은 서버의 물리적 위치와 객체의 저장 배치방법에 따라서 분류할 수 있다. 제어 서버와 저장 서버가 물리적으로 같은 노드상에 존재한다면 수평 구조이고 다른 노드상에 존재한다면 2 계층 구조이다. 그리고 저장 객체가 같은 노드상에 존재하면 독립 구조이고 여러 노드에 분산 저장되어 있다면 분산 구조이다[5].

2.3. VIA(Virtual Interface Architecture)

VIA 는 커널 버퍼로의 복사를 생략하여 사용자 영역의 버퍼에서 직접 NIC(Network Interface Card)으로 전송하는 사용자 수준의 무복사 기법을 제공한다. 그리고 VIA 는 Compaq, Intel, Microsoft 등의 회사가 참여하여 제안하였고 소프트웨어와 하드웨어 모두에서 구현이 가능하다.

- VIA 구조

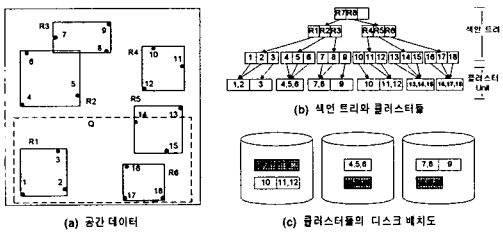


[그림 1] VIA 구조

VIA 는 User Agent, VI(Virtual Interface), Kernel Agent, Complete Queue 의 4 부분으로 나눌 수 있다. 사용자가 메시지를 전송하기 위해서는 먼저 메시지의 디스크립트를 만든 후 그 메시지를 VI 의 Send Queue 에 삽입한다. 만약 Data 가 있을 경우 디스크립터의 데이터 세그멘트에 데이터와 관련된 정보를 기록한다. Kernel Agent 는 Send Queue 의 Descriptor 에 따라서 메시지를 만들어 VI NIC 으로 전송한다. 그리고 디스크립트의 완료 여부를 Kernel Agent 가 Complete Queue 를 조사하여 알아낸다. 각각의 VI 는 송신을 위한 Send Queue 와 수신을 위한 Receive Queue 로 구성된다. Kernel Agent 는 실제 데이터의 전송과 주소 변환 등을 수행한다[그림 1][6].

2.4. Enhanced R-Tree

GIS 의 질의는 인접한 객체들을 동시에 읽도록 요구하는 경우가 대부분이다. 만약 인접한 객체들이 옆에 위치하도록 객체를 정렬하여 단말 노드에 저장하고, 또한 물리적인 블록의 크기를 넘지않는 한 하나의 블록에 여러 개의 객체를 저장한다. 데이터를 입력하기 전에 Hilbert Space Filling Curve 을 이용하여 인접도를 기준으로 정렬한다. 정렬된 데이터를 입력할 때 일반 R-tree 와 달리 단말 노드부터 만들어 나간다. 단말노드부터 입력이 이루어지는 이유는 인접도에 의해 정렬된 데이터를 물리적으로 저장할 때 실제로 인접하게 저장하기 위해서이다[그림 2].



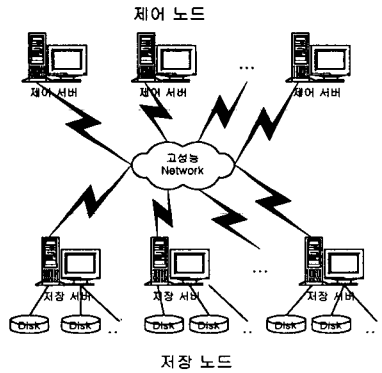
[그림 2] EPR(Enhanced Parallel R-tree)

하나의 노드에 저장되는 객체들은 물리적으로 연속된 블록에 저장이 되고, 하나의 블록 내에 그 블록의 경계를 넘지 않는 적당한 개수의 객체가 입력된다. 인접한 객체는 같은 질의에 의해 추출될 확률이 높기 때문에 한 블록 내에 인접한 객체들을 여러 개 넣게 되면 한번의 디스크 접근으로 여러 개의 객체들을 읽어 올 수 있기 때문이다. 하나의 단말 노드에 속한 객체들은 같은 디스크에 저장이 되고 각 단말 노드들은 Round Robin 방식으로 디스크들에 스트라이핑 된다. Round Robin 방식을 채택한 이유는 디스크의 부하 균형이 좋아지고, 하나의 단말 노드가 하나의 디스크에 저장되고, 하나의 질의는 보통 여러 개의 단말 노드를 포함하므로 디스크 이용율도 향상되기 때문이다[7].

3. 공간 데이터의 병렬성과 다중 사용자를 고려한 클러스트 시스템

3.1. 클러스트 시스템 구조

본 논문에서는 다중 사용자를 위해서 2 계층 분산 구조를 제안한다. 2 계층 분산 구조는 수평 구조에 비해 부하 균등에 있어서 유리하다. 그리고 다중 사용자를 위해서 지역성에 기반하여 공간 데이터를 저장 서버에 분산하여 배치하는 분산 구조를 제안한다. 그리고 각 노드간의 메시지 과부하를 줄이기 위해서 노드간 통신 프로토콜은 VIA(Virtual Interface Architecture)를 사용한다[그림 3].

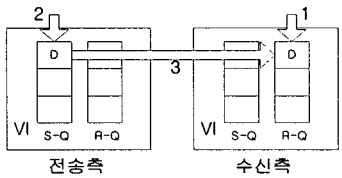


[그림 3] 클러스트 시스템 구조

3.1.1. 각 노드들 간의 메시지 전송 방법

모든 VI 는 초기 메시지 수신을 위해서 Receive Queue 에 적당량의 Receive 디스크립트를 삽입해놓는다. 그리고 모든 VI 는 하나의 Send 디스크립트를 수신 후 항상 Receive 디스크립트를 자동으로 삽입한다.

- VIA 에서의 일반적인 메시지 전송



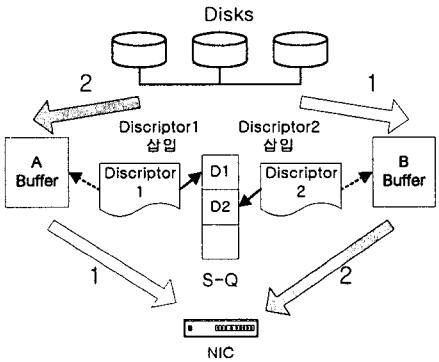
[그림 4] 데이터 없는 전송

전송측에서 제어 메세지를 포함하고 있는 Send 디스크립트를 Send Queue 에 삽입하고 미리 준비된 수신측의 Receive Queue 의 Receive 디스크립트를 이용하여 제어 메

시지를 수신한다. 그리고 제어 메시지의 내용에 따라서 처리해준다. 데이터가 없는 전송의 경우에는 순차적인 메시지 전송을 사용한다[그림 4].

• 데이터가 동반한 메시지 전송 기법

공간 데이터를 위한 클러스트 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

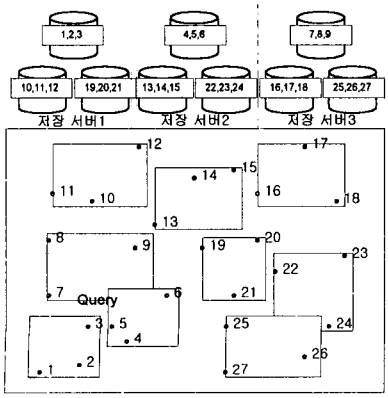
Send 디스크립트의 Data 세그먼트 영역을 설정한 후 전송측의 Send Queue 에 디스크립트를 삽입하고 메시지를 전송. 수신측의 미리 삽입된 Receive 디스크립트를 통하여 메시지를 수신한다. 그리고 각각의 설정된 Buffer 영역을 통하여 데이터를 전송을 시작한다.

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

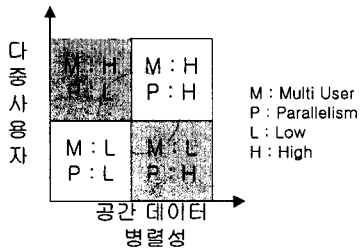
3.2. 지역성에 기반한 클러스트 시스템내의 공간 데이터 배치 기법

3.2.1. 데이터의 병렬성을 고려한 배치 방법



[그림 6] 병렬성을 고려한 배치 방법

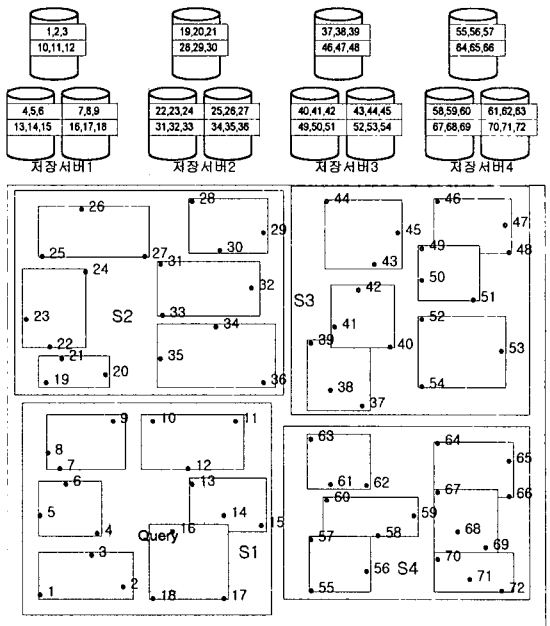
[그림 6]과 같이 공간 데이터의 병렬성을 고려하여 배치할 경우 전체 클러스트 시스템에 한 사용자의 요구를 각각의 저장 서버가 분담하여 처리하기에 적합하도록 공간데이터를 배치하였다. 인접한 데이터를 클러스트 팩킹 단위로 나누고 다시 인접한 클러스트 팩킹 단위를 서로 다른 저장 서버에 배치한다. 그 결과 [그림 6]와 같은 질의가 들어오면 모든 저장 서버에 사용자 요구를 분산하여 처리할 수 있다. 그러나 [그림 6]과 같은 처리를 할 경우, 다음 사용자의 요구가 하나라도 중복된 디스크의 자원을 요구할 경우에는 앞선 사용자의 요구가 끝나기를 기다려야 한다. 이와 같은 결과로 볼 때 클러스트 시스템에서의 공간 데이터의 병렬성과 다중 사용자의 요구처리 관계는 어느 정도 반비례의 관계에 있다고 할 수 있다[그림 7].



[그림 7] 다중 사용자와 공간 데이터 병렬성 관계

3.2.2. 병렬성과 다중사용자를 고려한 배치 방법

다중 사용자 요구와 공간 데이터의 병렬성을 둘 다 수용하기 위해서 저장서버 단위로 지역을 분할 한 후 그 지역 내에서는 EPR(Enhanced Parallel R-tree)를 이용하여 공간 데이터의 병렬성을 높인다[그림 6].



[그림 8] 지역성에 기반한 공간 데이터 배치 방법

[그림 8]과 같은 영역 질의가 요청된 경우 S1 영역은 저장서버 1 의 디스크들에 EPR(Enhanced Parallel R-tree)로 저장되어 있는 3 개의 디스크들에서 병렬적으로 데이터를 읽어온다. 그리고 그 다음 요구가 S1 지역이 아닌 다른 지역에 영역 질의가 요청된다면 더 이상의 지연 없이 바로 서비스가 될 수 있다.

4. 실험 및 성능 평가

4.1. 실험 환경

본 논문에서 제안한 시스템은 공간 데이터의 특성상 빈번한 데이터의 전송이 발생한다. 데이터를 전송할 때 전체 시스템의 성능을 고려한 적절한 데이터 전송 사이즈와 전송 횟수를 선택하기 위한 실험을 실시하였다. 기본적으로 VIA 와 다른 프로토콜간의 비교는 M-VIA 에서의 결과를 가정하였다[표 1][8].

[표 1] VIA 와 TCP 비교

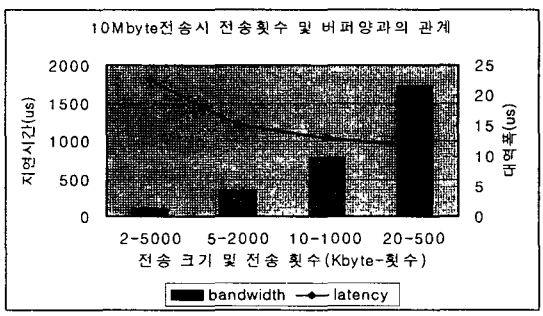
NIC 종류	Protocol	Latency(us)	Bandwidth(MB/s)
Packet Engines GNIC II	TCP	59	31
Packet Engines GNIC II	M-VIA	19	60
Tulip Fast Ethernet	TCP	65	11.4
Tulip Fast Ethernet	M-VIA	23	11.9

[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

본 논문의 실험을 위해서 하나의 제어 서버와 두개의 저장서버로 클러스트 시스템을 구축하였다.

4.2. 데이터를 동반한 메시지 전송 비교



[그림 9] 데이터 전송 시 버퍼량과 전송횟수 관계

전송 데이터의 양은 고정시키고 전송 횟수와 버퍼의 양만을 조절하였을 때는 버퍼의 양은 증가시키고 전송 횟수를 감소시킬수록 더 좋은 결과를 보이고 있다. 빈번한 디스크 접근으로 인하여 버퍼의 양이 작을수록 더 좋지 않은 성능을 보이고 있다. 그리고 전송 횟수가 증가함에 따라서 네트웍 대역폭은 약 23Mbyte 에서 약 14Mbyte 정도로 줄어들고 있다[그림 10].

5. 결론 및 향후 연구과제

본 논문에서는 공간 데이터의 병렬성과 클러스트 시스템의 부하 분산을 위한 시스템을 제안하였다. 부하 분산을 고려한 클러스트 시스템은 제어서버와 저장 서버가 각각 서로 다른 노드에 존재하는 2 계층 구조를 가지고 각각의 서버들간의 효율적인 메시지 전송을 위해서 SAN 기반의 VIA 를 통신 프로토콜로 선택하였다. 그리고 공간 데이터의 병렬성을 위해서 각각의 저장 서버에는 EPR 을 이용하여 공간 데이터를 배치하였고 사용자의 요구 분산을 위해서 공간 데이터의 지역성에 기반하여 데이터를 분산 저장하였다. 그리고 VIA 기반의 클러스트 시스템 내에서의 효율적인 메시지 및 데이터 전송을 위해서 이중 버퍼를 이용하여 메시지를 전송하였다. 그리고 적절한 데이터 전송 크기와 전송 횟수를 선택하기 위한 실험의 결과 전송 데이터의 크기가 증가함에 따라서 디스크 접근 횟수가 줄어들기 때문에 더 좋은 성능을 보이므로 공간데이터 블록을 물리적으로 인접하게 배치할 경우 더 좋은 성능을 나타낼 것이다.

향후에는 본 시스템의 안정성을 위한 재전송 메커니즘과 지역성에 기반한 공간 데이터의 병렬성과 다중사용자를 고려한 공간 데이터 배치 기법에 따른 연구에 계속될 것이다.

참고문헌

[1] Ibrahim Kamel and Christos Faloutsos, "Parallel R-trees," in Proc. Of ACM SIGMOD Conference, pp. 195-204, 1992.

[2] 김기홍, 차상균, "공간 색인의 다중 검색 경로 문제와 클러스트 저장 구조," 한국정보과학회 학술 발표 논문집, Vol. 23(I), pp. 39-42, 1996

[3] Jonathan Kay and Joseph Rasquale, "Profiling and Reducing Processing Overheads in TCP/IP," IEEE Transactions on Networking, vol.4, 1996

[4] 이창호, "대규모 공간 데이터를 위한 병렬 파일 시스템의 설계," 부산대학교 컴퓨터공학과 석사학위논문, 1999

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

[6] 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

[7] 이춘근, 김정원, 김영주, 정기동, "EPR : 지리정보 시스템을 위한 향상된 병렬 R-tree 색인 기법," 한국 정보처리학회 정보처리논문집, 제 6 권 제 9 호, 1999

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