EPR : 지리 정보 시스템을 위한 향상된 병렬 R-tree 색인 기법

이 춘 근† · 김 정 원† · 김 영 주†† · 정 기 동†††

요 약

본 논문은 병렬 입출력과 효율적인 디스크 접근을 이용하여 입출력 성능을 높임으로써 지리 정보 시스템의 질의 처리 성능을 향상시키는 것을 목적으로 한다. 동시에 접근할 가능성이 높은 인접한 공간 데이터를 디스크의 논리적 블록 단위로 패킹하여 하나 또는 연속적인 논리적 블록으로 클러스터링하면 한번의 디스크 접근으로 많은 공간 데이터를 읽을 수 있어 질의 처리에 따른 디스크 접근 횟수와 디스크 접근 오버 헤드를 줄임으로써 입출력 시간을 줄일 수 있다.

본 논문에서는 기존 Parallel R-tree 기법의 병렬 입출력 기법과 패킹 기반 클러스터링 기법을 결합하여 효율적인 입출력을 지원하는 EPR(Enhanced Parallel R-tee) 색인 기법을 제안한다. EPR 기법의 주요 특징은 다음과 같다. 첫째, 공간 데이터를 Hilbert space filling curve를 이용하여 인접도에 따라 정렬하여 패킹함으로써 상향식으로 R-tree를 생성한다. 둘째, 정렬된 공간 데이터를 패킹하여 하나 또는 연속적인 논리적 블록에 저장하는 패킹 기반 클러스터링을 통해 공간 데이터 클러스터를 구성한다. 셋째, 색인 및 공간 데이터 클러스터를 round-robin 스트라이핑 방식을 통해 다중 디스크에 분산 배치한다. EPR 기법과 기존 PR 기법의 성능을 비교한 결과, 공간 질의 처리 속도가 30% 이상 향상되었으며, 특히 논리적 블록의 크기가 클수록, 공간 데이터의 크기가 작을수록 질의 처리 성능이 향상되는 결과를 보였다.

EPR : Enhanced Parallel R-tree Indexing Method for Geographic Information System

Choon-Kun Yi† · Jeung-Won Kim† · Young-Ju Kim†† · Ki-Dong Chung†††

ABSTRACT

Our research purpose in this paper is to improve the performance of query processing in GIS(Geographic Information System) by enhancing the I/O performance exploiting parallel I/O and efficient disk access . By packing adjacent spatial data , which are very likely to be referenced concurrently, into one block or continuous disk blocks, the number of disk accesses and the disk access overhead for query processing can be decreased, and this eventually leads to the I/O time decrease. So, in this paper, we proposes EPR(Enhanced Parallel R-tree) indexing method which integrates the parallel I/O method of the previous Parallel R-tree method and a packing-based clustering method. The major characteristics of EPR method are as follows. First, EPR method arranges spatial data in the increasing order of proximity by using Hilbert space filling curve, and builds a packed R-tree by bottom-up manner. Second, with packing-based clustering in which arranged spatial data are clustered into continuous disk blocks, EPR method generates spatial data clusters. Third, EPR method distributes EPR index nodes and spatial data clusters on multiple disks through round-robin striping. Experimental results show that EPR method achieves up to 30% or more gains over PR method in query processing speed. In particular, the larger the size of disk blocks is and the smaller the size of spatial data objects is, the better the performance of query processing by EPR method is.

※ 본 연구는 1998년도 한국 학술진흥재단 대학부설연구소과제 연구비에 의하여 연구되었음.
† 준 회 원 : 부산대학교 대학원 전자계산학과
†† 정 회 원 : 부산대학교 전자계산학과
††† 종신회원 : 부산대학교 전자계산학과 교수
논문접수 : 1999년 6월 7일, 심사완료 : 1999년 8월 10일

1. 서 론

컴퓨터 하드웨어 및 소프트웨어의 발달로 정교하고 화려한 색상의 영상을 실시간으로 제공해 주는 비행 시뮬레이션이나 가상현실 등이 가능하게 되었다. 이러한 응용 프로그램들은 화면을 구성하거나 필요한 지리 정보를 추출하기 위해 방대한 양의 공간 데이터를 실시간으로 요구한다[1, 2, 3, 4, 5]. 이를 위해서는 지리 정보를 서비스 해주는 지리 정보 시스템(Geographical Information System : GIS)의 입출력 시스템에 많은 부하가 걸려 전체 시스템의 성능에 영향을 미치게 된다. 또한, 대용량 지리 정보 시스템의 경우 다수의 동시 사용자를 서비스 하기 위해 입출력 시스템에 상당한 부하가 걸리게 된다. 향후, 지리 정보 시스템이 보편화, 다양화, 전문화를 통해 더 많은 양의 공간 데이터를 관리하게 되면 입출력 시스템에 과중한 부하가 걸리게 될 것이다. 따라서, 입출력 시스템의 성능이 지리 정보 시스템의 전체 성능을 좌우하는 요인이 되므로 입출력 성능 향상을 위한 입출력 병렬화가 필수적으로 요구된다[3, 4, 6, 7, 8].

공간 데이터 객체는 다차원 속성을 가지며, 비공간 속성(인구, 이름, 용도 등)과 공간 속성(위치 등)으로 표현되고, 객체들을 검색하기 위해 교차(intersection), 포함(contain), 근접(proximity) 등의 공간 연산이 공간 데이터베이스에서 요구되는 등의 일반 데이터와는 상이한 특성을 가진다[9, 10, 11]. 그리고, 일반적으로 지리 정보 시스템은 병렬 입출력을 위해 다중 디스크를 사용하며, 단일 질의에 대해 다수의 디스크로부터 입출력을 병렬로 수행하여 응답 시간을 단축할 뿐만 아니라 입출력 성능을 향상시킨다[3, 4, 6, 7, 8]. 따라서, 지리 정보 시스템의 입출력 병렬화를 위해서는 공간 데이터 및 저장 장치의 특성을 적절하게 고려하여야 하며, 효율적인 입출력을 위한 색인화 및 다중 디스크에서의 분산 데이터 배치 등을 고려한 병렬 공간 색인 기법이 요구된다.

공간 데이터의 전체 크기는 대용량인데 반해 개별 공간 데이터의 크기는 수 바이트에서 수백 바이트 정도에 불과하여 일반적으로 디스크의 논리적 블록 크기에 비해 매우 작다[1, 2, 5, 13]. 따라서, 각각의 공간 데이터를 논리적 블록 단위로 저장하면 디스크 공간을 낭비하게 된다. 또한, 지리 정보 시스템의 질의는 공간상에서 인접한 공간 데이터를 동시에 요구하는 영역 질의가 대부분을 차지한다[10]. 이러한 특성을 고려해 볼 때 하나의 공간 데이터를 하나의 논리적 블록에 저장하기 보다는 공간상에서 인접한 여러 개의 공간 데이터를 하나의 논리적인 블록에 패킹하여 저장할 경우 디스크 접근 횟수와 디스크 접근에 따른 오버헤드를 줄임으로써 입출력 속도를 향상시킬 뿐만 아니라 입출력 효율을 높일 수 있다. 또한, 색인 트리(index tree)에서 단말 노드의 모든 슬롯을 패킹하여 채운다면 색인 트리의 높이와 검색해야 할 노드의 수가 감소하여 검색 속도도 높일 수 있다[1, 2, 5, 13]. 따라서, 공간 데이터의 효율적인 병렬 입출력을 위해서는 공간 데이터를 다수의 디스크에 분산 배치하기 위한 병렬 색인 기법과 다차원의 공간 데이터를 1차원으로 정렬하여 공간상에서 인접한 공간 데이터를 디스크 상에서 인접하게 저장함으로써 효율적인 입출력을 지원하는 패킹(packing) 기법과 클러스터링(clustering) 기법이 적용되어야 한다.

본 논문에서는 공간 데이터에 대한 병렬 색인 기법인 Parallel R-tree(PR)[6] 기법을 기반으로 입출력 효율성을 높이기 위해 색인 데이터 및 공간 데이터를 디스크의 논리적 블록 단위로 패킹하여 저장하는 패킹 기반 클러스터링 기법을 적용하는 EPR(Enhanced Parallel R-tree) 기법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 EPR기법과 관련된 연구들을 살펴보고, 3장에서는 EPR 기법의 연구 배경 및 기본 개념, PR 기법과의 비교, 그리고 알고리즘 등을 설명한다. 4장에서는 EPR 기법의 성능을 PR 기법과 비교하여 보여주고, 5장에서는 결론 및 향후 연구 방향을 제시한다.

2. 관련 연구

지리 정보 시스템은 대용량의 공간 데이터를 처리하고 다수의 동시 사용자들에게 실시간으로 서비스하기 위해 병렬 입출력을 지원하여야 하며, 이를 위해서는 효율적인 입출력을 위한 공간 색인 기법과 다중 디스크를 이용한 데이터 저장 기법이 고려되어야 한다.

초기 공간 색인 기법인 R-tree[9]는 다중 경로 문제[10, 11, 12, 14], 낮은 노드 활용률[1, 2, 5, 13] 등의 문제점을 내포하고 있었다. 다중 경로 문제를 개선하기 위해 R*-tree가 제안되었으며, 공간 데이터를 R-tree에 삽입할 때와 노드를 분할할 때 노드의 MBR(Minimum Bounding Rectangle : 객체를 포함하는 최소 크기의

사각형)이 최소가 되도록, 그리고 MBR의 둘레가 최소가 되도록 하여 20~30%의 성능 향상을 가져왔다[11].

노드 활용률에 있어서 R-tree가 50%를 보장하던 것을 100%의 노드 활용률을 보장하고, 노드의 MBR 사이에서의 겹침(overlapping)을 감소시켜 검색 효율을 높이는 패킹(packing) 알고리즘이 제안되었다[1, 2, 5, 13]. 패킹 알고리즘은 공간 데이터 특성이 정적이라는 가정하에 공간 데이터들을 1차원으로 정렬하여 단말 노드부터 노드의 슬롯을 채우면서 상향식으로R-tree를 생성함으로써 대부분의 노드들을 꽉 채워 노드 활용률을 높이고, 인접한 공간 데이터를 정적으로 클러스터링하여 다중 경로 문제를 완화 시켰다. 대표적인 패킹 알고리즘으로는 Packing R-tree[2]를 예로 들 수 있고, 이 패킹 알고리즘은 공간 데이터 정렬 방식으로 Hilbert Space Filling Curve[15]을 사용한다.

다중 디스크를 이용한 공간 데이터 저장 기법은 지리 정보 시스템에 관련된 다른 연구들에 비해 비교적 최근에 연구되기 시작한 분야로서 Parallel R-tree(PR)[6] 기법이 대표적이다. PR 기법은 일반 R-tree 구조에 다중 디스크에서의 분산 배치를 고려한 병렬 공간 색인 기법으로, 새로운 공간 데이터가 색인 트리의 단말 노드에 입력될 때 현재 그 노드에 존재하는 공간 데이터와의 인접도를 계산하여 가장 인접도가 떨어진 공간 데이터가 저장된 디스크에 저장하고, 이로써 영역 질의가 요구되면 인접한 공간 데이터를 다수의 디스크에서 병렬로 접근하여 입출력 속도를 향상시킨 색인 기법이다.

현재까지 제안된 병렬 공간 색인 기법은 "비효율적인 공간 데이터 저장 문제"를 가지고 있으며, 구체적인 문제점은 다음과 같다.

첫째, 개별 공간 데이터를 크기에 상관없이 하나의 논리적인 블록에 저장한다. 일반적으로 공간 데이터가 논리적인 블록의 크기보다 작아 여러 개의 공간 데이터를 하나의 논리적 블록에 저장할 수 있는데도 하나의 공간 데이터만을 저장함으로써 디스크 공간을 낭비한다. 하나의 논리적인 블록에 여러 개의 공간 데이터를 패킹하여 저장한다면 디스크 공간 효율을 높일 수 있다.

둘째, 영역 질의 기반의 입출력 병렬성만을 강조하고 있다. 인접한 공간 데이터를 다른 디스크에 분산 저장함으로써 일정한 영역의 공간 데이터를 동시에 요구하는 영역 질의를 처리할 때에 각 디스크로부터 공간 데이터를 병렬로 접근하여 입출력 속도를 향상시키는데 초점을 둠으로써 인접한 공간 데이터를 동일한 디스크에 클러스터링하여 저장함으로써 얻을 수 있는 입출력 효율성을 고려하지 않고 있다.

셋째, 저장 장치의 특성을 적절히 고려하지 못하고 있다. 병렬 공간 색인 기법은 좁은 영역 질의가 요구되더라도 모든 디스크에 대해 병렬적으로 접근하여 처리함으로써 디스크 접근 횟수가 증가할 뿐만 아니라 논리적인 블록의 연속적인 배치를 전제하지 않는 경우에 디스크 접근에 따른 오버헤드, 즉 디스크 탐색 시간과 회전 지연 시간 등이 증가하여 응답시간이 길어지고 입출력 효율도 낮아지게 된다.

이상에서 살펴본 병렬 공간 색인 기법의 문제점은 공간 데이터를 논리적인 디스크 블록에 일대일로 저장한다는 묵시적인 가정과 입출력 병렬성만을 강조함으로써 발생한다. 하나의 논리적인 블록에 여러 개의 인접한 공간 데이터를 패킹하여 클러스터링하고 다중 디스크에 적절히 분산 배치한다면 효과적인 병렬 공간 색인 기법을 얻을 수 있다.

3. EPR(Enhanced Parallel R-tree)

3.1 기본 개념

지리 정보 시스템의 질의는 대부분이 영역 질의이며, 영역 내에 포함되는 공간 데이터를 동시에 요구한다[1, 2, 5, 13]. 따라서, 공간 데이터를 인접도를 바탕으로 다중 디스크에 적절히 분산 배치함으로써 병렬 입출력을 통해 응답 시간을 단축할 수 있고, 동시에 접근할 가능성이 높은 색인 데이터와 공간 데이터를 각각 패킹을 통하여 클러스터링 함으로써 디스크 입출력 효율성을 높일 수 있다.

공간 데이터는 형태를 표현하기 위한 좌표와 MBR로 구성이 된다. MBR은 두개의 좌표 값을 나타내는 4개의 정수, 즉 16바이트로 고정되어 있고, 좌표 값은 표현하는 공간 데이터의 모양 복잡도나 표현하려는 정확도에 따라 크기가 일정하지 않고 작아 크기가 1KB를 넘는 공간 데이터가 많지 않다. 일반적으로 디스크의 논리적인 블록 크기는 4KB, 또는 8KB로 설정되므로 1KB를 넘지 않는 공간 데이터를 최소한 4개에서 8개까지 패킹하여 저장할 수 있다. 또한, 같은 영역 질의에서 추출될 확률이 높은 인접 공간 데이터를 인접도를 바탕으로 정렬하여 같은 논리적인 블록에 저장하거

나 물리적으로 연속적인 논리적인 블록들에 저장함으로써 디스크 접근 횟수를 감소시키고 디스크 접근 오버헤드, 즉 디스크 탐색 시간 및 회전 지연 시간 등을 줄임으로써 응답 시간을 줄일 뿐만 아니라 입출력 효율을 높일 수 있다.

그리고, 색인 데이터의 경우 단말 노드에서부터 패킹하여 저장함으로써 공간 색인 트리의 높이와 전체 노드 수를 줄여 검색 속도를 향상시킬 수 있다.

따라서, EPR 기법은 기본적으로 색인 데이터와 공간 데이터에 대해 각각 인접도를 바탕으로 하나의 논리적 블록 또는 연속적인 논리적 블록에 패킹하여 저장하는 패킹 기반 클러스터링(packing based clustering)을 통해 색인 노드 클러스터와 공간 데이터 클러스터를 생성하고, 클러스터 수준에서 다중 디스크에 분산 배치함으로써 디스크 접근 횟수를 줄이고 병렬 입출력으로 응답 시간을 단축한다.

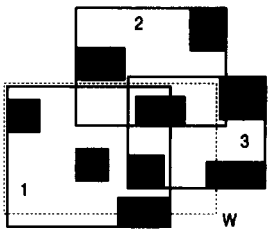
EPR 기법의 기본적인 알고리즘은 다음과 같다.

첫째, 공간 데이터를 Hilbert Space Filling Curve를 이용하여 인접도를 바탕으로 정렬한다. 그리고, 정렬된 공간 데이터를 패킹 기반 클러스터링을 통해 단말 노드부터 생성해가면서 공간 색인 트리를 구성한다. 패킹 기반 클러스터링은 노드 단위 및 클러스터 단위의 디스크 접근을 가능하게 하여 간단한 점 질의에서 복잡한 영역 질의까지 다양한 형태의 디스크 접근에 따른 오버헤드를 줄일 수 있다.

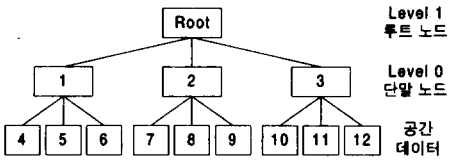
둘째, 하나의 클러스터에 속하는 공간 데이터, 즉, 하나의 단말 노드에 속하는 공간 데이터는 디스크상에서 물리적으로 연속 저장하기 위해 패킹하여 하나 또는 연속적인 논리적인 블록에 저장한다. 그리고, 병렬 입출력을 위해 색인 노드 및 공간 데이터 클러스터를 다중 디스크에 Round-Robin 스트라이핑 방식으로 분산 배치하고, 하나의 단말 노드에 속한 공간 데이터는 같은 디스크에 저장한다. 각 색인 노드들은 이미 공간 데이터를 인접도에 따라 정렬하여 패킹함으로써 구성되었기 때문에 Round-Robin 방식으로 분산 배치하는 것은 병렬성을 높이기 위해 공간 데이터간의 인접도를 계산하는 오버헤드를 줄일 수 있다. 그리고, 하나의 단말 노드에 속하는 공간 데이터를 하나의 디스크에 연속적으로 저장하는 것은 적은 양의 데이터를 요구하는 좁은 영역에 대한 질의에 대해 다수의 디스크에서의 병렬 입출력을 위한 동기화 오버헤드가 병렬 입출력에 따른 이점보다 크기 때문이다.

3.2 EPR과 PR의 비교

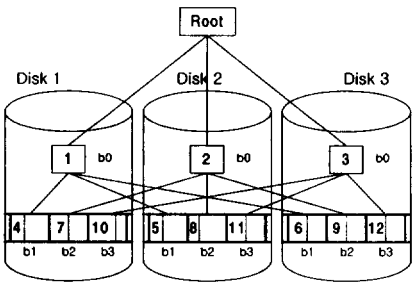
(그림 1)은 간단한 예를 통해 EPR의 디스크 배치 방식을 Parallel R-tree 기법과 비교하고 있으며, 공간 데이터에 대한 패킹 기반 클러스터링 효과를 보여주고 있다.



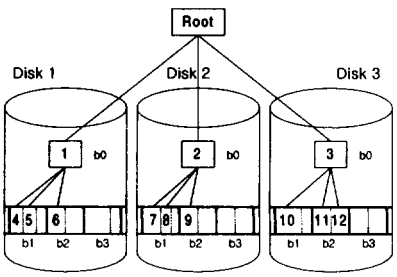
(a) 공간 데이터



(b) 높이가 2인 R-tree



(c) Parallel R-tree 기법에 따른 디스크 배치



(d) EPR 기법에 따른 디스크 배치

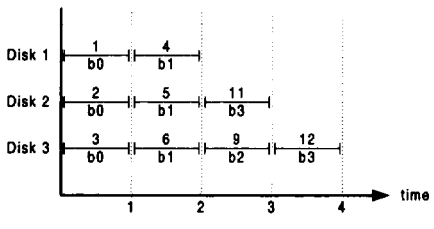
(그림 1) PR과 EPR의 색인 구조 비교

(그림 1)의 (a)는 하나의 공간 데이터 예로서 공간 색인을 이차원 평면으로 나타낸 것이고, (b)는 (a)에서 주어진 공간 색인을 트리 형식으로 표현한 R-tree이다. (그림 1)의 (c)는 (b)의 R-tree를 Parallel R-tree 기법으로 디스크에 저장한 예로서 공간 데이터의 크기에 무관하게 하나의 공간 데이터를 하나의 논리적인 블록에 저장하고, 공간상에 인접한 공간 데이터들을 다른 디스크에 배치한다. (그림 1)의 (d)는 (b)의 R-tree를 EPR 기법으로 디스크에 저장한 예로서 공간 데이터를 논리적인 블록 단위로 패킹하여 저장하고, 인접한 공간 데이터를 하나의 디스크에 연속적으로 배치한다.

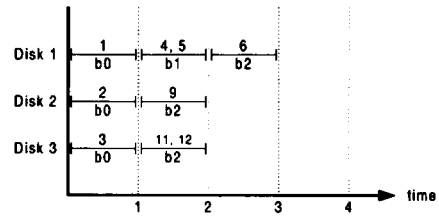
EPR과 PR 사이의 효율성을 비교하기 위해 간단한 질의 처리 과정을 살펴본다.

먼저, 전체 공간 데이터를 요구하는 질의에 대해서는 색인 데이터와 공간 데이터를 모두 읽기 위하여 PR 의 경우 각 디스크에서 4번의 디스크 블록 접근이 요구되고, EPR의 경우에는 3번의 디스크 블록 접근이 요구되어 EPR 기법이 효율적이다. 또한, "영역 1"에 대한 질의에 대해 공간 데이터만을 읽기 위하여 PR의 경우 각 디스크에서 1번의 디스크 블록 접근이 요구되는 반면에 EPR의 경우는 "디스크 1"에서 2번의 연속적인 디스크 블록 접근이 요구되어 디스크 접근 횟수 관점에서 PR 기법이 효율적이지만, 이러한 경우 접근하는 데이터 양이 적을 때는 병렬 입출력을 위한 디스크 동기화 오버헤드가 병렬 입출력에 따른 이점보다 커서 PR 기법이 비효율적일 수 있다.

(그림 1)의 (a)에서 점선에 해당하는 "질의 W"가 주어지는 경우 공간 데이터 4, 5, 6, 9, 11, 그리고 12가 요구되며, PR과 EPR에 따른 디스크 접근 패턴은 (그림 2)와 같다. PR의 경우 EPR에 비해 디스크 접근 횟수가 많고 비연속인 디스크 블록을 접근하기 때문에 비효율적이다.



(a) PR의 디스크 접근 패턴



(b) EPR의 디스크 접근 패턴

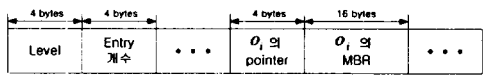
(그림 2) 질의 W를 위한 디스크 접근 패턴

3.3 노드 구조

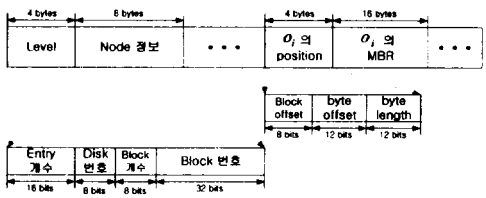
일반 R-tree는 비단말 노드와 단말 노드의 구조가 같다. 그러나, EPR에서는 패킹 기반 클러스터링과 다중 디스크에서의 분산 배치를 위해 단말 노드와 비단말 노드의 구조를 분리한다.

3.3.1 비단말 노드

비단말 노드는 (그림 3)의 (a)와 같이 일반 R-tree의 노드 구조와 동일하다. 엔트리 개수는 비단말 노드에 저장되는 공간 데이터 개수를 나타낸다. o_i의 *pointer*와 o_i의 MBR은 각각 비단말 노드에 저장되어 있는 i번째 자식 노드에 대한 포인터와 MBR을 나타낸다.



(a) 비단말 노드 구조



(b) 단말 노드 구조

(그림 3)ERP의 노드 구조

3.3.2 단말 노드

EPR 기법은 여러 개의 공간 데이터를 패킹하여 하나 또는 연속적인 논리적 블록에 저장하고 병렬 입출력을 위해 클러스터 단위로 다중 디스크에서 분산 배치함으로 단말 노드는 공간 데이터의 배치 정보, 즉 공간 데이터가 저장되어 있는 디스크 및 논리적 블록 등에 대한 위치 정보를 가지고 있어야 한다.

EPR에서의 단말 노드의 구조는 (그림 3)의 (b)와 같으며, 단말 노드를 구성하는 각 필드를 간단히 살펴보면 다음과 같다.

① **Level** : 공간 색인 트리에서 단말 노드의 레벨을 나타낸다.

② **Entry 개수** : 단말 노드에서 저장 정보를 저장하고 있는 공간 데이터 객체 수를 나타낸다.

③ **Disk 번호** : 단말 노드에 속한 공간 데이터를 저장하는 디스크 번호로서, 하나의 단말 노드에 속한 공간 데이터는 동일한 디스크에 저장한다.

④ **Block 개수** : 단말 노드에 속한 공간 데이터를 저장하는 논리적 블록의 개수를 나타내며, 클러스터 단위의 디스크 접근 시에 한번에 읽을 논리적 블록 개수로서 클러스터 크기를 의미한다.

⑤ **Block 번호** : 단말 노드에 속한 공간 데이터를 저장하는 연속적인 논리적 블록 중에 시작 논리적 블록의 번호를 나타낸다.

⑥ **o_i의 block offset** : 단말 노드에 속한 i번째 공간 데이터가 저장되어 있는 논리적 블록의 시작 논리적 블록에서의 block offset 주소를 나타낸다.

⑦ **o_i의 byte offset** : 단말 노드에 속한 i번째 공간 데이터가 논리적 블록 내에서 저장되어 있는 위치의 byte offset 주소를 의미하며, 16바이트 단위로 바운딩(bounding)하여 나타낸다. 즉, byte offset이 20이면 실제 공간 데이터가 저장되어 있는 위치는 320번째 바이트부터 시작된다.

⑧ **o_i의 byte length** : 단말 노드에 속한 i번째 공간 데이터의 크기를 byte 크기로 나타내며, byte offset 표현법과 같이 16바이트 단위로 바운딩(bounding)하여 나타낸다.

3.4 알고리즘

다음은 3.1절에서 제시한 EPR 기법의 기본적인 알고리즘을 바탕으로 EPR 기법에 의한 공간 색인 트리의 생성 연산 및 검색 연산에 대해 구체적인 알고리즘을 제시한 것이다.

3.4.1 생성 알고리즘

EPR 기법의 공간 색인 트리 생성 알고리즘은 크게 3단계를 거쳐 실행된다.

첫째, 공간 데이터를 인접도를 바탕으로 패킹 기반 클러스터링하기 위해 Hilbert 값을 계산하여 정렬한다.

둘째, 인접도에 따라 정렬된 공간 데이터에 대해 패킹 알고리즘을 이용하여 단말 노드를 생성하고, 하나의 단말 노드에 속한 공간 데이터는 패킹 기반 클러스터링을 통해 동일한 디스크의 연속적인 논리적 블록에 저장하되, 단말 노드 단위로 round-robin 스트라이핑 기법을 이용하여 다중 디스크에 분산 배치한다.

셋째, 앞에서 생성된 단말 노드에 패킹 알고리즘을 이용하여 상향식으로 비단말 노드를 생성하면서 EPR 공간 색인 트리를 생성한다.

단계 1. 공간 데이터 정렬 알고리즘

/* 인접성을 고려하여 클러스터링하기 위해 공간 데이터의 *Hilbert* 값을 계산하여 정렬한다. */

입력 : 정렬 되지 않은 공간 데이터

출력 : 정렬된 공간 데이터

알고리즘 :

1. 공간 데이터들의 *Hilbert* 값을 계산한다.
2. 공간 데이터를 *Hilbert* 값에 의해 오름차순으로 정렬한다.

단계 2. 단말 노드 생성 알고리즘

/* 정렬된 공간 데이터의 색인 정보(MBR, 디스크 위치 정보)를 단말 노드에 패킹하여 저장하고, 하나의 단말 노드에 속한 공간 데이터를 패킹 기반으로 클러스터링하여 클러스터 단위로 다중 디스크에 분산 배치한다. 단말 노드는 단계 3에 비단말 노드를 생성하기 위해 단말 노드 리스트에 저장한다. */

입력 : 정렬된 공간 데이터

출력 : 패킹된 단말 노드들의 연결 리스트(linked-list), 연속된 디스크 블록에 패킹하여 저장된 공간 데이터

변수 정의 :

n : 디스크 개수,

d : 디스크 번호 ($0 < d < n$),

bl_d : d번째 디스크의 블록 번호,

$N_{i,j}$: level i의 j번째 노드(level 0는 단말 노드),

NL_i : R-tree 색인을 구성하기 위해 $N_{i,j}$들을 저장하고 있는 level i 노드 연결 리스트,

$B(d, bl_d)$: d번 디스크의 bl_d번째 디스크 블록,

buffer : 공간 데이터를 패킹하여 디스크에 저장하기위한 임시 저장 장소로서 디스크의 논리적 블록과 동일한 크기를 갖는다.

알고리즘 :

1. d = 0; i = 0; j = 0 /* 변수 초기화 */
2. NL_i를 생성한다.
3. $N_{i,j}$를 생성한다.
4. bl_d를 할당 받는다.
   /* $N_{i,j}$의 모든 공간 데이터를 연속적으로 저장할 디스크 블록들의 시작 블록 */
5. 하나의 공간 데이터 색인 정보를 $N_{i,j}$에 입력한다.
6. *buffer*의 남은 공간이 공간 데이터를 저장하지 못하면, $B(d,bl_d)$에 저장한다; bl_d ++; *goto 8.*
7. *buffer*에 공간 데이터를 패킹하여 저장한다.
8. $N_{i,j}$가 꽉 채워져 있지 않으면 *goto 5.*
9. $N_{i,j}$가 꽉 채워져 있으면 NL_i에 입력한다;
   d = (d+1) mod n; j = j+1.
   /* d를 mod 연산을 통해 수정하는 것은 round-robin striping을 적용하기 위함 */
10. 공간 데이터가 남아있으면 *goto 3*
11. $N_{i,j}$가 비어 있지 않으면 NL_i에 입력한다.

단계 3. 비단말 노드 생성 알고리즘

/* 단계 2에서 생성된 단말 노드 리스트를 이용하여 상향식으로 R-tree 색인을 생성한다. 생성된 R-tree 색인은 디스크에 저장된다. */

입력 : 공간 데이터가 입력된 단말 노드들의 연결 리스트

출력 : 디스크에 저장된 R-tree 색인

변수 정의 :

i : 색인 트리의 level

알고리즘 :

1. i = 0; j=0; k=0. /* 변수 초기화 */
2. NL_{i+1}을 생성한다.
3. $N_{i+1,k}$을 생성한다.
4. NL_i에서 $N_{i,j}$을 추출한다.
5. $N_{i+1,k}$에 $N_{i,j}$의 인덱스 정보를 삽입한다; j = j+1.
6. NL_i가 비지 않고, $N_{i+1,k}$가 채워져 있지 않으면, *goto 4*
7. NL_i가 비지 않고, $N_{i+1,k}$가 채워져 있으면, NL_{i+1}에 $N_{i+1,k}$를 삽입한다; k = k+1; *goto 3.*
8. NL_{i+1}의 entry가 2 이상이면, i = i+1; *goto 2.*
9. NL_{i+1}을 근 노드로 하는 R-tree를 디스크에 저장한다.

3.4.2 검색 알고리즘

너비 우선 탐색(breadth first search) 방법으로 R-tree 색인을 검색하여 질의 영역에 포함되는 공간 데이터가 저장된 디스크 블록들을 찾고 읽기를 요구한다. 읽어진 디스크 블록에 저장된 공간 데이터들 중에 질의에서 요구되어진 공간 데이터를 클라이언트에 전송한다.

EPR 색인 트리 검색 알고리즘

/* 너비 우선 탐색 방법에 기반하여 R-tree 색인을 검색한다. */

입력 : 검색에 사용될 R-tree인덱스 ID 검색에 사용될 영역 질의

출력 : 질의 결과를 client에게 전송

변수 정의 :

Root : R-tree 인덱스의 근 노드

Q : 인덱스를 너비우선 검색하기 위한 노드를 저장하는 큐(queue)

알고리즘 :

1. *Root*를 *Q*에 삽입한다.
2. *Q*로부터 노드를 하나 가져온다.
3. 노드가 비단말 노드이면, 노드가 가지고 있는 모든 자식 노드를 검사하여 질의 영역에 포함되는 자식 노드를 *Q*에 삽입한다; *goto 2.*
4. 노드가 단말 노드이면, 단말 노드가 가진 공간 데이터들이 저장된 모든 블록을 읽는다.
5. 읽은 블록들에 포함된 공간 데이터 중에 질의 영역에 포함되는 공간 데이터를 클라이언트에게 전송한다; *goto 2.*

4. 실험 및 결과

4.1 실험 환경

실험에 사용된 시스템 환경은 <표 1>과 같다.

〈표 1〉 시스템 환경

시스템	Sun Ultra Sparc 1
CPU 속도	Ultra Sparc -147Mhz
주기억 장치	128M
하드 디스크	6GB(2GB SCSI HDD 3개) 7200 RPM, Avg. Seek Time : 9 msec Avg. Latency : 4.17 msec
운영 체제	Solaris 2.5.1
사용 언어	C 언어

그리고, 실험에 사용된 데이터 모델은 <표 2>와 같다.

〈표 2〉 실험 데이터 모델

디스크 페이지 크기	공간 데이터 개수	공간 데이터 크기 ~N (μ, δ^2)	질의 크기
1KB, 2KB, 3KB	10,000개, 20,000개, 40,000개	~ N(100,25), ~ N(200,50), ~ N(300,75), ~ N(400,100), ~ N(500,125), ~ N(600,150)	실험 공간 크기의 5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%, 45%, 50%

실험에 사용한 실험 공간은 100,000×100,000 픽셀을 가지는 지도 데이터로 가정하였다. <표 2>에서 제시된 변수는 입출력 시스템의 성능에 영향을 미치는 요인으로서, 디스크 페이지 크기는 디스크의 논리적 블록의 크기를, 공간 데이터 개수는 실험에 사용된 공간 데이터 객체 수를 나타내며, 공간 데이터 크기는 공간 데이터 객체를 정의하는 좌표 개수에 대한 확률 분포를 나타낸 것으로 정규 분포를 따른다고 가정한다. 실험은 디스크 페이지 크기, 공간 데이터 개수, 그리고 공간 데이터 크기를 조합하여 54가지의 데이터 모델을 만들고, 각 데이터 모델에 대해 10가지 크기의 영역 질의를 10회씩 수행하여 총 5400회의 실험을 실시하였다.

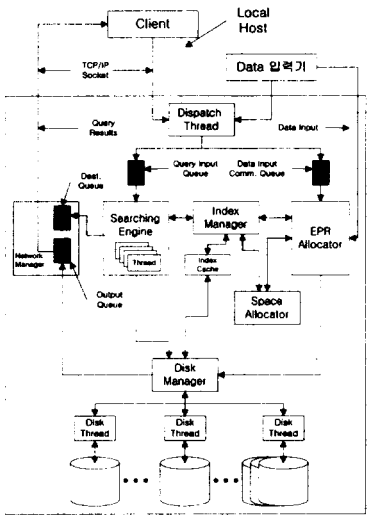
그리고, 실험은 EPR 기법의 성능 정도를 파악하기 위하여 PR 기법의 성능과 비교 분석하였다.

4.2 실험 모델

EPR 기법은 디스크에 공간 데이터를 연속적으로 저장하기 때문에 단일 영역 질의를 순차적으로 수행할 경우 EPR 기법이 PR 기법에 비해 유리하다. 따라서 실제 사용 환경과 유사하게 실험하기 위해 여러 사용자가 동시에 영역 질의를 요구하는 상황을 가정하고, 디스크를 순차적으로 접근하는 것이 아니라 임의적으로 접근하도록 하였다.

공간 데이터 입력기를 두어 54개의 데이터 모델에 따라 여러 개의 데이터 집합을 만들었으며, 생성된 데이터 집합에 대해 EPR Allocator를 이용하여 색인을 구성하고, 공간 데이터를 각 디스크에 분산 배치하였다. 색인들은 Index Manager가 관리하도록 하여 Searching engine에서 영역 질의를 수행할 때 필요한 색인을 사용할 수 있도록 하였다.

R-tree index ID와 질의 영역으로 구성되는 질의는 영역 질의 발생기를 두어 임의적으로 생성하도록 하였으며, 생성된 영역 질의를Searching engine에 포함된 search thread들에 할당하여 영역 질의에 포함된 공간 데이터를 찾고, Disk thread가 공간 데이터를 읽어 오도록 하였다.



(그림 6) EPR 시뮬레이션 모델

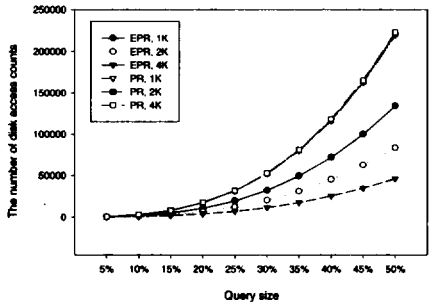
4.3 실험 결과

지리 정보 시스템의 입출력 시스템이 가지는 주요 성능은 빠른 응답 시간이고, 응답 시간에 가장 큰 영향을 미치는 요인이 디스크 액세스 횟수이므로, EPR 기법의 성능을 분석하기 위해 영역 질의에 대한 응답 시간과 디스크 접근 횟수를 측정하였다.

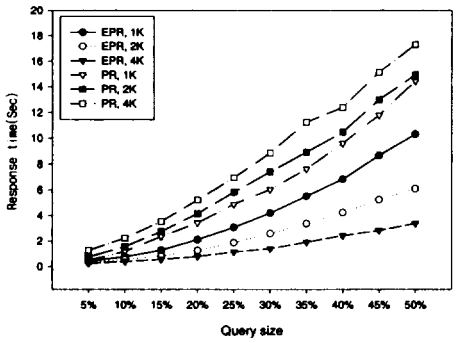
<표 3>은 전체적인 실험 결과를 요약하여 보여주고 있다. (그림 7)은 질의 영역 크기가 다른 10종류의 질의가 요구되었을 때 디스크 페이지 크기와 병렬 공간 색인 기법에 따른 디스크 접근 횟수를 보여주고 있다. 질의 영역이 증가함에 따라 질의 영역에 포함되는 공간 데이터의 개수도 증가하여 디스크 접근 횟수가 증가한다. PR 기법은 하나의 논리적 블록에 하나의 공간 데이터를 저장하기 때문에 디스크 페이지 크기가 커지더라도 영역 질의별로 동일한 디스크 접근 횟수를 보여주는 반면에 EPR 기법은 하나의 논리적 블록에 여러 개의 공간 데이터를 패킹하여 저장하므로 디스크 페이지 크기가 커질수록 논리적인 블록에 더 많은 공간 데이터가 저장할 수 있어 디스크 접근 횟수가 감소함을 보여준다.

〈표 3〉 실험 결과-평균 디스크 액세스 횟수와 평균 응답 시간

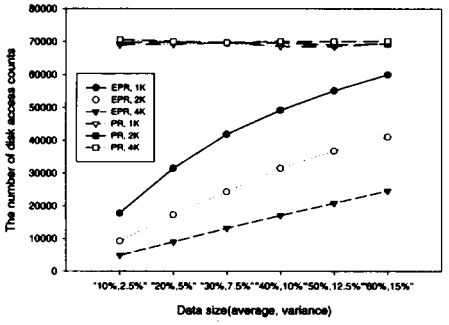
배치 방법	페이지 크기	평균 데이터 크기	평균 디스크 접근 횟수	평균 응답시간
EPR	1K	115	17717.67	2.148
		337	41908.33	4.318
		663	60009.67	5.897
	2K	115	9232.33	1.144
		337	24282.67	2.411
		663	41098.67	3.896
	4K	115	4873.00	0.678
		337	13144.67	1.356
		663	24576.00	2.275
PR	1K	115	69080.33	6.197
		337	69713.00	6.266
		663	69416.00	6.297
	2K	115	69733.33	7.151
		337	69573.33	7.057
		663	69385.00	6.955
	4K	115	70622.33	8.636
		337	69740.00	8.296
		663	70202.33	8.859



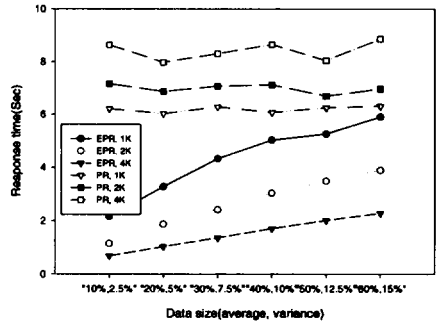
(그림 7) 질의 크기별 색인 방식과 디스크 페이지 크기에 따른 디스크 접근 횟수



(그림 8) 질의 크기별 색인 방식과 디스크 페이지 크기에 따른 응답 시간



(그림 9) 공간 데이터 크기별 색인 방식과 디스크 페이지 크기에 따른 디스크 접근 횟수



(그림 10) 공간 데이터 크기별 색인 방식과 디스크 페이지 크기에 따른 응답 시간

(그림 8)은 10종류의 크기별 영역 질의가 요구 되었을 때 디스크 페이지 크기와 병렬 공간 색인 기법에 따른 응답 시간을 보여주고 있다. EPR 기법의 경우 (그림 7)에서 보여주듯이 디스크 페이지 크기가 커질수록 디스크 접근 횟수가 감소하여 응답 시간이 감소하는 것을 알 수 있고, PR기법은 디스크 접근 횟수는 동일하나 읽어야 할 디스크 페이지 크기가 커지므로 응답 시간이 조금씩 증가하는 것을 알 수 있다.

(그림 9)와 (그림 10)은 영역 질의 요구가 들어 왔을 때, 공간 데이터 크기별 디스크 페이지 크기와 병렬 공간 색인 기법에 따른 디스크 접근 횟수와 응답시간을 보여주고 있으며, (그림 7)과 (그림 8)에서 보여주는 결과와 같은 경향을 보이고 있다.

(그림 10)에서 디스크 페이지 크기가 1K이고, 공간 데이터 크기가 60%일 때, 디스크 페이지 크기가 작고 공간 데이터 크기가 크므로 하나의 블록에 저장되는 평균 데이터 개수가 1에 접근하여 EPR 기법과 PR 기

법이 동등한 조건을 가지게 되나, EPR 기법의 경우 같은 단말 노드에 저장되어 있는 공간 데이터들이 동일한 디스크에 물리적으로 인접하게 배치되므로 디스크 헤드의 탐색 시간과 회전 지연 시간이 감소하여 PR 기법보다 응답 시간이 다소 좋게 나타난다. 이는 같은 조건 하에서도 PR 기법보다 EPR 기법의 성능이 더 효율적임을 보여준다.

실험 결과를 요약하면, EPR 기법은 공간 데이터 크기가 작고 디스크 페이지 크기가 클수록 하나의 논리적 블록에 저장되는 공간 데이터 수가 증가하여 훨씬 좋은 성능을 보여준다. 그리고, EPR 기법과 PR 기법이 동등한 조건을 갖는 환경, 즉 공간 데이터 크기가 크고 디스크 페이지 크기가 작을 때에도 EPR 기법이 PR 기법보다는 좋은 성능을 보여준다.

5. 결론 및 향후 연구과제

본 논문에서는 지리 정보 시스템의 입출력 병목 현상을 병렬 입출력과 효율적인 디스크 입출력을 통하여 줄이는 병렬 공간 색인 기법인 EPR 기법을 제안하였다.

EPR 기법은 개별 공간 데이터의 크기가 작고 영역 질의의 경우 인접한 공간 데이터를 동시에 요구한다는 특성을 이용하여 R-tree 색인을 패킹하고, 공간상에 인접한 공간 데이터를 패킹 기반 클러스터링을 통해 다중 디스크에 분산 배치하되 디스크의 연속적인 논리적 블록에 저장함으로써 디스크 입출력 성능을 향상시키고 디스크 접근 횟수를 줄여 응답 시간을 단축시킨다.

실험을 통하여 EPR 기법이 다른 병렬 공간 색인 기법과 비교하여 30% 이상 향상된 성능을 보임을 알 수 있었다.

향후에는 EPR 기법이 인접한 공간 데이터를 연속적인 논리적 블록에 저장하기 위한 비용을 최소화하고, EPR기법의 성능을 최대화하는 효율적인 디스크 공간 관리 기법과 공간 질의에 따라 디스크 접근을 최적화할 수 있는 기법 등에 대한 연구가 필요하다.

참 고 문 헌

[1] Roussopoulos N., and Leifker D., "Direct Spatial Search on Pictorial Database Using Packed R-trees," in Proc. of ACM SIGMETRICS Conf., 1985.

[2] Ibrahim Kamel and Christos Faloutsos, "On Packing R-trees," Second International Conf. on Information and Knowledge Management(CIKM), 1993.

[3] 김덕환, 정진완, "영역 질의를 위한 병렬 공간 색인 기법 및 공간 근접 디스크 할당 알고리즘," 한국정보과학회 가을 학술발표논문집, Vol.22(2), pp.123-126, 1995.

[4] N. Koudas, C. faloutsos, I. Kamel, "Declustering spatial database on a multi-computer architecture," In Proc. Of Extended Database Technologies(EDBT), pp.592-614, 1996.

[5] Scoot T. Leutenegger, Jeffrey M. Edgington, and Mario A. Lopez, "STR : A Simple and Efficient Algorithm for R-Tree Packing," in Proc. of the 1997 International Conf. on Data Engineering, 1997.

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

[7] Shashi Shekhar, Sivakumar Ravada, and Vipin Kumar, "Declustering and Load-Balancing Methods for Parallelizing Geographic Information Systems," in IEEE Transactions on Knowledge and Data Eng., Vol.10, No.4, 1998.

[8] Bernd Schnitzer, and Scott T. Leutenegger, "Master-Client R-trees: A New Parallel R-tree Architecture," the 11th International Conference on Scientific and Statistical Database Management(SSDBM 99), 1999.

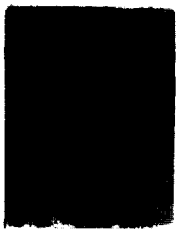
[9] Antonin Guttman, "R-trees : A dynamic index structure for spatial searching," in Proc. of ACM SIGMOD Conf., pp.47-57, 1984.

[10] Timos Sellis, Nick Roussopoulos, and Christos Faloutsos, "The R+-tree : A dynamic index for multi-dimensional objects," in Proc. of the 13th VLDB Conf., pp.507-518, 1987.

[11] Norbert Beckmann and Hans-Peter Kriegel, "The R*-tree : An efficient and robust access method for points and rectangles," in Proc. of ACM SOGMOD Conf., pp.322-331, 1990.

[12] 김기홍, 차상균, "공간 색인의 다중 검색 경로 문제와 클러스터 저장 구조," 한국정보과학회 학술

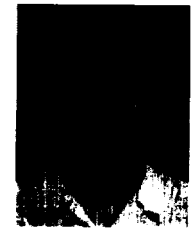
발표논문집, Vol.23(1), pp.39-42, 1996.
[13] Ibrahim Kamel and Christos Faloutsos, "Hilbert R-tree: An improved R-tree using fractals," in Proc. of International Conf. on Very Large Databases, 1994.
[14] Yvan J. Garcia R., Mario A. Lopez, and Scott T. Leutenegger, "On Optimal Node Splitting for R-trees," in Proc. of the 24th VLDB Conf., 1998.
[15] T. Bially, "Space-filling curves : Their generation and their application to bandwidth reduction," IEEE trans. on Information Theory, 1969.



이 춘 근
e-mail : tetris@hyowon.cc.pusan.ac.kr
1998년 경성대학교 전산통계학과 졸업(학사)
1998년~현재 부산대학교 전자계산학과 석사과정
관심분야 : GIS, 병렬 파일 시스템, 멀티미디어

김 정 원
e-mail : jwkim7@hyowon.cc.pusan.ac.kr
1995년 부산대학교 전자계산학과 졸업(학사)
1997년 부산대학교 전자계산학과 졸업(석사)
1997년~현재 부산대학교 전자계산학과 박사과정
관심분야 : 멀티미디어 파일 시스템, 병렬처리

김 영 주
e-mail : ygjukim@hyowon.cc.pusan.ac.kr
1988년 부산대학교 계산통계학과 졸업(학사)
1990년 부산대학교 계산통계학과 졸업(석사)
1990년~1995년 큐닉스컴퓨터 응용시스템연구소 근무
1999년 부산대학교 전자계산학과 졸업(박사)
관심분야 : 멀티미디어, 병렬파일시스템, 분산처리 등



정 기 동
e-mail : kdchung@hyowon.cc.pusan.ac.kr
1973년 서울대학교 졸업(학사)
1975년 서울대학교 대학원 졸업(석사)
1986년 서울대학교 대학원 계산통계학과 졸업(박사)
1990년~1991년 MIT, South Carolina 대학 교환교수
1978년~현재 부산대학교 전자계산학과 교수
1993년~현재 부산대학교 부설 컴퓨터 및 정보통신 연구소 운영위원
관심분야 : 병렬처리, 멀티미디어