

향상된 그리드 색인을 이용한 이동 객체의 연속 질의 처리*

*박용훈, 복정수**, 유재수*

충북대학교 정보통신공학과

e-mail: yhpark@netdb.chungbuk.ac.kr

ksbok@dbserver.kaist.ac.kr

yjs@cbucc.chungbuk.ac.kr

Continuous Range Queries Processing of Moving Objects using Enhanced Grid Indexing

Yong-Hun Park*, Kyung-Soo Bok**, Jae-Soo Yoo*

*Department of Computer & Communication Engineering,
Chungbuk National University

**Department of Computer, Korea Advanced Institute of Science
and Technology

요 약

본 논문은 위치 기반 서비스에서 중요한 질의 형태중의 하나인 연속 범위 질의를 효율적으로 처리하기 위한 질의 처리 기법을 제안한다. 본 논문에서는 기존 그리드 기반 질의 색인 기법을 변형하여 분할된 셀들을 그룹 단위로 관리하고 객체의 이동 변화에 따른 질의 결과의 변화를 빠르게 처리하기 위해 각 그룹에 포함된 질의들에 비트 식별자를 부여한다. 이러한 기법은 다수의 셀에 포함된 질의를 표현하기 위해 저장되는 질의 식별자의 수를 감소시킬 수 있으며 각 질의에 부여된 비트 식별자를 이용하여 객체의 갱신으로 인해 영향을 받는 질의를 빠르게 판단할 수 있다. 성능평가를 통해 제안하는 기법이 기존 질의 처리 기법에 비해 우수한 성능을 보여줄 것을 입증한다.

1. 서론

최근 휴대폰, PDA 또는 스마트폰 같은 모바일 기기의 발전과 위치를 인식할 수 있는 기술의 발전으로 위치 기반 서비스(LBS: Location Based Services)에 대한 활용이 증가되고 있다. 사용자의 위치에 관련된 정보제공, 광고, 주문배달 서비스, 쿠폰 서비스, 긴급경보 서비스, 물류관제 시스템 등이 그 대표적인 예이다. 최근, 이동 차량을 대상으로 하는 텔레매틱스 기술의 발전으로 위치 기반 서비스에 대한 중요성이 증가되고 있다.

질의의 결과를 특정 시간동안 모니터링 하는 질의를 연속 질의(Continuous Query)라 한다. 그리고 질의 대상이 범위에 관련된 질의를 연속 범위 질의(Continuous Range Query)라 한다. 지능형 운송 시스템(intelligent transportation systems) 또는 센서 기반 모니터링(sensor-based monitoring)과 같은 응용에서는 특정 영역 내에 존재하는 객체들의 변화를

계속적으로 추적하기 위한 연속 범위 질의가 활용되고 있다. 예를 들어, 교통사고 또는 정체를 발생시키는 사건들이 발생한다면 교통소통을 원활히 하기 위해 해당 구역 주변의 차량에게 해당 정보를 제공하거나 향상된 119 서비스로 화재나 폭발 사고 같은 위급 상황이 발생할 경우 구역별로 다양한 대처 정보를 모바일 기기를 통해 제공해 줄 수 있다.

연속 질의 처리는 변화되는 결과를 지속적으로 유지하기 위해 정보를 관리하는 서버에 많은 횟수의 접속과 검색을 수행해야 한다. 따라서 많은 검색 비용을 소요할 뿐만 아니라 서버 및 네트워크에 많은 부하를 초래할 수 있다. 최근 연속 범위 질의를 효과적으로 처리하기 위해 메모리 기반 질의 처리 기법이 많이 연구되고 있다[1,2,3]. 메모리 기반 질의 처리 기법들은 그리드 색인을 변형하여 질의를 색인하고 객체의 변화에 따라 질의 결과를 갱신한다. 그러나 이러한 기법들은 그리드로 분할된 각 셀에 질의의 식별자를 중복적으로 저장하기 때문에 많은 저장 공간을 소모한다. 또한, 객체의 갱신으로 인한 질의를 판단하기 위해 중복 저장된 질의 식별자

+ 본 연구는 2006년도 교육인적자원부 지방연구중심대학 육성사업과 한국과학재단 특정기초연구(과제번호 R01-2006-000-10809-0)의 지원에 의하여 수행되었음.

들을 계속적으로 비교하기 때문에 실제 영향을 받지 않는 질의도 연산에 포함된다.

본 연구에서는 기존 메모리 기반 연속 질의 처리 기법의 문제점을 해결하기 위한 새로운 질의 색인 및 질의 처리 기법을 제안한다. 제안하는 질의 색인 기법은 실시간적인 처리를 수행하기 위해 메인 메모리 기반의 그리드 구조를 가진 색인 기법으로 연속 질의를 처리하기 위한 연산 비용을 최소화하기 위해 제안하는 연속 질의 처리 기법은 비트 식별자(Bit ID)를 사용한다. 비트 식별자는 각 질의에 부여된 식별자로 그리드 색인의 각 셀은 질의들의 겹침 정보를 반영하는 비트 값을 가지고 있다. 이러한 비트 값을 통해 셀이 어떤 질의들에 포함되어져 있는지 계산 할 수 있으며 셀 사이의 비트 값을 비교하여 질의들 간의 포함관계를 파악할 수 있다. 이러한 처리 과정은 대량의 객체에 대해 질의를 처리할 경우 더욱 큰 성능 향상을 보이며 기존 메모리 기반 기법에 비해 연산 비용을 감소시킬 수 있다. 제안하는 연속 질의 처리 기법은 저장 공간과 비트 값의 길이를 효율적으로 관리하기 위해 셀들을 그룹 단위로 관리하고 그룹은 포함하는 질의들을 관리한다. 이로 인해, 질의 식별자들의 중복 저장을 최소화하여 색인의 유지비용을 감소시킨다.

본 논문의 나머지 구성은 다음과 같다. 먼저 2장에서 관련 연구에 대해 설명하고, 3장에서는 본 연구에서 제안하는 질의 색인 기법을 설명한다. 4장에서는 제안하는 기법의 성능평가 결과를 기술하고 마지막 5장에서는 결론 및 향후 연구에 대해서 기술한다.

2. 관련 연구

이동 객체를 대상으로 하는 연속 범위 질의의 처리는 크게 두 가지 방법으로 분류할 수 있다. 첫 번째는 객체 색인을 이용하는 방식으로 연속 범위 질의의 결과 집합을 유지하기 위해 일정 시간 간격으로 또는 객체의 정보가 갱신되었을 때 마다 모든 질의를 재실행 한다. 만약 30초 간격이나 1분 간격으로 최신의 정보를 갱신하고자 한다면, 동일한 형태의 질의가 30초나 1분 간격으로 또는 객체가 갱신된 횟수만큼 재실행 된다. 두 번째로는 질의 색인을 이용하는 방식으로 객체가 위치 갱신을 했을 시 질의 색인을 이용하여 어떤 질의로부터 벗어나고 어떤 질의로 새롭게 포함되는지를 검사하여 해당 질의의 결과셋을 갱신한다. 연속 질의 처리 기법은 일반적으로 두 번째 기법을 이용한다.

최근 연속 범위 질의를 처리하기 위한 질의 색인 기법에 관련된 많은 연구들이 진행되었다. 디스크 기반 색인 기법으로는 R-트리와 안정 영역(safe region) 개념을 이용한 Q-Index[4], 질의의 이동을 함께 고려한 SINA[5] 그리고 질의 영역을 구분하기 위해 color개념을 도입한 cGridex[6]를 들 수 있다.

메모리 기반 색인 기법으로는 셀 기반 색인(cell-based index)인 CQI-Index[1] 그리고 가상 타일 기반 색인(virtual tile-based index)인 VCR-Index[2]와 CES-Index[3]가 있다.

기존의 메모리 기반 색인 기법들은 접근 방법에 따른 성능상의 한계를 지니고 있다. 첫 번째로 객체가 어떤 질의의 범위로부터 벗어났는지 아니면 어떤 질의의 범위로 새롭게 들어갔는지를 알아내기 위해서 객체를 이전에 포함했던 질의들과 현재 포함하는 질의들을 질의 식별자를 이용하여 비교한다. 이때 객체의 식별자를 이용하여 비교 연산을 하기 때문에 효율적인 연산이라고 할 수 없다. 두 번째로 인덱스에서 질의의 정보를 유지하기 위해 질의의 식별자를 중복적으로 유지하기 때문에 저장 공간의 낭비를 초래한다.

3. 제안하는 기법

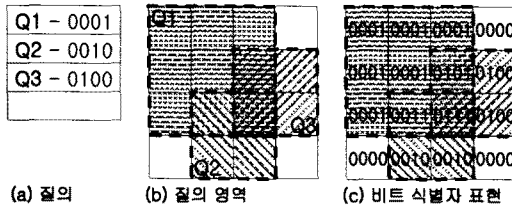
본 절에서는 기존 메모리 기반 연속 질의 처리 기법들의 문제들을 획기적으로 개선하여 보다 적은 저장 공간으로 보다 빠른 연산을 수행하는 질의 처리 기법을 제안한다. 제안하는 기법은 실시간적인 처리를 수행하기 위해 메인 메모리 기반의 그리드 구조를 변형하여 질의 색인을 수행하고 연속 질의를 처리하기 위한 연산 비용을 최소화하기 위해 비트 식별자(Bit ID)를 사용한다. 비트 식별자는 각 질의에 부여된 식별자로서 그리드 색인의 각 셀은 질의들의 겹침 정보를 반영하는데 비트 식별자의 조합으로 이루어진 비트 값을 이용한다. 이러한 비트 값을 통해 셀이 어떤 질의들에 포함되어져 있는지 계산 할 수 있으며 두 셀 사이의 비트 값을 비교하여 질의들의 포함관계를 비교 할 수 있다.

제안하는 연속 질의 처리 기법은 저장 공간과 비트 값의 길이를 효율적으로 관리하기 위해 셀들을 그룹 단위로 관리하고 그룹은 포함하는 질의들을 관리한다. 이로 인해, 질의 식별자들의 중복 저장을 최소화하여 색인의 유지비용을 크게 감소시킨다.

3.1 비트 식별자

비트 식별자는 각 질의에 부여된 식별자로서 그리드의 각 셀은 질의들의 겹침 정보를 비트 식별자들의 조합으로 만들어진 비트 값을 이용하여 표현한다. (그림 1)는 그리드 영역에서 질의가 관리되고 각 셀과 질의들 간의 비트 값을 이용한 관계 표현에 대해 나타내고 있다. (그림 1)의 (a)는 관리되는 질의 식별자들과 각 질의에 할당되는 비트 식별자에 대해 나타내고, (b)는 그리드 영역에서의 질의 영역을 표현한 것이다. (c)는 그리드를 구성하는 각 셀의 비트 값을 이용한 질의와의 관계 표현을 나타낸다. 예를 들어, Q1의 비트 식별자가 0010, Q2의 비트 식별자가 0010 그리고 Q3의 비트 식별자가 0100이라고 할 때, 아무런 질의와도 겹치지 않는 셀의 비트 값은 0000, 오직 Q1과 겹치는 부분의 셀의 비트 값은

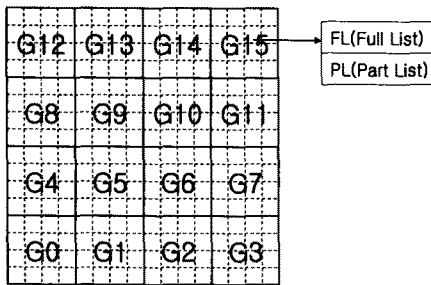
0001, Q3과 Q1에 겹치는 부분의 셀은 0101이다. 반대로 셀의 비트 값을 이용하면 겹침 관계에 있는 질의들을 식별할 수 있다.



(그림 1) 비트 식별자

3.2 그룹 단위의 셀 관리

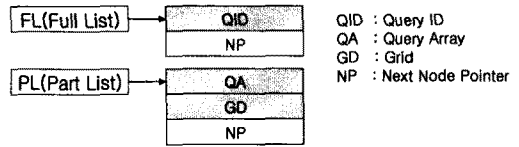
비트 식별자 표현 방식을 이용하여 전체 그리드에 모든 질의들을 표현할 경우 각 셀에 포함된 질의를 표현하기 위해 유지해야 하는 비트 값의 길이가 증가되고 이러한 비트 값의 길이에 따라 연산 비용의 증가와 저장 공간의 낭비를 초래할 수 있다. 따라서 제안하는 기법에서는 성능에 큰 영향을 주는 비트 값의 길이를 효율적으로 관리하기 위해 셀들을 그룹 단위로 관리한다. (그림 2)는 전체 그리드에 대해 그룹을 관리하는 기법을 나타낸 것이다. 각 그룹에서는 그룹과 겹치는 질의들을 질의 리스트로 관리하는데 이러한 질의 리스트는 FL(Full List)와 PL(Part List) 두 가지로 구분된다. FL은 그룹영역과 겹치는 질의 중에서 완전히 그룹을 포함하는 질의에 대한 리스트이고, PL은 그룹에 부분적으로 겹치는 질의를 관리하는 리스트이다.



(그림 2) 그룹단위의 셀 관리

각 그룹에서 관리하는 FL과 PL의 자료 구조는 (그림 3)과 같다. PL은 QA(Query Array), GD(Grid) 그리고 NP(Next Node Pointer)로 구성한다. QA는 그룹을 부분적으로 포함하는 질의의 식별자를 저장하고, 배열 상의 위치에 따른 비트 식별자를 지정한다. GD는 크기가 1인 셀들로 구성되어 있으며, 각 셀은 그룹에 부분적으로 겹치는 질의들과의 포함정보를 나타내는 비트 값을 유지한다. NP는 QA에 오버플로우가 발생할 경우 다음 PL 리스트를 접근하기 위한 포인터이다. FL은 QA(Query Array)와

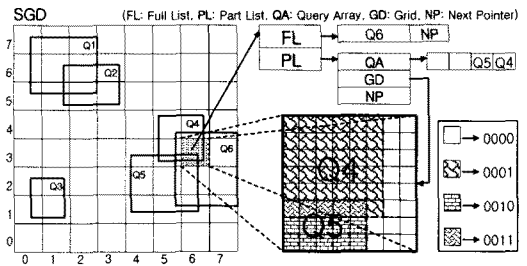
NP(Next Node Pointer)로 구성된다. FL은 그룹을 완전히 포함하는 질의들의 식별자를 관리한다.



(그림 3) FL과 PL의 자료구조

3.3 전체 색인 구조

(그림 4)는 연속 범위 질의 처리를 위해 본 연구에서 제안하는 기법의 전체적인 색인 구조를 표현하고 있다. 전체 그리드(SGD)는 크기가 8x8인 64개의 그룹으로 구성되어 있고, 6개의 질의(Q1~6)가 등록된 상태이다.



(그림 4) 전체 색인 구조

예를 들어, (6, 3) 위치에 있는 그룹으로 그룹과 겹치는 질의는 총 3개이다. 그중에서 Q6은 그룹을 완전히 포함하므로 FL에 등록 되어있다. 또한, Q4와 Q5는 그룹을 부분적으로 겹치므로 PL에 삽입된다. PL에 삽입 되어진 질의는 QA에 입력되고, 각 질의는 QA 내의 위치에 따른 비트 식별자를 갖는다. Q4의 비트 식별자는 0001이고, Q5의 질의 식별자는 0010이다. PL의 GD는 포함하는 셀을 관리하는데, Q4와 Q5에 포함되는 셀의 비트 값에 각각 질의의 비트 식별자 값을 더한다. 그래서 아무런 질의에도 포함되지 않는 셀의 비트 값은 0000, Q4에 또는 Q5에 포함되는 셀의 비트 값은 각각 0001과 0010이다. 그리고 Q4와 Q5에 모두 포함되는 셀의 비트 값은 0011이다.

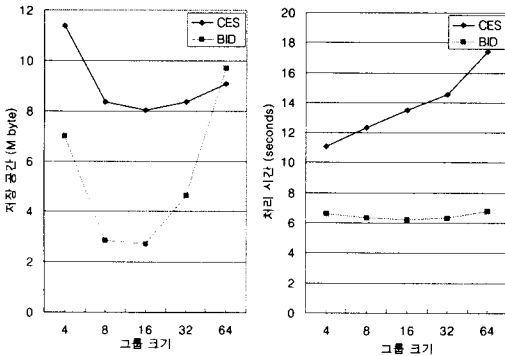
3.4 질의 처리

제안하는 연속 범위 질의 처리는 질의 색인 기법을 이용하여 객체가 이동시 영향을 받는 질의를 빠르게 검색하고 결과 집합을 변경하는 방식으로 수행한다. 질의 처리 기법은 객체가 한 그룹 안에서 이동했을 경우와 다른 그룹으로 이동 했을 경우를 구분하여 다른 처리 방법을 수행한다. 객체가 한 그룹 안에서 이동했다면, 그 그룹의 상에서 객체의 이전 위치를 포함하는 셀의 비트 값과 현재 위치를 포함하는 셀의 비트 값을 비교하여 갱신할 질의를 찾아낸다. 만약 객체가 다른 그룹으로 이동했다면, 객체의

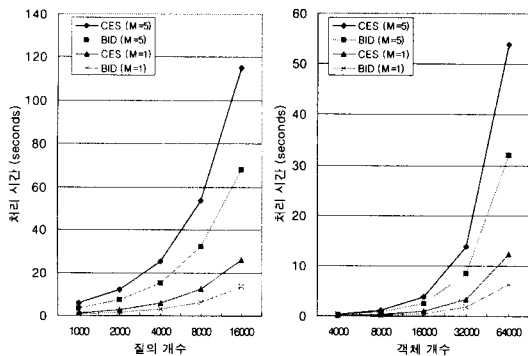
이전위치를 포함하는 모든 질의와 현재 위치를 포함하는 모든 질의를 비교하여 갱신할 질의를 찾아낸다.

4. 성능 평가

성능 평가를 위해 가장 최근에 연구된 CES-Index[6]와 처리 시간과 저장 공간에 대한 비교 분석을 수행한다. 실험은 펜티엄 4 CPU 3.0G, RAM 1G, 운영 체제는 윈도우즈 XP에서 수행되었으며 구현은 C언어를 사용하였다. 실험에서 전체 그리드 크기는 512×512, 그룹의 크기는 8×8, 객체의 개수는 64000, 질의의 가로 세로 크기는 1에서 80, 질의 개수는 8000을 기본으로 질의는 랜덤하게 생성시켜 수행하였다. 본 논문에서 제안하는 색인 기법은 BID로 나타내었다. (그림 5)와 (그림 6)는 CES와 질의 처리 시간에 대한 성능 비교를 수행한 결과이다.



(그림 5) 그룹 크기별 저장 공간과 처리 시간



(그림 6) 질의의 개수와 객체 개수에 따른 처리시간

(그림 5)는 그룹의 크기 지정에 따른 저장 공간과 처리 시간에 대한 성능 비교를 나타낸다. BID의 경우 그룹 크기의 지정의 저장 공간에 영향을 많이 미치지만 전체적으로 CES보다 적은 저장 공간을 사용하고, 보다 빠른 처리를 수행한다. (그림 6)는 질의의 개수와 객체의 개수에 따른 질의 처리 시간에 대한 성능 비교를 나타낸다. 모든 상황에서 BID가 CES보다 좋은 빠른 처리를 수행하는 것을 볼 수 있

다. M은 이동 객체들의 최대 이동 속도로서 M이 증가함에 따라 성능은 전체적으로 낮아지지만 CES와 BID사이의 별다른 특징을 보이지 않는다.

5. 결론 및 향후 연구

본 논문에서는 기존 메모리 기반 연속 질의 처리 기법의 문제점을 해결하기 위해 새로운 질의 색인 및 연속 질의 처리 기법을 제안하였다. 제안한 질의 색인 기법은 그리드 기반 색인 구조로 질의가 포함된 셀을 빠르게 검색할 수 있다. 또한, 연속 질의 처리 기법은 질의 색인 기법을 이용하여 객체의 이동 변화에 따라 질의 결과의 변경을 빠르게 처리한다. 향후 연구로 다양한 실험을 수행할 예정이며 이동하는 질의에 대한 연속 질의 처리 기법에 대한 연구를 수행할 예정이다.

참고문헌

- [1] D. V. Kalashnikov, S. Prabhakar, S. E. Hambrusch, "Main Memory Evaluation of Monitoring Queries Over Moving Objects", Distributed and Parallel Databases, Vol.15, No.2, pp.117-135, 2004
- [2] K. L. Wu, S. K. Chen, and P. S. Yu, "Processing continual range queries over moving objects using VCR-based query indexes" In Proc. of IEEE Int. Conf. on Mobile and Ubiquitous Systems : Networking and Services, Aug. 2004
- [3] K. L. Wu, S. K. Chen and P. S. Yu, "On Incremental Processing of Continual Range Queries for Location-Aware Services and Applications", Proc. Annual International Conference on Mobile and Ubiquitous Systems, pp.261-269, 2005
- [4] S. Prabhakar, Y. Xia, D. V. Kalashnikov, W. G. Aref and S. E. Hambrusch, "Query Indexing and Velocity Constrained Indexing : Scalable Techniques for Continuous Queries on Moving Objects", IEEE Transactions on Computers, Vol.51, No.10, pp.1124-1140, 2002
- [5] M. F. Mokbel, X. Xiong and W. G. Aref, "SINA: Scalable Incremental Processing of Continuous Queries in Spatio-temporal Databases", Proc. the ACM SIGMOD International Conference on Management of Data, pp.623-634, 2004
- [6] X. Wang, Q. Zhang, W. Sun, W. Wang and B. Shi, "cGridex: Efficient Processing of Continuous Range Queries over Moving Objects", Proc. th International Conference on Advances in Web-Age Information Management, pp.345-356, 2005