

RFID 미들웨어에서 연속질의 처리를 위한 질의 색인 기법*

석수욱⁰ 박재관 홍봉희

부산대학교 컴퓨터 공학과

{seok1968⁰, jkpack, bhong}@pusan.ac.kr

A Query Indexing Method for Processing Continuous Queries in RFID Middleware Systems

Su wook Seok⁰, Jae kwan Park, Bong hee Hong

Department of Computer Science and Engineering, Pusan National University

요 약

EPCglobal은 RFID와 관련된 다양한 분야의 표준화를 주도하고 있으며, 응용 표준으로써 Tag 정보의 운용을 위한 RFID 미들웨어(Savant)를 제시하였다. 특히, Savant의 RIED는 Tag 정보를 저장하고 다양한 질의 처리를 제공하는 MMDBMS이다. 그러나 RIED는 User-Driven 방식만 지원하므로 연속 질의를 처리하기 위해 주기적으로 질의를 수행하게 된다. 따라서 주기적인 질의 수행으로 인한 심각한 성능 저하가 발생하며, 실시간 처리를 요구하는 질의를 수행하지 못한다.

이 논문에서는 RFID 미들웨어의 질의 처리 문제를 해결하기 위하여 연속 질의에 적합한 질의 색인 구조를 제시한다. 이 색인은 질의를 색인의 데이터로, 데이터를 색인의 질의로 Swap하여 데이터에 독립적인 검색 성능을 보장한다.

1. 서 론

차세대 IT혁명을 이끌 기술로써 RFID 관련 기술에 대한 연구 개발이 진행되고 있다. 특히, 리더로부터 수집된 대용량의 EPC 데이터들을 필터링 및 전달하기 위한 미들웨어 기술에 대한 관심이 증대되고 있다. EPCglobal의 RFID 미들웨어 규격인 Savant[1]는 TMS(Task Management System)를 포함하고 있다. TMS는 수집된 데이터들을 기반으로 일련의 태스크들을 수행하는 역할을 한다. TMS는 태스크 처리를 위해 Savant내의 메인메모리 기반 데이터베이스인 RIED(Real-time In-memory Event Database)로 SQL 질의처리를 요청한다. 이러한 질의들은 일정 시간범위 동안 수행되어야 하는 연속 질의[2]의 특성을 가진다. 최근 이러한 연속 질의를 처리하는 핵심 기술로써 질의 색인 기법에 대한 연구가 진행되고 있다. 본 논문은 RFID 미들웨어에 등록되는 태스크들의 실시간 처리를 위한 질의 색인 기법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 질의 색인에 관한 관련연구를 기술한다. 3장에서는 현재 RFID 미들웨어의 질의처리 방법의 비효율성과 기존 질의 색인 기법의 문제점을 정의한다. 4장에서 질의 색인의 적용방법과 색인 기법을 제시한다. 마지막으로 5장에서 결론을 맺는다.

2. 관련 연구

지속적으로 입력되는 대량의 데이터 스트림에 대한 연속 질의를 효율적으로 처리하기 위해 데이터 스트림 관리 시스템(Data Stream Management System: DSMS)[3]의 개발이 진행되고 있다. DSMS는 데이터 스트림을 저장하지 않는 대신 연속 질의를 등록하여 입력 스트림에 대해 실시간으로 질의 결과를 얻는 방식을 사용한다.

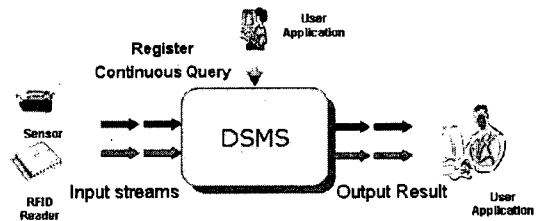


그림 1 데이터 스트림 관리 시스템(DSMS)

등록된 다수의 연속 질의들을 실시간으로 처리하기 위해서는 입력 스트림과 관계되는 질의를 빠르게 검색하는 것이 필요하다. 이를 위해 연속 질의를 색인하는 기술이 연구개발 되고 있다. 연속 질의를 색인하는 방법은 질의의 Predicate(조건절)을 이용하는 것이다. 대표적인 질의 색인 구조로는 CEI-based Query Index [4]와 VCR-Index[5]가 있다.

CEI-based Query Index는 1차원 속성을 가지는 질의를 색인한다. CEI(Containment-Encoded Interval)라는 가상 분할 구조를 이용하여 1차원의 Predicate Interval을 Overlap이 없도록 분할 저장한다. 이때 정수 연산만 사용하여 메인 메모리 환경에서 부동소수점 연산으로 인한 부하를 줄이고 있다.

VCR-Index는 2차원의 영역 질의를 색인한다. 이동체에 대한 영역 질의의 Predicate은 2차원의 사각형으로 표현이 되는데 이것을 VCR(Virtual Construct Rectangle)이라는 가상의 사각형 구조로 분할 저장한다.

3. 문제 정의

EPCglobal의 RFID 미들웨어인 Savant는 TMS를 통해 사용자가 등록한 태스크를 처리한다. 이때 TMS는 태스크 처리를 위한 SQL

* 이 논문은 교육인적자원부 지방연구중심대학육성사업 "차세대 물류IT기술연구사업단"의 지원에 의하여 연구되었음.

질의를 생성하여 주기적으로 실시간 메모리 데이터 구조인 RIED에 질의 처리를 요청한다. 이러한 주기적인 질의 수행은 TMS의 태스크 스케줄링 부하를 야기하며 질의 처리에 필요한 데이터의 삽입이 없을 때에도 질의를 수행해야 하는 문제점이 있다. 또한 주기적인 방식은 실시간 질의 결과를 보장하지 못한다.

이러한 비효율성을 질의 색인을 사용하여 개선할 수 있다. 하지만 기존의 질의 색인들은 RFID 미들웨어 환경의 질의를 고려하지 않았다. RFID 미들웨어 환경에서 연속 질의의 Predicate는 ReaderID, tagEPC 두 가지 속성을 가진다. ReaderID는 순서를 가지지 않기 때문에 interval이 될 수 없으며 tagEPC는 case단위의 이동을 고려 시 interval 또는 point가 될 수 있다. 즉, RFID 미들웨어 환경에서 연속 질의의 Predicate는 2D Interval 또는 2D Point가 된다. CEI-based Query Index의 경우 1차원의 interval을 다루기 때문에 2차원 Predicate를 가지는 질의에는 적용이 불가능하다. VCR-Index의 경우 2차원의 영역을 가지는 Predicate를 사용하고 있지만 이동체에 대한 영역 질의를 위해 설계되었기 때문에 분할 구조가 2D Interval에 적합하지 않고 삽입되는 데이터 스트림에 대한 질의 검색(정질의) 시 불필요한 질의들이 선택될 수 있는 단점이 있다.

4. RFID 미들웨어를 위한 질의 색인 기법
4.1 질의 색인 적용 방법 제안

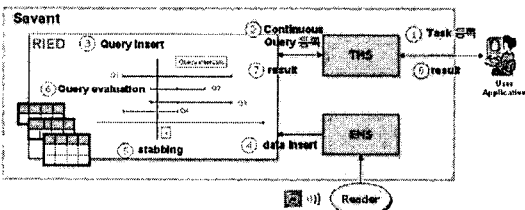


그림 2 질의 색인 적용 방법

그림 2는 질의 색인을 RFID 미들웨어 시스템에 적용하는 방법을 보이고 있다. 먼저 TMS에 등록된 태스크들의 처리를 위해 필요한 연속 질의를 RIED에 등록한다. 이렇게 등록된 질의들을 RIED의 질의 색인에 삽입한다. 색인에 삽입하는 시간은 연속 질의의 시작 시간이 된다. 이때 리더로부터 데이터가 RIED에 삽입이 되면 해당 데이터를 필요로 하는 질의가 무엇인지를 질의 색인을 이용하여 검색한다. 마지막으로 검색 결과에 해당하는 질의들을 실시간으로 수행 후 TMS로 결과를 반환한다.

4.2 질의 색인 기법

RFID 미들웨어 환경에서 연속 질의의 Predicate를 먼저 분석하도록 하자. 본 논문이 제안하는 질의 색인은 ReaderID, tagEPC 두가지 속성으로 이루어지는 질의들을 대상으로 한다.

```
SELECT * FROM Observation[start time, end time]
WHERE readerID = r1 and tagEPC >=e1 and tagEPC <= e10
```

그림 3 RFID 환경의 연속 질의 예

위의 그림 3은 연속 질의의 예를 보여주고 있다. 본 논문에서는 색인되는 질의의 WHERE절이 예와 같이 오직 논리곱(Conjunctive)들로만 구성되는 것으로 가정한다. 이때 연속 질의의 Predicate는 아래의 그림 4와 같이 2차원의 Point 또는 Interval로 표현할 수 있다. 색인에 삽입되는 데이터는 (QID, readerID, tagEPC⁺, tagEPC⁻)로 구성되며 tagEPC의 From과 To가 같을 때 2D Point가 된다.

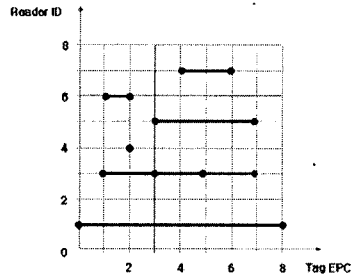


그림 4 연속 질의의 2D Interval Predicate

질의 색인 방법은 VCR-hashing기법에 기반한다. 위의 2D Interval Predicate를 미리 정의된 가상 구조를 이용하여 분할하고 매칭되는 가상 구조의 리스트에 삽입하는 방식이다. 가상 구조는 Predicate의 데이터 표현 방법과 유사한 2D Interval을 사용한다. 그림 5는 2차원의 가상 선분 구조를 보여주고 있다.



그림 5 2차원의 가상 선분 구조

tagEPC를 x라 하고 readerID를 y라고 할 때 가상 선분 구조는 그림 5와 같이 (x, y, length)로 표현된다. 이때 x, y의 범위는 각 축의 Max값을 각각 R_x, R_y라 할 때 0 ≤ x < R_x, 0 ≤ y < R_y가 된다. length는 tagEPC의 범위를 나타내며 2의 지수 승으로 표현한다. 따라서 위의 구조는 (x, y, 2^k)로 바꾸어 표현할 수 있다. 이때 i의 범위는 0 ≤ i ≤ k이며 2^k는 length의 최대값이 된다. 이러한 가상 선분 구조는 각 점마다 k+1개씩 존재한다. 가상 선분 구조 (x, y, 2ⁱ)의 ID는 각 점당 선분 구조의 개수를 N이라 하고 (0,0), (1,0), ..., (R_x-1,0), (0,1), ...,과 같은 순서로 ID번호를 부여할 경우, N(x+yR_x) + i가 된다. 그림 6은 (0,0)에서 length의 최대값이 8이라고 할 때 네 가지 경우의 가상 선분 구조 예제를 보여주고 있다. length의 최대값이 8일 때 length는 2⁰, 2¹, 2², 2³이 될 수 있으며 이때 각각의 ID는 위의 식 N(x+yR_x) + i로 구할 수 있다. 예에서 (0,0,2)의 경우 N=4, x=0, y=0, R_x=8, i=1이므로 위의 식에 대입하면 ID는 1이 된다.

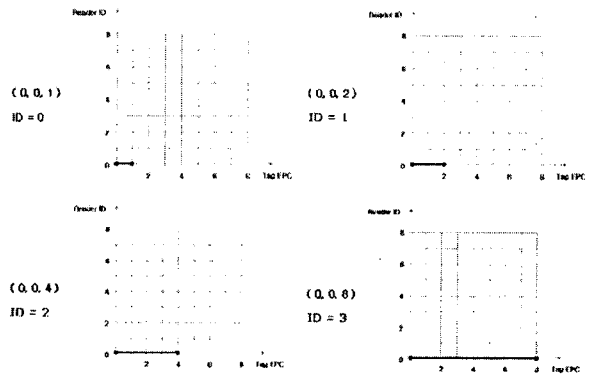


그림 6 (0,0)이 가지는 가상 선분 구조

새로운 질의가 등록되어 색인에 삽입이 발생하면 먼저 해당 질의의 Predicate를 왼쪽부터 매칭되는 가장 큰 가상 선분 구조로 분할해 간다. 그림 7의 예에서 Q1이 삽입되는 경우 가상선분(1,0,

4)와 (5.0.2) 두 개로 분할 된다. 이때 가상 선분의 ID는 6, 21이고 그림 7의 왼쪽과 같이 ID 6, 21에 해당하는 버킷의 리스트에 삽입한다. 그림 8은 삽입 알고리즘을 보이고 있다.

삭제 방법은 삽입 시와 동일하다. 분할 선분의 리스트를 따라 가면서 삭제할 QID와 매칭되는 데이터를 삭제하면 된다.

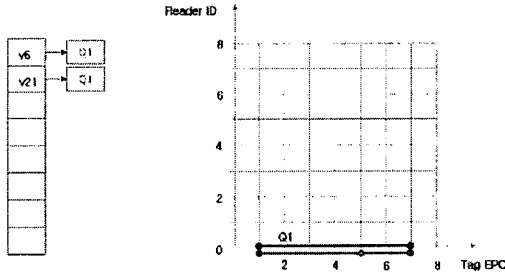


그림 7 질의 Q1의 삽입 예

```

Algorithm Insert( qid, readerID, tagEPC+, tagEPC+ )
Begin
x = tagEPC, y = readerID, length = tagEPC+ - tagEPC-
while (length > 0) do
    VI = findMaxVI(x, y, length)
    Insert(VI, qid)
    length = length - VI.length
    x = x + VI.length
end while
end
    
```

그림 8 삽입 알고리즘

검색은 리더가 RFID 미들웨어로 보내는 데이터에 의해 발생한다. 이러한 데이터는 질의 색인의 공간상에서 점으로 표현이 되므로 검색은 곧 점집의가 된다. 그림 9는 데이터 i가 삽입 될 때의 질의 검색 예를 보여주고 있다. 먼저 Data i를 포함할 수 있는 모든 가상 선분 구조를 찾는다. 그리고 해당 선분 구조에 해당하는 리스트를 따라가면 매칭되는 질의를 찾을 수 있다.

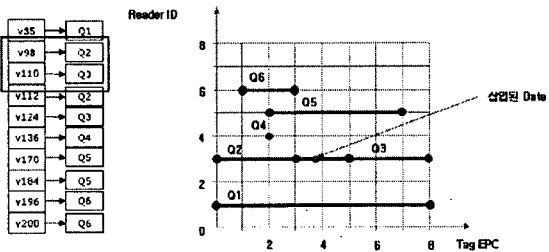


그림 9 데이터 i를 이용한 검색(점집의) 예

그림 11은 Data i의 좌표가 (a,b)일 때, 점(a,b)를 포함하는 모든 가상 선분 구조들을 보여주고 있다. 가시성을 위해 같은 선상에 위치하는 선분들을 아래로 퍼뜨려 놓았다. 그림 11을 보면 length가 4인 구조가 5개, 2인 구조가 3개, 1인 구조가 2개인 것을 확인할 수 있다. 즉, 점(a,b)를 포함하는 가상 선분 구조들은 일정한 패턴을 가지는데 그것은 length가 2^k인 선분 구조는 2^k+1개 존재한다는 것이다. 따라서 그림 10과 같이 단순화시킬 수 있다. 그림 12는 이러한 특징을 이용한 검색 알고리즘을 보이고 있다.

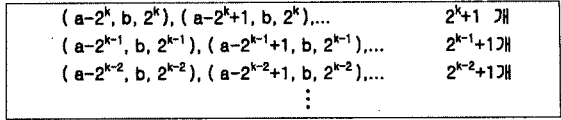


그림 10 점(a,b)를 포함하는 VI의 패턴

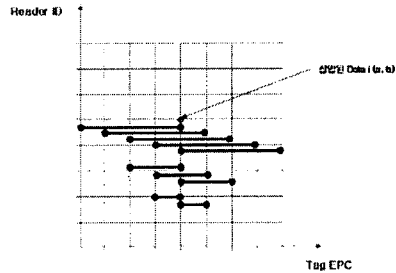


그림 11 점(a,b)를 포함하는 모든 VI

```

Algorithm Search( rid : readerID, tid : tagEPC )
Begin
exp = k, a = rid, b = tid
while (exp >= 0) do
    x = a - 2exp, y = b, length = 2exp
    n = 2exp + 1
    while (n > 0) do
        if (x >= 0) then
            result += getVList(x,y,length)
        end if
        x = x + 1, n = n - 1
    end while
    exp = exp - 1
end while
end
    
```

그림 12 검색 알고리즘

5. 결론

본 논문에서는 RFID 미들웨어의 실시간 태스크 처리를 위해 질의 색인 기법을 RIED에 적용하는 방법을 제시하였고 RFID환경의 연속 질의 Predicate을 위한 색인 기법을 제시하였다. 제시한 색인 기법은 질의를 색인의 데이터로, 데이터를 색인의 질의로 Swap하여 데이터에 독립적인 질의 검색 성능을 보장한다. 향후 연구로는 기존의 질의 색인 기법과의 평가를 통한 성능 분석이 필요하다.

6. 참고 문헌

- [1] Oat Systems & MIT Auto-ID Center, "The Savant Version 0.1(Alpha)", Auto-ID Center, 2002.
- [2] S. R. Madden, M. A. Shah, J. M. Hellerstein, and V. Raman. "Continuously adaptive continuous queries over streams." In Proc. of ACM SIGMOD, 2002.
- [3] 원종호, 이미영, 김명준, "유비쿼터스 컴퓨팅 환경을 위한 RFID 기반 센서 데이터 처리 미들웨어 기술 동향", 전자통신동향분석, 제 19권 제 5호 pp 21-30, 2004.
- [4] Kun-Lung Wu, Shyh-Kwei Chen, Philip S. Yu, "Interval query indexing for efficient stream processing.", CIKM 2004, pp 88-97, 2004.
- [5] Kun-Lung Wu, Shyh-Kwei Chen, Philip S. Yu, "VCR indexing for fast event matching for highly-overlapping range predicates.", SAC 2004, pp740-747, 2004.