RFID 미들웨어를 위한 질의 색인 기법에 관한 연구[1]

A Study of Query Indexing Scheme for RFID Middleware

박재관*, 홍봉희

(Jaekwan Park, Bonghee Hong)

부산대학교 컴퓨터공학과

{jkpack, bhhong}@pusan.ac.kr

요 약 최근 이동체 또는 센서 데이터에 대한 연속 질의를 처리하기 위해 다양한 질의 색인 기법들이 연구되고 있다. 그러나 RFID 미들웨어는 이전 연구에서 고려되지 않은 특징이 존재한다. 첫째, 질의 색인에 삽입되는 질의는 Industry, Product, Serial의 세 부분으로 구성되며, 각 값은 범위 값으로 표현될 수 있기 때문에 색인 공간에서 비 연속적인 간격의 조각들이 반복해서 나타난다. 둘째, TagID가 가지는 Industry, Product, Serial의 값은 포함관계를 가진다. 즉, 하나의 Industry에 여러 Product가 존재하며, 하나의 Product에 여러 Serial Item이 존재한다. 따라서 이러한 특징을 고려하지 못하는 기존의 질의 색인을 적용하면 다수의 질의 조각을 삽입하게 되어 색인의 성능이 급격히 저하되는 문제점이 있다.

이 논문에서는 RFID 미들웨어의 실시간 필터링 및 컬렉션을 위해서 Tag 스트림에 대한 질의 색인 기법을 제안한다. 이 논문은 ECSpec 필터링 범위의 3단계 계층 구조 특징을 고려한 코드 분할(Code Segmented) 도메인을 제시하고 이를 위한 그리드 기반 색인 구조를 제안한다. 또한 ECSpec의 필터링 패턴 특징에 의해 나타나는 질의 조각의 반복 현상을 정의하고 이를 위한 질의 처리 기법을 제시한다.

1. 서 론

EPCglobal은 RFID(Radio Frequency Identification) 기술을 이용하여 전 세계의 물류환경을 통합하기 위한 Global Network 표준화를 진행하고 있다. EPCglobal Network의 핵심으로써 RFID 미들웨어인 ALE(Application Level Event)는 하위 리더에서부터 상위 EPCIS(EPC Information Service)까지의 정보 교환을 위한 인터페이스 표준화에 중점을 두고 있다[1].

RFID 리더로부터 미들웨어로 수집되는 EPC 태그 정보들은 센서 네트워크의 데이터 스트림과 매우 유사한 특성을 가진다. 수많은 태그 정보는 여러 리더에서 인식되면서 연속적으로 시간의 순서에 따라 끊임없이 RFID 미들웨어 시스템으로 전달된다.

<그림 1> RFID 미들웨어 시스템

[1] 이 논문은 교육인적자원부 지방연구중심대학육성사업 (차세대물류IT기술연구사업단)의 지원 에 의하여 연구되었음.

애플리케이션은 미들웨어에 수집되는 수많은 EPC 태그 정보들 중에서 필요한 데이터들만을 전달받기 위해 그림 1과 같이 미들웨어에 ECSpec(Event Cycle Specification)을 등록한다. ECSpec은 어떤 리더로부터 어떤 EPC 태그 정보들을 필터링하여 보고받을 것인지에 대한 내용을 포함하고 있다. 이러한 ECSpec은 일정 시간의 사이클 동안 삽입되는 이벤트 데이터들에 대해 필터링과 컬렉션을 반복적으로 처리하여 ECReports(Event Cycle Reports)를 생성하게 된다. 즉, ECSpec은 일정 시간 범위 동안 계속적으로 수행되어야 하는 연속질의(Continuous Query)[2]와 유사한 특성을 가진다.

센서 네트워크의 스트림 데이터처럼 질의에 비해 데이터가 월등히 많을 경우에는 질의를 색인하고 데이터를 색인의 질의로 처리하는 질의 색인 기법이 좋은 성능을 발휘한다[2]. 질의 색인은 실시간으로 삽입되는 데이터 스트림을 필요로 하는 질의가 무엇인지를 빠르게 검색할 수 있도록 한다.

본 논문에서는 RFID 미들웨어인 ALE에서 ECSpec의 이벤트 필터링을 실시간으로 처리하기 위한 질의 색인 기법을 제안한다. RFID 미들웨어는 기존 환경과 다른 두 가지 특성을 가진다. 첫째, 질의 색인에 삽입되는 질의는 Industry, Product, Serial의 세 부분으로 구분되는데, 각 값은 특정 범위의 값으로 표현될 수 있기 때문에 색인 공간에서 연속적이지 않은 간격(Interval)의 조각(segment)들이 반복해서 나타나게 된다. 둘째, TagID가 가지는 Industry, Product, Serial의 값은 서로 포함관계를 가진다. 즉, 하나의 Industry에 여러 Product가 존재하며, 하나의 Product에 여러 Serial Item이 존재한다.

이 논문에서 제시하는 질의 색인은 세 가지 포함 관계를 구조적으로 표현할 수 있는 그리드 기반 3 단계 계층 구조를 가진다. 또한 간격의 조각들로 분석 및 정의하여 하나의 간격으로 수행할 수 있는 구조 및 질의 처리 방법을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 질의 색인에 관한 관련 연구를 기술하고 3장에서는 대상 환경과 문제점을 정의한다. 4장에서 이 논문에서 제안하는 질의 색인을 설명하고, 5장에서 결론을 맺는다.

2. 관련 연구

등록된 다수의 연속 질의들을 실시간으로 수행하기 위해서는 삽입되는 스트림의 개별 정보에 해당하는 질의를 빠르게 검색하는 것이 필요하다. 이를 위해 최근 질의를 색인하는 기법이 연구되고 있다[3,4,5]. 질의 색인은 실시간 처리를 위해 메인 메모리 기반 구조를 가진다. 질의를 색인하는 방법은 질의의 조건절 즉, Predicate를 삽입에 이용하는 것이다. 검색은 색인 영역 상에서 데이터 스트림이 점으로 표현되기 때문에 오직 점질의(Point Query)만 존재한다. 대표적인 다차원 질의 색인 구조로는 VCR(Virtual Construct Rectangle) Index[3,4]와 CQI(Cell-based Query Indexing) Index[5]가 있다.

VCR-Index는 이동체에 대한 영역 질의를 색인한다. 영역 질의의 Predicate는 2차원의 사각형으로 표현이 되는데 이것을 미리 정의된 VCR(Virtual Construct Rectangle)이라는 가상의 사각형 구조를 이용하여 분할하고 해당 구조의 ID List에 질의의 ID를 삽입하는 방식을 사용한다. 검색 방법은 질의 점을 포함하는 모든 분할 구조의 ID List를 탐색하는 것이다.

CQI-Index는 셀 기반의 그리드 방식을 적용한 방법으로 이동체에 대한 영역 질의를 색인한다. 전체 영역을 일정 크기의 셀로 분할하고 각 셀을 위한 2개의 ID List인 Full List와 Part List를 가진다. Part List는 분할된

영역이 셀에 부분적으로 겹치는 경우를 위한 리스트이다. 메인 메모리 기반의 색인으로 R-tree보다 높은 질의 검색 성능을 보장하지만 검색 시 Part List로부터 얻어진 결과 질의의 셋이 실제로 이동체를 포함하는지 비교하기 위한 정제 단계가 필요하기 때문에 검색 성능이 저하되는 단점이 있다.

3. 대상 환경 및 문제 정의

3.1 대상 환경

애플리케이션은 RFID 필터링 및 컬렉션 미들웨어인 ALE로 ECSpec을 등록하여 원하는 Tag 정보를 획득할 수 있다. ECSpec은 어떤 리더로부터 읽혀진 EPC 정보를 원하는지에 대한 논리적 리더명(Logical Reader Name, LRN), 어느 시간 간격동안 데이터를 수집할 것인지에 대한 ECBoundary Spec, 어떤 EPC 패턴의 데이터들을 필터링할 것인지에 대한 필터링 패턴(Filtering Pattern) 등을 포함한다. <그림 2>는 EPCglobal의 Specification에서 제시한 ECSpec의 예이다.

9.5.1 ECSpec: Door42PassThrough			
Description:	TODO: Description of why included		
Parameters:			
readers	DockDoor42		
includeSpecInReports			
ECBoundarySpec:			
startTrigger		stopTrigger	
repeatPeriod	50 MS	duration	10 MS
stableSetInterval			
ECReportSpec			
reportName	DockDoorPassThrough	reportSet	ADDITIONS
reportIfEmpty	false	reportOnlyOnChange	true
output	MEMBERS		
fmt			
group			

<그림 2> ECSpec의 예

<그림 3>은 <그림 2>의 ECSpec을 연속질의 즉, CQL(Continuous Query Language)로 표현한 것이다. LRN을 ReaderID로 표현하고 필터링 패턴을 TagEPC로 표현하였다.

```
SELECT epc FROM DataStream[Range now, now+10]
WHERE ReaderID = DockDoor42 AND TagEPC = 20.*.*
REPEAT INTERVAL 50
```

<그림 3> CQL로 표현한 ECSpec의 예

위의 <그림 3>에서와 같이 WHERE절의 Predicate를 구성하는 요소는 ReaderID와 TagEPC가 된다. 이때 ReaderID는 여러 개의 물리적 리더명(Physical Reader Name, PRN)의 집합으로 구성되며 축 상에서 PRN인 EPC(예: urn:epc:id:gid-96:1.1.1)로 표현된다. LRN을 구성하는 PRN들은 서로 discrete한 특성을 가지기 때문에 ReaderID는 축 상에서 점으로 표현 된다. TagEPC는 필터링 패턴이 특정값 또는 범위값으로 구성되기 때문에 점(Point) 또는 간격(Interval)으로 표현 된다. 따라서 ECSpec을 위한 질의의 Predicate은 ReaderID와 TagEPC로 구성되는 2차원의 Interval로 표현된다.

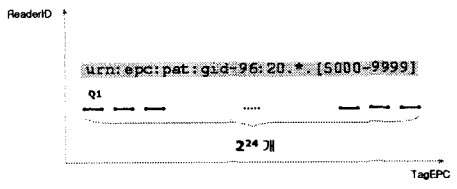
	Header	General Manager Number (Company)	Object Class (Product)	Serial Number (Serial)
GID-96	8 0011 0101 (Binary Value)	28 268,435,455 (Max decimal)	24 16,777,215 (Max decimal)	36 68,719,476,735 (Max decimal)

<그림 4> 96 bit EPC(GID-96)의 인코딩

TagEPC는 <그림 4>와 같이 회사, 제품, 아이템의 세 부분으로 나뉘며, 하나의 회사에 다수의 제품이, 하나의 제품에 다수의 아이템이 존재하도록 구성하여 포함관계를 가진다.

3.2 문제 정의

하나의 ECSpec의 필터링 패턴을 연속 질의로 변환하면 아래 <그림 5>와 같이, 하나의 질의가 여러 개의 조각들로 분리된 형태로 표현된다. 이것은 EPC 코드가 세 부분으로 분할되어 있기 때문에 발생하게 된다.



<그림 5> 여러 조각으로 구성되는 연속 질의의 예

기존의 색인에서 <그림 5>와 같은 질의를 삽입할 경우, 2^{24}개의 조각을 삽입해야 하는

문제가 있다.

ECSpec에 기술되는 EPC Code는 세 부분으로 구성되며, 포함관계를 가지는 특징이 있다. 이러한 코드간의 관계를 가지는 도메인 환경을 이 논문에서는 코드 분할 도메인(Code Segmented Domain)이라고 정의한다. 기존의 질의 색인 또는 공간 색인은 대부분 x, y, t의 서로 독립적인 도메인을 대상으로 하므로 코드의 포함관계를 고려하지 못하므로 본 환경에 적용할 경우 질의의 삽입 및 검색 성능이 떨어지게 된다.

4. Grid 기반 Multi-Layer Interval Index

이 장에서는 코드 분할 도메인을 가지는 RFID 미들웨어의 태그 정보 필터링 및 수집을 위한 Grid 기반 Multi-Layer Interval(이하, MLI) Index를 기술한다. 색인을 설명하기 위해 먼저 아래의 용어들을 정의한다.

- QI(Query Interval) : 질의 색인에 삽입되는 질의로써, 2차원의 Interval로 표현되는 데이터
- DI(Discrete Interval) : lower bound에서 upper bound까지의 범위가 여러 개의 조각들로 분리되어 있으며, 분리된 조각들은 일정한 패턴으로 구성되는 Interval
- DPI(Discrete Pattern Interval) : lower bound에서 upper bound까지의 범위가 1개 이상의 패턴으로 구성되는 Interval로서, 1개 이상의 DI로 구성되는 Interval
- MBI(Minimum Bounding Interval) : DI의 lower bound와 upper bound를 연결한 가상 Interval

4.1 Basic Idea

메인 메모리기반의 질의 색인은 Tree 계열보다 Grid 계열 색인이 더 나은 성능을 보인다[5]. TagEPC는 Industry, Product, Serial의 독립적이며, 포함관계를 가지는 구조로 구성되는 특징이 있다.

이 논문에서는 ReaderID와 Industry로 구성되는 2차원의 1^{st}-layer Grid, Product로 구성되는 1차원의 2^{nd}-layer Grid, Serial로 구성되는 1차원의 3^{rd}-layer Grid의 3단계 그리드 기반 색인 구조를 제안한다. 3단계 색인 구조는 하나의 그리드 구조를 사용하는 기존 색인 보다 사용되는 Directory 수가 감소하는 효과가 있다. 예를 들어, GID-96 코드 체계를 사용하고 그리드의 크기가 2^{10}이라고 가정할 때, 기존 고정 그리드의 경우 $2^{78}(2^{88}/2^{10})$개의 Directory 이 존재하지만 3단계 GRID의 경우 약 $2^{26}(2^{28}/2^{10} + 2^{24}/2^{10} + 2^{36}/2^{10})$개의 Directory 만이 존재하게 된다.

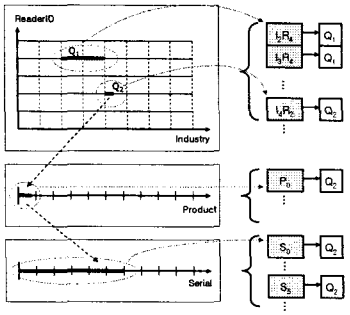
RFID 질의 색인에 삽입되는 QI는 다양한 형태의 DPI가 존재한다. 하나의 연속 질의를 표현한 DPI를 개별 조각으로 처리하는 것은 비효율적이다. DI의 조각들이 동일한 크기를 가지며, 조각간의 간격이 동일하다는 특징을 이용한다. 즉, DI의 조각들의 반복이 나타나는 2^{nd} , 3^{rd} Layer Grid에 DI를 1회 삽입하고 검색 시에 후처리 정제 단계를 통한 최종 결과셋을 도출한다.

4.2 Index Structure

질의 색인에 삽입되는 데이터는 ECSpec에 의해 생성되는 Interval이며, 질의는 개별 Tag 인식에 의해 발생하는 점질의만 존재한다. 이러한 특징에 의해 색인은 포함관계 단위로 개별 그리드를 구성하여, Directory를 공유하더라도 원하는 결과를 탐색할 수 있다.

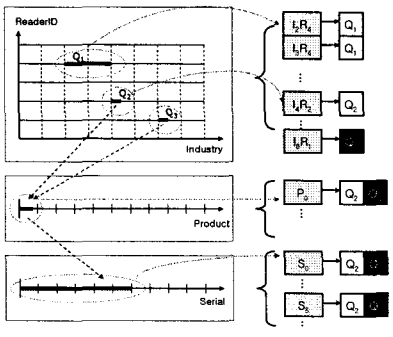
그림 6은 이 논문에서 제안하는 Multi-Layer Interval Index의 구조를 나타내고 있다. 1^{st} Layer는 Industry축과 ReaderID축으로 구성되는 2차원의 공간을 그리드 셀로

분할한다. 2nd Layer는 Product축의 1차원 공간을 그리드 간격으로 분할한다. 3rd Layer는 Serial축의 1차원 공간을 그리드 간격으로 분할한다. 이 논문에서 각 Layer의 분할 방식은 Fixed Grid의 고정 크기 분할 방식을 따르며, Grid File의 동적 분할 방식도 적용 가능하다. 각 Layer는 색인에 삽입되는 데이터와 겹치는 셀 또는 간격에 대한 Directory만 동적으로 생성 및 유지한다.



<그림 6> Multi-Layer Interval Index 구조

Product와 Serial을 전체 범위를 포함하고, Industry만 특정 값을 가진 Q_1이 색인에 삽입되면 1st Layer에만 Q_1을 위한 Directory를 생성하고 삽입한다. Serial의 일정 범위를 가지는 Q_2가 삽입되면, 먼저 1st Layer에 삽입하고 2nd Layer에 삽입한 후 마지막으로 3rd Layer에 삽입한다. 이것은 색인에 대한 검색의 과정이 상위 Layer에서 하위 Layer로 Top Down 방식으로 이루어지면서 결과셋을 줄여가는 방식을 가능하게 한다.



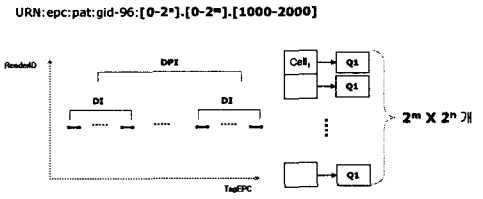
<그림 7> Directory 공유의 예

<그림 7>은 3단계 Layer로 분리함으로써 색인의 다수 데이터가 Directory 을 공유하는 예를 보여준다. Q_2와 Industry는 다르지만, Product, Serial 범위는 동일한 Q_3가 삽입되면 1st Layer에 Directory 을 생성하고 삽입한다. 그러나 2nd와 3rd Layer는 Q_2와 동일하므로 새로운 Directory 을 생성하지 않고 Q_3만 추가함으로써 삽입을 완료한다. 즉, 하위 Layer로 갈수록 Directory 의 재활용 빈도는 높아진다.

4.3 Discrete Pattern Interval

ECSpec에 의해 생성되는 Query Interval(QI)은 연속적인 Interval, DI, DPI 중 하나로 표현된다. 특히, DI와 DPI는 기존 색인을 사용할 경우, 다수의 삽입 연산이 필요하다.

<그림 8>은 Industry, Product, Serial의 값이 범위를 가질 때 Grid File에 2^m X 2^n 개의 삽입이 발생함을 보여준다.



<그림 8> DPI의 예

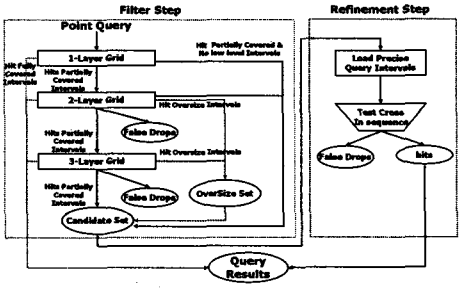
DI와 DPI는 MLI Index에서 효과적으로 색인할 수 있다. DI를 구성하는 조각들의 크기, 조각간의 간격은 동일하다. 즉, DI를 MLI Index를 삽입하게 되면 동일한 Directory에 동일한 Query ID를 중복해서 삽입하게 된다. 이것을 피하기 위해서 각 Directory에 DI를 위한 별도의 메모리 공간을 할당하고 1회만 삽입한다.

DPI의 경우, DI의 MBI가 동일한 크기, 동일한 간격으로 구성됨을 알 수 있다.

따라서 DI의 방법과 유사하게 MBI를 1회만 삽입하도록 한다. 즉, 다수 삽입 연산을 수행한 후 검색하기 보다는, 고부하 삽입 연산을 줄이면서 저부하 정제 단계의 작업을 늘림으로써 전체적인 색인 성능을 향상시킨다.

4.4 Search

검색은 리더에 의해 태그 정보가 인식될 때마다 반복적으로 수행되며 질의 색인에 대한 점질의로 수행된다. 검색방법은 그림 9와 같이 필터 단계와 정제 단계의 두 단계로 진행된다.



<그림 9> Filter and Refinement Step

필터링은 해당 점을 포함할 수 있는 각 Layer의 Grid 셀을 찾고 해당 구조의 Directory에 연결된 ID List를 저장한다. 1st Layer에서 3rd Layer로 Top-Down Search 방식으로 진행되며, 1st Layer에서 탐색된 Candidate Set 이외의 것이 2nd or 3rd Layer에서 탐색되더라도 검색 결과로 포함되지 않는다. 즉, 하위 Layer로 내려가면서 Candidate Set을 줄여나가는 방식으로 검색이 진행된다.

정제는 DI와 DPI에 대한 검증을 수행하는 단계이다. DI의 경우, 점질의의 tagEPC에서 DI의 lower bound를 뺀 값에 대하여 DI의 Size와 Gap을 더한 값으로 Moduler한 결과값이 Size보다 작거나 같으면 Hit이며, 아닐 경우 Drop이다. DPI도 유사한 연산으로 처리되며, 매우 단순하고 빠른 연산으로 검증할 수 있다.

5. 결론 및 향후 연구

본 논문에서는 RFID 미들웨어의 실시간 필터링 및 컬렉션을 위한 질의 색인 기법을 제안하였다. 제시한 색인 기법은 EPC 코드가 가지는 3단계 포함관계를 색인에 반영하여, 다수의 질의가 등록될 때 Directory의 공유를 통하여 공간 활용도를 높이는 방법을 제안하였다. 또한 ECSpec을 연속 질의로 변환했을 때 발생하는 비 연속적인 Interval 문제를 해결하는 방법을 제시하였다.

향후 연구로써 반복적으로 활성화, 비활성화되는 연속질의를 효과적으로 처리할 수 있는 기법을 연구하는 것이 필요하다.

<참고문헌>

[1] K.Traub, S.Bent, T.Osinski, S.N.Peretz, S.Rehling, S.Rosenthal, B.Tracey, "The Application Level Event (ALE) Specification, Version 1.0", EPCglobal , 2005.

[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] 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.

[4] Kun-Lung Wu, Shyh-Kwei Chen, Philip S. Yu: Processing Continual Range Queries over Moving Objects Using VCR-Based Query Indexes. MobiQuitous 2004: 226-235

[5] D.V.Kalashnikov, S.Prabhakar, W.G.Aref, and S.E.Hambrusch. Efficient evaluation of continuous range queries on moving objects. In Proc. of 13th DESA, 2002.