실시간 RFID 미들웨어에서의 태그 데이터 고속 필터링 방법

박미선*, 김용진**, 유민수**
*한양대학교 지능형로봇학과
**한양대학교 전자컴퓨터통신공학부
e-mail : {mspark,yjkim,msryu}@rtcc.hanyang.ac.kr

A method for high-speed event processing in the real-time RFID middleware systems

Mi Sun Park*, Yong Jin Kim**, Minsoo Ryu **
*Dept. of Intelligence Robotics, Hanyang University
**Dept. of Electronics & Computer Engineering, Hanyang University

요 약

RFID 시스템의 미들웨어는 태그에서 생성된 방대한 양의 데이터를 리더를 통해 전달받는다. RFID 미들웨어는 이러한 데이터를 정제하여 응용 애플리케이션에 전달하는 기능을 담당한다. 하지만 태그 데이터 정제 과정에서 발생되는 지연 시간은 RFID 미들웨의 응답성을 저하시킨다. 본 논문은 EPCglobal 의 RFID 미들웨어 표준인 ALE 에 의거하여 태그 데이터에 대한 다수의 필터링 조건들이 주어진 RFID 미들웨어 환경에서 실시간으로 수집되는 대용량의 태그에 대한 고속 필터링 엔진을 설계한다. 이를 위하여 Intermediate node 들이 key 값을 저장하는 Binary Search Tree 형태를 구성하여 태그를 필터링하는 방법을 제안한다. 결과로써 기존의 순차적인 RFID 데이터 필터링에 비해 고속의 필터링 성능을 보이며 특히 필터의 수가 증가할수록 필터링의 효율이 높아짐을 보인다.

1. 서론

RFID 미들웨어는 다수의 센싱 리더기 등 장비를 관리하거나, 이기종의 RFID 환경에서 발생하는 대량의 가공되지 않은 데이터를 수집, 필터링하여 의미 있는 정보로 변환하여 응용 소프트웨어 등에 필요한 정보를 제공하는 소프트웨어 플랫폼으로 정의할 수 있다[1].

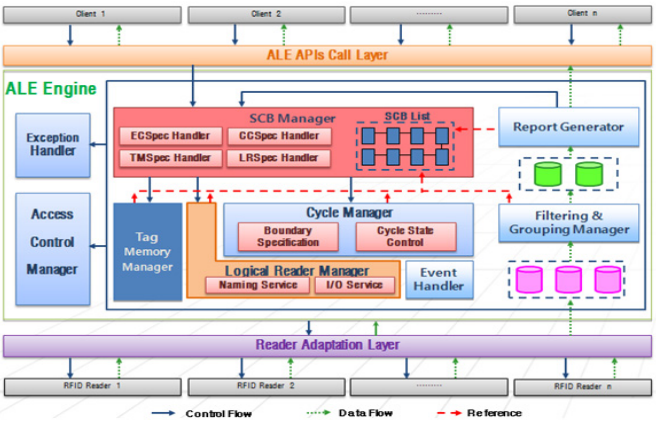
RFID 미들웨어의 핵심 기능은 불필요한 태그 정보를 제거하고 필요한 정보를 정제하는 필터링이라 할 수 있다. 본 논문은 대용량 태그 데이터에 대해 고속의 필터링 작업을 수행하기 위한 RFID 필터링 모듈 구현 방안을 제안한다. 물론 제안된 모듈은 EPCglobal의 RFID 미들웨어 표준인 ALE에 준하여 설계되었다.

2. 관련 연구 및 문제 정의

본 장에서는 ALE 표준을 따르는 RFID 미들웨어 구조와 EPCglobal 리더 프로토콜에서 정의한 RFID 태그 구조와 ALE 표준 태그 필터에 대하여 살펴본다.

2.1 RFID 미들웨어 아키텍처 설계

설계한 RFID 미들웨어의 아키텍처 인터페이스는 2 개의 Layer 로 구분된다. Client 와 ALE 의 통신을 위한 ALE APIs Call Layer, RFID Reader 와 ALE 통신을 위한 Reader Adaptation Layer 가 있다. 그리고 ALE Engine 은 6 개의 Manager, 2 개의 Handler, Report Generator 가 있다.



(그림 1) RFID 미들웨어 구조

그림 1 에서 나타내고 있는 Filtering&Grouping Manager 는 태그 데이터의 수집과 필터링을 담당한다.

2.2 EPC 태그 구조와 ALE 필터 패턴 분석

표준 EPC 태그 데이터는 Header 와 Domain Identifier 로 구성되며, Domain Identifier 은 피리어드로 분리된 Domain Manager, Objcet Class, Serial Number 필드로 구성된다. GID-96 태그의 경우 전체가 96 비트이며 헤더 부분이 8 비트, 회사 필드가 28 비트, 제품 필드가 24 비트, 시리얼번호 필드가 36 비트로 구성된다[3].

ALE 인터페이스와 관련된 주요 자료형으로 ECSpec

과 ECReport 가 있다[2]. ECSpec 에 포함되는 ECFilterSpec 은 포함 패턴(IncludePattern)과 제외 패턴(ExcludePattern)의 두가지 패턴을 가지고 있다. 포함 패턴은 수집되어야 할 태그 데이터를 나타내고, 제외 패턴은 제외되어야 할 태그 데이터를 나타낸다. 필터 패턴은 각 단일값(fix value), 범위([최소값-최대값]), 전체범위(*)로 나타낼 수 있는 3 가지 표현 형식으로 표기될 수 있다.

<표 1> 필터 패턴의 예(GID-96)

urn:epc:pat:gid-96:20.300.400	코드번호가 20 인 회사로 객체 클래스번호 30 에 속하며 일련번호가 400 에 해당하는 제품
urn:epc:pat:gid-96:20.*.[1-50]	일련번호가 1 에서 50 까지로 코드번호가 20 인 회사의 제품
urn:epc:pat:gid-96:*.*.*	GID-96 포맷을 따르는 모든 제품

본 논문에서는 필터 패턴이 가지는 특성을 분석하여, 고속의 태그 필터링을 위해 Tree 를 이용한 필터링 방법을 제안한다.

3. 제안하는 RFID 미들웨어 필터링 기법

제안하는 방법의 주요 목적은 필터링의 불필요한 매칭 과정을 제거하는 것이다. 패턴 검색과정을 최소화하는 방법으로 필터 패턴을 Binary Pattern Search Tree 로 구성하여 필터링하는 방법을 제안한다.

3.1 트리의 형태

패턴 필터링에 사용될 트리는 Intermediate node 들이 key 값과 Level 값을 저장하는 이진 검색 트리 형태로 구성된다. 하지만 트리의 Depth 가 깊어져 검색 횟수가 증가하는 문제가 있다. 이를 해결하기 위해 필터 패턴의 각 필드값들의 표준편차를 계산하여 가장 작은 값을 가지는 필드를 Level y 로 선정한다. 이는 트리의 Depth 를 감소시켜 패턴 검색의 횟수를 감소시킨다. key 값은 선택된 y 필드값들의 중간값으로 설정한다. 필터 패턴이 더 이상 나누어지지 않을 때까지 각 노드의 필터패턴으로 Level 과 key 값을 계산하여 트리 노드를 구성한다.

3.2 필터링 방법

태그 데이터는 Level 번째 필드값과 트리 노드의 key 값을 비교하여 Binary Pattern Search Tree 를 검색한다. 태그 필터링은 트리 검색 과정이 최종 leaf node 에 다다르면 leaf node 가 가지는 필터 패턴과 태그 데이터를 비교하여 매칭 결과를 얻을 수 있다. 필터링 방법은 다음 2 단계로 요약할 수 있다.

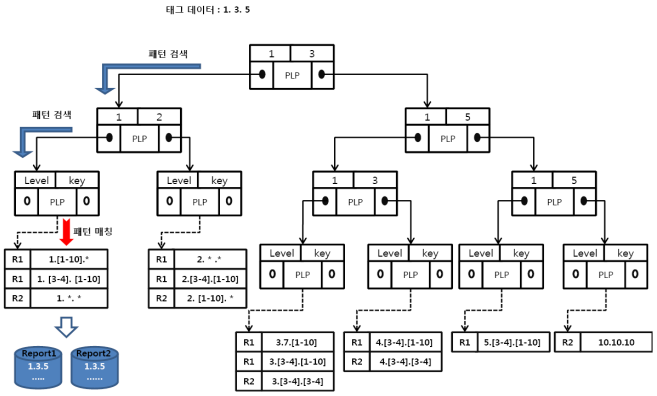
- 1단계 : 태그 데이터의 y번째 필드의 값이 key 보다 작으면 왼쪽 child를 검색하고, 크거나 같으면 오른쪽 child를 검색한다.
- 2단계 : Leaf node에 다다르면 필터 패턴과 태그 데이터를 비교 검사한다. 일치하는 태그 데이터는 해당 리포트에 저장한다.

4. 구현 및 평가

본 장에서는 논문에서 제안하는 방법을 구현하고 모의 실험을 통하여 평가를 한다. 실제적인 구현은 자바 LinkedList 클래스를 사용하여 Binary Pattern Search Tree 를 구현하였다.

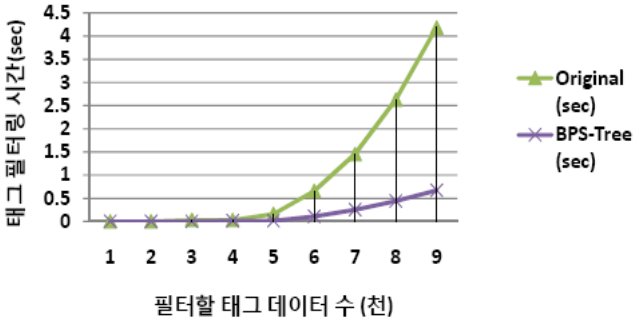
4.1 필터링 속도 비교를 통한 성능 평가

두개의 Report 가 가지는 8 개의 필터패턴으로 트리를 구성하고 최소 1 천개에서 최대 9 천개의 태그 데이터를 필터링하였다. 일반적으로 수행되는 순차적 패턴 필터링 방법(이하 Original)과 제안하는 필터링 방법(이하 BPS-Tree)으로 실험하였다.



(그림 2) 필터링 예시

주어진 7 개의 테스트 패턴으로 6 개의 node 로 구성된 3depth 의 Binary Pattern Search Tree 가 형성되었다. 각 단말 node 에는 패턴검색에 대한 결과 패턴들이 리스트로 형성된다.



(그림 3) 필터링 속도 비교

필터링의 실행 시간을 비교해보면, 그림 3 과 같이 제안하는 BPS-Tree 필터링 방법의 경우 태그의 수가 증가할 수록 필터링의 성능이 높아짐을 알 수 있다.

5. 결론 및 향후 방향

본 논문에서는 먼저 ALE 스펙을 분석하고 RFID 미들웨어의 필터링 기능에 중점을 두어 효율적인 필터링 방안에 대하여 모색하였다. 제안한는 방법은 필터링 성능 평가시 기존의 순차적인 방법 보다 속도적인 측면에서 우수함을 알수 있었다.

향후 연구 방향으로 본 논문의 내용을 실제 응용 서비스에서의 효율성을 입증할 예정이다.

참고문헌

[1] EPCglobal, “The EPCglobal Architecture Framework,” Final Version of 1 July, 2005.
[2] EPCglobal, “Application Level Events (ALE) Standards Version 1.1” Standard Specification, Febuary, 2008.
[3] EPCglobal, “Tag Data Standards Version 1.4” Standard Specification, June, 2008.